Penalized Regression

Ridge, Lasso, ElasticNet

SYS 6018 | Spring 2021

penalized.pdf

Contents

1 Shrinkage and Penalized Regression Intro

1.1 Required R Packages

We will be using the R packages of:

- `MASS` for ridge regression
- `glmnet` for ridge, lasso, and elasticnet regression
- `tidyverse` for data manipulation and visualization

```
library(MASS)
library(glmnet)
library(broom)
library(tidyverse)
```

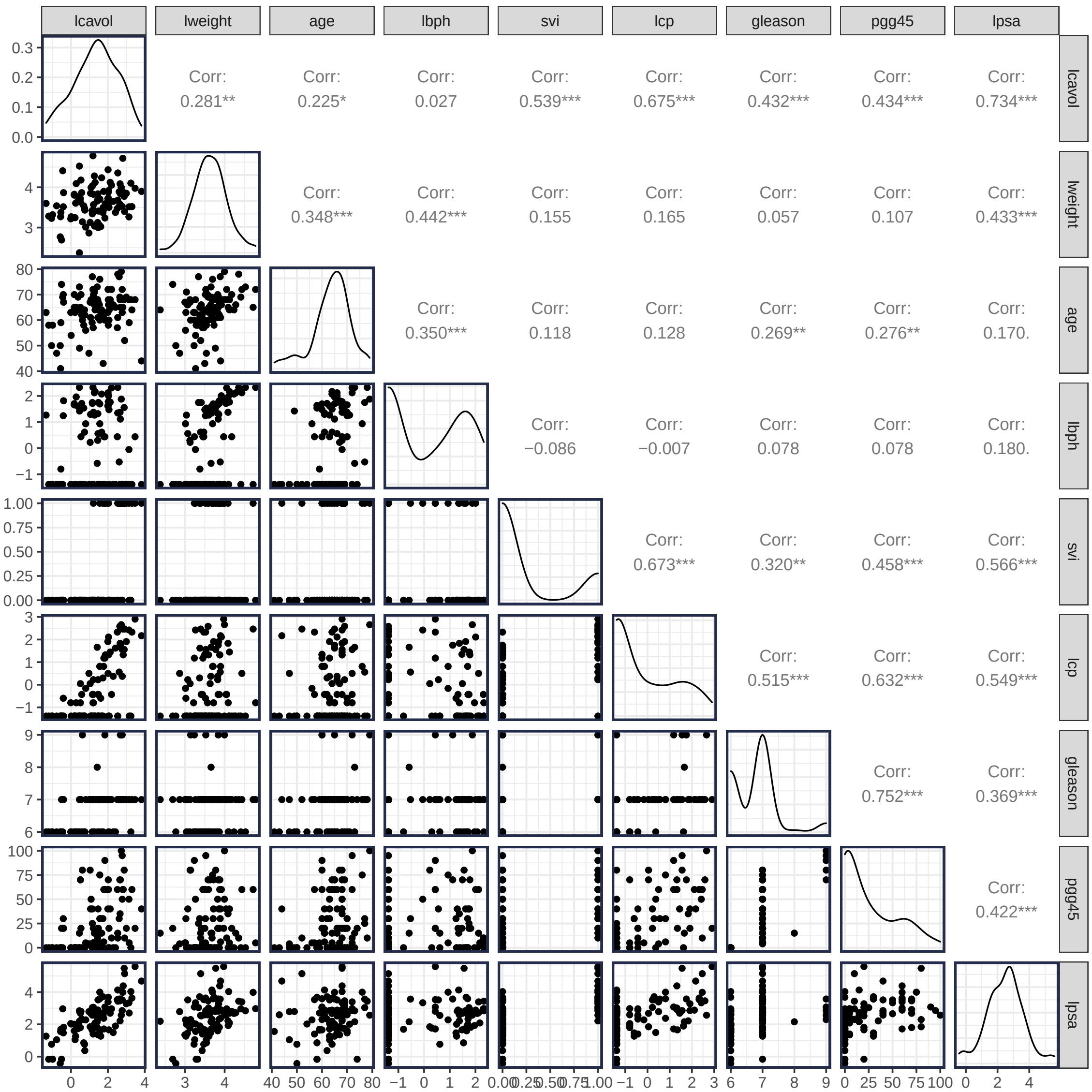
> Some of the figures in this presentation are taken from "An Introduction to Statistical Learning, with applications in R" (Springer, 2013) with permission from the authors: G. James, D. Witten, T. Hastie and R. Tibshirani.

1.2 Prostate Cancer Data

The Elements of Statistical Learning (ESL) text has a description of a prostate cancer dataset used in a study by Stamey et al. (1989). They examined the correlation between the level of prostate-specific antigen and a number of clinical measures in men who were about to receive a radical prostatectomy.

The variables are:

- log cancer volume (`lcavol`)
- log prostate weight (`lweight`)
- `age`
- log of the amount of benign prostatic hyperplasia (`lbph`)
- seminal vesicle invasion (`svi`)
- log of capsular penetration (`lcp`)
- Gleason score (`gleason`)
- percent of Gleason scores 4 or 5 (`pgg45`)
- *outcome variable* is the log of prostate-specific antigen, (`lpsa`)

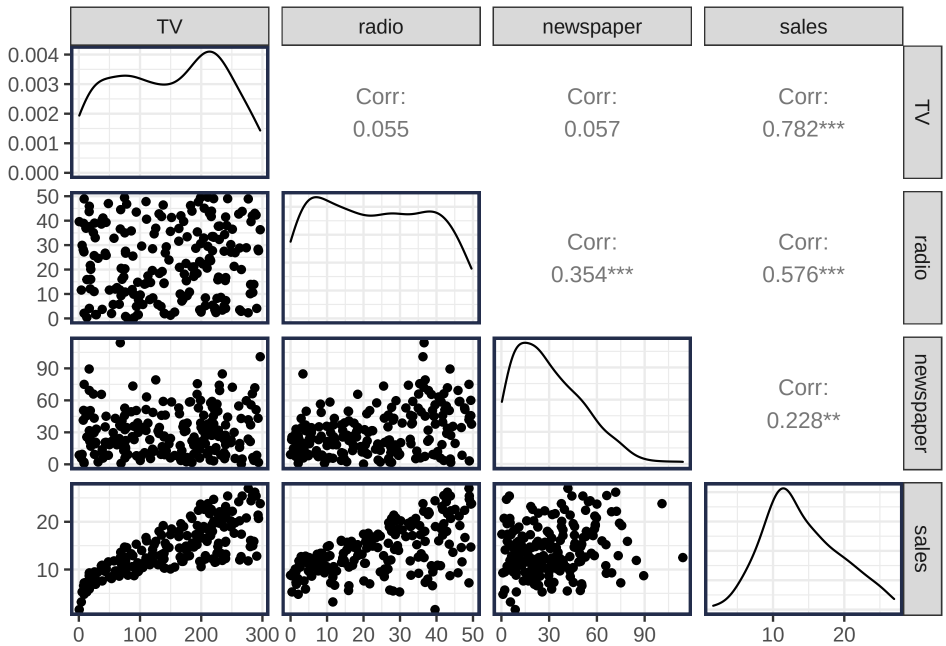


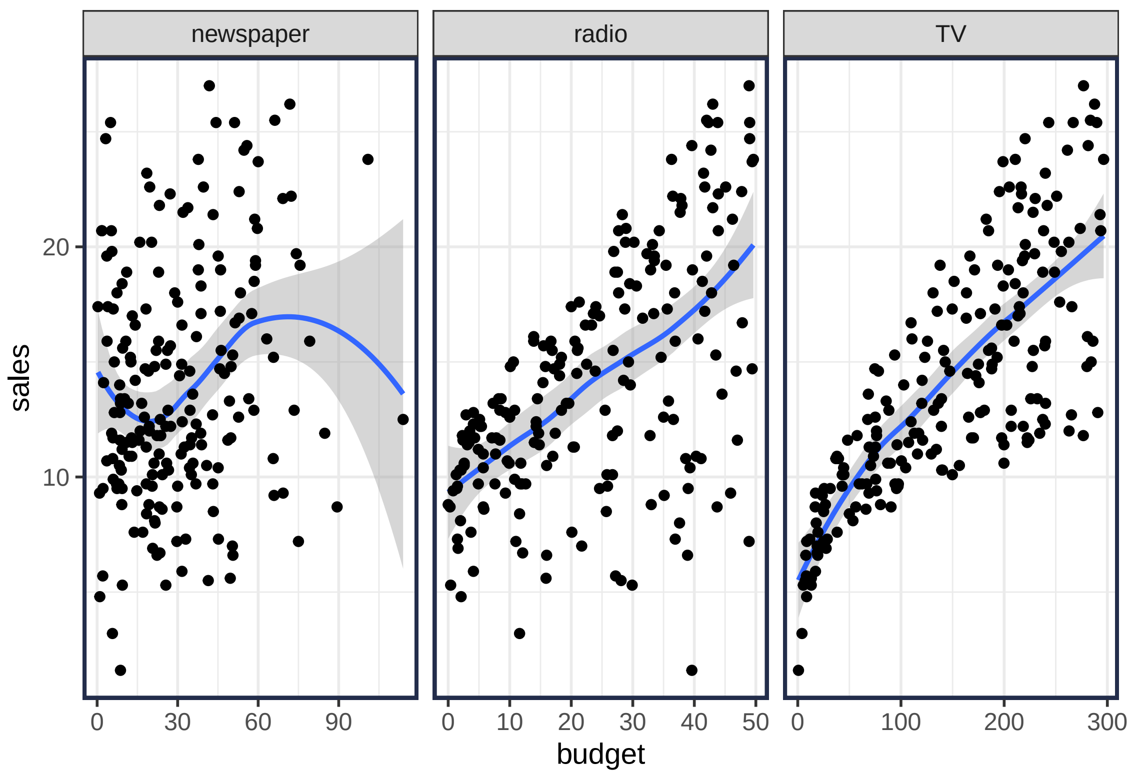
1.3 Advertising Data

The Introduction to Statistical Learning (ISL) text has some data on advertising.

These data give the sales of a product (in thousands of units) under advertising budgets (in thousands of dollars) of TV, radio, and newspaper.

The goal is to predict `sales` for a given `TV`, `radio`, and `newspaper` budget.





1.4 Linear Regression (OLS)

The standard generic form for a linear regression model is

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots, + \beta_p X_p + \epsilon$$

- Y is the response or dependent variable
- $X_1, X_2, \ldots, X_p$ are called the p explanatory, independent, or predictor variables

- the greek letter ϵ (epsilon) is the random error variable
- For example:

$$\text{sales} = \beta_0 + \beta_1 \times (\text{TV}) + \beta_2 \times (\text{radio}) + \beta_3 \times (\text{newspaper}) + \text{error}$$

Training data is used to estimate the *model parameters* or *coefficients*.

$$\left[\begin{array}{cccccc|c} x_{11} & x_{12} & \cdot & \cdot & \cdot & x_{1p} & y_1 \\ x_{21} & x_{22} & \cdot & \cdot & \cdot & x_{2p} & y_2 \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ x_{n1} & x_{n2} & \cdot & \cdot & \cdot & x_{np} & y_n \end{array}\right]$$

Producing the predictive model:

$$\hat{y}(x_1, x_2, \ldots, x_p) = \hat{\beta}_0 + \hat{\beta}_1 x_1 + \hat{\beta}_2 x_2 + \ldots, + \hat{\beta}_p x_p$$

- where $\hat{\beta}_j$ are the weights assigned to each variable
- these weights are the values that minimize the residual sum of squares (RSS) for predicting the training data
- For example:

$$\widehat{\text{sales}} = 2.939 + 0.046 \times (\text{TV}) + 0.189 \times (\text{radio}) \times -0.001 \times (\text{newspaper})$$

- The *complexity* of an OLS regression model is the *number of estimated parameters*
  - it is $p + 1$ (using the notation above), where the $+1$ is added for the intercept.

1.5 Estimation

- The weights/coefficients (β) are the *model parameters*
- OLS uses the weights/coefficients that minimize the RSS loss function over the training data

$$\begin{aligned} \hat{\beta} &= \arg\min_{\beta} \ \text{RSS}(\beta) \qquad \text{Note: } \beta \text{ is a } \textit{vector} \\ &= \arg\min_{\beta} \ \sum_{i=1}^{n} (y_i - f(\mathbf{x}_i; \beta))^2 \\ &= \arg\min_{\beta} \ \sum_{i=1}^{n} (y_i - \beta_0 - \beta_1 x_{i1} - \beta_2 x_{i2} + \ldots + \beta_p x_{ip})^2 \end{aligned}$$

1.5.1 Matrix notation

$$f(\mathbf{x}; \beta) = \mathbf{x}^{\mathsf{T}} \beta$$

$$Y = \begin{bmatrix} Y_1 \\ Y_2 \\ \vdots \\ Y_n \end{bmatrix} \qquad X = \begin{bmatrix} 1 & X_{11} & X_{12} & X_{13} & \ldots & X_{1p} \\ 1 & X_{21} & X_{22} & X_{23} & \ldots & X_{2p} \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & X_{n1} & X_{n2} & X_{n3} & \ldots & X_{np} \end{bmatrix} \qquad \beta = \begin{bmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_p \end{bmatrix}$$

$$\mathrm{RSS}(\beta) = (\mathrm{Y} - \mathrm{X}\beta)^\mathsf{T}(\mathrm{Y} - \mathrm{X}\beta)$$

$$\begin{aligned}
\frac{\partial \mathrm{RSS}(\beta)}{\partial \beta} &= 2X^\mathsf{T}(Y - X\beta) \\
&\implies X^\mathsf{T}Y = X^\mathsf{T}X\beta \\
&\implies \boxed{\hat{\beta} = (X^\mathsf{T}X)^{-1}X^\mathsf{T}Y}
\end{aligned}$$

1.5.2 OLS in R with `lm()`

```
#-- Fit OLS
prostate.lm = lm(lpsa~., data=prostate.train)
prostate.lm %>% broom::tidy()
#> # A tibble: 9 x 5
#>   term        estimate std.error statistic    p.value
#>   <chr>          <dbl>     <dbl>     <dbl>      <dbl>
#> 1 (Intercept)   0.429     1.55       0.276 0.783
#> 2 lcavol        0.577     0.107      5.37  0.00000147
#> 3 lweight       0.614     0.223      2.75  0.00792
#> 4 age          -0.0190    0.0136    -1.40  0.168
#> 5 lbph          0.145     0.0705     2.06  0.0443
#> 6 svi           0.737     0.299      2.47  0.0165
#> # ... with 3 more rows
```

1.6 Some Problems with least squares estimates

There are a few problems with using least squares estimation (OLS) to estimate the regression parameters (coefficients)

- *Prediction Accuracy*
  - the least squares estimates in high dimensional data often have low bias but large variance.
  - Prediction accuracy can sometimes be improved by shrinking or setting some coefficients to zero.
  - By doing so we sacrifice a little bit of bias to reduce the variance of the predicted values, and hence may improve the overall prediction accuracy.
  - Some predictors may not have any predictive value and only increase noise
- *Interpretation*: With a large number of predictors, we often would like to determine a smaller subset that exhibit the strongest effects. In order to get the "big picture", we are willing to sacrifice some of the small details
  - When $p > n$ least squares won't work at all

1.7 Improving Least squares

We will examine 3 standard approaches to improve on least squares estimates

1. Subset Selection
   - Only use a subset of predictors, but estimate with OLS
   - Examples: *best subsets*, *forward step-wise*
2. Shrinkage/Penalized/Regularized Regression
   - Instead of an "all or nothing" approach, shrinkage methods force the coefficients closer toward 0.

- Examples: *ridge*, *lasso*, *elastic net*

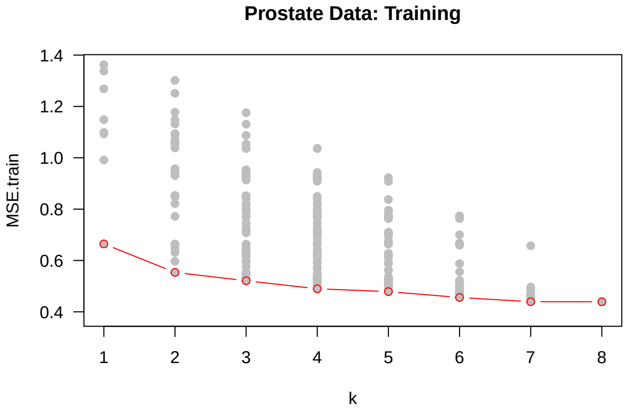
3. Dimension Reduction with Derived Inputs
    - Use a subset of linearly transformed predictors
    - Examples: *PCA*, *PLS*

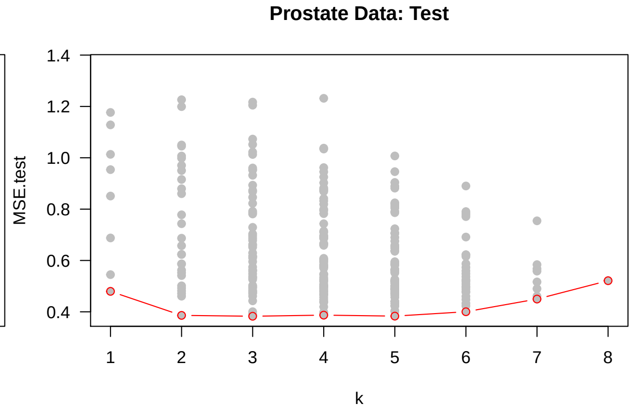
All three methods introduce some additional bias in order to reduce variance and *hopefully* improve prediction.

2 Subset Selection

Subset selection methods attempt to find the best subset of predictors to use in the model

- Best Subsets finds the best (usually in terms of minimum RSS/MSE) combination of k predictors
  - k is the tuning parameter
- Stepwise Selectors takes a greedy approach by sequentially adding (forward) or deleting (backward) the predictor that most improves the fit
  - This is a computational necessity for high dimensional data





Subset selection methods remove predictors by setting their coefficients to 0 (e.g., $\hat{\beta} = 0$)

- These "all or nothing" approaches can be very unstable. A small change in the data can completely change the model

predictor	lm	best_subset	bootstrap
(Intercept)	0.43	-1.05	-0.33
lcavol	0.58	0.63	0.51
lweight	0.61	0.74	0.54
age	-0.02	0.00	0.00
lbph	0.14	0.00	0.14
svi	0.74	0.00	0.67
lcp	-0.21	0.00	0.00
gleason	-0.03	0.00	0.00
pgg45	0.01	0.00	0.00

3 Shrinkage Methods

Instead of an "all or nothing" approach, shrinkage methods force the coefficients closer toward 0.

- Usually this is accomplished through penalized regression where a penalty is imposed on the size of the coefficients
- Equivalently, the size of the coefficients are *constrained* not to exceed a threshold

The general framework is

$$\hat{\beta} = \arg\min_{\beta} \{l(\beta) + \lambda P(\beta)\}$$

where

- $l(\beta)$ is the loss function (e.g. mean squared error, negative log-likelihood)
- $\lambda \geq 0$ is the strength of the penalty
- $P(\beta)$ is the penalty term (as a function of the model parameters)

3.1 Two Representations

The penalized optimization (Lagrangian form)

$$\hat{\beta} = \arg\min_{\beta} \{l(\beta) + \lambda P(\beta)\}$$

An equivalent representation is (constrained optimization)

$$\begin{aligned}\hat{\beta} &= \arg\min_{\beta} \; l(\beta) \qquad \text{subject to} \;\; P(\beta) \leq t \\ &= \arg\min_{\beta:\, P(\beta)\leq t} \; l(\beta)\end{aligned}$$

3.2 Penalties

Examples penalties:

- Ridge Penalty

$$P(\beta) = \sum_{j=1}^{p} |\beta_j|^2 = \beta^{\mathsf{T}}\beta$$

- Lasso Penalty

$$P(\beta) = \sum_{j=1}^{p} |\beta_j|$$

- Best Subsets

$$P(\beta) = \sum_{j=1}^{p} |\beta_j|^0 = \sum_{j=1}^{p} 1_{(\beta_j \neq 0)}$$

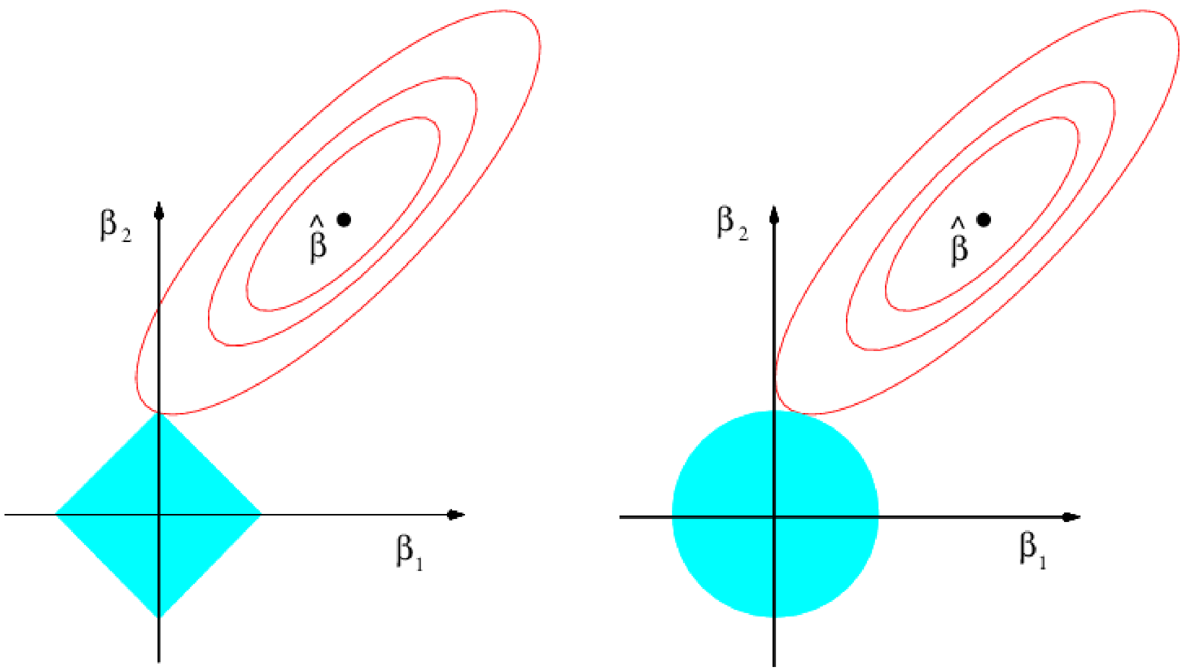


Figure 1: Contours of the error and constraint functions for the lasso (left) and ridge (right). The solid blue areas are the constraint regions, $|\beta_1| + |\beta_2| \leq t$ and $\beta_1^2 + \beta_2^2 \leq t^2$, while the red ellipses are the contours of the RSS (residual sum of squares).

4 Ridge Regression

For ridge regression

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

$$P(\beta) = \sum_{j=1}^{p} |\beta_j|^2 \qquad \text{(Notice that the intercept, } \beta_0\text{, is not penalized)}$$

Watch for how software defines the loss. Some use MSE, others use SSE = n*MSE.

So the ridge solution becomes:

$$
\begin{aligned}
\hat{\beta}_\lambda^{\text{ridge}} &= \arg\min_\beta \frac{1}{n} \sum_{i=1}^n \left(y_i - \beta_0 - \sum_{j=1}^p x_{ij}\beta_j \right)^2 + \lambda \sum_{j=1}^p |\beta_j|^2 \\
&= \arg\min_\beta \mathrm{MSE}(\beta) + \lambda \sum_{\mathrm{j}=1}^{\mathrm{p}} |\beta_\mathrm{j}|^2 \\
&= \arg\min_\beta \sum_{i=1}^n \left(y_i - \beta_0 - \sum_{j=1}^p x_{ij}\beta_j \right)^2 + n\lambda \sum_{j=1}^p |\beta_j|^2 \\
&= \arg\min_\beta \mathrm{RSS}(\beta) + \mathrm{n}\lambda \sum_{\mathrm{j}=1}^{\mathrm{p}} |\beta_\mathrm{j}|^2
\end{aligned}
$$

Your Turn #1 : Ridge Regression

1. What happens when $\lambda = 0$?
2. What happens when $\lambda \uparrow \infty$?
3. Why is it important to scale the predictor variables?

vspace*{.5in}

4.1 Ridge Regression - Estimation

The usual set-up is to center and scale the predictor variables $\mathbf{X}$ and center the response $\mathbf{y}$ (so $\bar{y} = 0$, $\bar{\mathbf{x}}_j = 0$, and $\mathbf{x}_j^\mathsf{T}\mathbf{x}_j = c$, $\forall j$).

- This is assumed in the equations below
  - By centering all variables and response, the intercept is 0 (i.e., $\hat{\beta}_0 = 0$) and can be excluded.
  - This is useful because the intercept should **not** be penalized
- The scaling of the predictor variables is important because it forces all variables/features to be treated equally.
  - Think of the penalty term
  - Also, we will see that the ridge penalty forces coefficients from correlated predictors towards each other

Center and Scale

Consider the advertising data

```
#-- Unmodified model
lm.all = advert %>%
  lm(formula = sales ~ TV + radio + newspaper, data = .)   # fit model

lm.all %>% tidy()  # get tidy output
#> # A tibble: 4 x 5
#>   term        estimate std.error statistic  p.value
#>   <chr>          <dbl>     <dbl>     <dbl>    <dbl>
#> 1 (Intercept)  2.94      0.312       9.42  1.27e-17
#> 2 TV           0.0458    0.00139    32.8   1.51e-81
#> 3 radio        0.189     0.00861    21.9   1.51e-54
#> 4 newspaper   -0.00104   0.00587    -0.177 8.60e- 1
```

4.1.1 Center X:

$$\tilde{x}_{ij} = x_{ij} - \bar{x}_j$$

- This changes the intercept: $\tilde{\beta}_0 = \bar{y}$

```
#-- Center X
advert %>%
  mutate_at(vars(-sales), function(x) x - mean(x)) %>%    # center all X's
  lm(formula = sales ~ TV + radio + newspaper, data = .) %>%
  tidy()
#> # A tibble: 4 x 5
#>   term        estimate std.error statistic   p.value
#>   <chr>          <dbl>     <dbl>     <dbl>     <dbl>
#> 1 (Intercept) 14.0       0.119     118.     9.15e-184
#> 2 TV           0.0458    0.00139    32.8    1.51e- 81
#> 3 radio        0.189     0.00861    21.9    1.51e- 54
#> 4 newspaper   -0.00104   0.00587    -0.177 8.60e-  1

#-- Check
mean(advert$sales) # mean sales
#> [1] 14.02
```

4.1.2 Scale X:

$$\tilde{x}_{ij} = \frac{x_{ij}}{s_j}$$

- This changes the slope: $\tilde{\beta}_j = s_j \hat{\beta}_j$

```
#-- Scale X
advert %>%
  mutate_at(vars(-sales), function(x) x/sd(x)) %>%  # scale (by std.dev)
  lm(formula = sales ~ TV + radio + newspaper, data = .) %>%
  tidy()
#> # A tibble: 4 x 5
#>   term        estimate std.error statistic  p.value
#>   <chr>          <dbl>     <dbl>     <dbl>    <dbl>
#> 1 (Intercept)   2.94       0.312      9.42  1.27e-17
#> 2 TV            3.93       0.120     32.8   1.51e-81
#> 3 radio         2.80       0.128     21.9   1.51e-54
#> 4 newspaper    -0.0226     0.128     -0.177 8.60e- 1

#-- Check
(sd.x = advert %>% summarize_at(vars(-sales), sd) %>% as.matrix())
#>          TV radio newspaper
#> [1,] 85.85 14.85     21.78

(coef(lm.all) * c(1, sd.x)) %>% round(2)
#> (Intercept)          TV        radio   newspaper
#>        2.94        3.93         2.80       -0.02
```

4.1.3 Center Y:

$$\tilde{y}_i = y_i - \bar{y}$$

- This changes the slope: $\tilde{\beta}_0 = \hat{\beta}_0 - \bar{y}$

– $\tilde{\beta}_0 = 0$ if $\tilde{x}$ and $\tilde{y}$ are centered

```
#-- Center Y
advert %>%
  mutate(sales = sales - mean(sales)) %>%  # center y
  lm(formula = sales ~ TV + radio + newspaper, data = .) %>%
  tidy()
#> # A tibble: 4 x 5
#>   term         estimate std.error statistic  p.value
#>   <chr>           <dbl>     <dbl>     <dbl>    <dbl>
#> 1 (Intercept) -11.1       0.312     -35.5   2.28e-87
#> 2 TV            0.0458    0.00139    32.8   1.51e-81
#> 3 radio         0.189     0.00861    21.9   1.51e-54
#> 4 newspaper    -0.00104   0.00587    -0.177 8.60e- 1

#-- check
coef(lm.all)[1] - mean(advert$sales)
#> (Intercept)
#>      -11.08

#-- Center Y and X
advert %>%
  mutate(sales = sales - mean(sales)) %>%  # center y
  mutate_at(vars(-sales), function(x) x - mean(x)) %>%    # center all X's
  lm(formula = sales ~ TV + radio + newspaper, data = .) %>%
  tidy()
#> # A tibble: 4 x 5
#>   term         estimate std.error statistic  p.value
#>   <chr>           <dbl>     <dbl>     <dbl>    <dbl>
#> 1 (Intercept)  8.74e-17   0.119    7.33e-16 1.00e+ 0
#> 2 TV           4.58e- 2   0.00139  3.28e+ 1 1.51e-81
#> 3 radio        1.89e- 1   0.00861  2.19e+ 1 1.51e-54
#> 4 newspaper   -1.04e- 3   0.00587 -1.77e- 1 8.60e- 1
```

The optimization function can be written:

$$\begin{aligned}\hat{\beta}_\lambda^{\text{ridge}} &= \arg\min_\beta \ \ell(\beta) + \lambda P(\beta) \\ &= \arg\min_\beta \ J(\beta; \lambda)\end{aligned}$$

where

$$J(\beta) = (\mathbf{y} - \mathbf{X}\beta)^\mathsf{T}(\mathbf{y} - \mathbf{X}\beta) + \lambda\beta^\mathsf{T}\beta$$

And the solution must satisfy

$$\frac{\partial J(\beta)}{\partial\beta} = \frac{\partial l(\beta)}{\partial\beta} + \lambda\frac{\partial P(\beta)}{\partial\beta} = \mathbf{0}$$

For ridge regression, this becomes

Ridge Regression Solution

The Matrix Cookbook has some common matrix and vector derivative expressions.

4.2 Ridge Regression Properties

$$\hat{\beta}^{\text{ridge}} = \left(\mathbf{X}^\mathsf{T}\mathbf{X} + \lambda \mathbf{I_p}\right)^{-1} \mathbf{X}^\mathsf{T}\mathbf{Y}$$

- Ridge regression always works, even when $\mathbf{X}$ is not full rank because $\mathbf{X}^\mathsf{T}\mathbf{X} + \lambda \mathbf{I_p}$ is always invertible for $\lambda > 0$
- For $0 < \lambda < 2\sigma^2 / \sum_j |\beta_j|^2$, ridge regression has a lower mean square prediction error than least squares (Theobald 1974)!
- (Quasi) Bayesian Interpretation: If $\beta \sim N(0, \tau^2 \mathbf{I_p})$ is the prior distribution, and τ and the standard deviation σ are assumed known, then the posterior mode (and hence mean since Gaussian) of β, given the data, is

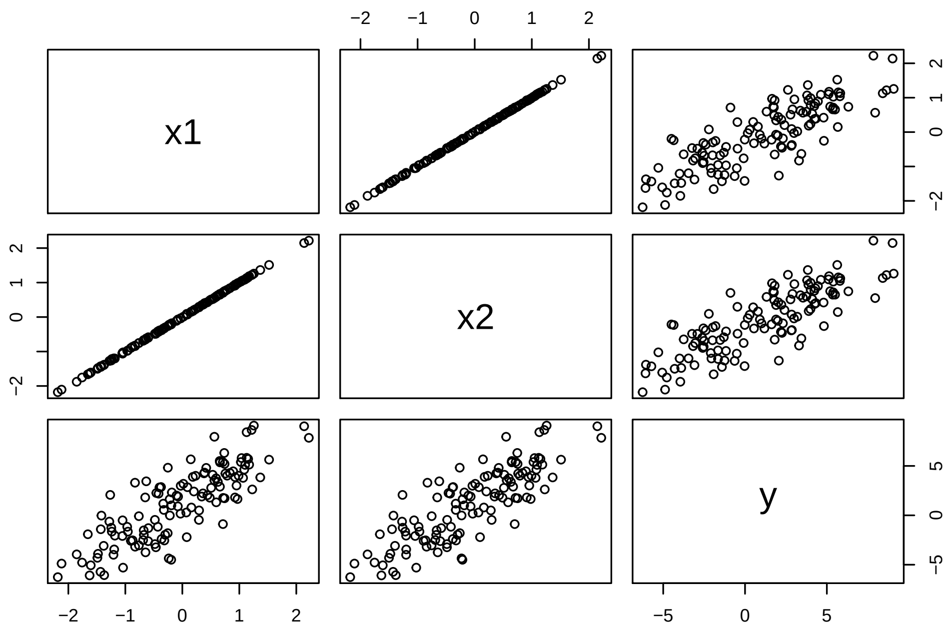
$$\mathrm{E}[\beta | \mathcal{D}] = \left(\mathbf{X}^\mathsf{T}\mathbf{X} + \frac{\sigma^2}{\tau^2}\mathbf{I_p}\right)^{-1} \mathbf{X}^\mathsf{T}\mathbf{Y}$$

  which is equivalent to using $\lambda = \sigma^2/\tau^2$

4.3 Ridge Regression Example

Consider a problem with strong multicollinearity:

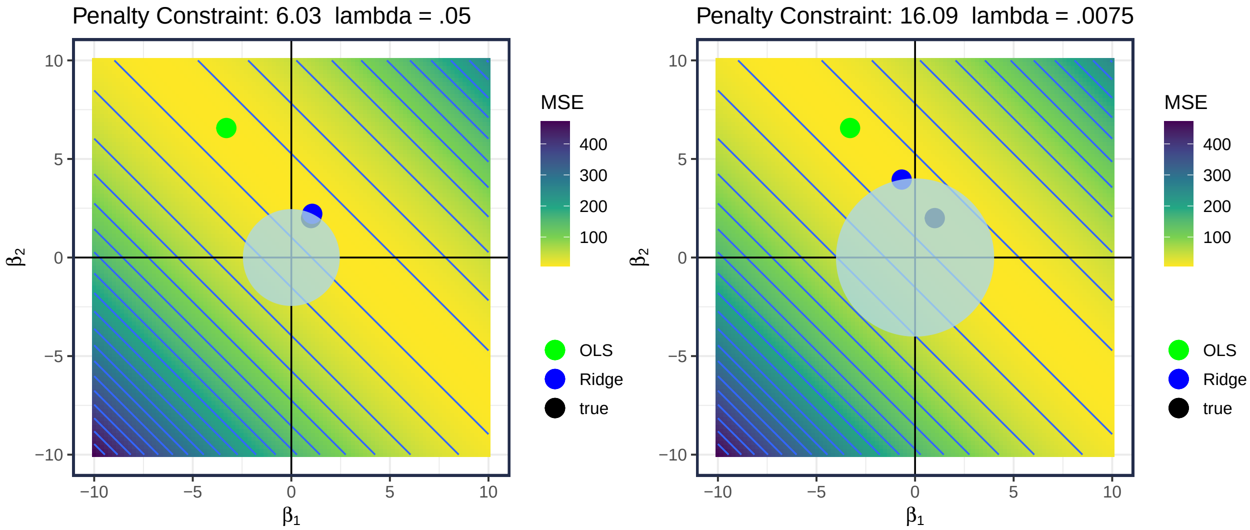
```
#-- Generate Data
set.seed(10)
n = 125
x1 = rnorm(n)
x2 = rnorm(n, mean=x1, sd=.01)
cor(x1,x2)                              # strong correlation
#> [1] 0.9999
y = rnorm(n, mean=1+1*x1+2*x2, sd=2)   # f(x) = 1 + 1x_1 +2x_2

#-- Pairs Plot
pairs(cbind(x1, x2, y))
```




```
#-- OLS estimation
fit.lm = lm(y~x1 + x2)
coef(fit.lm)
#> (Intercept)          x1          x2
#>       1.380      -3.296       6.569
```

predictor	true	ols	ridge(lambda=.05)
(Intercept)	1	1.38	1.38
x1	1	-3.30	1.06
x2	2	6.57	2.22

- Notice that the OLS coefficients have negative signs and large magnitude but a small constraint (penalty) produces a much closer result.
- That is, a small ridge penalty controlled the high variance.



4.4 Ridge Regression Complexity and Tuning parameter λ

The *tuning parameter* for a ridge regression model is the λ that controls the strength of penalty

$$\hat{\beta}_{\lambda}^{\text{ridge}} = \arg\min_{\beta}\ \text{RSS}(\beta) + \lambda \sum_{j=1}^{p} |\beta_j|^2$$

- $\lambda = 0$ gives β^{LS}
- $\lambda = \infty$ gives $\beta_j = 0 \quad j = 1, \ldots, p$
- As λ goes up, variance decreases and bias increases.

The *effective degrees of freedom*, $\text{df}(\lambda)$ is the trace of the hat matrix, H_λ

EDOF Calculation

4.5 Ridge Regression Solution Paths

Ridge Regression introduces a set of models indexed by λ

- $\lambda = 0$ gives β^{LS}

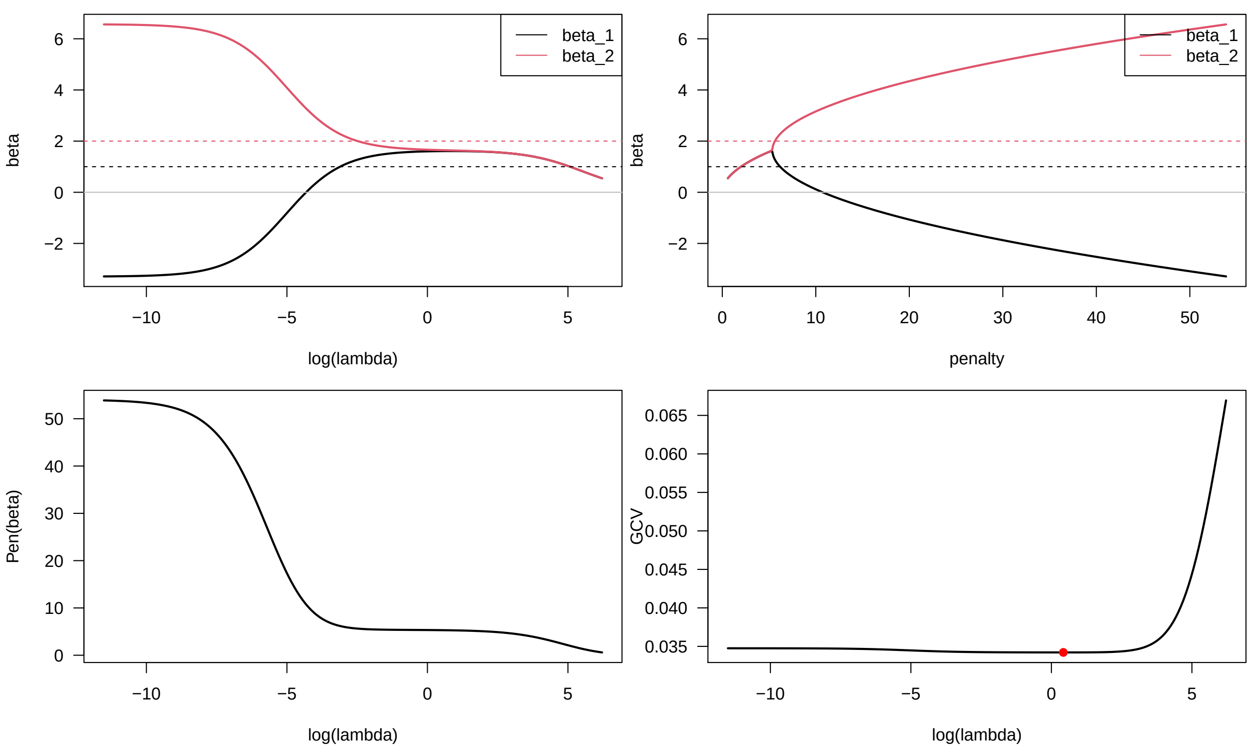
- $\lambda = \infty$ gives $\beta_j = 0 \quad j = 1, \ldots, p$
- As λ goes up, variance decreases and bias increases.

It can be illustrative to plot the *coefficient path* against:

- λ or $\log(\lambda)$
- $P(\hat{\beta}_j(\lambda)) = \sum_{j=1}^{p} |\hat{\beta}_j(\lambda)|^2$
- $df(\lambda)$ (effective degrees of freedom)

4.6 Ridge Path Analysis for Correlated Data Example

lam	intercept	x1	x2	penalty	GCV
500.0000	1.198	0.5465	0.546	0.5968	0.0669
84.1791	1.309	1.2262	1.226	3.0060	0.0388
14.1722	1.361	1.5493	1.552	4.8104	0.0344
2.3860	1.373	1.6104	1.636	5.2704	0.0342
0.4017	1.375	1.5567	1.716	5.3664	0.0342
0.0676	1.376	1.1978	2.079	5.7547	0.0343
0.0114	1.377	-0.1789	3.454	11.9659	0.0344
0.0019	1.379	-2.1908	5.465	34.6616	0.0346
0.0003	1.380	-3.0677	6.341	49.6140	0.0347
0.0001	1.380	-3.2564	6.529	53.2334	0.0348

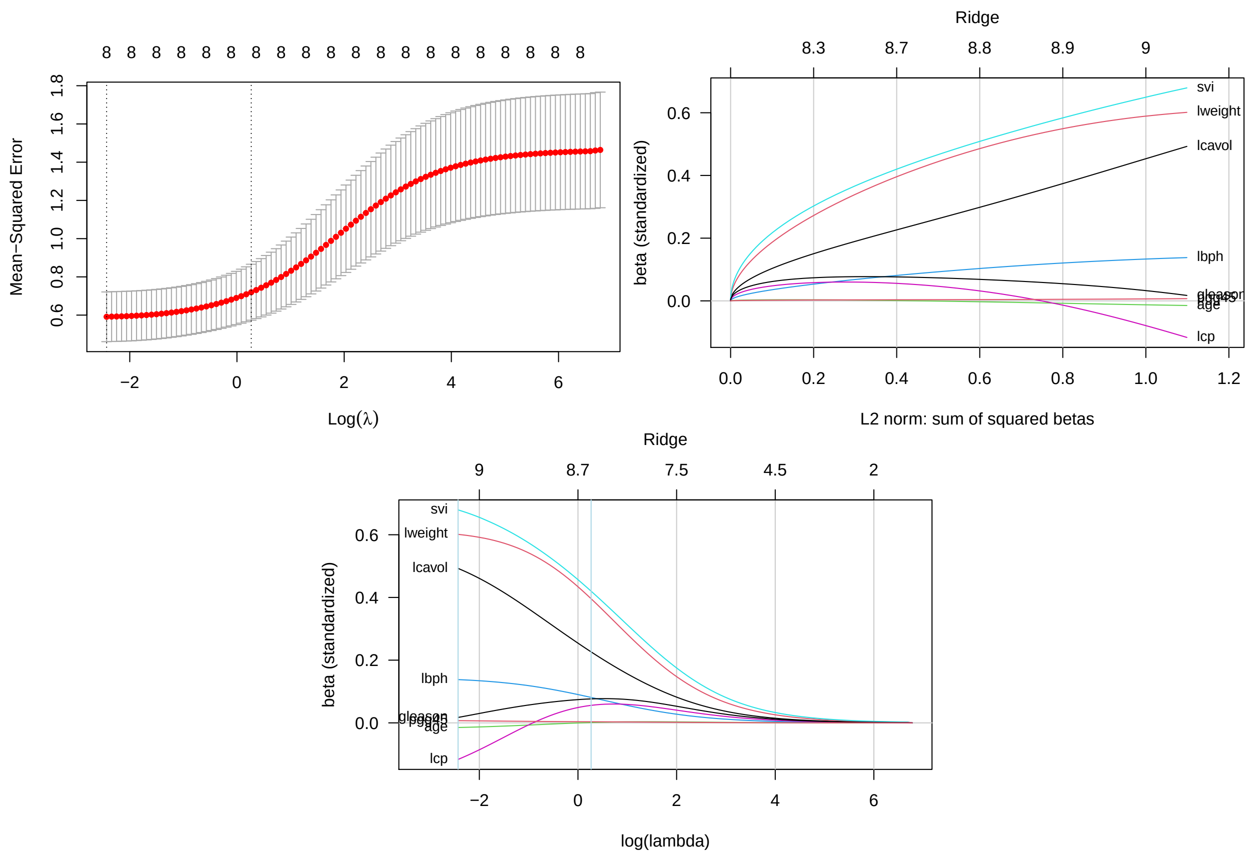


And we find that the λ that gives the lowest *generalized cross-validation* error is $\lambda^{\text{ridge}} = 1.53$ ($\log \lambda^{\text{ridge}} = 0.42$), which gives estimated coefficients of

predictor	ridge
Intercept	1.374
beta1	1.608
beta2	1.649

Note: Ridge tends to shrink correlated predictors together.

4.6.1 Ridge Regression for Prostate Data



Cross-validation suggests using $\lambda = 0.088$ (or $\log \lambda = -2.432$).

This λ corresponds to (scaled) coefficients:

predictor	ols	ridge
(Intercept)	2.45	0.10
lcavol	0.72	0.49
svi	0.31	0.68
lweight	0.29	0.60
pgg45	0.28	0.01
lbph	0.21	0.14
gleason	-0.02	0.02
age	-0.14	-0.01
lcp	-0.29	-0.12

Note: these coefficients are based on the scaled input data (each predictor has sample variance of 1).

4.7 Ridge Regression functions in R

There are several R packages that have functions for ridge regression.

- The `MASS` package has the function `lm.ridge()`
- The `glmnet` package has the functions `glmnet()` and `cv.glmnet()`
  - Use `alpha=0` for ridge regression
  - We will use this function for the *lasso* and *elasticnet* models
  - Note: input matrices; does not handle formulas! But check out the `glmnetUtils` R package which does allow formulas.
  - Check out An Introduction to `glmnet` for more information on using `glmnet` package for ridge, lasso, and elasticnet penalized regression.
- All of these functions will center, scale, and transform the output back to the original units, so you do not need to do any of the scaling yourself

5 Lasso

5.1 The Lasso

For lasso regression

$$
\begin{aligned}
l(\beta) &= \sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j \right)^2 \\
P(\beta) &= \sum_{j=1}^{p} |\beta_j| \qquad \text{(Notice that } \beta_0 \text{ is not penalized)}
\end{aligned}
$$

The lasso solution becomes:

$$
\hat{\beta}^{\text{ridge}} = \arg\min_{\beta} \; \sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j \right)^2 + \lambda \sum_{j=1}^{p} |\beta_j|
$$

Why is it important to scale the predictor variables?

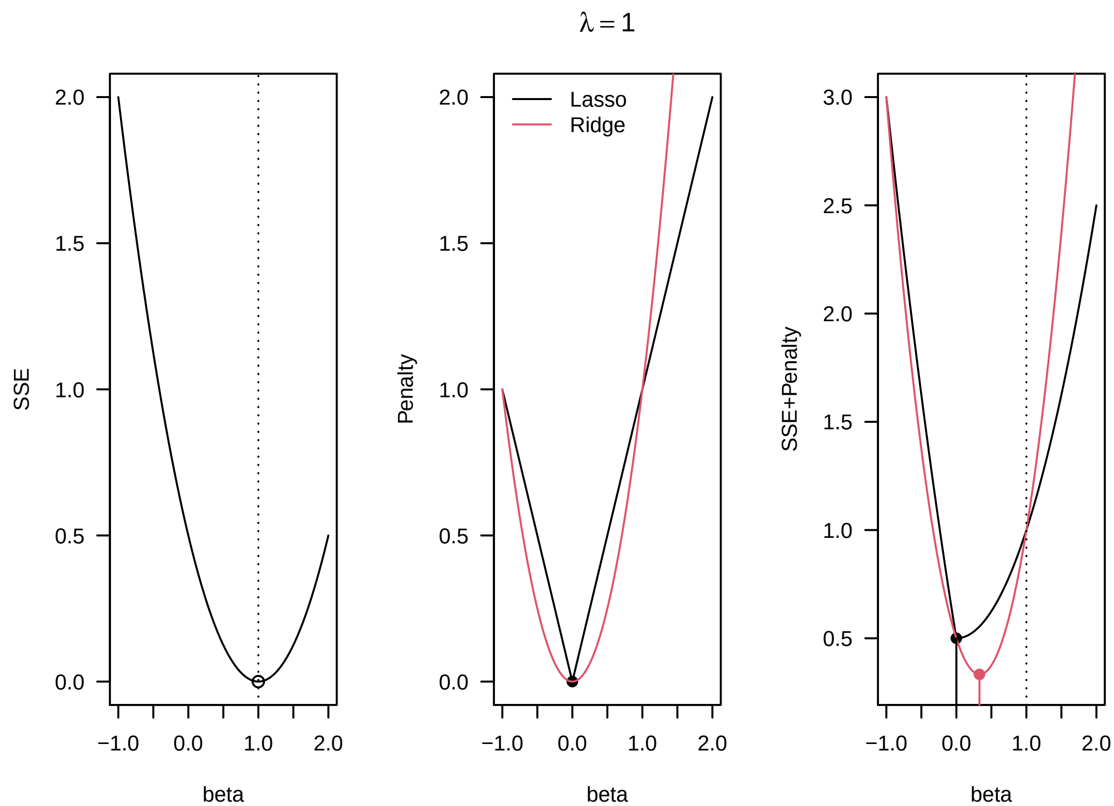
5.2 Lasso Penalty

- By using a L_1 penalty, lasso penalty can shrink some coefficients all the way to 0 (unlike the ridge penalty)
- This effectively removes predictors from the model (like the stepwise procedures), but in a type of continuous fashion
- Lasso stands for "Least Absolute Shrinkage and Selection Operator"

5.3 Example of 1D Lasso Selection

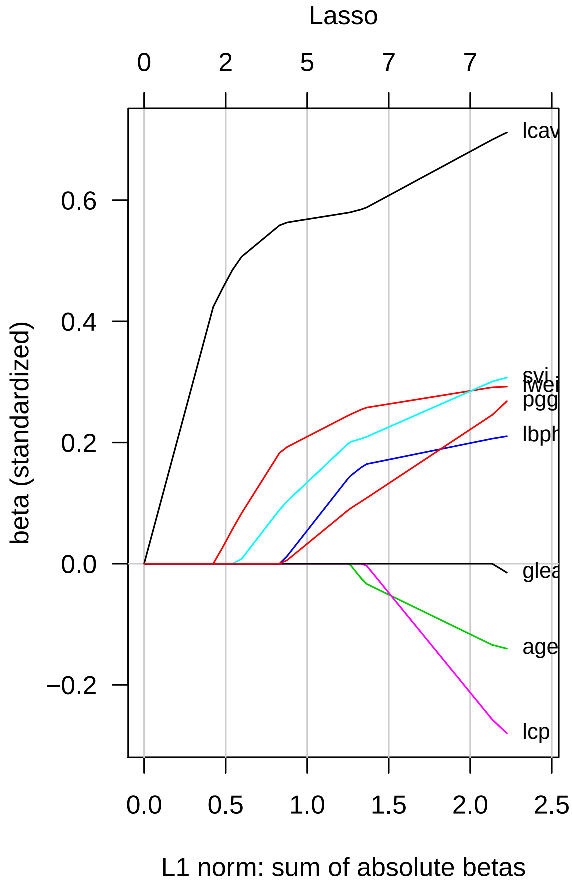
Suppose the simplified setting of fitting a loss function of $l(\beta) = \frac{1}{2}(1-\beta)^2$.

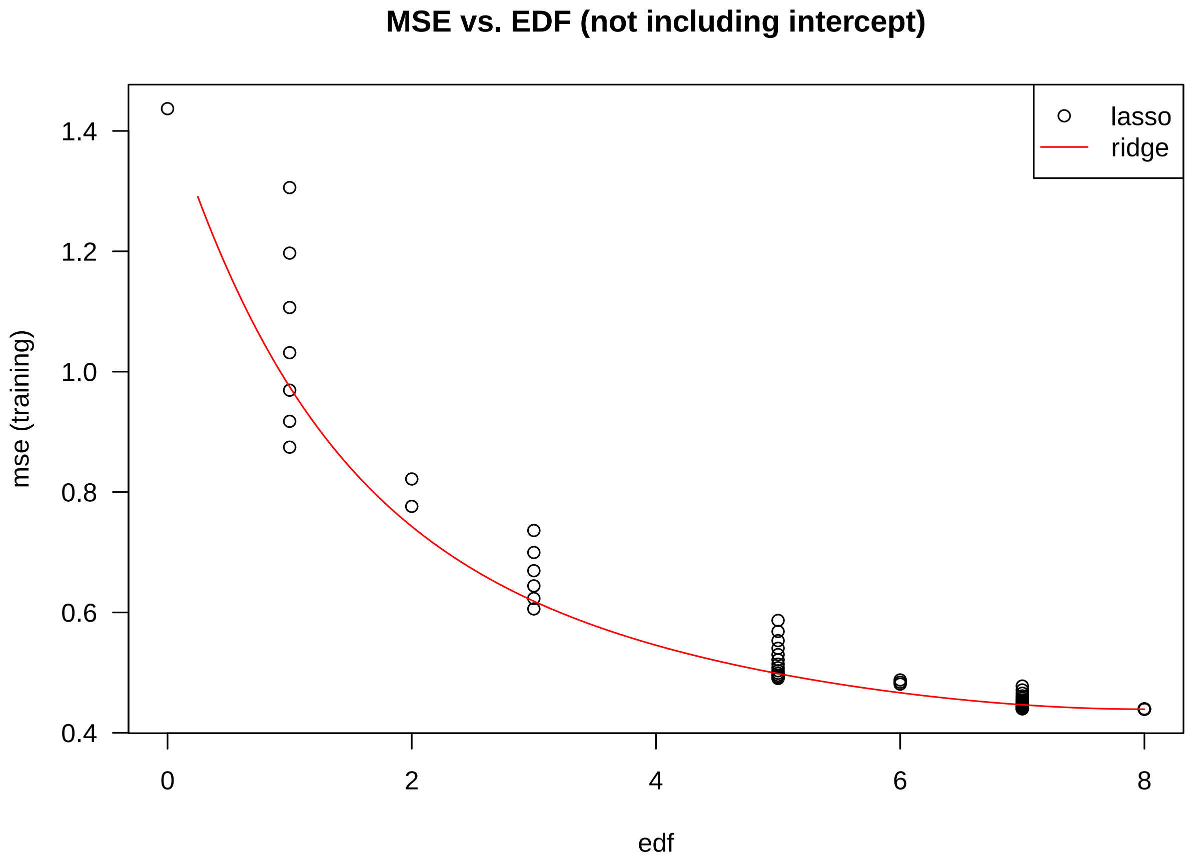
- This loss is the squared deviation from 1.
- The lasso penalty is $|\beta|$.
- The objective function is $l(\beta) + \lambda|\beta|$



5.4 Comparing Lasso and Ridge Regression

Prostate Cancer Data from ESL book: Figs 3.8, 3.10 and Table 3.3

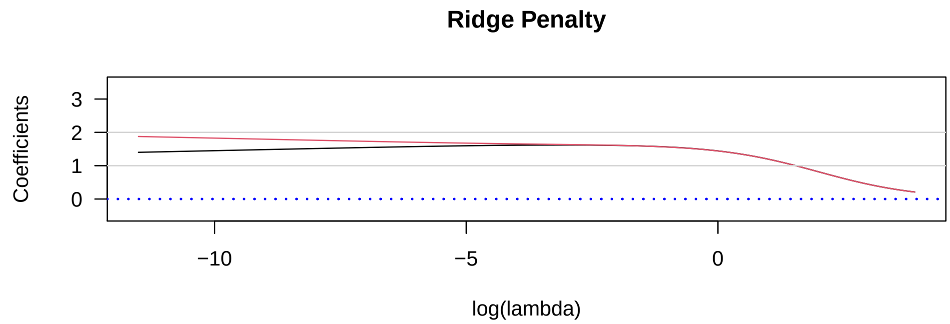

Lasso
0
2
5
7
7
lcav
svi
lwei
pgg
lbph
glea
age
lcp
beta (standardized)
0.6
0.4
0.2
0.0
−0.2
0.0
0.5
1.0
1.5
2.0
2.5
L1 norm: sum of absolute betas
Ridge
3.3
5.3
6.6
7.6
lcav
svi
lwei
pgg
lbph
glea
age
lcp
beta (standardized)
0.6
0.4
0.2
0.0
−0.2
0.0
0.2
0.4
0.6
0.8
1.0
L2 norm: sum of squared betas

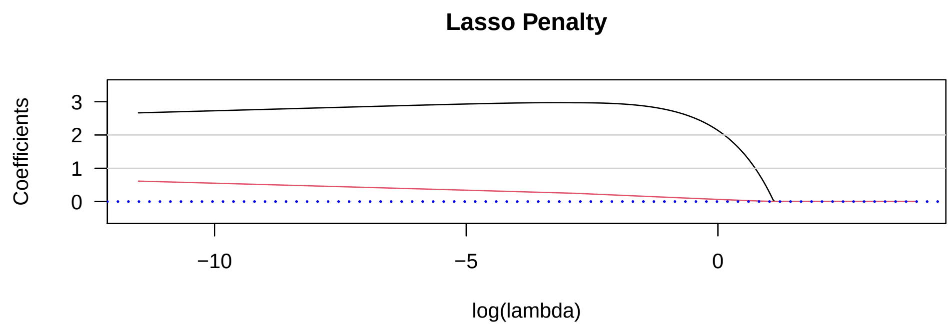


See lasso.R for the code that produced the plots.

5.4.1 Example with Strong Correlation

$$\boxed{Y = 1 + 1X_1 + 2X_2 + \epsilon}$$





predictor	true	ols	ridge	lasso
(Intercept)	1	1.38	1.36	1.37
x1	1	-3.30	1.58	3.15
x2	2	6.57	1.57	0.11

Ridge and Lasso using λ_{min} from cross-validation.

Ridge tends to shrink correlated predictors together Lasso tends to choose one.

5.5 Effective Number of Parameters for Lasso

- Unlike ridge regression, the lasso is *not* a linear smoother. There is no way to write $\hat{\mathbf{y}} = \mathbf{H}\mathbf{y}$.
- Thus, estimating the effective degrees of freedom is not based on trace of hat matrix.
- It turns out that the number of non-zero coefficients is a decent approximation of the effective number of parameters
- We can use this value ($df = \sum_j \mathbb{1}(|\beta_j| > 0)$) in AIC/BIC/GCV for selecting λ
  - Note: the df is not continuous in λ, so the min SSE model would have smallest λ within the set with $df = k$

5.6 Elastic Net

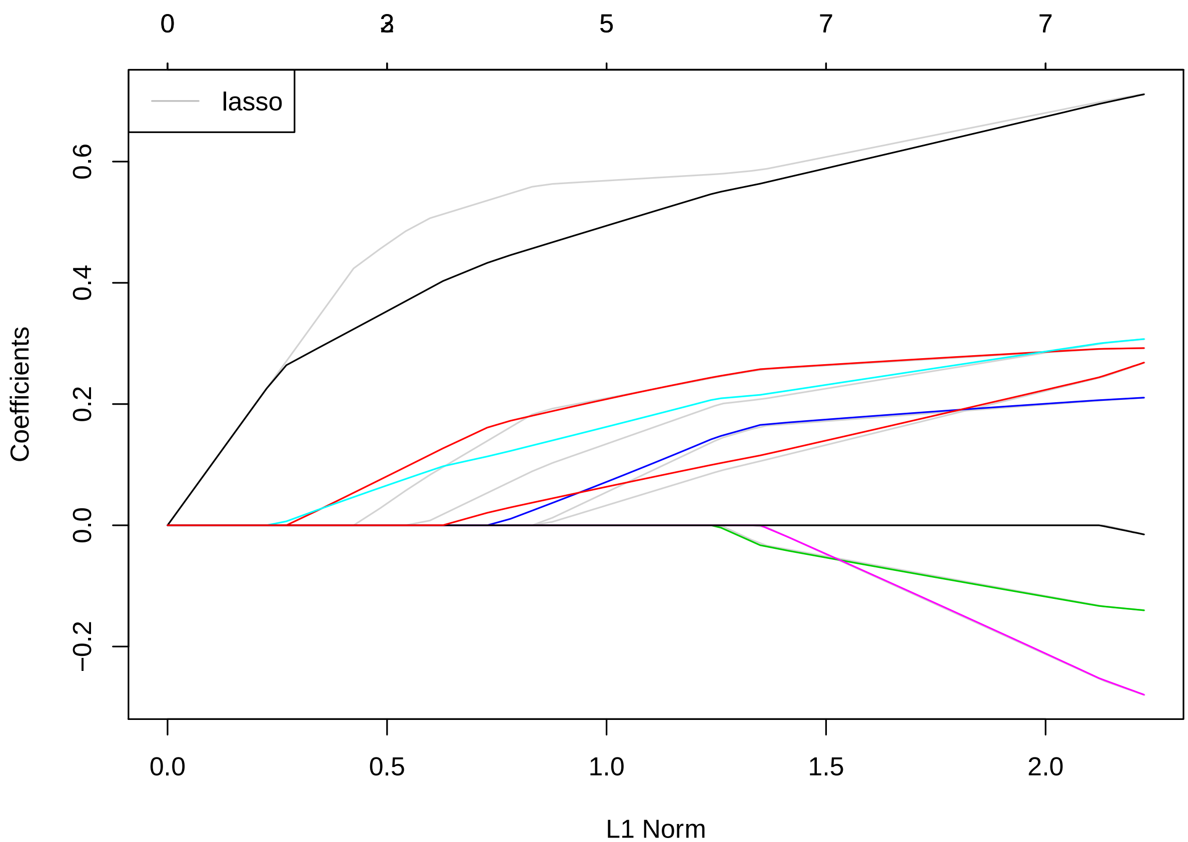
The Elastic Net Penalty can help with selection (like lasso) and shrinks together correlated predictors (like ridge).

$$P(\beta, \alpha) = \sum_{j=1}^{p} \alpha\beta_j^2 + (1-\alpha)|\beta_j| \qquad \text{Eq 3.54 on pg 73 of ESL}$$

$$P(\beta, \alpha) = \sum_{j=1}^{p} \frac{(1-\alpha)}{2}\beta_j^2 + \alpha|\beta_j| \qquad \text{glmnet R package}$$

5.6.1 Comparing Elastic Net to Lasso and Ridge

Elastic Net with $\alpha = 0.5$



5.7 Categorical Predictors in Penalized Regression

1. How does lasso/ridge treat categorical predictors?
2. How does lasso/ridge treat interaction terms?
3. How does lasso/ridge treat basis expansions of a single variable, e.g. polynomial?

5.7.1 Dummy Coding and Model Matrix

See the R formula interface document for details on using the `model.matrix()` to convert a data frame to a model matrix for use in `glmnet()` family of functions.

5.8 Group Lasso

- L groups of predictors
  - categorical variable with 3 levels will be in a group of 3 predictors
- Let X_l be $n \times p_l$ matrix of group l predictors
- β_l is $p_l \times 1$ group coefficients

$$J(\beta) = \ell(\beta) + P(\beta, \lambda)$$

$$\ell(\beta) = \left\| Y - \beta_0 1 - \sum_{l=1}^{L} X_l \beta_l \right\|_2^2$$

$$P(\beta, \lambda) = \sum_{l=1}^{L} \sqrt{p_l} \, \|\beta_l\|_2$$

6 More Resources

- glmnet tutorial
- broom tutorial
  - Using broom with glmnet