STAT 224 Lecture 18

Ridge and Lasso Regressions

Yibi Huang

Bias-Variance Tradeoff

In Chapter 11 Variable Selections (L17.pdf), we showed that

$$\begin{aligned}\text{MSE}(\hat{\beta}_j) &= \mathrm{E}\left[(\hat{\beta}_j - \beta_j)^2\right] \\ &= \mathrm{E}\left[(\hat{\beta}_j - \mathrm{E}[\hat{\beta}_j])^2\right] + (\mathrm{E}[\hat{\beta}_j] - \beta_j)^2 \\ &= (\text{Variance of } \hat{\beta}_j) + (\text{Bias of } \hat{\beta}_j)^2\end{aligned}$$

- OLS estimates for β_j's are unbiased
- However, the variances of OLS estimates $\hat{\beta}_j$ can be large when
  - the number of predictors is large, or when
  - the predictors are multicollinear
- Is there a way to reduce the variance of $\hat{\beta}_j$, possibly at the cost of increased bias?

Shrinkage Estimates (aka. Regularization)

- OLS estimates $\hat{\beta}_j$ have no upper bound, and hence is susceptible to very high variance
- By **shrinking** the OLS estimates $\hat{\beta}_j$ toward 0, we can often substantially reduce the variance at the cost of a negligible increase in bias, substantially improving the accuracy of prediction for future observations
- **Shrinkage** is called “Regularization” in Machine Learning
- Two common shrinkage estimates are
  - Ridge regression
  - Lasso (Least Absolute Shrinkage and Selection Operator)

OLS v.s. Ridge v.s. Lasso

Ordinary Least Square minimizes:

$$\sum_{i=1}^{n}(y_i - \hat{\beta}_0 - \hat{\beta}_1 x_{i1} - \ldots - \hat{\beta}_p x_{ip})^2$$

Ridge Regression minimizes:

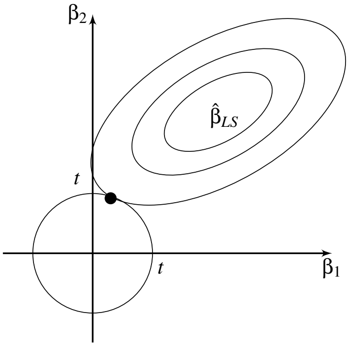
$$\sum_{i=1}^{n}(y_i - \hat{\beta}_0 - \hat{\beta}_1 x_{i1} - \ldots - \hat{\beta}_p x_{ip})^2 \quad \text{with the constraint} \ \sum_{j=1}^{p} \hat{\beta}_j^2 \leq t$$

Lasso mininizes:

$$\sum_{i=1}^{n}(y_i - \hat{\beta}_0 - \hat{\beta}_1 x_{i1} - \ldots - \hat{\beta}_p x_{ip})^2 \quad \text{with the constraint} \ \sum_{j=1}^{p} \left|\hat{\beta}_j\right| \leq t$$

Note there is no constraint placed on the magnitude of the intercept $\hat{\beta}_0$.

Geometric Illustration of Ridge and Lasso Estimates



- Ellipses are the contours of $\sum_{i=1}^{n}(y_i - \hat{\beta}_0 - \hat{\beta}_1 x_{i1} - \hat{\beta}_2 x_{i2})^2$, which centered at the OLS estimates $(\hat{\beta}_{1,OLS}\,,\ \hat{\beta}_{2,OLS}\,)$.
- (Left) Ellipse intersects the circle of radius t at the Ridge estimate.
- (Right) Ellipse intersects the square $(|\hat{\beta}_1| + |\hat{\beta}_2| < t)$ at the Lasso estimate

Equivalent Forms of Ridge and Lasso

By the Lagrange multiplier methods, minimizing $\sum_{i=1}^{n}(y_i - \hat{\beta}_0 - \hat{\beta}_1 x_{i1} - \ldots - \hat{\beta}_p x_{ip})^2$ under the constraints

$$\sum_{j=1}^{p} \hat{\beta}_j^2 \leq t \quad \text{or} \quad \sum_{j=1}^{p} \left|\hat{\beta}_j\right| \leq t$$

is equivalent to

Ridge Regression, minimizing

$$\sum_{i=1}^{n}(y_i - \hat{\beta}_0 - \hat{\beta}_1 x_{i1} - \ldots - \hat{\beta}_p x_{ip})^2 + \lambda \sum_{j=1}^{p} \hat{\beta}_j^2$$

Lasso, minimizing:

$$\sum_{i=1}^{n}(y_i - \hat{\beta}_0 - \hat{\beta}_1 x_{i1} - \ldots - \hat{\beta}_p x_{ip})^2 + \lambda \sum_{j=1}^{p} \left|\hat{\beta}_j\right|$$

Tuning Parameter λ or t

Both Ridge and Lasso have a **tunning parameter** λ (or t)

- The Ridge estimates $\hat{\beta}_{j,\lambda,Ridge}$'s and Lasso estimates $\hat{\beta}_{j,\lambda,Lasso}$ depend on the value of λ (or t)

λ (or t) is the **shrinkage parameter** that controls the size of the coefficients

- As $\lambda \downarrow 0$ or $t \uparrow \infty$, the Ridge and Lasso estimates become the OLS estimates
- As $\lambda \uparrow \infty$ or $t \downarrow 0$, Ridge and Lasso estimates shrink to 0 (intercept only model)

Ridge and Lasso Estimates Are NOT Scale Invariant

Say we change the unit of a predictor X_j from inches to feet

$$X'_j = X_j/12$$

its coefficient would be scaled as

$$\beta'_j = 12\beta_j$$

so that the product $\beta'_j X'_j = \beta_j X_j$ stays unchanged.

However, the Ridge and Lasso estimates are not scaled accordingly

$$\hat{\beta}'_{j,\lambda,Ridge} \neq 12\hat{\beta}_{j,\lambda,Ridge}, \quad \hat{\beta}'_{j,\lambda,Lasso} \neq 12\hat{\beta}_{j,\lambda,Lasso}$$

since large β's are penalized

Must Standardize Predictors Before Applying Ridge and Lasso

As Ridge and Lasso estimates are not scale invariant, by convention, we **standardize** all predictors

$$Z_j = \frac{X_j - \overline{X}_j}{s_j}, \quad j = 1, \ldots, p,$$

where s_j is the sample SD of X_j. before applying Ridge and Lasso.

That is, all predictors X_j's in Ridge and Lasso regression are assumed to have mean 0 and variance 1.

Ridge Estimates Are Biased but Have Smaller Variance

- Recall OLS estimate for $\boldsymbol{\beta} = (\beta_0, \beta_1, \ldots, \beta_p)^T$ is $(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{Y}$
- One can show Ridge estimate for $\boldsymbol{\beta}$ is $(\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I}_p)^{-1}\mathbf{X}^T\mathbf{Y}$
  - Keep in mind that $\mathbf{X}$ is standardized
    that each predictor has mean 0 and variance 1
- Expected value for the Ridge estimate for $\boldsymbol{\beta}$ can be shown to be

$$(\mathbf{I}_p + \lambda\mathbf{X}^T\mathbf{X})^{-1}\boldsymbol{\beta} \neq \boldsymbol{\beta}$$

- If all predictors are standardized and uncorrelated,

$$\hat{\beta}_{j,\lambda,Ridge} = \frac{1}{1+\lambda}\hat{\beta}_{j,OLS}$$

- Smaller variance than OLS estimates,
- Variance of $\hat{\beta}_{j,\lambda,Ridge}$ is much smaller than $\hat{\beta}_{j,OLS}$ when the data have **multicollinearity** problem

Properties of Lasso Estimates

- No close form formula for the Lasso estimates
- Also biased (toward 0)
- Smaller variance than OLS estimates
- NOT perform as well as Ridge when data have **multicollinearity** problem
- Greatest advantage of Lasso: **Sparsity** (See next page)

Sparsity of Lasso Estimates

- In a model with many predictors

$$Y = \beta_0 + \beta_1 X_1 + \ldots + \beta_p X_p + \varepsilon$$

  we may believe many of the β_j's are actually 0.
- Hence, we seek a set of sparse solutions
- Lasso estimates will set some coefficients exactly equal to 0 when λ is large (or when t is small)

So the LASSO will perform model selection for us!

How to Choose λ?

- We need a disciplined way of choosing λ
- Obviously want to choose λ that minimizes the mean squared error
- Issue is part of the bigger problem of **variable selection**

Choosing λ Using Cross-Validation

- If we have a good model, it should predict well when we have new data
- Data are hence split into 2 parts — **training data** and **test data**
- For each λ, use the training set to fit (train) a model and than use the model to predict values in the test set and compute the rooted mean square error (RMSE)

$$\sqrt{\sum_{\text{test data}} (y_i - \hat{y}_i)^2/n}, \quad \text{where } n = \text{size of the test data}$$

- Choose the λ that has the smallest RMSE
- The training set and test set should be chosen randomly
  - May split the whole data into several different training set and test set and compute the mean of the RMSE for different splits

Ridge and Lasso Regression in R

Ridge Regression in R

Recall the Equal Educational Opportunity (EEO) Data in the slides `L16.pdf`.

Data: http://www.stat.uchicago.edu/~yibi/s224/data/P236.txt

- `ACHV`: Student achievement index (higher values are better)
- `FAM`: Faculty credentials index
- `PEER`: the influence of their peer group in the school
- `SCHOOL`: School facility/resource index

```
EEO = read.table("P236.txt", h=T)
```

Ridge Regression in R

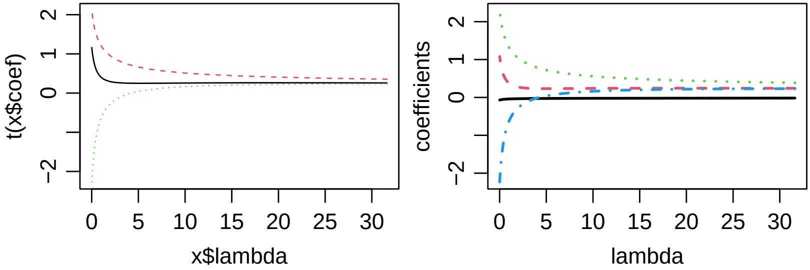
The `lm.ridge()` function in the `MASS` library can perform the Ridge Regression.

The `lambda` (λ) value(s) must be specified. The following gives the Ridge estimates for the intercept β_0 and the coefficients β_j for `FAM`, `PEER`, and `SCHOOL` for $\lambda = 1$, 5, and 10 respectively.

```
library(MASS)
lm.ridge(ACHV ~ FAM + PEER + SCHOOL, data=EEO, lambda=c(1,5,10))
                   FAM   PEER   SCHOOL
 1 -0.04055 0.3769 1.3205 -0.62767
 5 -0.02708 0.2318 0.7230  0.04196
10 -0.02355 0.2384 0.5568  0.16240
```

We can try more values of `lambda` and plot how the coefficients shrink as `lambda` grows larger:

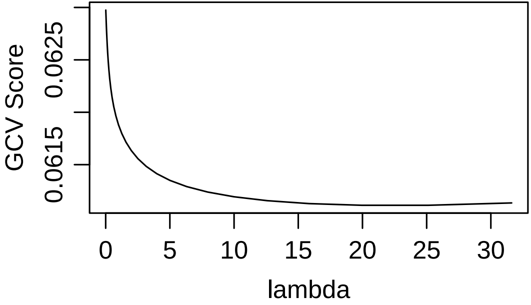
```
EEO.rg = lm.ridge(ACHV ~ FAM + PEER + SCHOOL, data=EEO,
                  lambda=10^seq(1.5, -2, by = -.1))
par(mai=c(0.6,0.6,0.01,0.01), mgp=c(2,0.7,0))
plot(EEO.rg)
matplot(EEO.rg$lambda, coef(EEO.rg), type = "l", lwd=2,
        xlab = "lambda", ylab = "coefficients")
```



Selecting λ Using Cross-Validation

For each λ, the `lm.ridge()` function computes the generalized cross-validation (GCV), similar to cross-validation using RMSE based on training data and test data.

```
par(mai=c(0.6,0.6,0.01,0.01), mgp=c(2,0.7,0))
plot(EEO.rg$lambda, EEO.rg$GCV, type = 'l',
     xlab = "lambda", ylab = "GCV Score")
```



The best `lambda` (among those `lambda`'s specified in `EEO.rg`) can be selected automatically to be 19.95.

```
select(EEO.rg)
modified HKB estimator is 0.3786
modified L-W estimator is 4.082
smallest value of GCV  at 19.95
```

Setting `lambda` at the optimal value 19.95 that minimize the GCV, the Ridge estimates for coefficients of the EEO data can be obtained as follows.

```
lm.ridge(ACHV ~ FAM + PEER + SCHOOL, data=EEO, lambda=19.95)
                    FAM       PEER    SCHOOL
-0.02034  0.24403  0.44264  0.21867
```

The Ridge estimates of the 3 coefficients are all positive, which makes more sense than the OLS estimates below that asserts better `SCHOOL` facility has a negative impact on students' performance.

```
lm(ACHV ~ FAM + PEER + SCHOOL, data=EEO)$coef
(Intercept)         FAM        PEER      SCHOOL
   -0.06996     1.10126     2.32206    -2.28100
```

The 3 Ridge estimates all have smaller magnitudes than corresponding OLS estimates.

Example (Meat Spectroscopy Data)

Data: 215 samples of finely chopped pure meat (Ch11 in *Linear Models with R* (2014) by J Faraway)

A Tecator near-infrared spectrometer was used to measure the spectrum of light transmitted through each sample of meat. The spectrum gives the absorbance at 100 wavelengths in the range 850-1050 nm. Since determining the fat content via analytical chemistry is time consuming, we would like to build a model to predict the fat content of new samples using the 100 absorbances which can be measured more easily.

```
meatspec = read.table(
  "http://www.stat.uchicago.edu/~yibi/s224/data/meatspec.txt",
  header=TRUE)
```

The first 100 variables are the 100 absorbances of different wave lengths. The 101th variable `fat` is the fat content determined via analytical chemistry.

Lasso in R

The `meatspec` data contain $n = 215$ observations but have $p = 100$ predictors.

Lasso is most useful for problems with much larger numbers of predictors like `meatspec`.

The `lars()` function in the `lars` library (installation required) can perform the Lasso Regression.

We first split the `meatspec` data into training data and test data

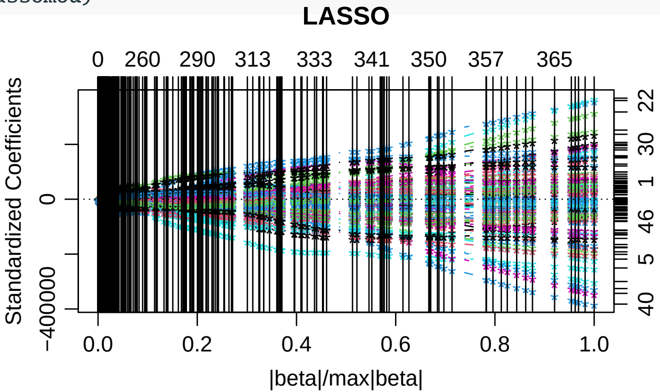
```
trainmeat = meatspec[1:172,]
testmeat = meatspec[173:215,]
```

We compute the Lasso fit for the training data:

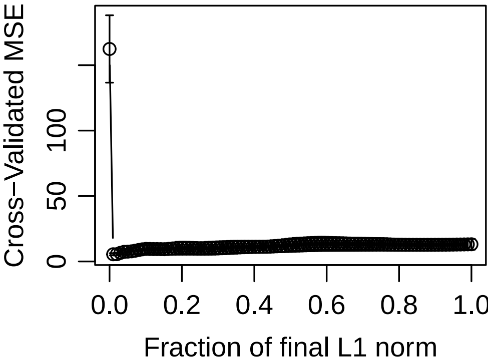
```
trainy = trainmeat$fat
trainx = as.matrix(trainmeat[,-101])
library(lars)
lassomod = lars(trainx,trainy)
```

Below is the plot of the estimated coefficients as a function of t.

```
plot(lassomod)
```



```r
par(mai=c(0.6,0.6,0.01,0.01), mgp=c(2,0.7,0))
set.seed(123)  # you can change the value within `set.seed()`
cvout = cv.lars(trainx, trainy)
```




```r
cvout$index[which.min(cvout$cv)]
[1] 0.0101
```

The best t selected by cross-validation is $t = 0.0101$.

Setting t at the optimal value 0.0101 determined by cross-validation, the Lasso estimates for coefficients of the meat data can be obtained as follows.

```
predlars = predict(lassomod, s=0.0101, type="coef", mode="fraction")
predlars$coef
      V1       V2       V3       V4       V5       V6       V7       V8
    0.00  -137.11     0.00     0.00     0.00     0.00     0.00     0.00
      V9      V10      V11      V12      V13      V14      V15      V16
    0.00     0.00     0.00   249.46     0.00     0.00     0.00     0.00
     V17      V18      V19      V20      V21      V22      V23      V24
    0.00     0.00     0.00     0.00     0.00     0.00     0.00  -266.12
     V25      V26      V27      V28      V29      V30      V31      V32
    0.00     0.00     0.00     0.00     0.00  1827.73     0.00     0.00
     V33      V34      V35      V36      V37      V38      V39      V40
    0.00 -4255.89     0.00     0.00  1931.28  1383.86     0.00     0.00
     V41      V42      V43      V44      V45      V46      V47      V48
    0.00 -1202.58     0.00     0.00   867.18   324.93   131.61     0.00
     V49      V50      V51      V52      V53      V54      V55      V56
-1102.57   -15.74     0.00     0.00     0.00   189.47     0.00     0.00
     V57      V58      V59      V60      V61      V62      V63      V64
```

We can see that only 20 coefficients have non-zero Lasso estimates.

```
sum(predlars$coef != 0)
[1] 20
```

Here are the 20 variables non-zero estimates.

```
predlars$coef[predlars$coef != 0]
      V2       V12       V24       V30       V34       V37       V38       V42
 -137.11    249.46   -266.12   1827.73  -4255.89   1931.28   1383.86  -1202.58
     V45       V46       V47       V49       V50       V54       V61       V71
  867.18    324.93    131.61  -1102.57    -15.74    189.47    205.20   -223.67
     V79       V89       V96      V100
   80.76     27.26    -96.87     81.65
```