Shrinkage Methods

2020-05-11

Contents

Chapter 1

Motivation

1.1 The ordinary least squares model

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

is the most commonly used method to describe the relationship between the response variable Y and a set of variables X. However, the OLS model is not perfect and it faces two major criticism and one major drawback:

- First, the OLS model could result in low prediction accuracy because it often has large variance and low bias.
- The second weakness is efficient interpretation. If we have a dataset with large number of predictors, we would usually choose a small subset of predictors which are strongly associated with the response variable; thus, it seems to be a loss that we still include the weak predictors in our model.
- Also, OLS does not work when there are more predictors than data points. That being said, it does not generate a unique solution($p > n$).

However, the linear regression model is easy to regress, interpret, and perform inference; therefore, we would still use linear regression method but we want to modify it in a way such that we could have higher prediction accuracy and stronger interpretation.

1.2 Several concepts for better linear regression model

One of the key questions to find the best model is to determine if we should include a variable in our model or not. The process of making decision on each

variable is called variable selection. There are two different ways of selecting variables: discrete and continuous selection process. We would briefly talk about both solutions in the following session.

The first method is subset selection:

- Best subset selection algorithm: This is a sort of brute force technique because it is computationally expensive. In order to perform best subset selection, we generate the all possible subsets of p variables in a dataset, and then we can select the one with the least error.
- backward stepwise selection: This is a more feasible version of subset selection. We first generate the full model of the dataset. Then among all the features, we test the deletion of each variable using a chosen model fit criterion, deleting the variable (if any) whose loss gives the most statistically insignificant deterioration of the model fit, and repeating this process until no further variables can be deleted without a statistically insignificant loss of fit.
- More stepwise selection: forward stepwise selection, bidirectional elimination and so on (Tibshirani et al., 2001).

Problems with subset selection algorithms:

- They are mostly computationally expensive.
- They are discrete selection processes(1 variable at a time).
- The algorithms only do locally(each step) best choices. Once a variable is deleted, it is never going back in.

1.3 Shrinkage:

Shrinkage refers to the shrinking coefficients as we include another penalty term for large coefficients. As we increase the penalty, all coefficients would shrink towards zero. However, among all those variables, the coefficients of the predictors which are worse at explaining the variation in the response variable would decrease faster than others. Therefore, as the penalty increases, we could gradually see how each coefficient decrease, which can support us choose the important variables. There are two different kinds of shrinkage algorithms:

- Lasso (L1 shrinkage methods): performs both shrinking coefficients and variable selection(shrink some variables to 0). Since it uses l_1 norm of β, we can also call this L1 shrinkage methods.

- Ridge (L2 shrinkage methods): only shrinks the coefficients but does not perform variable selection. It uses l_2 norm of β, hence it is also called L2 shrinkage methods.

There are certain benefits associated with the continuous variable selection. First, the results would be less variable than the discrete process if there is a little change in the dataset (Tibshini, 1996). Second, it is more computatioinally efficient compared to the best subset selection process. Therefore, we want to dive into this method and discuss its advantages and limitations.

1.4 Literature Review

In 1996, Robert Tishirani posted `Regression Shrinkage and Selection Via the Lasso`, which was the first time Lasso was brought into statistics. Later, Tibshirani together with other well known statisticians published `An introduction to statistical learning: with applications in R`, which includes simple explaination of Lasso and Ridge Regression. In a more complicated version of textbook Tibshirani and other statisticians published, `The Elements of statistical learning: data mining, inference, and prediction`, they included a much more complete and comprehensive perspective on general Shrinkage techniques, including a Bayesian interpretation of the Lasso. Most of our work will be based on our understanding from this textbook.

1.5 Outline

In section 2 and 3, we will be covering the definition of ridge regression and how ridge regression performs shrinkage. In section 4 and 5, we will be stating the definition of Lasso and how Lasso performs both shrinkage and variable selection. In section 6, we will implement ridge regression and Lasso on some tidied data that are commonly used for machine learning to show the continous process that these shrinkage methods provide. In section 7, we will be studying Bayesian interpretation of Lasso. In section 8, we are going to cover some limitations of shrinkage methods and go through some other related topics.

Chapter 2

Ridge Regression

2.1 Definition

Recall that the ordinary least squares model estimates coefficients with the goal of minimizing the sum of the squared residuals:

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

Since ridge regression penalizes for large coefficients and uses l_2 norm of coefficients as the penalty term, we know that its estimatation of coefficients should minimize the combination of RSS and sum of the squared coefficients:

2.2 Ridge Regression:

$$\hat{\beta}^{ridge} = arg\min_{\beta}\{\sum_{i=1}^{N}(y_i - \beta_0 - \sum_{j=1}^{p}x_{ij}\beta_j)^2 + \lambda\sum_{j=1}^{p}\beta_j^2\} \quad (1)$$

Which is the same as:

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

where there is a one-to-one corresondence between the parameters λ and t.

λ is the tuning parameter, and $\lambda\sum_{j=1}^{p}\beta_j^2$ is the skrinkage penalty. If λ is zero, then the penalty term equals to zero, and we would minimize the RSS, which is equivalent as ordinary least squares model; if λ is large, then the penalty

term would be larger as well, the coefficients would be smaller; therefore, if λ approaches infinity, then the coefficients would approach zero. The selection of the optimal tuning parameter is often based on the evidence which is the MSE, and we would discuss the choice of tuning parameter later in the application.

Chapter 3

How Ridge Works

Recall from Linear Algebra that if we want to solve for $x\beta = y$, we can do:

$$x^T x\beta = x^T y$$

We then take the inverse of $\boldsymbol{x}^T\boldsymbol{x}$ from the LHS, and multiply both sides by the inverse:

$$\beta = (x^T x)^{-1} x^T y$$

Then we have come up with a least squares solution for x in $x\beta = y$. From here then we derive that: $\hat{\beta}_{OLS} = (X^T X)^{-1} X^T y$. We will use a very similar approach to show how the shrinkage work for ridge regression.

3.0.1 Univariate example

Let's consider a very simple model: $y = \beta x + \epsilon$, with an L2(ridge regression) penalty on $\hat{\beta}$ and a least-squares loss function on $\hat{\epsilon}$, where $\hat{\epsilon}$ is the estimator for prediction error and $\hat{\epsilon} = \sum_{i=1}^{N}(y_i - x_i\beta)$. We can then expand the expression for sum of squared residuals to be minimized as:

$$\hat{\beta} = arg\min_{\beta}\{\sum_{i=1}^{N}(y_i - x_i\beta)^2 + \lambda\sum_{j=1}^{p}\beta_j^2\}$$

Which if we transform into a Matrix form, gives:

$$\hat{\beta} = arg\min_{\beta}\{(\vec{y} - \vec{x}\hat{\beta})^T(\vec{y} - \vec{x}\hat{\beta}) + \lambda\hat{\beta}^2\}$$

Which we can further expand into:

$$\hat{\beta} = arg\min_{\beta}\{\vec{y}^T\vec{y} - 2\vec{y}^T\vec{x}\hat{\beta} + \hat{\beta}\vec{x}^T\vec{x}\hat{\beta} + \lambda\hat{\beta}^2\}$$

Now if we take the derivative w.r.t $\hat{\beta}$ and set equal to 0, we can calculate the the estimator for ridge regression coefficients $\hat{\beta}$:

$$-2\vec{y}^T\vec{x} + 2\vec{x}^T\vec{x}\hat{\beta} + 2\lambda\hat{\beta} =_{set} 0$$

Where we can obtain:

$$\hat{\beta} = \vec{y}^T\vec{x}(\vec{x}^T\vec{x} + \lambda)^{-1}$$

3.0.2 Higher dimension ridge regression

Note that here we are using an univariate example(1-dimensional y and 1-dimensional x), in a more complicated case, we calculate the sum of squared residual in the same manner:

$$RSS(\lambda) = (\boldsymbol{y} - \boldsymbol{X}\hat{\beta})^T(\boldsymbol{y} - \boldsymbol{X}\hat{\beta}) + \lambda\hat{\beta}^T\hat{\beta}$$

Where we can get:

$$\hat{\beta} = (\boldsymbol{X}^T\boldsymbol{X} + \lambda\boldsymbol{I}_{pp})^{-1}\boldsymbol{X}^T\boldsymbol{Y}$$

Which is very similar to the result from the univariate example.

3.0.3 The tuning parameter

To further understand why λ helps shrink the coefficients, we are going to use singular value decomposition(SVD) to examine the effect of λ. Let the singular value decomposition of the $(n \times p)$-dimensional design matrix X be:

$$\boldsymbol{X} = \boldsymbol{U}_x\boldsymbol{D}_x\boldsymbol{V}_x^T$$

where $\boldsymbol{D}_x$ is an $(n \times n)$-dimensional diagonal matrix with the singular values, $\boldsymbol{U}_x$ is an $(n \times n)$-dimensional matrix with columns containing the left singular vectors (denoted u_i), and $\boldsymbol{V}_x$ is a $(p \times n)$-dimensional matrix with columns containing the right singular vectors (denoted V_i). The columns of $\boldsymbol{U}_x$ and $\boldsymbol{V}_x$ are orthogonal:

$$\boldsymbol{U}_x^T\boldsymbol{U}_x = \boldsymbol{I}_{nn} = \boldsymbol{V}_x^T\boldsymbol{V}_x$$

Recall that the OLS estimator is:

$$\hat{\beta}_{OLS} = (X^TX)^{-1}X^Ty$$

We can then rewrite the fitted values from OLS as:

$$\begin{aligned}
\boldsymbol{X}\hat{\beta}_{OLS} &= \boldsymbol{U}_x\boldsymbol{D}_x\boldsymbol{V}_x^T(\boldsymbol{V}_x\boldsymbol{D}_x\boldsymbol{U}_x^T\boldsymbol{U}_x\boldsymbol{D}_x\boldsymbol{V}_x^T)^{-1}\boldsymbol{V}_x\boldsymbol{D}_x\boldsymbol{U}_x^T y \\
&= \boldsymbol{U}_x\boldsymbol{D}_x\boldsymbol{V}_x^T(\boldsymbol{V}_x\boldsymbol{D}_x\boldsymbol{D}_x\boldsymbol{V}_x^T)^{-1}\boldsymbol{V}_x\boldsymbol{D}_x\boldsymbol{U}_x^T y \\
&= \boldsymbol{U}_x\boldsymbol{D}_x\boldsymbol{V}_x^T\boldsymbol{V}^{T-1}_x\boldsymbol{D}_x^{-1}\boldsymbol{D}_x^{-1}\boldsymbol{V}_x^{-1}\boldsymbol{V}_x\boldsymbol{D}_x\boldsymbol{U}_x^T y \\
&= \boldsymbol{U}_x\boldsymbol{D}_x\boldsymbol{D}_x^{-1}\boldsymbol{D}_x^{-1}\boldsymbol{D}_x\boldsymbol{U}_x^T y \\
&= \boldsymbol{U}_x\boldsymbol{U}_x^T y \\
&= \sum_{j=1}^{p}\boldsymbol{u}_j(1)\boldsymbol{u}_j^T\boldsymbol{y}
\end{aligned}$$

We can also rewrite $\hat{\beta}^{ridge}$ as:

$$\begin{aligned}
\hat{\beta} &= (\boldsymbol{X}^T\boldsymbol{X} + \lambda\boldsymbol{I}_{pp})^{-1}\boldsymbol{X}^T\boldsymbol{Y} \\
&= (\boldsymbol{V}_x\boldsymbol{D}_x\boldsymbol{U}_x^T\boldsymbol{U}_x\boldsymbol{D}_x\boldsymbol{V}_x^T + \lambda\boldsymbol{I}_{pp})^{-1}\boldsymbol{V}_x\boldsymbol{D}_x\boldsymbol{U}_x^T\boldsymbol{Y} \\
&= (\boldsymbol{V}_x\boldsymbol{D}_x^2\boldsymbol{V}_x^T + \lambda\boldsymbol{V}_x\boldsymbol{V}_x^T)^{-1}\boldsymbol{V}_x\boldsymbol{D}_x\boldsymbol{U}_x^T\boldsymbol{Y} \\
&= (\boldsymbol{V}_x[\boldsymbol{D}_x^2 + \lambda\boldsymbol{I}_{nn}]^{-1}\boldsymbol{V}_x^T)^{-1}\boldsymbol{V}_x\boldsymbol{D}_x\boldsymbol{U}_x^T\boldsymbol{Y} \\
&= (\boldsymbol{V}_x^{-T}[\boldsymbol{D}_x^2 + \lambda\boldsymbol{I}_{nn}]^{-1}\boldsymbol{V}_x^{-1})\boldsymbol{V}_x\boldsymbol{D}_x\boldsymbol{U}_x^T\boldsymbol{Y} \\
&= \boldsymbol{V}_x^{-T}[\boldsymbol{D}_x^2 + \lambda\boldsymbol{I}_{nn}]^{-1}\boldsymbol{D}_x\boldsymbol{U}_x^T\boldsymbol{Y}
\end{aligned}$$

Then the ridge solutions becomes:

$$\begin{aligned}
\boldsymbol{X}\hat{\beta} &= \boldsymbol{X}\boldsymbol{V}_x^{-T}[\boldsymbol{D}_x^2 + \lambda\boldsymbol{I}_{nn}]^{-1}\boldsymbol{D}_x\boldsymbol{U}_x^T\boldsymbol{Y} \\
&= \boldsymbol{U}_x\boldsymbol{D}_x\boldsymbol{V}_x^T\boldsymbol{V}_x^{-T}[\boldsymbol{D}_x^2 + \lambda\boldsymbol{I}_{nn}]^{-1}\boldsymbol{D}_x\boldsymbol{U}_x^T\boldsymbol{Y} \\
&= \boldsymbol{U}_x\boldsymbol{D}_x(\boldsymbol{D}_x^2 + \lambda\boldsymbol{I}_{nn})^{-1}\boldsymbol{D}_x\boldsymbol{U}_x^T\boldsymbol{Y} \\
&= \sum_{j=1}^{p}\boldsymbol{u}_j\frac{d_j^2}{d_j^2+\lambda}\boldsymbol{u}_j^T\boldsymbol{y}
\end{aligned}$$

Note that since $\lambda \geq 0$, we have $\frac{d_j^2}{d_j^2+\lambda} \leq 1$. Ridge regression computes the coordinates of $\boldsymbol{y}$ with respect to the orthonormal basis U. Then it shrinks these coordinates by the factor $\frac{d_j^2}{d_j^2+\lambda}$. This means that variable with smaller singular value (d_j) is shrunk more than those with larger singular values. Note that variables with smaller singular values explains less of the variation of our sample(less important). This could be proved using the sample covariance matrix.

Chapter 4

Lasso

Least absolute shrinkage and selection operator, aka Lasso, is similar to the ridge regression but its constraint is estimated with the sum of the absolute value of the coefficient:

$$\hat{\beta}^{Lasso} = arg\min_{\beta}\{\sum_{i=1}^{N}(y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j)^2 + \lambda\sum_{j=1}^{p}|\beta_j|\} \quad (1)$$

Which is the same as:

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

Notice that in Lasso, the penalty term now becomes the sum of absolute value of the coefficients instead of sum of squares of the coefficients.

4.0.1 Standardizing the dataset

Considering lasso regression (Find x to minimize the following term):

$$||x\beta - y||^2 + \lambda||x||_1$$

By standardizing the dataset, the columns of A have zero mean and unit norm and the columns of b have zero mean. By making the means of columns zero, we can get rid of the intercept. Besides, since the norms of the columns in A is one, some column would no longer have smaller coefficients because of their bigger norm, which leads us incorrently conclude that this column can not explain x well. Therefore, by putting all columns into the same scale, we can use the differences in the magnitudes of the elements of x that are directly related to the "wiggliness" of the explanatory function Ax, which is, loosely speaking, what the regularization tries to control.

Chapter 5

How Lasso performs variable selection

Again, just as we discussed how shrinkage worked, we'll use the univariate model to illustrate.

5.0.1 univariate example

Again we have a simple model: $y = \beta x + \epsilon$, with an L1 penalty(Lasso regression) on $\hat{\beta}$ and a least squares loss function on $\hat{\epsilon}$. We can then expand the expression for sum of squared residuals to be minimized as:

$$\hat{\beta} = arg\min_{\beta}\{\sum_{i=1}^{N}(y_i - x_i\beta)^2 + \lambda\sum_{j=1}^{p}|\beta_j|\}$$

For convenience, we will now substitute $2\lambda^* = \lambda$, since $2\lambda^* = \lambda$ has a one-to-one relationship, this change does not affect the result. We can then rewrite $\hat{\epsilon}$ in the matrix form:

$$\hat{\beta} = arg\min_{\beta}\{(\vec{y} - \vec{x}\hat{\beta})^T(\vec{y} - \vec{x}\hat{\beta}) + 2\lambda^*|\hat{\beta}|\}$$

Which we can further expand into:

$$\hat{\beta} = arg\min_{\beta}\{\vec{y}^T\vec{y} - 2\vec{y}^T\vec{x}\hat{\beta} + \hat{\beta}\vec{x}^T\vec{x}\hat{\beta} + 2\lambda^*|\hat{\beta}|\}$$

Unlike the ridge regression case where we can take a derivative directly and set it equal to zero so that we can minimize the residual, here we have a $|\hat{\beta}|$ term which makes such procedure painful. Thus, we will instead compare different cases w.r.t. $\hat{\beta}$. Case1: $\hat{\beta} \geq 0$, $|\hat{\beta}| = \hat{\beta}$ Since we are assuming that $\hat{\beta} \geq 0$, it is

the same as assuming $\vec{y}^T\vec{x} \geq 0$(x's and y's have a positive relationship). In this case, we can rewrite the above expression as:

$$\hat{\epsilon} = arg\min_{\beta}\{\vec{y}^T\vec{y} - 2\vec{y}^T\vec{x}\hat{\beta} + \hat{\beta}\vec{x}^T\vec{x}\hat{\beta} + 2\lambda^*\hat{\beta}\}$$

Then we can take the derivative w.r.t. $\hat{\beta}$ and set it equal to zero:

$$-2\vec{y}^T\vec{x} + 2\vec{x}^T\vec{x}\hat{\beta} + 2\lambda^* =_{set} 0$$

Where we can obtain a solution for $\hat{\beta}$:

$$\hat{\beta} = (\vec{y}^T\vec{x} - \lambda^*)(\vec{x}^T\vec{x})^{-1}$$

Obviously by increasing λ^*, we can eventually achieve $\hat{\beta} = 0$ at $\lambda^* = \vec{y}^T\vec{x}$. However, it is tricky to think about what happens when we increase λ^* once $\vec{y}^T\vec{x} = 0$. The thing is that increasing λ^* at this point will not drive $\hat{\beta}$ negative, because once the estimator $\hat{\beta}$ becomes negative, the derivative of the penalty term estimator becomes:

$$-2\vec{y}^T\vec{x} + 2\vec{x}^T\vec{x}\hat{\beta} - 2\lambda^* =_{set} 0$$

where the flip in the sign of λ^* is due to the absolute value function before taking the derivative. Thus, we have a new solution for $\hat{\beta}$:

$$\hat{\beta} = (\vec{y}^T\vec{x} + \lambda^*)(\vec{x}^T\vec{x})^{-1}$$

This solution, however, is inconsistent with our premises $\hat{\beta} < 0$, since we have assumed that the least squares solution is greater than or equal to zero($\vec{y}^T\vec{x} \geq 0$), and $\lambda^* \geq 0$. For this solution, the sum of squared residual does not have an minimum anymore. Thus, we will just stick at $\hat{\beta} = 0$, even if $\lambda^* > \vec{y}^T\vec{x}$. Intuitively, when we assume that the least squares solution is negative with $\hat{\beta} < 0$, the logic is the same that we stick with $\hat{\beta} = 0$ once it reaches zero. Note that so far we've only talked about a univariate Lasso example. When having a dataset that has multiple dimensions, as we keep increasing the value of λ(or λ^*), some of the features or variables will be zeroed out just as $\hat{\beta}$ shown above while some other features are shrinked toward zero but not yet reduced to zero. Therefore, the Lasso does variable selection in this manner, by shrinking some of the coefficients to zero while some non-zero.

5.0.2 Shrinkage explanation from geometric perspective

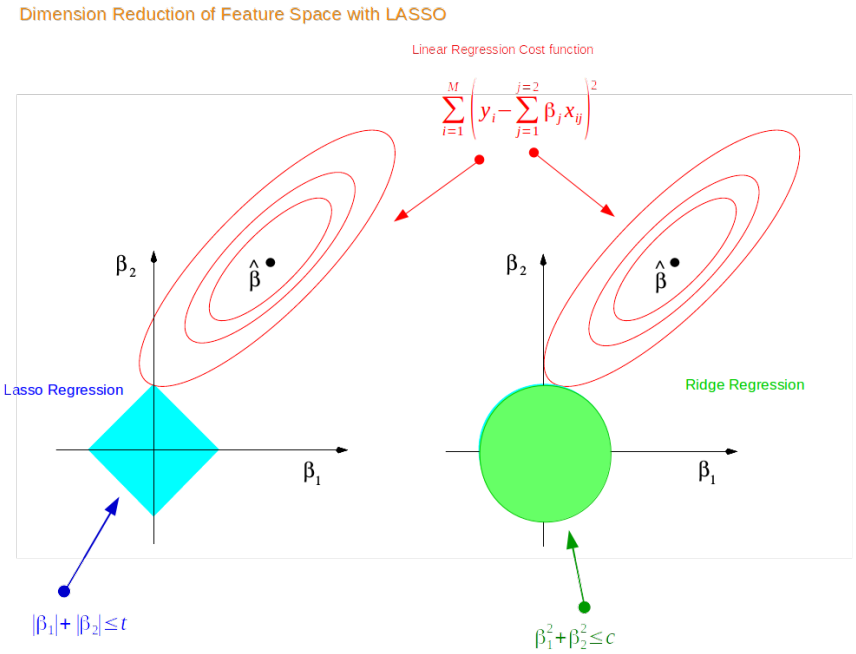
Now suppose that we have a dataset with 2 features. We first train a liner model for the data, and suppose β_1 and β_2 are the coefficients for the two features. Thus by the definition of the ridge regression constraint, we have:

$$\beta_1^2 + \beta_2^2 \leq t$$

Similarly, for the Lasso regression contraint, we have:

$$|\beta_1|+|\beta_2| \leq t$$

Thus, if we plot the two constraints in a 2-dimensional coordinate system, we have:



Note that the red contours are the vector space for the coefficient estimator $\hat{\beta}$. Originally, the constraint and the vector space are separate; as we increase the constraint limit t, eventually the two constraints will hit the vector space, and thus achieving optimization for both the ridge regression model and the Lasso regression model. Since the 2-dimensional Lasso constraint has corners(diamond shape), if the constraint hit the vector space on one such corner, one of the two features gets dropped and shrinked to zero. In higher dimensional spaces, the diamond shape becomes rhomboid where there are more corners and more possibility for dropping one or more variables. In contrast, the constraint for ridge regression is a circle, where all points on or within the circle resemble a linear combination of the two features. Thus, technically speaking, when the ridge regression constraint hit the vector space and achieve optimization, one variable could be shrunk very close to zero while the other remain slightly shrunk, it is impossible for ridge regression to achieve variable selection in reality. Similarly, in higher dimensional space, the constraint for ridge regression becomes a sphere and the situation is very similar to what we have in the two dimensional space.

Chapter 6

Applications of Ridge and Lasso

6.1 Cross Validation

One of the key features of a good model is prediction accuracy. If our prediction varies a lot when we use the model on other datasets, then our model is bad at making prediction, and our model is overfitting to that the training dataset. Therefore, we also want to estimate the prediction accuracy of our models to decide whether it is good at making predictions; and one way of estimation prediction accuracy is cross validation.

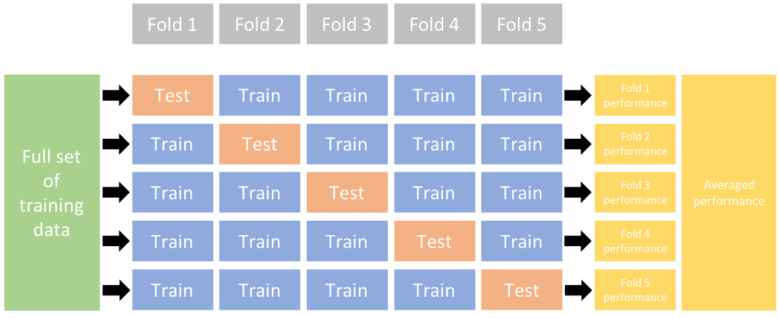


Figure 2.4: Illustration of the k-fold cross validation process.

(from Hands-On Machine Learning with R 2.4 by Bradley Boehmke & Brandon Greenwell)

Cross validation randomly divides the data set into k folds, k-1 training groups to which models will be fitted and 1 testing group to test the model. For example, in the image above, the dataset is divided into 5 groups, the blue ones as the

training groups and the pink one as the test group. The process is repeated 5 times and averaging the RMSE for each time gives the cross-validation RMSE of the model. In our simulation, k=10 will be used to minimize the variability and imporve the accuracy of the models.

6.2 Dataset

This dataset is obtained from https://raw.githubusercontent.com/juliasilge/supervised-ML-case-studies-course/master/data/cars2018.csv. The goal is to find how different features of a car affect its miles per gallon (MPG). The outcome variable is numerical while the predictor variables are a mix of numerical and categorical variables. The dataset is splitted into training data with 70% of the data and test data with 30%. In this case, we would use the training data to fit our linear model and test its prediction accuracy with the remaining data.

6.3 OLS

We first fit an OLS model to the training data. The result shows that the coefficient for Gears is 0.1338 and the coefficient for Cylinders is 0.5918. The RMSE for this model is 3.0272.

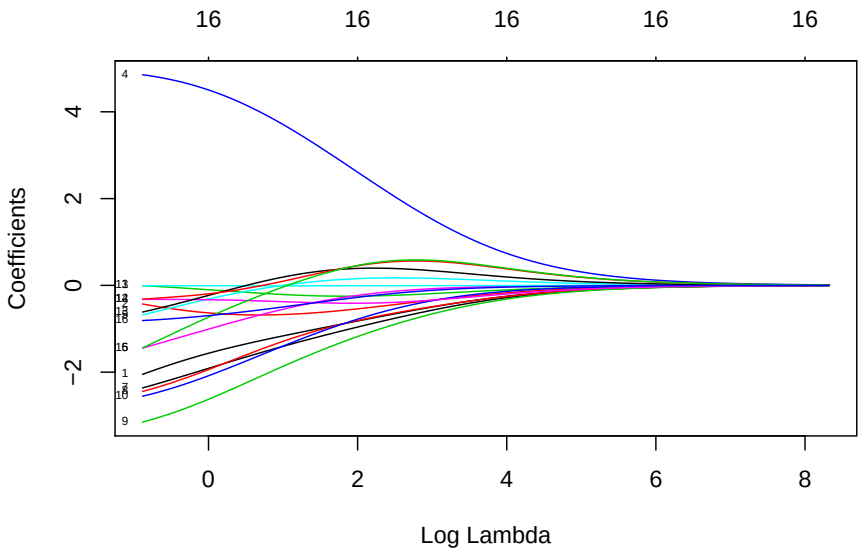
6.4 RIDGE

Secondly, we fit a ridge model to the training data. We try with a range of lambda values and use the bestTune$lambda function to find the lambda value that minimizes the RMSE. Then we plug in this lambda value and make a new ridge model. We get a RMSE of 3.0743, which is slightly higher than the OLS RMSE. The coefficient for Cylinders is shrunk to -0.4284 and the one for Gear is shrunk to -0.0064, but none of the coefficients is set to 0 though some of them are close to 0. The graph below shows the shrinkage of coefficients. We can see the as lambda increases, all of them get close to 0 but not equal to 0.

```
set.seed(455)
cars_ridge <- train(
  MPG ~ .,
  data = cars_train,
  method = "glmnet",
  trControl = trainControl(method = "cv",
                           number = 10),
  tuneGrid = data.frame(alpha = 0,
```

```
                          lambda = 10^seq(-3,0, length = 100)),
  na.action = na.omit
)

cars_ridge$bestTune$lambda
```



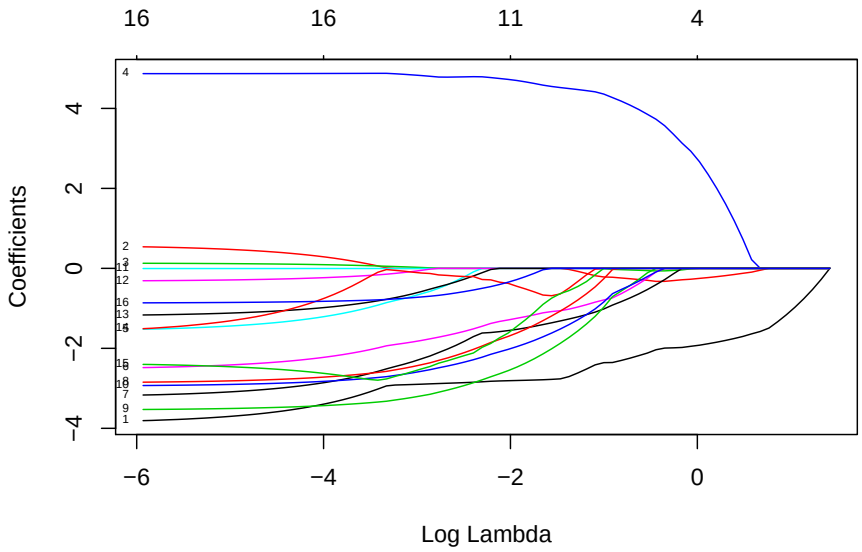
6.5 LASSO

We repeat the same process to fit the LASSO model. In terms of coding, the only difference is to change the alpha value in the tuneGrid from 0 to 1. We get a RMSE of 3.0255, which is the smallest. Some coefficients such as Cylinders and Gears are set to 0. It is interesting that the coefficient TransmissionManual is set to 0 while TransmissionCVT, the other term of the same categorical variable, is not. This leads to one of the limitation of LASSO models when analyzing categorical variables. From the graph showing the shrinkage of coefficients, we can see the as lambda increases, all of them get to exactly 0.

```
set.seed(455)
cars_lasso <- train(
  MPG ~ .,
  data = cars_train,
  method = "glmnet",
```

```
  trControl = trainControl(method = "cv",
                           number = 10),
  tuneGrid = data.frame(alpha = 1,
                        lambda = 10^seq(-3, -2, length = 100)),
  na.action = na.omit
)

cars_lasso$bestTune$lambda
```




```
## 17 x 1 sparse Matrix of class "dgCMatrix"
##                                                          1
## (Intercept)                                     39.75545077
## Displacement                                    -2.83657321
## Cylinders                                        .
## Gears                                            .
## TransmissionCVT                                  4.79076756
## TransmissionManual                               .
## AspirationTurbocharged/Supercharged             -1.44680028
## `Lockup Torque Converter`Y                      -1.61692083
## Drive2-Wheel Drive, Rear                        -1.94168304
## Drive4-Wheel Drive                              -2.78768269
## DriveAll Wheel Drive                            -2.22100559
## `Max Ethanol`                                   -0.00222014
## `Recommended Fuel`Premium Unleaded Required      .
```

```
## `Recommended Fuel`Regular Unleaded Recommended -0.10788122
## `Intake Valves Per Cyl`                         -0.27837487
## `Exhaust Valves Per Cyl`                        -1.97358320
## `Fuel injection`Multipoint/sequential ignition -0.49852192
```

6.6 Test data

We also apply the models to the test data to obtain the test RMSEs.

6.7 Summary

RMSE	ols_train	ridge_train	lasso_train	ols_test	ridge_test	lasso_test
bodyfat dataset	4.319528	4.487984	4.250632	5.023873	4.885821	4.901642
cars daraset	3.027263	3.074365	3.025527	2.542560	2.561062	2.529108

From the summary table above, we can see that for this dataset, Lasso always gives the smallest RMSE no matter for the training group or the testing group. Also, according to the graphs of shrinking coefficients, we can see that some coefficients can reach zero for Lasso regression; while in the graph for ridge regression, all coefficients decrease towards zero but they never reach the x-axis. By shrinking some coefficients to zero, Lasso regression can also help us to select variables. Therefore, Lasso regression is also better than ridge when we need to select key variables.

Chapter 7

Bayesian Interpretation

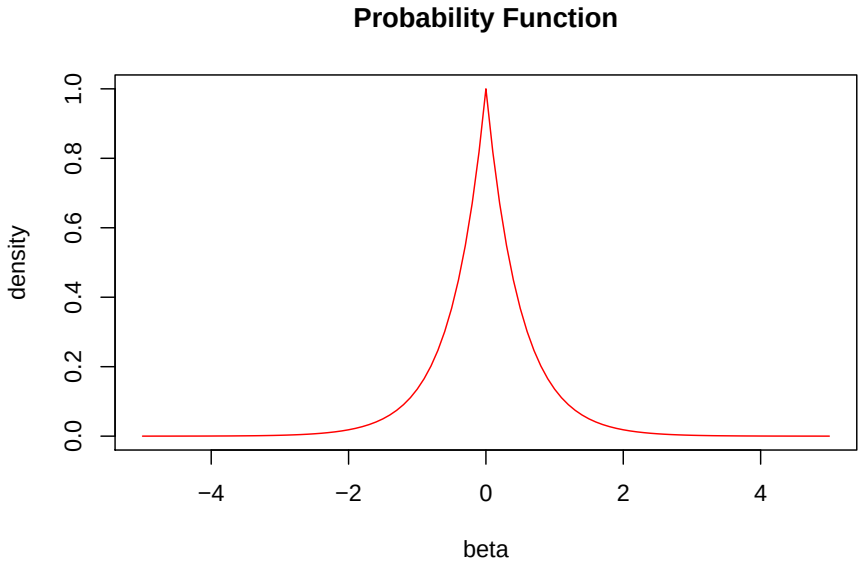
7.1 Bayesian Interpretation of Lasso

We can show that $\hat{\boldsymbol{\beta}}_{LASSO}$ has a Bayesian interpretation. In particular, we can show that is a Bayes estimator for $\boldsymbol{\beta}$ assuming a multivariate Normal likelihood for $\mathbf{y}$:

$$f(y \mid \boldsymbol{\beta}, \sigma^2) \sim N(\mathbf{X}\boldsymbol{\beta}, \sigma^2\mathbf{I}),$$

and an independent double exponential (aka Laplace) prior for $\boldsymbol{\beta}$:

$$f(\beta_j) = \left(\frac{\lambda}{2}\right) \exp(-\lambda|\beta_j|)$$



The above is an example of what the distribution of the double exponential (aka Laplace) prior looks like. From the grapg, we can see that as it goes towards the two sides, the density gradually goes to exactly 0. This explains why it is possible for Lasso coefficients to shrink towards 0 and ultimately be exactly 0.

7.2 Posterior

The posterior distribution for $\boldsymbol{\beta}$ (assuming $\sigma^2 = 1$ for simplicity):

$$
\begin{aligned}
g(\boldsymbol{\beta} \mid y) &\propto f(y \mid \boldsymbol{\beta}, \sigma^2) f(\boldsymbol{\beta}) \\
&= f(y \mid \boldsymbol{\beta}) \prod_{i=1}^{p} f(\beta_j), \text{ by independence assumption} \\
&= (2\pi)^{-n/2} \exp\left(-\frac{1}{2}(\mathbf{y} - \mathbf{X}\boldsymbol{\beta})^\top (\mathbf{y} - \mathbf{X}\boldsymbol{\beta})\right) \left(\frac{\lambda}{2}\right)^p \exp\left(-\lambda \sum_{j=1}^{p} |\beta_j|)\right) \\
&\propto \exp\left(-\frac{1}{2}(\mathbf{y} - \mathbf{X}\boldsymbol{\beta})^\top (\mathbf{y} - \mathbf{X}\boldsymbol{\beta})\right) \exp\left(-\lambda \sum_{j=1}^{p} |\beta_j|)\right) \\
&= \exp\left(-\frac{1}{2}(\mathbf{y} - \mathbf{X}\boldsymbol{\beta})^\top (\mathbf{y} - \mathbf{X}\boldsymbol{\beta}) - \lambda \sum_{j=1}^{p} |\beta_j|)\right)
\end{aligned}
$$

7.3 Posterior Mode

Maximizing the posterior distribution, or minimizing the $-\log$ posterior, leads to $\hat{\beta}_{LASSO}$.

$$\begin{aligned}
& \arg\max \ \exp\left(-\frac{1}{2}(\mathbf{y}-\mathbf{X}\boldsymbol{\beta})^\top(\mathbf{y}-\mathbf{X}\boldsymbol{\beta})-\lambda\sum_{j=1}^{p}|\beta_j|\right) \\
= \ & \arg\max \ -\frac{1}{2}(\mathbf{y}-\mathbf{X}\boldsymbol{\beta})^\top(\mathbf{y}-\mathbf{X}\boldsymbol{\beta})-\lambda\sum_{j=1}^{p}|\beta_j| \\
= \ & \arg\min \ \frac{1}{2}(\mathbf{y}-\mathbf{X}\boldsymbol{\beta})^\top(\mathbf{y}-\mathbf{X}\boldsymbol{\beta})+\lambda\sum_{j=1}^{p}|\beta_j| \\
= \ & \arg\min \ \text{-log posterior}
\end{aligned}$$

$$\begin{aligned}
\arg\min \ \text{-log posterior} &= \arg\min -\log(\exp\left(-\frac{1}{2}(\mathbf{y}-\mathbf{X}\boldsymbol{\beta})^\top(\mathbf{y}-\mathbf{X}\boldsymbol{\beta})-\lambda\sum_{j=1}^{p}|\beta_j|\right)) \\
&= \arg\min -\left(-\frac{1}{2}(\mathbf{y}-\mathbf{X}\boldsymbol{\beta})^\top(\mathbf{y}-\mathbf{X}\boldsymbol{\beta})-\lambda\sum_{j=1}^{p}|\beta_j|\right) \\
&= \arg\min \frac{1}{2}(\mathbf{y}-\mathbf{X}\boldsymbol{\beta})^\top(\mathbf{y}-\mathbf{X}\boldsymbol{\beta})+\lambda\sum_{j=1}^{p}|\beta_j| \\
&= \hat{\beta}_{LASSO}, \text{ by definition}
\end{aligned}$$

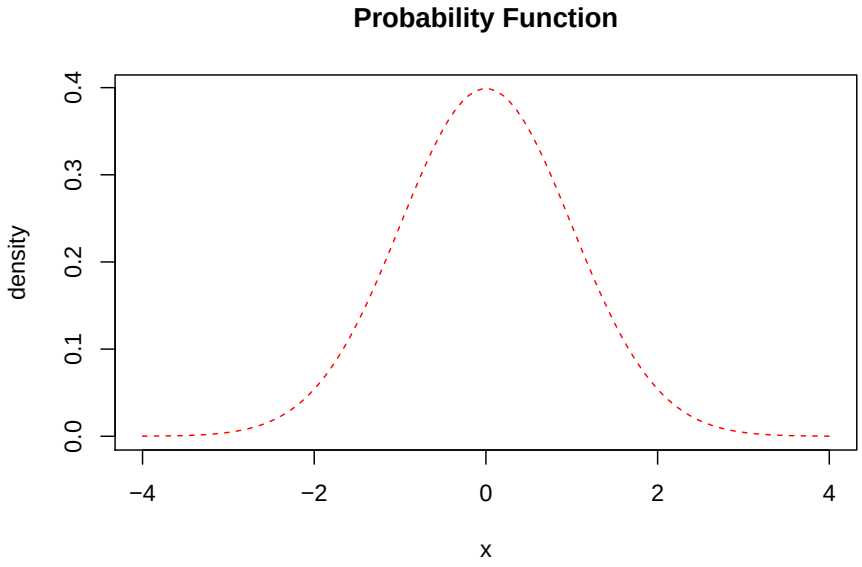
7.4 Bayesian Interpretation of Ridge

In a similar way, we can show that $\hat{\boldsymbol{\beta}}_{RIDGE}$ has a Bayesian interpretation by using a different prior. In particular, we can show that is a Bayes estimator for $\boldsymbol{\beta}$ assuming a multivariate Normal likelihood for $\mathbf{y}$:

$$f(y \mid \boldsymbol{\beta}, \sigma^2) \sim N(\mathbf{X}\boldsymbol{\beta}, \sigma^2\mathbf{I}),$$

and an independent normal prior for $\boldsymbol{\beta}$:

$$f(\beta_j) = \left(\frac{\lambda}{\sigma^2\sqrt{2\pi}}\right)\exp(-\frac{1}{2}(\frac{\lambda\beta_j}{\sigma^2})^2)$$



The above is an example of what the distribution of the normal prior looks like. From the grapg, we can see that as it goes towards the two sides, the density gradually shrinks towards 0. This explains why it is possible for Ridge coefficients to shrink towards 0 and cannot goes to exactly 0.

7.5 Posterior

The posterior distribution for $\boldsymbol{\beta}$ (assuming $\sigma^2 = 1$ for simplicity):

$$
\begin{aligned}
g(\boldsymbol{\beta} \mid y) &\propto f(y \mid \boldsymbol{\beta}, \sigma^2) f(\boldsymbol{\beta}) \\
&= f(y \mid \boldsymbol{\beta}) \prod_{i=1}^{p} f(\beta_j), \text{ by independence assumption} \\
&= (2\pi)^{-n/2} \exp\left(-\frac{1}{2}(\mathbf{y} - \mathbf{X}\boldsymbol{\beta})^\top (\mathbf{y} - \mathbf{X}\boldsymbol{\beta})\right) \left(\frac{\lambda}{\sqrt{2\pi}}\right)^p \exp(-\frac{\lambda^2}{2} \sum_{j=1}^{p} \beta_j^2) \\
&\propto \exp\left(-\frac{1}{2}(\mathbf{y} - \mathbf{X}\boldsymbol{\beta})^\top (\mathbf{y} - \mathbf{X}\boldsymbol{\beta})\right) \exp\left(-\lambda^* \sum_{j=1}^{p} \beta_j^2\right) \\
&= \exp\left(-\frac{1}{2}(\mathbf{y} - \mathbf{X}\boldsymbol{\beta})^\top (\mathbf{y} - \mathbf{X}\boldsymbol{\beta}) - \lambda^* \sum_{j=1}^{p} \beta_j^2\right)
\end{aligned}
$$

7.6 Posterior Mode

Maximizing the posterior distribution, or minimizing the $-\log$ posterior, leads to $\hat{\beta}_{RIDGE}$.

$$\begin{aligned}
& \arg\max \ \exp\left(-\frac{1}{2}(\mathbf{y}-\mathbf{X}\boldsymbol{\beta})^\top(\mathbf{y}-\mathbf{X}\boldsymbol{\beta}) - \lambda^* \sum_{j=1}^{p}\beta_j^2\right) \\
= \ & \arg\max \ -\frac{1}{2}(\mathbf{y}-\mathbf{X}\boldsymbol{\beta})^\top(\mathbf{y}-\mathbf{X}\boldsymbol{\beta}) - \lambda^* \sum_{j=1}^{p}\beta_j^2 \\
= \ & \arg\min \ \frac{1}{2}(\mathbf{y}-\mathbf{X}\boldsymbol{\beta})^\top(\mathbf{y}-\mathbf{X}\boldsymbol{\beta}) + \lambda^* \sum_{j=1}^{p}\beta_j^2 \\
= \ & \arg\min \ \text{-log posterior}
\end{aligned}$$

$$\begin{aligned}
\arg\min \ \text{-log posterior} &= \arg\min -\log(\exp\left(-\frac{1}{2}(\mathbf{y}-\mathbf{X}\boldsymbol{\beta})^\top(\mathbf{y}-\mathbf{X}\boldsymbol{\beta}) - \lambda^* \sum_{j=1}^{p}\beta_j^2\right)) \\
&= \arg\min -\left(-\frac{1}{2}(\mathbf{y}-\mathbf{X}\boldsymbol{\beta})^\top(\mathbf{y}-\mathbf{X}\boldsymbol{\beta}) - \lambda^* \sum_{j=1}^{p}\beta_j^2\right) \\
&= \arg\min \frac{1}{2}(\mathbf{y}-\mathbf{X}\boldsymbol{\beta})^\top(\mathbf{y}-\mathbf{X}\boldsymbol{\beta}) + \lambda^* \sum_{j=1}^{p}\beta_j^2 \\
&= \hat{\beta}_{RIDGE}, \text{ by definition}
\end{aligned}$$

7.7 Limitations of Ridge and Lasso

- According to our empirical results, we find that both ridge and lasso are better at making predictions than the OLS model.
- The coefficients in Lasso can reach zero while those in ridge can never be zero. If we want to perform variable selection, lasso might be a better choice.
- One important drawback of Lasso and ridge is that it's very hard/impossible to do proper inference (i.e., get confidence intervals and p-values)—you really just get estimates and that's it.
- The other thing is that we usually cannot give a good explanation for the coefficients and the entire model, because the coefficients are shrunk.

Some might say we could use Lasso simply as a variable selection tool and we can use the Lasso result to generate an OLS which is easier to explain. But note that there is possibility that Lasso would give us wrong information, based on assumptions we have made about the data.

- In real examples, when dealing with categorical variables(those with more than 10 sub-categories), Lasso often does weird thing as it treats each of the sub-category as an individual variable, that is, it selects some of the sub-categories while think others to be irrelevant. This also makes it hard to interpret.

7.8 Reference

- Hastie, T., Friedman, J., & Tisbshirani, R. (2017). The Elements of statistical learning: data mining, inference, and prediction. New York: Springer.
- James, G., Witten, D., Hastie, T., & Tibshirani, R. (2017). An introduction to statistical learning: with applications in R. New York: Springer.
- Tibshirani, R. (1996). Regression Shrinkage and Selection Via the Lasso. Journal of the Royal Statistical Society: Series B (Methodological), 58(1), 267–288. doi: 10.1111/j.2517-6161.1996.tb02080.x
- Wieringen, W. N. van. (2020, January 18). Lecture Notes on Ridge Regression. PDF.