6 Linear Model Selection and Regularization

In this chapter we mainly discuss about the methods which multiple linear regression model can be improved. These methods include,

- Prediction Accuracy: If $p > n$ then there is no longer a unique least squares coefficient estimate: the variance is infinite so the method cannot be used at all. By *constrain* or *shrinking* the estimated coefficients, we can often substantially reduce the variance at the cost of a negligible increase in bias.
- Model Interpretability: Some approaches for automatically performing feature selection or variable selection. That is, for excluding irrelevant variables from a multiple regression model (Lasso).

6.1 Subset Selection

6.1.1 Best Subset Selection

To perform best subset selection, we fit a separate *least squares regression* for each possible combination of the p predictors.

Algorithm 1 Best subset selection

```
1: Let $\mathcal{M}_0$ denote the *null* model, which contain no predictors. This model simply predicts the sample mean for each observations.
2: for $k = 1, 2, \ldots, p$ do
3:     (a) Fit all $\binom{p}{k}$ models that contain exactly $k$ predictors.
4:     (b) Pick the best among these $\binom{p}{k}$ models, and call it $\mathcal{M}_k$. Here best is defined as having the smallest RSS, or equivalently largest $R^2$
5: end for
6: Select a single best model from among $\mathcal{M}_0, \ldots, \mathcal{M}_p$ using cross-validated prediction error, $C_p$, $(AIC), BIC$, or adjusted $R^2$.
```

In Algorithm 1, Step 2.(b) we select a best model in round k by computing RSS or R^2. However, a low RSS or a high R^2 indicates a model with a low training error. In this case we use cross-validated prediction error, $C_p, (AIC), BIC, adj - R^2$ to select among $\mathcal{M}_0, \ldots, \mathcal{M}_p$ in order to evaluate our model by computing the test error.

- Drawbacks

  1. We have to choose the best model among the 2^p models, which is computationally expensive.

- Advantages

  1. The approach guaranties to choose the best model since it evaluates all model combinations.

6.1.2 Stepwise Selection

6.1.2.1 Forward Stepwise Selection

Algorithm 2 Forward Stepwise Selection

```
Let $\mathcal{M}_0$ denote the *null* model, which contain no predictors.
**for** $k = 1, 2, \dots, p-1$ **do**
    (a) Consider all $p-k$ models that augment the predictors in $\mathcal{M}_k$ with on additional predictor.
    (b) Choose the best among these $p-k$ models, and call it $\mathcal{M}_{k+1}$. Here best is defined as having smallest RSS or highest $R^2$
**end for**
Select a single best model from among $\mathcal{M}_0, \dots, \mathcal{M}_p$ using cross-validated prediction error, $C_p, (AIC), BIC$, or adjusted $R^2$.
```

- Drawbacks

  1. For instance, suppose that in a given data set with $p = 3$. We already knew that the best one-variable model is X_1 and the best two-variables model is X_2 and X_3. Now, apply the forward stepwise selection and get X_1 for $\mathcal{M}_1$. However, forward stepwise selection will fail in $\mathcal{M}_2$, since X_1 shall be retained in $\mathcal{M}_2$ according to the implementation of forward stepwise and this violate what we know of the best two-variable model(X_2, X_3).
  2. Not guaranteed to yield the best model containing a subset of the p predictors.

- Advantages

  1. The approach is more efficient than best subset method.

6.1.2.2 Backward Stepwise Selection

Algorithm 3 Backward Stepwise Selection

```
Let $\mathcal{M}_p$ denote the $full$ model, which contain all $p$ predictors.
for $k = p, p-1 \dots, 1$ do
    (a) Consider all $k$ models that contain all but one of the predictors in $\mathcal{M}_k$, for a total of $k-1$ predictors.
    (b) Choose the best among these $k$ models, and call it $\mathcal{M}_{k-1}$. Here best is defined as having smallest RSS or highest $R^2$
end for
Select a single best model from among $\mathcal{M}_0, \dots, \mathcal{M}_p$ using cross-validated prediction error, $C_p, (AIC), BIC$, or adjusted $R^2$.
```

- Drawbacks

  1. $n > p$, otherwise the full model can not be fitted.
  2. Not guaranteed to yield the best model containing a subset of the p predictors.

- Advantages

  1. The approach is more efficient than best subset method.

6.1.2.3 Hybrid Approaches

This method is similar to stepwise regression. After adding each new variable, we may also remove any variables that no longer provide an improvement in the model fit (p-value $<$ 0.5).

6.1.3 Choosing the Optimal Model

In order to select the best model with respect to test error, we need to estimate this test error. There are two common approaches:

1. We can indirectly estimate test error by making as adjustment to the training error to account for the bias due to overfitting (C_p, AIC, BIC, $adj - R^2$) .
2. We can directly estimate the test error, using either a validation set approach or a cross-validation approach (k-Fold CV).

6.1.3.1 C_p, AIC, BIC, $adj - R^2$

$$\begin{aligned} C_p &= \frac{1}{n}(\text{RSS} + 2d\hat{\sigma}^2) \\ \text{AIC} &= \frac{1}{n\hat{\sigma}^2}(\text{RSS} + 2d\hat{\sigma}^2) \\ \text{BIC} &= \frac{1}{n}(\text{RSS} + \log(n)d\hat{\sigma}^2) \\ \text{Adjusted } R^2 &= 1 - \frac{\text{RSS}/(n-d-1)}{\text{TSS}/(n-1)} \end{aligned}$$

6.1.3.2 Validation and Cross-Validation

Compares to C_p, AIC, BIC, and $adj - R^2$; validation and cross-validation provides a direct estimate of the test error , and makes fewer assumptions about the true underlying model.

6.2 Shrinkage Methods

In the previous section, our strategy to preform model selection is to fit multiple subsets of models and choose a best one by evaluating measurements such as C_p or Cross-Validation error. In this section, we tried to shrinks the coefficient estimates or even shrinks towards zero to choose the most preventative predictors. It turns out that shrinking the coefficient estimates can significantly reduce their variance. The two best-known techniques are ridge regression and the lasso.

6.2.1 Ridge Regression

The estimation of coefficients is done by introducing a $\ell 2$ regulation in the original RSS model. The $\hat{\beta}_\lambda^{ridge}$ of ridge regression takes the form,

$$\hat{\beta}_\lambda^{ridge} = \arg\min_\beta \sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{ij} \right)^2 + \lambda \sum_{j=1}^{p} \beta_j^2 \tag{6.1}$$

It's simple to get the closed form of $\hat{\beta}_\lambda^{ridge}$ by first order differential,

$$\hat{\beta}_\lambda^{ridge} = (\lambda \mathbf{I}_D + \mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y}$$

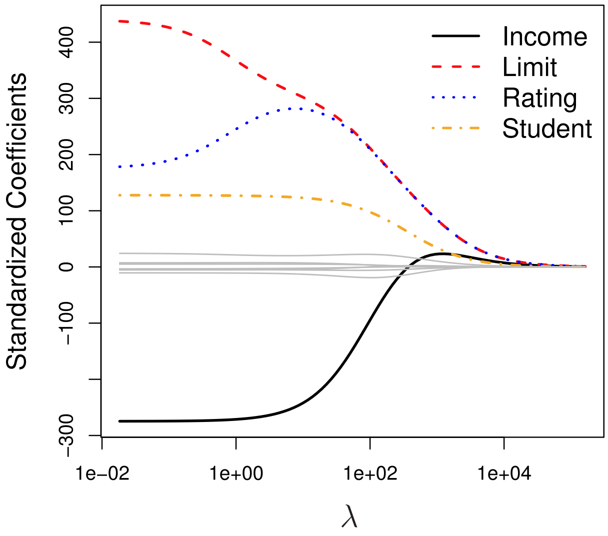

Figure 1

Figure 1 shows an illustration of the ridge regression optimization from (6.1). When $\lambda = 0$, the penalty term has no effect, and ridge regression will produce the least squares estimates. However, as $\lambda \rightarrow \infty$, the impact of the shrinkage penalty grows, and the ridge regression coefficient estimates will approach zero.

6.2.1.1 Scale equivariant

Consider the following scenario,

1. First, fit X_j by least square errors. This will give us $\hat{\beta}_j$
2. Second, we multiply X_j by a constant c and fit the parameters by least square errors again. Now, the $\hat{\beta}_j$ will takes the transformation as the original $\hat{\beta}_j \times \frac{1}{c}$

In other words, regardless of how X_j is scaled, $X_j\hat{\beta}$ will remain the same. Since X_j are in different scale, the value for β_j range widely. This will impact the ridge penalty term and lead the penalty to a high variance. Ridge regression regularize the linear regression by imposing a penalty on the size of coefficients. Thus the coefficients are shrunk toward zero and toward each other. But when this happens and if the independent variables does not have the same scale, the shrinking is not fair. Two independent variables with different scales will have different contributions to the penalized terms, because the penalized term is a sum of squares of all the coefficients. To avoid such kind of problems, very often, the independent variables are centred and scaled in order to have variance 1. In this case, we should standardizing predictors before we apply ridge regression. The standardizing formula takes the form,

$$\tilde{x}_{ij} = \frac{x_{ij}}{\sqrt{\frac{1}{n}\sum\limits_{i=1}^{n}(x_{ij} - \bar{x}_j)^2}}$$

Assume our test data are represent as z_{ij}, the standardizing of test data takes the form,

$$\tilde{z}_{ij} = \frac{z_{ij}}{\sqrt{\frac{1}{n}\sum\limits_{i=1}^{n}(x_{ij} - \bar{x}_j)^2}}$$

6.2.2 Lasso

As with ridge regression, the lasso shrinks the coefficient estimates towards zero. In other words, Lasso also performs variable selection when λ is sufficiently large.

Another way to describe Lasso:

$$\begin{aligned} &\underset{\beta}{\text{minimize}} \quad Z = \sum_{i=1}^{n}\left(y_i - \beta_0 - \sum_{j=1}^{p}\beta_j x_{ij}\right)^2 \\ &\text{subject to} \\ &\qquad \sum_{j=1}^{p}|\beta_j| \leq s \end{aligned} \tag{6.2}$$

When $p = 2$, then (6.2) indicates that the lasso coefficient estimates have the smallest RSS out of all points that lie within the diamond defined by $|\beta_1| + |\beta_2| \leq s$. The illustration of the formula in Figure 2.

We can think of a set of coefficients $\boldsymbol{\beta}$ are constrained by s from (6.2). So, if s is extremely large, we will get the least square estimates while the constraint is not very restrictive. On the other hand, if s is approximately to zero, some of the estimates we get will be zero, since the constraint is highly restrictive.

6.2.2.1 The Variable Selection Property of the Lasso

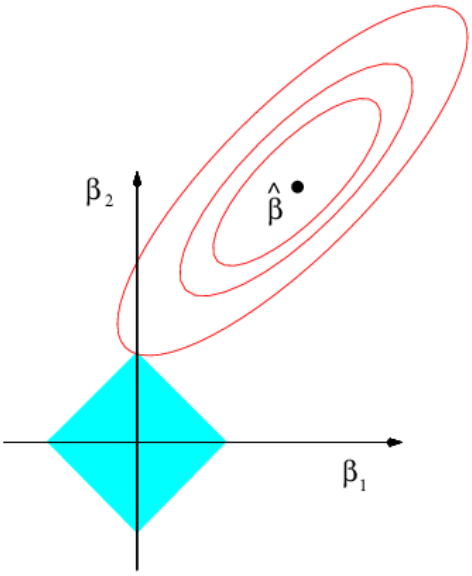

Figure 2

The least squares estimates is marked as $\hat{\beta}$, while the blue diamond represent the lasso constraints. If s is sufficiently large, then the constraint regions will cover $\hat{\beta}$, and the lasso estimates will be the same as $\hat{\beta}$.

The ellipses that are centred around $\hat{\beta}$ represent regions of constant RSS. As the ellipses expand away from $\hat{\beta}$, the RSS increases. The lasso constraint has corners at each of the axes, so the ellipse will often intersect the constraint region at an axis. When this occurs, one of the coefficients will equal zero. In higher dimensions, many of the coefficient estimates may equal zero simultaneously. In Figure 2 , the intersection occurs at $\beta_1 = 0$, so the resulting model will only include β_2.

6.2.2.2 A simple case for the Lasso

Assume we are performing regression without an intercept, $n = p$ and $\mathbf{X}$ a diagonal matrix with 1's on the diagonal and 0's in all off-diagonal elements. With these assumptions, the least squares takes for form,

$$\hat{\beta} = \arg\min_{\beta} \sum_{j=1}^{p} (y_j - \beta_j)^2 \tag{6.3}$$

In this case, the least squares estimator is given by,

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

In this setting, the lasso amounts to finding the coefficients such that

$$\hat{\beta}_{\lambda}^{lasso} = \arg\min_{\beta} \sum_{j=1}^{p} (y_j - \beta_j)^2 + \lambda \sum_{j=1}^{p} |\beta_j| \tag{6.4}$$

One can show that

$$\frac{\partial}{\partial \beta_j} \text{RSS} = -2(y_j - \beta_j) \tag{6.5}$$

Adding in the penalty term, we find that the subderivative is given by

$$\begin{aligned} \frac{\partial}{\partial \beta_j} f(\boldsymbol{\beta}) &= -2(y_j - \beta_j) + \lambda \frac{\partial}{\partial \beta_j} |\beta_j| \\ &= \begin{cases} -2(y_j - \beta_j) + \lambda & \text{if } \beta_j > 0, \\ [-2y_j - \lambda, -2y_j + \lambda] & \text{if } \beta_j = 0, \\ -2(y_j - \beta_j) - \lambda & \text{if } \beta_j < 0, \end{cases} \end{aligned}$$

Depending on the value of y_j , the solution to $\frac{\partial}{\partial \beta_j} f(\boldsymbol{\beta}) = 0$ can occur at 3 different values of β_j , as follows:

$$\beta_j = \begin{cases} y_j - \frac{\lambda}{2} & \text{if } y_j > \frac{\lambda}{2}, \\ 0 & \text{if } -\frac{\lambda}{2} < y_j < \frac{\lambda}{2}, \\ y_j + \frac{\lambda}{2} & \text{if } y_j < -\frac{\lambda}{2}, \end{cases}$$

If $\lambda = 1$, then the two dimensional figure of β_j vs. y_j,

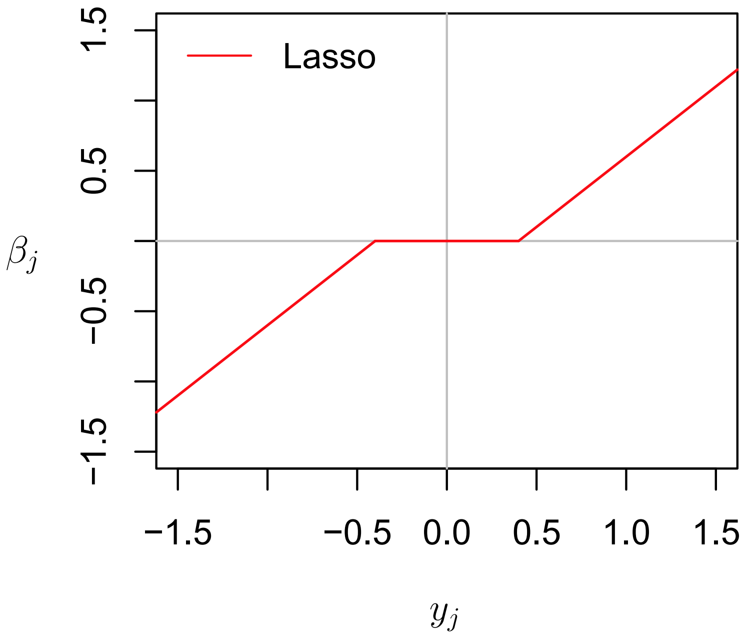

Figure 3

6.2.2.3 $\ell 1$ regularization: algorithms

Now, we derive the general form of lasso, the RSS,

$$\text{RSS} = \sum_{i=1}^{n} (y_i - \sum_{j=1}^{p} \beta_j x_{ij})^2 \tag{6.6}$$

The partial derivative of β_j on RSS,

$$\begin{aligned}
\frac{\partial}{\partial \beta_j}\mathrm{RSS} &= 2\sum_{i=1}^{n}\left(y_i - \sum_{j=1}^{p}\beta_j x_{ij}\right)(-x_{ij}) \\
&= 2\sum_{i=1}^{n}\left(y_i - \sum_{z\neq j}^{p}\beta_z x_{iz} - \beta_j x_{ij}\right)(-x_{ij}) \\
&= 2\sum_{i=1}^{n} x_{ij}^2 \beta_j - 2\sum_{i=1}^{n} x_{ij}(y_i - \sum_{z\neq j}^{p}\beta_z x_{iz}) \\
&= a_j\beta_j - c_j \\
a_j &= 2\sum_{i=1}^{n} x_{ij}^2 \\
c_j &= 2\sum_{i=1}^{n} x_{ij}(y_i - \sum_{z\neq j}^{p}\beta_z x_{iz})
\end{aligned}$$

where z is the index represent the coefficients without j. Adding in the penalty term, we find that the subderivative is given by,

$$\begin{aligned}
\frac{\partial}{\partial \beta_j} f(\boldsymbol{\beta}) &= (a_j\beta_j - c_j) + \lambda\frac{\partial}{\partial \beta_j}|\beta_j| \\
&= \begin{cases} (a_j\beta_j - c_j) - \lambda & \text{if } \beta_j < 0, \\ [-c_j - \lambda, -c_j + \lambda] & \text{if } \beta_j = 0, \\ (a_j\beta_j - c_j) + \lambda & \text{if } \beta_j > 0, \end{cases}
\end{aligned}$$

Depending on the value of c_j , the solution to $\frac{\partial}{\partial \beta_j} f(\boldsymbol{\beta}) = 0$ can occur at 3 different values of β_j , as follows:

$$\hat{\beta}_j = \begin{cases} \frac{c_j+\lambda}{a_j} & \text{if } c_j < -\lambda, \\ 0 & \text{if } -\lambda < c_j < \lambda, \\ \frac{c_j-\lambda}{a_j} & \text{if } c_j > \lambda, \end{cases}$$

We can write this as follows:

$$\hat{\beta}_j = \mathrm{soft}(\frac{c_j}{a_j}; \frac{\lambda}{a_j})$$

where

$$\mathrm{soft}(a;\delta) \triangleq \mathrm{sign}(a)(|a| - \delta)_+$$

Note that $x_+ = \max(x, 0)$ is the positive part of x and the definition of sign,

$$\text{sign}(x) = \begin{cases} -1 & \text{if } x < 0, \\ 0 & \text{if } x = 0, \\ 1 & \text{if } x > 0, \end{cases}$$

6.2.2.4 Shooting algorithm

Algorithm 4 Shooting algorithm

```
Initialize $\boldsymbol{\beta} = (\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I})^{-1}\mathbf{X}^T\mathbf{y}$;
repeat
    for $j = 1, \dots, D$ do
        $a_j = 2\sum_{i=1}^{n} x_{ij}^2$
        $c_j = 2\sum_{i=1}^{n} x_{ij}(y_i - \boldsymbol{\beta}^T\mathbf{x}_i + \beta_j x_{ij})$
        $\beta_j = \text{soft}(\frac{c_j}{a_j}; \frac{\lambda}{a_j})$
    end for
until calculate loss function to check converged;
```

The problem with Shooting Algorithm is that it only updates one variable at a time, so can be slow to converge. LARS can compute $\hat{\beta}(\lambda)$ for all possible values of λ in an efficient manner. To see more reference on LARS and lasso. [1]

Reference

張翔 (2012), 《提綱契領學統計》, 四版, 鼎茂圖書

[1] Lasso for logistic regression (see Yuan et al. 2010). And see (Schmidt et al. 2009; Yuan et al. 2010; Yang et al. 2010) for some recent surveys

Friedman, J., Hastie, T., & Tibshirani, R. (2001). *The elements of statistical learning* (Vol. 1). Springer, Berlin: Springer series in statistics.

James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013). *An introduction to statistical learning* (Vol. 6). New York: springer.

Wasserman, L. (2013). *All of statistics: a concise course in statistical inference.* Springer Science & Business Media.

J Li, Linear Discriminant Analysis, http://sites.stat.psu.edu/~jiali/course/stat597e/notes2/lda.pdf