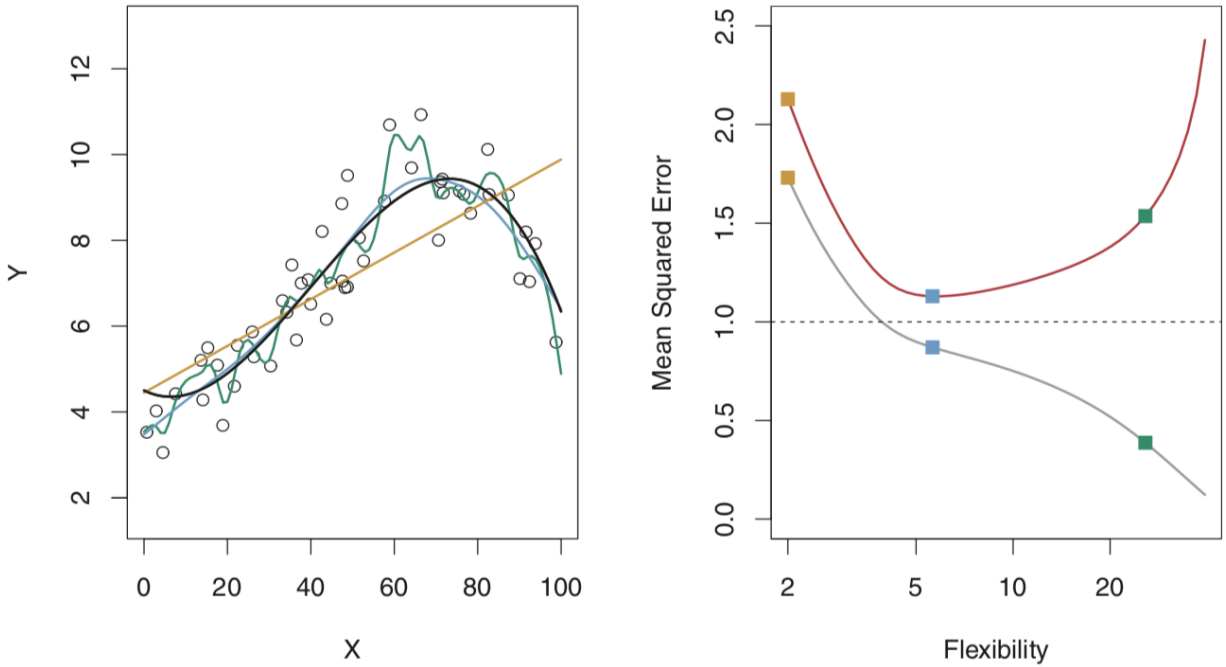
STAT562 L9 Regularization

Beidi Qiang

SIUE

Recall the following curve, which shows the loss (error) for both the training set and test set against the model complexity.



You see that training loss gradually decreases, but test loss eventually goes up when we overfit. We could prevent overfitting by penalizing complex models, a principle called regularization.

Regularization Methods

Instead of simply aiming to minimize loss, we'll now minimize loss plus complexity, which is called structural risk minimization:

$$minimize\Big(Loss(Data|Model) + \lambda \times Complexity(model)\Big)$$

Our training optimization algorithm is now a function of two terms: the loss term, which measures how well the model fits the training data, and the regularization term, which measures model complexity.

Two common ways:

- Model complexity as a function of the weights of all the features in the model: use L_2 penalty for model weights.
- Model complexity as a function of the total number of features with nonzero weights: use L_1 penalty for model weights.

Regularization rate, λ

In practice, we'd like to tune the overall impact of the regularization term by multiplying its value by a scalar known as λ. This is also called the regularization rate.

- Increasing the λ value strengthens the regularization effect.
- When choosing a λ value, the goal is to strike the right balance between simplicity and training-data fit.
- The ideal value of λ is data-dependent, so you'll need to do some tuning, using a validation set or method like using cross-validation.

L2 Regularization and Ridge Regression

Suppose we have a model with weights $w_1, \cdots, w_p$.

$$\text{L2 regularization term} = ||\mathbf{w}||^2 = w_1^2 + \cdots + w_p^2$$

Apply this to the linear regression problem, where the model is

$$\hat{y} = \beta_0 + \beta_1 x_1 + \cdots + \beta_p x_p$$

Previously, we find the estimates of β using least squares estimates, which are the values minimizing

$$\text{SSE (sum of squared error)} = \sum_{i=1}^{n} (y_i - \beta_0 - \beta_1 x_{i1} + \cdots \beta_p x_{ip})^2$$

Adding the L2 regularization, we now find estimates of βs by minimizing

$$SSE + \lambda \sum_{i=1}^{p} \beta_j^2$$

This is called a **Ridge Regression**.

Remarks: Ridge Regression

- Ridge regression is sensitive to the scale of predictors. We recommend standardizing predictors by their standard deviation.
- Bias-Variance trade-off: As λ increases, the flexibility of the ridge regression decreases, leading to decreased variance but increased bias.
- Will shrink estimated coefficients towards 0 but will not reach 0 exactly. Ridge regression will include all predictors in the final model.

L1 Regularization and the LASSO

In some problems, feature vectors can often be high-dimensional. It would be nice to encourage weights to drop to exactly 0 where possible. A weight of exactly 0 essentially removes the corresponding feature from the model. To achieve this, we introduce L1 regularization, that is penalizing the absolute value of all the weights.

$$\text{L1 regularization term} = |\mathbf{w}| = |w_1| + \cdots + |w_p|$$

Apply this regularization to the linear regression problem, we find estimates of βs by minimizing

$$SSE + \lambda \sum_{i=1}^{p} |\beta_j|$$

This is called **LASSO**.

Comparing Ridge and LASSO Constraints

An alternative formulation for Ridge and the Lasso is to minimize SSE subject to the constraint $\sum(\beta_j)^2 \leq s$ or $\sum|\beta_j| \leq s$, respectively.

- Consider $p = 2$. The constraint with LASSO is a diamond shape and the constraint with Ridge is a circle.

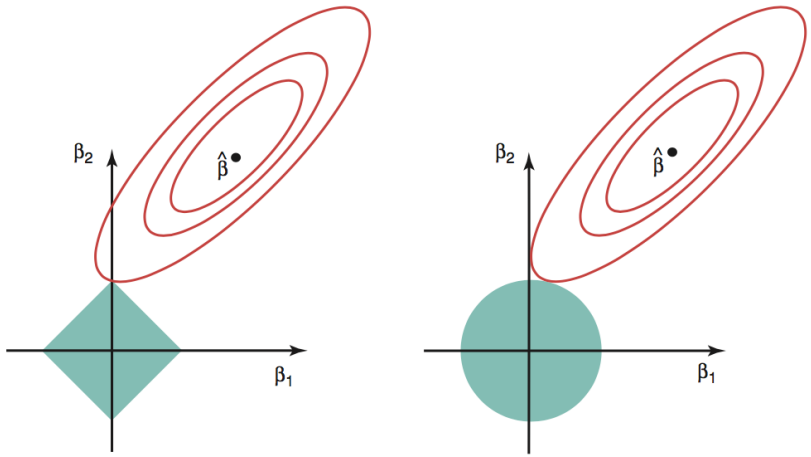


FIGURE 6.7. *Contours of the error and constraint functions for the lasso* (left) *and ridge regression* (right). *The solid blue areas are the constraint regions,* $|\beta_1| + |\beta_2| \leq s$ *and* $\beta_1^2 + \beta_2^2 \leq s$, *while the red ellipses are the contours of the RSS.*

Comparing Ridge and LASSO Cont.

- Lasso has similar behavior to ridge regression, in that as λ increases, the variance decreases and the bias increases.
- Lasso performs variable selection. It produces simpler models that are more interpretable.
- We expect the LASSO to perform better in a setting where a small number of predictors have substantial coefficients. And Ridge regression will perform better when the response is a function of many predictors, all with coefficients of roughly equal size.

Example: Credit Data

```
> library(glmnet)
> x=model.matrix(Balance~Income+Limit+Rating+Student,data=Credit)[,-1]
> y=Credit$Balance
> ridge10=glmnet(x,y,alpha=0,lambda=10, standardize=T)
> coef(ridge10)
5 x 1 sparse Matrix of class "dgCMatrix"
                     s0
(Intercept) -482.630613
Income        -7.138192
Limit          0.127221
Rating         1.920824
StudentYes   411.816682
> ridge80=glmnet(x,y,alpha=0,lambda=80, standardize=T)
> coef(ridge80)
5 x 1 sparse Matrix of class "dgCMatrix"
                      s0
(Intercept) -349.6774040
Income        -3.7327840
Limit          0.0996093
Rating         1.4981519
StudentYes   350.2088421
> lasso10=glmnet(x,y,alpha=1,lambda=10, standardize=T)
> coef(lasso10)
5 x 1 sparse Matrix of class "dgCMatrix"
                      s0
(Intercept) -467.3726020
Income        -6.5616366
Limit          0.1155487
Rating         1.9674922
StudentYes   385.6329164
> lasso80=glmnet(x,y,alpha=1,lambda=80, standardize=T)
> coef(lasso80)
5 x 1 sparse Matrix of class "dgCMatrix"
                      s0
(Intercept) -205.25360726
Income         .
Limit          0.02819233
Rating         1.62980111
StudentYes   132.79423756
```

Selecting the Tuning Parameter

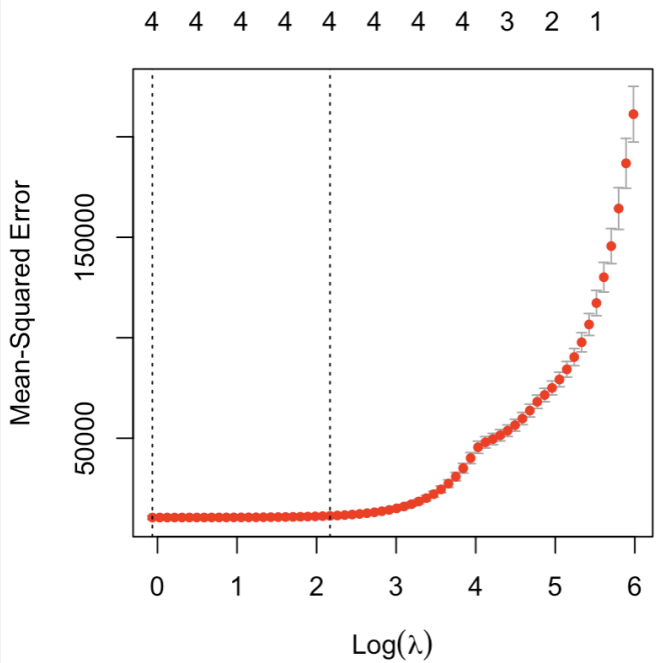
Cross-validation provides a simple way to selection tuning parameter. We choose a grid of λ values, and compute the cross-validation error for each value of λ.

```
> library(glmnet)
> x=model.matrix(Balance~Income+Limit+Rating+Student,data=Credit)[,-1]
> y=Credit$Balance
> cv.lasso=cv.glmnet(x,y,alpha=1)
> plot(cv.lasso)
> cv.lasso

Call:  cv.glmnet(x = x, y = y, alpha = 1)

Measure: Mean-Squared Error

    Lambda Index Measure    SE Nonzero
min  0.938    66   10523 931.2       4
1se  8.745    42   11367 729.3       4
> cv.lasso$lambda.min
[1] 0.9376683
>
```



Applying Regularization to Logistic Regression

You may also apply the L1 or L2 regularization in logistic regression when we have a classification problem.

```
> x=model.matrix(default~.,data=Default)[,-1]
> y=Default$default
> lasso=glmnet(x,y,alpha=1,family="binomial",lambda=0.005, standardize=T)
> coef(lasso)
4 x 1 sparse Matrix of class "dgCMatrix"
                      s0
(Intercept) -9.113074995
studentYes  -0.119292143
balance      0.004561439
income       .
> cv=cv.glmnet(x,y,alpha=1,family="binomial",standardize=T,type.measure="class")
> cv

Call:  cv.glmnet(x = x, y = y, type.measure = "class", alpha = 1, family = "binom
ial",      standardize = T)

Measure: Misclassification Error

      Lambda Index Measure       SE Nonzero
min 0.000870    47  0.0266 0.001973       3
1se 0.006137    26  0.0282 0.001837       2
> plot(cv)
>
```