Data Mining

HW#6: Linear Model Selection & Regularization

학번	182STG18
이름	이하경
제출일	2019.04.24

Description

Linear Model Selection and Regularization

선형모형은 간단한 구조임에도 불구하고 해석이 용이하고 종종 예측에서 좋은 퍼포먼스를 보인다. 선형 모형에서 회귀계수들의 Least Squares 방법으로 추정된 최소 제곱 추정치(OLS)는 기댓값이 실제 회귀계수와 일치하는 불편추정량이며 동시에 가장 작은 분산을 가지는 BLUE(Best Linear Unbiased Estimator)이다.

그러나 설명변수가 많은 경우 분산의 불안정성을 해소하고 많은 변수들 중 필요없는 변수를 제거하거나 영향을 줄임으로써 모형의 해석과 정확도 향상을 기대할 수 있다. 본 과제에서는 OLS 추정 방법을 변형한 다양한 추정 방법들의 적합 과정을 알아보고 실제 데이터에 적용해 예측력을 비교한다.

Subset Selection	Best Subset Regression Stepwise Regression Forward Selection Backward Elimination	p개의 변수들 중 반응변수에 대한 예측력이 좋을 것이라고 예상되는 변수의 조합을 선택하여 선택한 변수들에 대해 OLS 방법으로 회귀계수를 추정한다.
Shrinkage (Regularization)	Ridge Lasso	p개의 변수들을 모두 사용해 적합하지만 목적함수에 회귀계수 크기에 대한 penalty를 추가해 일부 설명변수들의 회귀계수를 0에 가깝게 축소한다.
Dimension Reduction	PCR PLS	p개의 설명변수들의 선형결합으로 이루어진 M개의 component들을 사용해 모형을 적합함으로써 변수들에 대한 정보 손실을 줄이면서도 차원을 축소할 수 있다.

Results

Chapter 6 Lab

6장의 Lab에서는 위에서 설명한 다양한 방법의 모형 선택 방법을 R에서 직접 실행하기 위한 몇 가지 패키지를 설치하고 내장된 함수들을 사용해 두개의 데이터 셋에 대해 선형 회귀모형을 찾고 CV Error을 계산해 모형 별 성능을 비교하였다.

Subset Selection	library(leaps)	regsubsets	p개의 설명변수의 가능한 모든 조합에 대해 p가 동일한 모형들 중에서는 가장 RSS가 작은 모형을 하나씩 찾고 이 p개의 모형들 중 R^2, Adjusted R^2, C_p, BIC 등을 비교해 최적 모형을 찾는다.
Shrinkage	library(glmnet)	glmnet	alpha = 0일 경우 Ridge, alpha = 1일 경우 Lasso 방법으로 지정한 lambda의 범위에 따라 회귀계수를 추정한다.
		cv.glmnet	K-fold CV를 통해 가장 CV Error가 작은 lambda를 찾을 수 있다.
Dimension Reduction	library(pls)	pcr	validation = 'CV'를 지정함으로써 K-fold CV를 이용해 validation error를 가장 작게 하는 components의 개수를 결정할 수 있다.
		plsr	
		validationplot	사용하는 Components의 개수에 따른 CV Error을 나타내는 그래프

Exercise 6.1 Best Subset, Forward Stepwise, Backward Stepwise Selection

(a) Which of the 3 models with k predictors has the smallest training RSS?

Forward Selection은 Null 모형 또는 Full 모형에서부터 변수를 하나씩 추가, 제거하므로 Best Subset Selection에 비해 고려하는 경우의 수가 적다. 따라서 Forward와 Backward가 찾는 모든 모형을 포함해 p개의 설명변수들의 가능한 조합을 모두 고려하는 Best Subset Selection 방법이 가장 training RSS가 작을 것이다.

(b) Which of the 3 models with k predictors has the smallest test RSS?

Test set 에 대한 RSS 는 3 가지 모형 중 어떤 모형이 가장 작을지 알 수 없으며 세 모형 모두 가능하다. Best Subset Selection 은 모든 조합 중 Training RSS 를 가장 줄이는 모형을 찾을 수 있지만 과대적합되는 경우 오히려 Test RSS 가 크게 나타날 수 있다.

(c) True or False

i. The predictors in the k-variable model identified by forward stepwise are a subset of the predictors in the (k+1)-variable model identified by forward stepwise selection.	**True**
ii. The predictors in the k-variable model identified by backward stepwise are a subset of the predictors in the (k+1)-variable model identifies by backward stepwise selection.	**True**
iii. The predictors in the k-variable model identified by backward stepwise are a subset of the predictors in the (k+1)-variable model identifies by forward stepwise selection.	**False**
iv. The predictors in the k-variable model identified by forward stepwise are a subset of the predictors in the (k+1)-variable model identifies by backward stepwise selection.	**False**
v. The predictors in the k-variable model identified by best subset are a subset of the predictors in the (k+1)-variable model identifies by best subset selection.	**False**

Forward Stepwise 에서는 k 개의 변수를 포함한 모형에서 k+1 번째로 모형에 포함할 변수를 결정하므로 k 개로 이루어진 모형은 항상 k+1 개의 변수를 포함한 모형의 subset 이다.

Backward Stepwise 에서는 k+1 개의 변수를 포함한 모형에서 제거할 하나의 변수를 선택하여 k 개로 이루어진 모형을 찾으므로 이 역시 k+1 개 모형의 subset 이다.

Exercise 6.2

(a) The Lasso, relative to least squares, is (iii) less flexible and hence will give improved prediction accuracy when its increase in bias is less than its decrease in variance.

(b) The Ridge Regression, relative to least squares, is (iii) less flexible and hence will give improved prediction accuracy when its increase in bias is less than its decrease in variance.

Ridge 와 Lasso 모두 회귀계수의 크기에 대해 penalty term 을 목적함수에 추가해 OLS 보다 덜 복잡한 모형을 적합한다. 이 목적함수를 최소화하는 추정치들은 OLS 와 달리 bias 가 존재하지만 회귀계수들의 분산을 크게 줄임으로써 더 안정적인 추정을 할 수 있다.

(c) Non-Linear methods relative to least squares are (ii) more flexible and hence will give improved prediction accuracy when its increase in variance is less than its decrease in bias.

비선형 모형(NLS)은 OLS 에 비해 더 복잡한 모형이 되며 모형 적합에 사용되는 데이터에 대해 더 정확한 추정이 가능하나 복잡한 모형일수록 회귀계수 추정치의 bias 는 줄어들고 variance 는 증가한다.

Exercise 6.5

$$\mathbf{Y} = \mathbf{X\beta} + \boldsymbol{\epsilon}$$

$$\begin{bmatrix} \mathbf{y_1} \\ \mathbf{y_2} \end{bmatrix} = \begin{bmatrix} \mathbf{1} & \boldsymbol{x_{11}} & \boldsymbol{x_{12}} \\ \mathbf{1} & \boldsymbol{x_{21}} & \boldsymbol{x_{22}} \end{bmatrix} \begin{bmatrix} \boldsymbol{\beta_0} \\ \boldsymbol{\beta_1} \\ \boldsymbol{\beta_2} \end{bmatrix} + \begin{bmatrix} \boldsymbol{\epsilon_1} \\ \boldsymbol{\epsilon_2} \end{bmatrix}, \qquad \rightarrow \hat{\boldsymbol{\beta}}_{\mathbf{0}} = \mathbf{0}$$

$$\mathbf{x_{11}} = \mathbf{x_{12}}, \qquad \mathbf{x_{21}} = \mathbf{x_{22}}, \qquad \mathbf{y_1} + \mathbf{y_2} = \mathbf{0}, \qquad \mathbf{x_{11}} + \boldsymbol{x_{21}} = \mathbf{0}$$

(a) Ridge Optimization Problem

$$\text{To Minimize} \quad (y_1 - \beta_1 x_{11} - \beta_2 x_{12})^2 + (y_1 - \beta_1 x_{11} - \beta_2 x_{12})^2 + \lambda(\beta_1^2 + \beta_2^2)$$

(b) Ridge Coefficients Estimates $\hat{\boldsymbol{\beta}}_{\mathbf{1}} = \hat{\boldsymbol{\beta}}_{\mathbf{2}}$

$$L_{Ridge} = (y_1 - \beta_1 x_{11} - \beta_2 x_{12})^2 + (y_1 - \beta_1 x_{11} - \beta_2 x_{12})^2 + \lambda(\beta_1^2 + \beta_2^2)$$

$$= \left(y_1 - (\beta_1 + \beta_2)x_{11}\right)^2 + \left(y_2 - (\beta_1 + \beta_2)x_{21}\right)^2 + \lambda(\beta_1^2 + \beta_2^2) = 2\left(y_1 - (\beta_1 + \beta_2)x_{11}\right)^2 + \lambda(\beta_1^2 + \beta_2^2)$$

$$= 2y_1^2 - 4\beta_1 y_1 x_1 - 4\beta_2 y_1 x_1 + \beta_1^2 x_{11}^2 + 2\beta_1\beta_2 x_{11}^2 + \beta_2^2 x_{11}^2 + \lambda\beta_1^2 + \lambda\beta_2^2$$

$$i)\ \frac{\partial L_{Ridge}}{\partial \beta_1} = -4y_1 x_1 + 2x_{11}^2\beta_1 + 2x_{11}^2\beta_1 + 2\lambda\beta_1 = 0$$

$$ii)\ \frac{\partial L_{Ridge}}{\partial \beta_2} = -4y_1 x_1 + 2x_{11}^2\beta_2 + 2x_{11}^2\beta_2 + 2\lambda\beta_2 = 0$$

$$(\because)\ \lambda\hat{\beta}_1 = \lambda\hat{\beta}_2, \quad \hat{\beta}_1 = \hat{\beta}_2$$

(b) Lasso Optimization Problem

$$\text{To Minimize} \quad (y_1 - \beta_1 x_{11} - \beta_2 x_{12})^2 + (y_1 - \beta_1 x_{11} - \beta_2 x_{12})^2 + \lambda(|\beta_1| + |\beta_2|)$$

(b) Ridge Coefficients Estimates $\hat{\boldsymbol{\beta}}_{\mathbf{1}} = \hat{\boldsymbol{\beta}}_{\mathbf{2}}$

$$L_{Lasso} = (y_1 - \beta_1 x_{11} - \beta_2 x_{12})^2 + (y_1 - \beta_1 x_{11} - \beta_2 x_{12})^2 + \lambda(|\beta_1| + |\beta_2|)$$

$$i)\ \frac{\partial L_{Lasso}}{\partial \beta_1} = -4y_1 x_1 + 2x_{11}^2\beta_1 + 2x_{11}^2\beta_1 + \lambda\frac{|\beta_1|}{\beta_1} = 0$$

$$ii)\ \frac{\partial L_{Lasso}}{\partial \beta_2} = -4y_1 x_1 + 2x_{11}^2\beta_2 + 2x_{11}^2\beta_2 + \lambda\frac{|\beta_2|}{\beta_2} = 0$$

$(\because)\ \frac{|\hat{\beta}_1|}{\hat{\beta}_1} = \frac{|\hat{\beta}_2|}{\hat{\beta}_2}$, $\hat{\beta}_1$과 $\hat{\beta}_2$의 부호가 동일하다면 해가 여러 개 존재한다.

Exercise 6.9 College Data Set

(a) Split the data set into a training set and a test set

모형 적합에 동일하게 사용할 training set 과 예측오차를 평가할 test set 을 각각 50%로 분할하였다. College 데이터는 총 777 개의 관측치를 포함하므로 training set 388 개, test set 389 개로 나누었다.

(b) Least Squares Fit & Test Error

Coefficients

(Intercept)	PrivateYes	Accept	Enroll	Top10perc	Top25perc	F.Undergrad	P.Undergrad	Outstate
78.152	-757.228	1.680	-0.624	67.457	-22.375	-0.061	0.048	-0.092
Room.Board	**Books**	**Personal**	**PhD**	**Terminal**	**S.F.Ratio**	**perc.alumni**	**Expend**	**Grad.Rate**
0.245	0.091	0.059	-8.890	-1.720	-5.752	-1.467	0.035	7.576
							Test MSE	**1108531**

(c) Ridge Regression Fit, with λ chosen by CV & Test Error

Coefficients

(Intercept)	PrivateYes	Accept	Enroll	Top10perc	Top25perc	F.Undergrad	P.Undergrad	Outstate
-711.641	-577.008	1.051	0.433	34.712	-0.975	0.048	0.036	-0.030
Room.Board	**Books**	**Personal**	**PhD**	**Terminal**	**S.F.Ratio**	**perc.alumni**	**Expend**	**Grad.Rate**
0.267	0.236	-0.029	-2.473	-6.512	0.364	-9.203	0.053	0.267
							Test MSE	**1037616**
							Best λ	**450.743**

최적의 λ 을 찾기 위해 10-Fold CV 를 실행해 CV Error 가 가장 작은 값을 선택하였다.

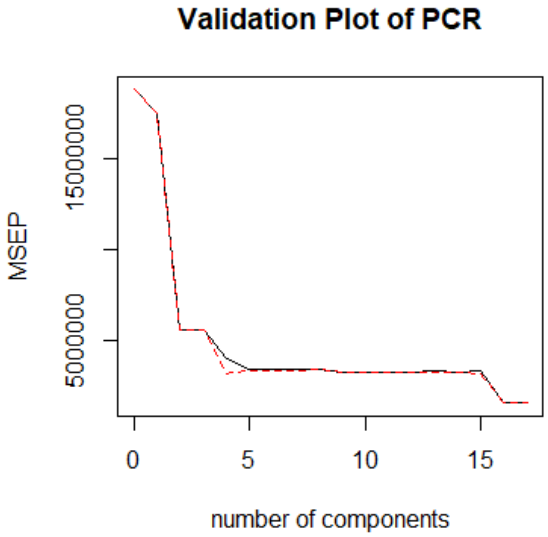
(d) Lasso Regression Fit, with λ chosen by CV & Test Error

Coefficients

(Intercept)	PrivateYes	Accept	Enroll	Top10perc	Top25perc	F.Undergrad	P.Undergrad	Outstate
160.052	-524.793	1.560	-0.454	50.098	-9.659	-0.009	**0.000**	-0.057
Room.Board	**Books**	**Personal**	**PhD**	**Terminal**	**S.F.Ratio**	**perc.alumni**	**Expend**	**Grad.Rate**
0.195	0.019	0.003	-4.617	-3.139	**0.000**	-2.166	0.033	0.195
							Test MSE	**1030941**
							Best λ	**24.621**

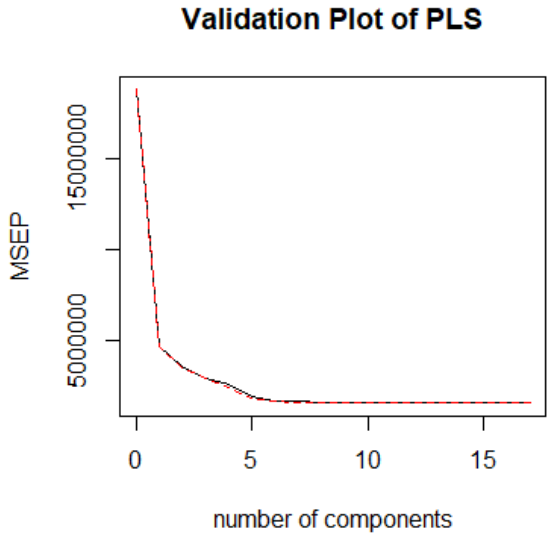
추정된 회귀계수가 0 이 아닌 설명변수는 총 15 개이다.

(e) PCR Fit, with M chosen by CV & Test Error



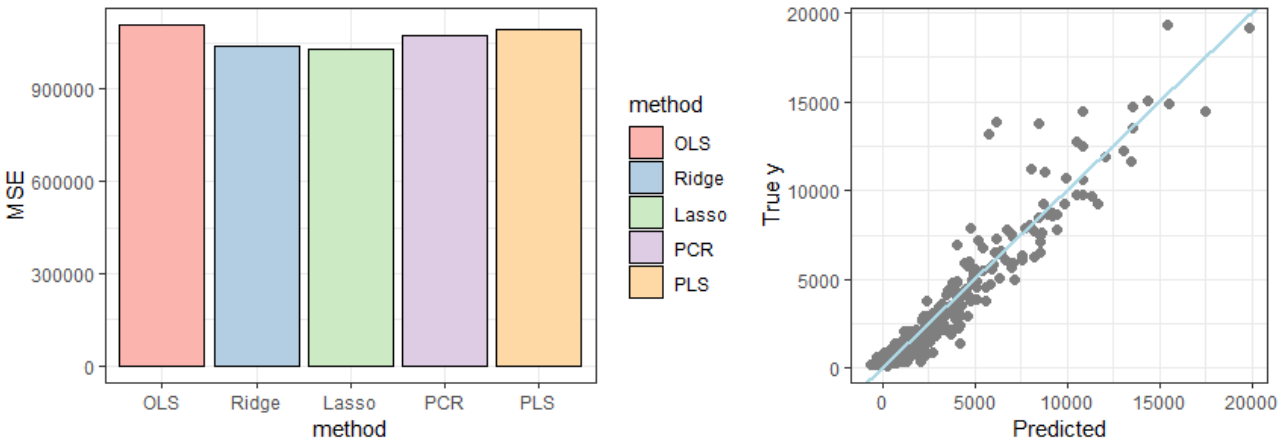
Test MSE	**1075293**
Best M	**16**

(f) PLS Fit, with M chosen by CV & Test Error



Test MSE	**1093438**
Best M	**10**

(g) Summary of Results



5 가지 모형의 Test Error 을 비교하였을 때 OLS 모형에서의 Test Error 보다 나머지 4 가지 방법에서의 최종 선택된 모형의 Test Error 가 더 낮았다. 방법 별로 큰 차이는 보이지 않았지만 Ridge 와 Lasso 방법을 사용하였을 때 Test Error 가

PCR 과 PLS 를 사용하였을 때보다 더 낮았다. 그 중 Lasso 모형의 Test Error 가 가장 작으므로 해당 모형에서 추정된 Test set 의 예측치와 실제 값의 산점도를 그려보았을 때 점들이 대부분 직선 주변에 분포하여 예측이 적절하게 되었다고 할 수 있다.

Exercise 6.11 Boston Data Set (Predict per capita crime rate)

(a) Results of some regression methods

먼저 validation approach 를 위해 Boston 데이터를 50%의 training set 과 50%의 test set 으로 분할하고, train 데이터만을 이용해 각 모형을 적합하였다.

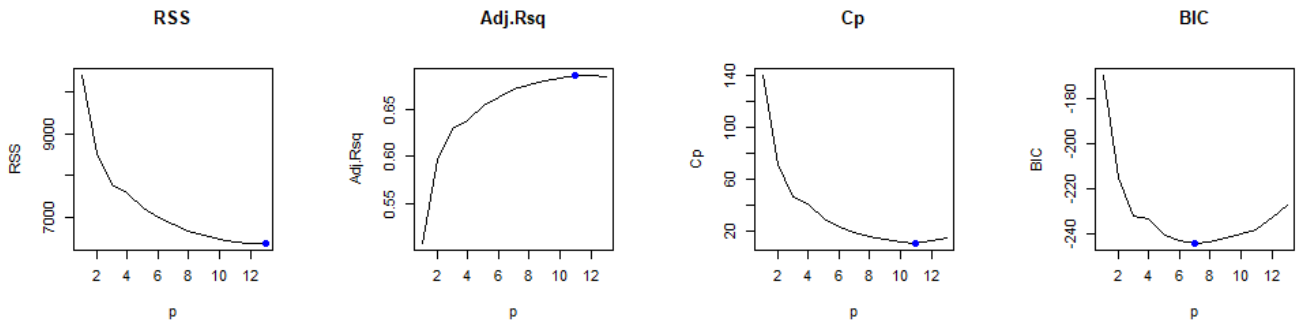
(1) OLS

Coefficients

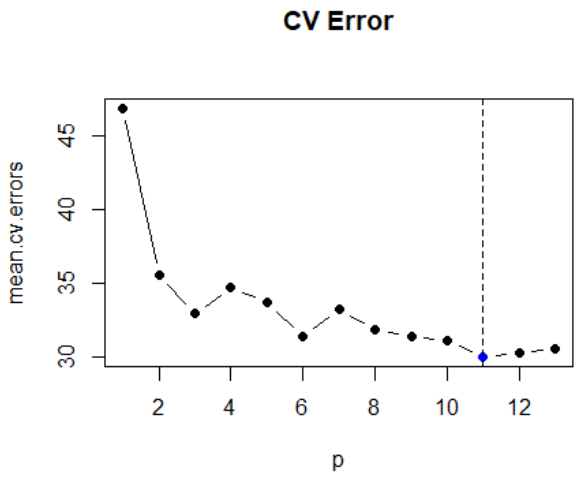
(Intercept)	crim	zn	indus	chas	nox	rm
38.915	-0.159	0.042	0.066	2.170	-20.906	3.865
age	**dis**	**rad**	**tax**	**ptratio**	**black**	**lstat**
0.001	-1.397	0.413	-0.015	-1.013	0.008	-0.503
					Test MSE	**19.056**

(2) Best Subset Selection

p=1 부터 p=19 까지 p 의 개수가 동일한 모형들 사이에서는 RSS 가 가장 작은 모형을 뽑고 추출된 19 개 모형의 Adjusted R^2, Cp, BIC 등을 비교하였다. RSS 는 동일한 데이터에 대한 적합에서는 p 가 커질수록 계속해서 감소한다. Adjusted R^2, Cp 기준으로는 p=11 모형이 제일 좋았고 BIC 기준으로는 p=7 모형이 제일 좋았다. Cp 와 AIC 는 동일한 모형을 선택하므로 BIC 기준 최적 모형이 Cp 기준 최적 모형과 비교했을 때 더 간단한 모형을 선택하는 것을 알 수 있다. 이것은 BIC 가 모형의 복잡성에 대한 penalty 를 비교적 크게 주기 때문이다.



또한 K=10 fold CV 를 위해 training set 을 다시 10 개의 set 으로 나누고 각 k 번째 set 에 대한 test error 을 계산하고 p 에 따른 평균 CV error 을 비교하였다. 이 경우 역시 p=11 인 모형에서 가장 퍼포먼스가 좋았다.



변수 선택방법을 'forward', 'backward'로 설정했을 때 역시 최종 선택된 모형은 동일하였다.
Best Subset Regression 의 최종 모형에 포함된 11 개의 변수 및 계수는 다음과 같다.

Coefficients

(Intercept)	crim	zn	chas	nox	rm	dis
38.521	-0.159	0.040	2.278	-19.532	3.835	-1.441
rad	**tax**	**ptratio**	**black**	**lstat**		
0.396	-0.014	-0.995	0.008	-0.500		
					Test MSE	**18.964**

(3) Ridge

Coefficients

(Intercept)	crim	zn	indus	chas	nox	rm
28.644	-0.128	0.029	-0.008	2.414	-14.047	4.038
age	**dis**	**rad**	**tax**	**ptratio**	**black**	**lstat**
-0.001	-1.003	0.180	-0.005	-0.868	0.007	-0.471
					Test MSE	**19.246**
					Best λ	**0.720**

10-fold CV 로 CV error 가 가장 작은 λ를 선택하고 Ridge 모형의 추정에 이용하였다.

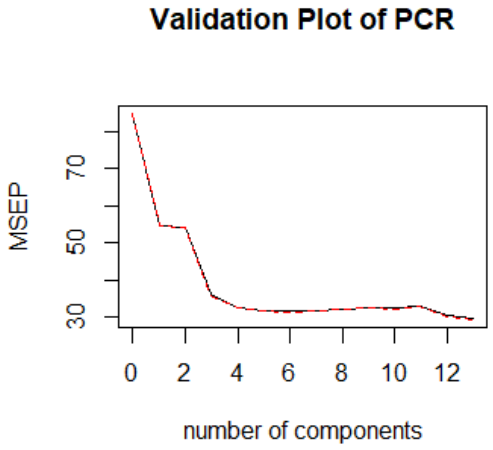
(4) Lasso

Coefficients

(Intercept)	crim	zn	indus	chas	nox	rm
34.353	-0.142	0.031	**0.000**	2.252	-17.770	3.974
age	**dis**	**rad**	**tax**	**ptratio**	**black**	**lstat**
0.000	-1.248	0.270	-0.008	-0.954	0.007	-0.503
					Test MSE	**18.876**
					Best λ	**0.052**

Lasso 역시 10-fold CV 로 최적의 λ 를 찾았다.
p 개의 변수를 모두 사용하는 Ridge 와 다르게 일부 변수들의 계수가 0 이 되어 Lasso 가 변수 선택의 기능이 있음을 확인할 수 있다.

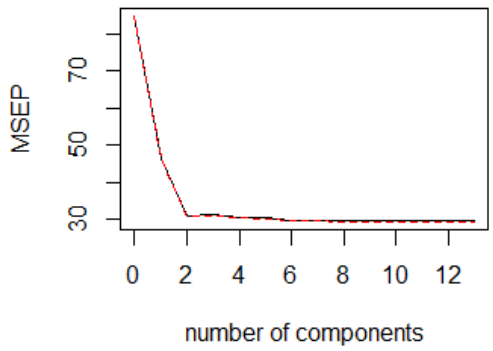
(5) PCR



Test MSE	**19.056**
Best M	**13**

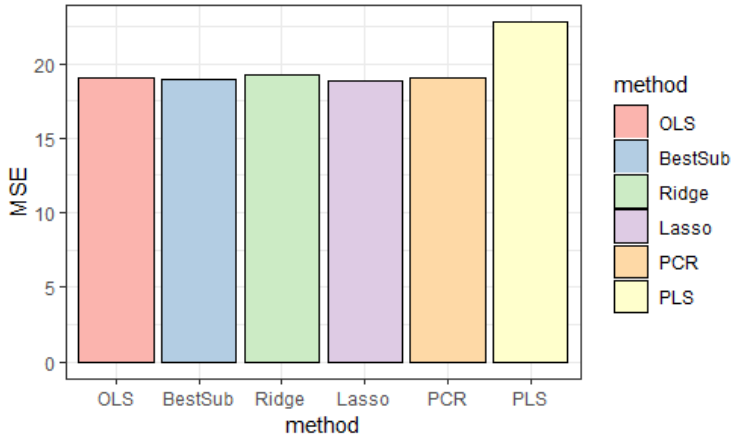
Boston 데이터에 대한 PCR 의 결과 component 를 설명변수의 개수와 동일하게 13 개까지 사용했을 때의 퍼포먼스가 가장 좋으므로 차원 축소의 효과는 없었다.

(6) PLS



Test MSE	**22.789**
Best M	**9**

(b) Propose a model that seem to perform well on the data (using Validation Approach)



Test Error(MSE) 비교 결과 OLS에 비해 Best Subset (p=11) 모형과 Lasso 모형이 비슷하게 더 작았다. 최종 Lasso 모형의 회귀계수 중 indus, age의 계수가 0이므로 best subset 모형과 동일한 변수들이 선택된 것을 알 수 있다. Lasso 모형의 Test Error가 가장 작으므로 Boston data set에서는 Lasso 모형의 퍼포먼스가 가장 좋은 것으로 예상된다.

(c) Does your chosen model involve all of the features in the data set?

최종 Lasso 모형은 p 개의 변수를 모두 사용한 OLS 추정 모형과 다르게 일부 회귀계수의 크기를 0 으로 축소하여 모형의 복잡성을 줄이는 효과가 있다. 따라서 indus, age 변수는 반응변수 예측에 활용되지 않으므로 모든 feature 을 다 포함한 모형이라고 할 수 없다.

Discussion

선형 모형의 회귀계수 추정에서 고려 가능한 다양한 방법으로 모형을 적합하고 서로 예측오차를 비교해 최종모형을 선택하는 예제들을 직접 실습하였다. 모형에 포함되는 변수가 많아질수록, 즉 모형이 복잡해질수록 회귀계수 추정치의 bias 는 줄어들지만 Variance 는 증가한다. (Bias-Variance Trade Off) Ridge 와 Lasso Regression 에서 추정된 회귀계수들은 기댓값이 실제 회귀계수와 다른 biased 추정치지만 약간의 bias 를 허용하는 대신 분산을 크게 줄일 수 있어 예측의 정확도를 높이길 기대하는 shrinkage 방법이다. Exercise 6.9 와 6.11 에서 일부 회귀계수의 크기를 0 으로 줄인 Lasso 모형이 고려한 모든 모형들 중 Test MSE 가 가장 작아 퍼포먼스가 좋음을 확인하였다. Best Subset Regression 은 고려 가능한 변수들의 조합에 대해 RSS 및 CV error 을 비교할 수 있으므로 역시 예측 퍼포먼스가 뛰어난 모형을 찾을 수 있었다.

그러나 데이터에 따라 고려한 모형들 중 어떤 모형에서 예측의 정확도가 가장 높을 지는 정해져 있지 않으며 예상할 수 없다. 주어진 데이터를 이용한 Cross-Validation 을 여러 번 실행해 최적 모형을 선택한다면 알려지지 않은 새로운 데이터에 대한 예측력 또한 좋을 것이라고 예상한다.

[Appendix] R code

Exercise 6.9

```
fix(College)
X <- data.matrix(College[,-2])
y <- College$Apps
MSE.College <- function(pred, test) mean((College$Apps[test]-pred)**2)

# (a) split the data set
set.seed(1)
train <- sample(1:nrow(College), nrow(College)/2)
test <- (-train)

# (b) least squares
(lmFit <- lm(Apps ~ ., College[train,]))
lmPred <- predict(lmFit, newdata = College[test,])
(mse.lm <- MSE.College(lmPred, test))
round(coef(lmFit), 3)
```

```
# (c) ridge
set.seed(1)
(cv.ridge <- cv.glmnet(X[train,], y[train], alpha = 0))
(best.lambda.ridge <- cv.ridge$lambda.min)
ridgeFit <- glmnet(X[train,], y[train], alpha = 0)
ridgePred <- as.numeric(predict(ridgeFit, type = "response", s = best.lambda.ridge, newx = X[test,]))
(mse.ridge <- MSE.College(ridgePred, test))

ridgeCoef <- predict(ridgeFit, type = "coefficients", s = best.lambda.ridge, newx = X[test,])[1:ncol(X),]
round(ridgeCoef, 3)

# (d) lasso
set.seed(1)
(cv.lasso <- cv.glmnet(X[train,], y[train], alpha = 1))
(best.lambda.lasso <- cv.lasso$lambda.min)
lassoFit <- glmnet(X[train,], y[train], alpha = 1)
lassoPred <- as.numeric(predict(lassoFit, type = "response", s = best.lambda.lasso, newx = X[test,]))
(mse.lasso <- MSE.College(lassoPred, test))

lassoCoef <- predict(lassoFit, type = "coefficients", s = best.lambda.lasso, newx = X[test,])[1:ncol(X),]
round(lassoCoef, 3)

# (e) PCR
set.seed(1)
summary(pcrFit <- pcr(Apps ~ ., data = College[train,], scale = TRUE, validation = "CV"))
validationplot(pcrFit, val.type = 'MSEP', main = 'Validation Plot of PCR') # M = 16

pcrFit <- pcr(Apps ~ ., data = College[train,], ncomp = 16)
pcrPred <- predict(pcrFit, X[test,], ncomp = 16)
(mse.pcr <- MSE.College(pcrPred, test))

# (f) PLS
set.seed(1)
summary(plsFit <- plsr(Apps ~ ., data = College[train,], scale = TRUE, validation = "CV"))
validationplot(plsFit, val.type = 'MSEP', main = 'Validation Plot of PLS') # M = 10

plsFit <- plsr(Apps ~ ., data = College[train,], ncomp = 10)
plsPred <- predict(plsFit, X[test,], ncomp = 10)
(mse.pls <- MSE.College(plsPred, test))

# (g)
mse.summary <- data.frame(method = c('OLS', 'Ridge', 'Lasso', 'PCR', 'PLS'),
                          MSE = c(mse.lm, mse.ridge, mse.lasso, mse.pcr, mse.pls))
mse.summary$method <- factor(mse.summary$method, mse.summary$method)
ggplot(mse.summary) + theme_bw() + scale_fill_brewer(palette = 'Pastel1') +
  geom_col(aes(method, MSE, fill = method), color = 'black')
ggplot() +theme_bw() + geom_point(aes(lassoPred, y[test]), color = 'gray50', size = 2) +
  geom_abline(aes(intercept = 0, slope = 1), col = 'lightblue', size = 1) +
  labs(x = 'Predicted', y = 'True y')
```

Exercise 6.11

```
fix(Boston)
X <- data.matrix(Boston[,-14])
y <- Boston$medv
MSE.Boston <- function(pred, test) mean((Boston$medv[test]-pred)^2)

# (a)
set.seed(10)
```

```
train <- sample(1:nrow(Boston), nrow(Boston)/2)
test <- (-train)

# OLS
lmFit <- lm(medv ~ ., Boston[train,])
lmPred <- predict(lmFit, Boston[test,])
(mse.lm <- MSE.Boston(lmPred, test))
round(coef(lmFit), 3)

# Best Subset
predict.regsubsets <- function(object, newdata, id, ...) {
  form <- as.formula(object$call[[2]])
  mat <- model.matrix(form, newdata)
  coefi <- coef(object, id = id)
  xvars <- names(coefi)
  mat[,xvars] %*% coefi
}
# stepwise
regFit <- regsubsets(medv ~ ., Boston[train,], nvmax = 13)
regSummary <- summary(regFit)
par(mfrow = c(1,4))
plot(regSummary$rss, type = 'l', pch = 19, xlab = 'p', ylab = 'RSS', main = 'RSS')
which.min(regSummary$rss)
points(13, regSummary$rss[13], col = 'blue', pch = 16)
plot(regSummary$adjr2, type = 'l', pch = 19, xlab = 'p', ylab = 'Adj.Rsq', main = 'Adj.Rsq')
which.max(regSummary$adjr2)
points(11, regSummary$adjr2[11], col = 'blue', pch = 16)
plot(regSummary$cp, type = 'l', pch = 19, xlab = 'p', ylab = 'Cp', main = 'Cp')
which.min(regSummary$cp)
points(11, regSummary$cp[11], col = 'blue', pch = 16)
plot(regSummary$bic, type = 'l', pch = 19, xlab = 'p', ylab = 'BIC', main = 'BIC')
which.min(regSummary$bic)
points(7, regSummary$bic[7], col = 'blue', pch = 16)

plot(regFit, scale = 'Cp')
coef(regFit, 11)

dtrain <- Boston[train,]
k <- 10
set.seed(1)
folds <- sample(1:k, nrow(dtrain), replace = TRUE)
cv.errors <- matrix(NA, k, 13, dimnames = list(NULL, paste(1:13)))
for(j in 1:k) {
  best.fit <- regsubsets(medv ~ ., data = dtrain[folds!=j,], nvmax = 13)
  for(i in 1:13) {
    pred <- predict(best.fit, dtrain[folds==j,], id = i)
    cv.errors[j,i] <- mean((dtrain$medv[folds==j]-pred)^2)
  }
}
(mean.cv.errors <- apply(cv.errors, 2, mean))

par(mfrow = c(1,1))
plot(mean.cv.errors, type = 'b', pch = 19, xlab = 'p', main = 'CV Error')
which.min(mean.cv.errors)
points(11, mean.cv.errors[11], col = 'blue', pch = 19)
abline(v = 11, lty = 2)

regPred <- predict(regFit, Boston[test,], 11)
(mse.best <- MSE.Boston(regPred, test))
round(coef(regFit, 11), 3)
```

```
# Ridge
set.seed(10)
(cv.ridge <- cv.glmnet(X[train,], y[train], alpha = 0))
(best.lambda.ridge <- cv.ridge$lambda.min)
ridgeFit <- glmnet(X[train,], y[train], alpha = 0)
ridgePred <- as.numeric(predict(ridgeFit, type = "response", s = best.lambda.ridge, newx = X[test,]))
(mse.ridge <- MSE.Boston(ridgePred, test))

ridgeCoef <- (predict(ridgeFit, type = "coefficients", s = best.lambda.ridge, newx = X[test,]))[1:14,]
round(ridgeCoef, 3)

# Lasso
set.seed(10)
(cv.lasso <- cv.glmnet(X[train,], y[train], alpha = 1))
(best.lambda.lasso <- cv.lasso$lambda.min)
lassoFit <- glmnet(X[train,], y[train], alpha = 1)
lassoPred <- as.numeric(predict(lassoFit, type = "response", s = best.lambda.lasso, newx = X[test,]))
(mse.lasso <- MSE.Boston(lassoPred, test))

lassoCoef <- (predict(lassoFit, type = "coefficients", s = best.lambda.lasso, newx = X[test,]))[1:14,]
round(lassoCoef, 3)

# PCR
set.seed(10)
pcrFit <- pcr(medv ~ ., data = Boston[train,], scale = TRUE, validation = "CV")
summary(pcrFit)
validationplot(pcrFit, val.type = 'MSEP', main = 'Validation Plot of PCR') # M = 13

pcrFit <- pcr(medv ~ ., data = Boston[train,], ncomp = 13)
pcrPred <- predict(pcrFit, X[test,], ncomp = 13)
(mse.pcr <- MSE.Boston(pcrPred, test))

# PLS
set.seed(10)
plsFit <- plsr(medv ~ ., data = Boston[train,], scale = TRUE, validation = "CV")
summary(plsFit)
validationplot(plsFit, val.type = 'MSEP', main = 'Validation Plot of PLS') # M = 9

plsFit <- plsr(medv ~ ., data = Boston[train,], ncomp = 9)
plsPred <- predict(plsFit, X[test,], ncomp = 9)
(mse.pls <- MSE.Boston(plsPred, test))

# (b)
mse.summary <- data.frame(method = c('OLS', 'BestSub', 'Ridge', 'Lasso', 'PCR', 'PLS'),
                          MSE = c(mse.lm, mse.best, mse.ridge, mse.lasso, mse.pcr, mse.pls))
mse.summary$method <- factor(mse.summary$method, mse.summary$method)
ggplot(mse.summary) + theme_bw() + scale_fill_brewer(palette = 'Pastel1') +
  geom_col(aes(method, MSE, fill = method), color = 'black')
ggplot() +theme_bw() + geom_point(aes(lmPred, y[test]), color = 'gray50', size = 2) +
  geom_abline(aes(intercept = 0, slope = 1), col = 'lightblue', size = 1) +
  labs(x = 'Predicted', y = 'True y')
```