2019 학년도 1학기
DATA MINING
HW6

과목명	데이터마이닝
담당교수명	송종우 교수님
제출일	2019.04.24
학번	182STG27
이름	임지연

Ⅰ. Description

이제까지 데이터에 linear 모형을 주로 공부했는데, linear 방법은 Additive, Non-linear 모형에 비하여 ‘추론’ 에서의 장점이 있다. 이제까지는 최적의 linear model을 찾기 위해서 Least Suqare 방법을 이용했지만 예측의 정확도와 모델 해석력 관점에서 최적의 모형을 찾기 위하여 다른 방법을 살펴볼 것이다. 넓은 관점에서 Subset selection, Shirinkage, Dimension reduction 방법을 공부할 것이다. 각각에 해당하는 모형의 원리를 이해하고 Lab 에 설명되어 있는 코드를 실행해 보고 결과를 이해한 후 연습문제를 풀어볼 것이다.

Ⅱ. Implementation

Question 1.

Lab : 1.Subset Selection Method, 2.Ridge Regression and Lasso, 3.PCR and PLS Regression 의 코드를 실행해보고 감상문을 써라

Hitters 데이터를 사용하여 야구선수의 Salary를 예측해 보았다. Linear Model Selection and Regularization 방법인 All possible Regression, Forward Selection, Backward Selection, Ridge, Lasso, PCR, PCA, PLS 방법에 대하여 실습을 하면서 방법을 이해하고 장단점을 이해할 수 있었다. R에서 cv.glmnet 함수를 이용해 대표적인 Shirinkage 방법인 Ridge, Lasso 모형을 적합할 수 있다는 것을 공부했다. 이 함수는 데이터를 넣어주면 CV 방법을 이용해 가장 최적의 λ 값을 구해준다. 하지만 사용 전 x, y 를 행렬, 벡터 형식으로 만들어 준 다음 적합해야 한다는 것을 알게 되었다. 다음으로는 대표적인 Dimension Reduction 방법인 PCR, PLS방법에 대해 pcr, plsr 함수를 이용해 PCR, PLS 모형을 적합해보았다. Validation = “CV” 옵션을 사용하면 CV 방법을 이용할 수 있다. 마지막으로 각각의 방법을 서로 비교해 보았다.

Question 2.

6.8 Excercises - Example 1,2, 5, 9, 11 풀어라

[Example 1]

We perform best subset, forward stepwise, and backward stepwise selection on a single data set. For each approach, we obtain p + 1 models, containing 0, 1, 2, . . . , p predictors. Explain your answers:

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

▶ Best Subset, Forward stepwise, Backward stepwise 방법 중 Best Subset 모델이 가장 작은 RSS 값을 가질 것이다. 왜냐하면 모든 가능한 경우를 고려하기 때문이다.

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

▶ n<p 일 경우 Best Subset 방법은 overfitting 할 수 있지만 그에반해 Forward Selection 은 좋은 성능을 낼 수 있다. 따라서 test error 의 경우에는 어떤 모형이 좋다고 할 수 없으며 상황에 맞게 적절한 방법을 사용해야 한다.

(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 identified 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 identified 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 identified 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 identified by best subset selection.

▶ FALSE

[Example 2]

For parts (a) through (c), indicate which of i. through iv. is correct. Justify your answer.

(a) The **lasso**, relative to **least squares**, is:

i. More flexible and hence will give improved prediction accuracy when its increase in bias is less than its decrease in variance.

ii. More flexible and hence will give improved prediction accuracy when its increase in variance is less than its decrease in bias.

iii. Less flexible and hence will give improved prediction accuracy when its increase in bias is less than its decrease in variance.

iv. Less flexible and hence will give improved prediction accuracy when its increase in variance is less than its decrease in bias.

▶ iii. Lasso 방법은 Least Square 방법에서 제한 조건을 둬서 모수를 추정하는 방법이다. 따라서 덜 flexible 하며, 약간의 Bias 를 증가시켜서 Variance 를 크게 낮추는 방법이다.

(b) Repeat (a) for ridge regression relative to least squares.

▶ iii. 앞서 (a) 와 마찬가지로, Ridge Regression 방법도 Lasso 처럼 제한조건을 둬서 모수를 추정하는 방법이고 Bias-Variance trade off 에 의해 약간의 Bias 를 증가시켜서 Variance 를 크게 낮추는 방법을 선택하는 것이다.

(c) Repeat (a) for non-linear methods relative to least squares.

▶ ii . non-linear 방법의 경우 Laeast Square 방법과 비교하여 더 flexible 한 방법이다. Variance 를 증가시켜 Bias 를 낮추는 방법이다. Flexible 하기 때문에 overfitting 에 주의해야 한다.

[Example 5]

It is well-known that ridge regression tends to give similar coefficient values to correlated variables, whereas the lasso may give quite different coefficient values to correlated variables. We will now explore this property in a very simple setting. Suppose that n = 2, p = 2, $x_{11}=x_{12}$, $x_{21}=x_{22}$. Furthermore, suppose that $y_1+y_2=0$ and $x_{11}+x_{21}=0$, $x_{12}+x_{22}=0$, so that the estimate for the intercept in a least squares, ridge regression, or lasso model is zero: $\hat{\beta}_0$= 0.

(a) Write out the ridge regression optimization problem in this setting.

- ridge regression 은 $\sum_{i=1}^{n}\left(y_i-\beta_0-\sum_{j=1}^{p}\beta_j x_{ij}\right)^2+\lambda\sum_{j=1}^{p}\beta_j^2=\mathrm{RSS}+\lambda\sum_{j=1}^{p}\beta_j^2$ 값을 최소화하는 방법이다. 따라서 적절한 lambda 를 선택해서 test MSE 가 최소가 되도록 한다.
  즉 본 문제에서는 $(y_1-\hat{\beta}_1x_1-\hat{\beta}_2x_1)^2+(y_2-\hat{\beta}_1x_2-\hat{\beta}_2x_2)^2+\lambda(\hat{\beta}_1^2+\hat{\beta}_2^2)$ 값을 최소화하는 값을 찾는 것과 같다.

(b) Argue that in this setting, the ridge coefficient estimates satisfy $\hat{\beta}_1 = \hat{\beta}_2$.

- Let, S= $(y_1-\hat{\beta}_1x_1-\hat{\beta}_2x_1)^2+(y_2-\hat{\beta}_1x_2-\hat{\beta}_2x_2)^2+\lambda(\hat{\beta}_1^2+\hat{\beta}_2^2)$
  $\frac{\partial}{\partial\beta_1}S$ =0, $\frac{\partial}{\partial\beta_2}S$ =0 을 만족하는 두 식을 풀면 $\hat{\beta}_1 = \hat{\beta}_2$ 를 만족함을 알 수 있다.

(c) Write out the lasso optimization problem in this setting.

- lasso regression 은 $\sum_{i=1}^{n}\left(y_i-\beta_0-\sum_{j=1}^{p}\beta_j x_{ij}\right)^2+\lambda\sum_{j=1}^{p}|\beta_j| =\mathrm{RSS}+\lambda\sum_{j=1}^{p}|\beta_j|$ 값을 최소화하는 방법이다. Ridge 와 마찬가지로 적절한 lambda 를 선택해서 test MSE 가 최소가 되도록 한다.
  즉 본 문제에서는 $(y_1-\hat{\beta}_1x_1-\hat{\beta}_2x_1)^2+(y_2-\hat{\beta}_1x_2-\hat{\beta}_2x_2)^2+\lambda(|\hat{\beta}_1|+|\hat{\beta}_2|)$ 값을 최소화하는 값을 찾는 것과 같다.

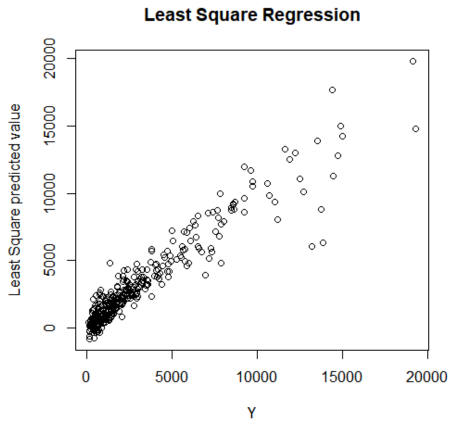
(d) Argue that in this setting, the lasso coefficients $\hat{\beta}_1$ and $\hat{\beta}_2$ are not unique—in other words, there are many possible solutions to the optimization problem in (c). Describe these solutions.

- $\frac{\partial}{\partial\beta}(\lambda|\beta|) = \lambda\frac{|\beta|}{\beta}$ 를 풀면 $\lambda\frac{|\beta_1|}{\beta_1} = \lambda\frac{|\beta_2|}{\beta_2}$ 식이 유도되고 따라서 β_1, β_2 값은 같은 부호를 갖게 된다.

[Example 9]

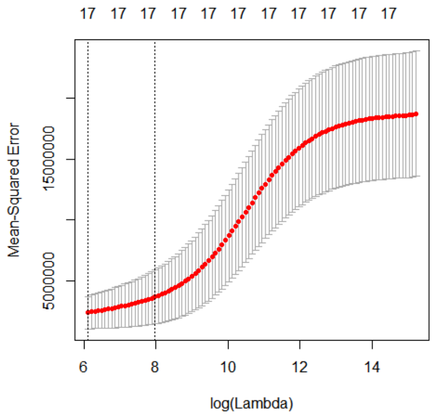
In this exercise, we will predict the number of applications received using the other variables in the College data set. (a) Split the data set into a training set and a test set.

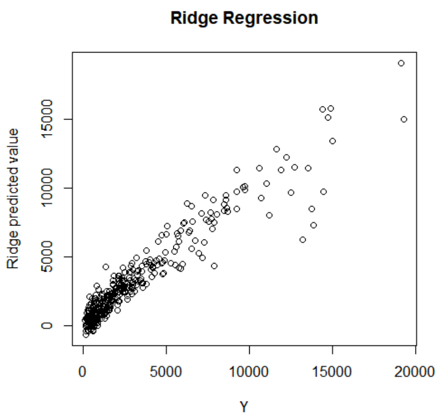
(b) Fit a linear model using least squares on the training set, and report the test error obtained.



▶ Least Square Regression 결과 test MSE 값은 1,108,531 로, 왼쪽 그래프는 실제 Y 값과 Least Square 방법에 의해 추정된 Y 값의 그래프이다.

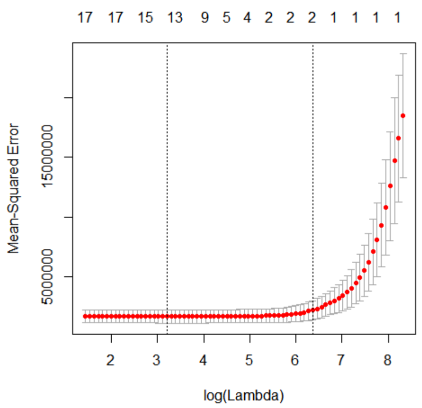
(c) Fit a ridge regression model on the training set, with λ chosen by cross-validation. Report the test error obtained.

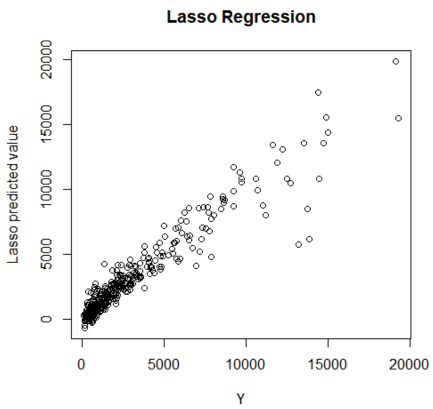




▶ cv.glmnet 함수를 이용해 Cross-Validation 방법에 의해 error 가 가장 작은 best λ 를 추정해 본 결과 λ=450.7435 일 때로 이 때의 test MSE 는 1,037,616 로 나타났다.

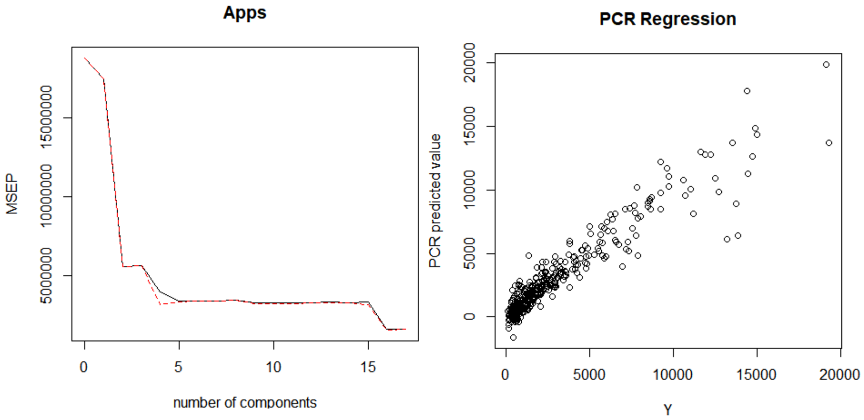
(d) Fit a lasso model on the training set, with λ chosen by crossvalidation. Report the test error obtained, along with the number of non-zero coefficient estimates.





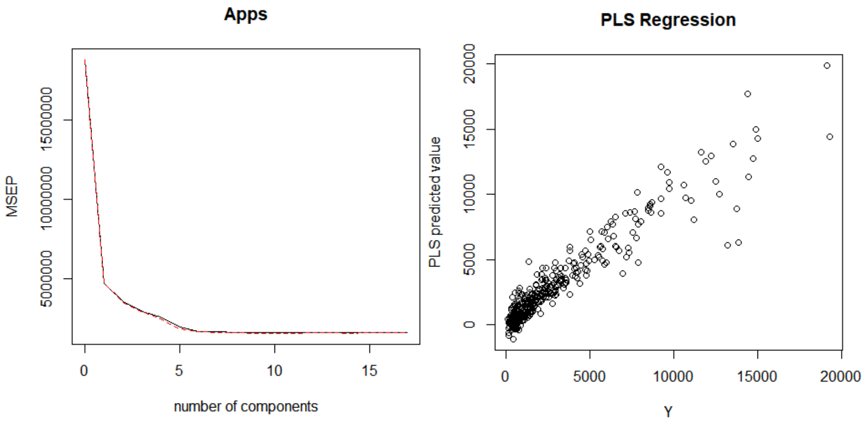
▶ (c)와 마찬가지로 cv.glmnet 함수를 이용한 best λ 는 24.6, 이 때의 test MSE 는 1,030,941 이다. 또한 변수 18 개에 대해서 0 이 아닌 회귀계수 값은 16 개로 나타났다.

(e) Fit a PCR model on the training set, with M chosen by crossvalidation. Report the test error obtained, along with the value of M selected by cross-validation.



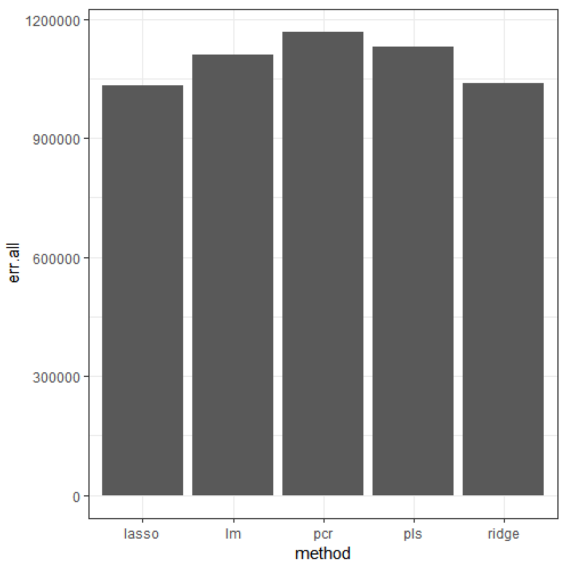
▶ pcr 방법을 사용해 본 결과 CV 값이 가장 작은 경우는 M= 16 일 때 test MSE 는 1273 값을 가진다.

(f) Fit a PLS model on the training set, with M chosen by crossvalidation. Report the test error obtained, along with the value of M selected by cross-validation.



▶ pls 방법을 사용해 본 결과 CV 값이 가장 작은 경우는 M = 11 일 때 test MSE 는 1279 값을 가진다.

(g) Comment on the results obtained. How accurately can we predict the number of college applications received? Is there much difference among the test errors resulting from these five approaches?



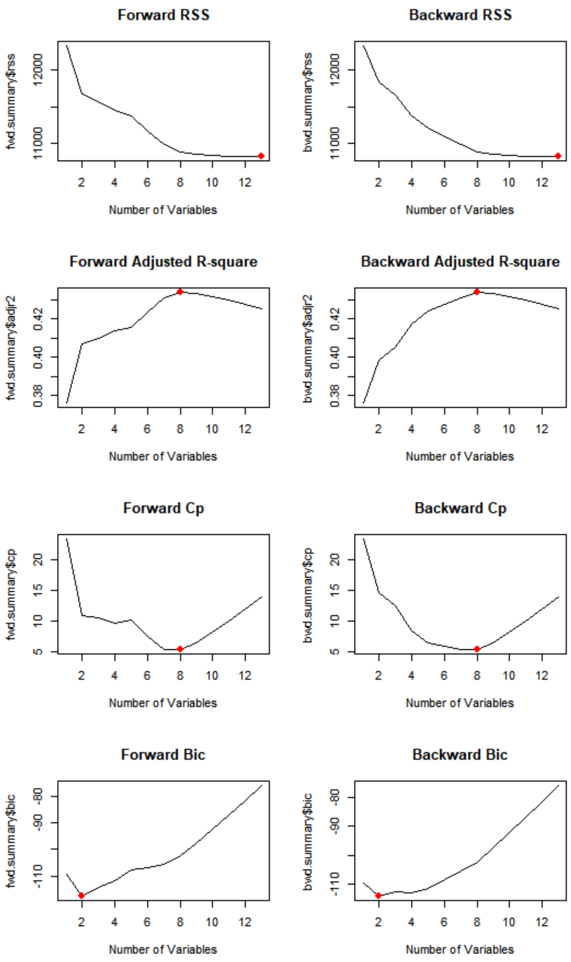
▶ 위에서 5 가지 방법을 이용해 적합해 본 결과 Lasso, Ridge 의 test MSE 값이 가장 작은 것을 알 수 있었다. PCR, PLS 방법은 Least Square Regression 방법보다도 test MSE 값이 높았다. 따라서 접수된 지원서의 수를 예측하는 데 있어 Lasso Model 이 가장 값을 잘 예측할 것이라고 기대할 수 있다.

[Example 11]

We will now try to predict per capita crime rate in the Boston data set.

(a) Try out some of the regression methods explored in this chapter, such as best subset selection, the lasso, ridge regression, and PCR. Present and discuss results for the approaches that you consider.

▶ 데이터를 train, test set 으로 분할 후 train data set 으로 모델을 적합한 후 CV 방법 기준에서 CV MSE 값을 서로 비교하여 가장 작은 값을 갖는 모형을 살펴보았다. 본 문제에서 사용한 Regression method 는 Forward Selection, Backward Elimination, Ridge, Lasso, PCR, PLS 방법이다. 적합 결과는 아래와 같다.



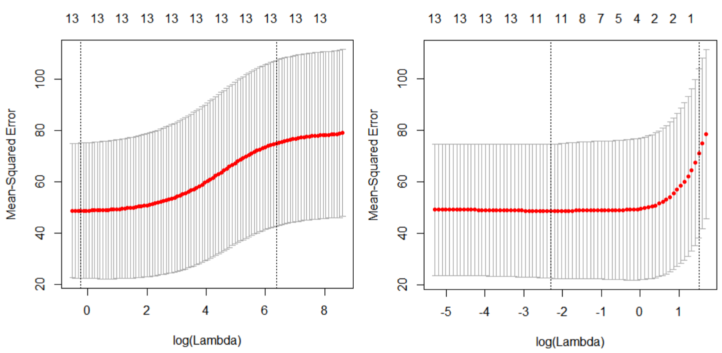
▶ 먼저 Best subset selection 방법 중 Forward, Backward 방법에 대해 p(변수 개수)를 1 부터 13 까지 변화함에 따라서 RSS, Adjusted Rsquare, Cp, Bic 값을 살펴보았다.

- RSS 의 경우 p=13 일 때의 값이 가장 작은 값을 갖는 것으로 나타났다. RSS 값은 p 가 증가할수록 항상 단조감소한다.

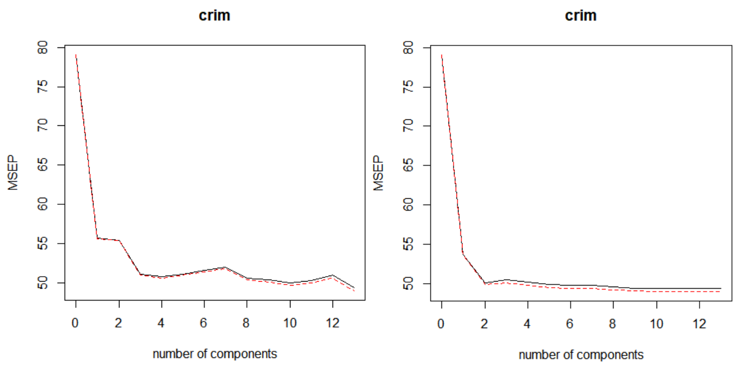
- Adjusted R square 의 경우는 p = 8 일 때 가장 큰 값을 갖는 것으로 나타났다. 기존 R square 는 p 값이 증가함에 따라서 단조증가하는데, 이러한 단점을 보완해 준 방법이다.

- Cp 의 경우는 p=8 일 때 가장 작은 값을 갖는 것으로 나타난다.

- BIC 기준에서는 p=2 일 때가 가장 작은 값을 갖는데, 이 방법은 다른 방법보다 p 값에 더 penalty 를 부여하기 때문에 더 간단한 모형이 선택된다.

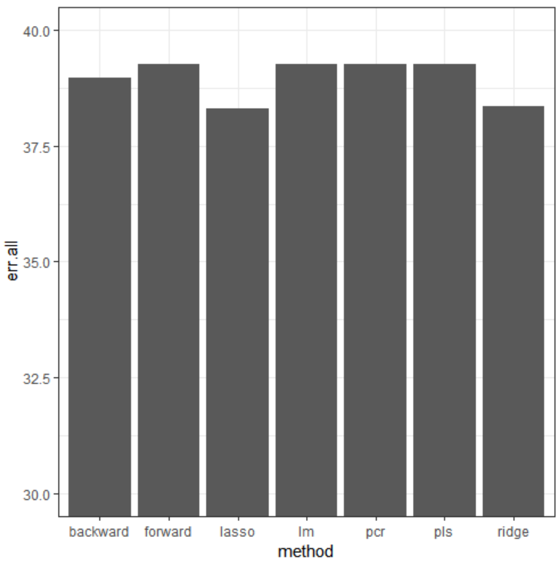


▶ 각각 Ridge Regression, Lasso 방법으로 적합한 MSE 결과이다. 각각의 best λ 는 0.79, 0.1 이다. 특히 Lasso 방법에서 총 14 개의 변수 중 0 이 아닌 회귀계수 값을 갖는 변수는 총 12 개이다.



▶ 각각 PCR, PLS 방법으로 적합한 MSE 결과이다. PCR 방법은 M=13 일 때 가장 작은 MSE 값을 가지며 PLS 방법은 M=10 일 때 가장 작은 MSE 값을 가진다.

(b) Propose a model (or set of models) that seem to perform well on this data set, and justify your answer. Make sure that you are evaluating model performance using validation set error, crossvalidation, or some other reasonable alternative, as opposed to using training error.



▶ 위에서 CV 방법을 이용해 6 가지의 Regression 방법을 데이터에 적합한 후 test error 를 나타내 본 결과, lasso 방법이 가장 작은 값을 갖는 것을 알 수 있다. 따라서, 범죄율을 예측하는 데 있어 Lasso 모형이 가장 성능이 좋을 것이라고 기대할 수 있다.

(c) Does your chosen model involve all of the features in the data set? Why or why not?

▶ (b) 에서 최적의 방법으로 Lasso 모형이 선택되었는데, 추정된 회귀 계수 값을 살펴봤을 때 총 14 개의 변수 중 age, tax 의 회귀계수 값이 0 으로 축소되었다. 따라서 12 가지의 변수만이 모형에 포함되어 이음을 알 수 있다.

Ⅲ. Discussion

이번 과제를 하면서 최적의 모형을 찾기 위한 몇 가지 방법을 알 수 있었다. 모형의 성능과 예측력을 좋게 하는 모형을 찾기 위한 방법의 특징과 장단점을 공부할 수 있었다.

Subset selection 방법의 경우 all possible regression, Forward selection, Backward Elimination, stepwise 가 있는데 이 중 Forward Selection 은 n <p 인 경우 사용할 수 있는 방법이며 이 경우 n개의 모형을 고려할 수 있게 된다. 모형을 선택할 때 항상 test error 기준에서 낮은 모형을 선택해야 한다.

Shirinkage 방법에는 Ridge Regression, Lasso 가 있는데 이 방법은 회귀계수 추정치를 제한하거나 regularize 하는 방법이다. 회귀계수 추정치의 분산을 매우 줄일 수 있다는 큰 장점이 있고 적합 전 표준화를 해야 한다. Ridge Regression 은 기존 RSS에 shirinkage penalty를 부여한다. 여기서 lambda 가 등장하는데 lambda 값이 커질수록 penalty 의 영향이 커져서 회귀계수 추정치는 0으로 수렴하며 l2 norm 이 감소할 것이다. 따라서 bias-variance trade-off에 의해 bias 가 약간 커져도 분산이 크게 감소하게 한다. 계산에 대한 부담도 적은 방법이다. 또한 Ridge는 최종 모형에 p개 설명변수를 모두 포함하기 때문에 모델 해석력 측면에서 좋지 않다. Lasso 방법은 l1 norm 을 사용한다. Lambda가 충분히 클수록 회귀계수 중 일부를 0으로 수렴하게 하여 변수선택처럼 기능한다. 따라서 모델 해석력이 좋은 sparse 한 모형이 선택된다. 두 방법 중 test error 기준에서 어떤 방법이 더 성능이 좋다고 할 수 없다. Ridge는 p를 같은 비율로 축소하며 Lasso는 모든 계수들을 같은 양만큼 0으로 축소한다.

마지막으로 차원축소 방법에는 PCA, PCR, Partial Regression 방법이 있다. 먼저 PCA 방법은 unsupervised learning 방법이며, 표준화해 준 뒤 사용한다. 첫 번째 주성분을 데이터가 가장 크게 변하는 방향으로 한다. 즉 분산이 가장 큰 방향으로 설정하고, 두 번째 주성분은 첫 번째 주성분과 uncorrelated 되며 가장 큰 분산을 갖는 변수들의 linear combination으로 설정한다. 즉 첫 번째 주성분과 두 번째 주성분은 서로 직교하며 이 의미는 서로 상관관계가 없다는 것을 의미한다. PCR 방법은 M개의 주성분이 데이터 내 대부분의 분산과 y와의 상관관계를 설명하는 데 충분하다는 idea로부터 시작된다. 또한 마찬가지로 변수들을 표준화해 줘야 한다. Ridge 방법은 PCR의 연속버전이다. Partial Regression 방법은 PCR의 supervised learning 버전이라고 할 수 있다. PCR이 설명변수들을 잘 설명하는 방향의 주성분을 찾는 것이라면 PLS 방법은 설명변수와 반응변수를 모두 잘 설명하는 방향을 찾는 것이다.

Ⅳ. Appendix – R code

```
### 9번
# a
dim(College)
set.seed(1)
trainid = sample(1:nrow(College), nrow(College)/2)
train = College[trainid,]
test = College[-trainid,]
y.train = train$Apps
y.test = test$Apps

# b
# Least Squar Model
lm.fit = lm(Apps~., data=train)
lm.pred = predict(lm.fit,test) # predict function 안에 data = 이런거 넣지 않기
lm.mse = mean((lm.pred - y.test)^2)
par(mfrow= c(1,1))
plot(test$Apps,lm.pred, main="Least Square Regression", xlab="Y", ylab ="Least Square predicted value")

# c
# Ridge Regression Model
x = model.matrix(Apps ~., College)[,-1]
y = College$Apps
set.seed(1)
ridge.fit = cv.glmnet(x[trainid,], y.train, alpha = 0)
plot(ridge.fit)
ridge.bestlam = ridge.fit$lambda.min
ridge.pred = predict(ridge.fit, s = ridge.bestlam, newx = x[-trainid,])
ridge.mse = mean((ridge.pred - y.test)^2)
plot(test$Apps,ridge.pred, main="Ridge Regression", xlab="Y", ylab ="Ridge predicted value")

# d
# Lasso Model
set.seed(1)
lasso.fit = cv.glmnet(x[trainid,], y.train, alpha = 1)
plot(lasso.fit)
lasso.bestlam = lasso.fit$lambda.min
lasso.pred = predict(lasso.fit, s = lasso.bestlam, newx = x[-trainid,])
lasso.mse = mean((lasso.pred - y.test)^2)
lasso.coef = predict(lasso.fit, type = "coefficients", s =lasso.bestlam)[1:ncol(College),]
lasso.coef
length(lasso.coef[lasso.coef != 0])
length(lasso.coef)
plot(test$Apps,lasso.pred, main="Lasso Regression", xlab="Y", ylab ="Lasso predicted value")

# e
# PCR Model
set.seed(1)
pcr.fit = pcr(Apps ~., data =train, scale = TRUE, validation = "CV")
validationplot(pcr.fit, val.type = "MSEP")
summary(pcr.fit)          # CV 값이 16comps 일 때 1273로 가장 작음
write.csv(summary(pcr.fit), "C:/Users/jeeyeon/Desktop/데마/HW6/table1.csv")
pcr.pred = predict(pcr.fit, x[-trainid,], ncomp = 16)
pcr.mse = mean((pcr.pred - y.test)^2)     # 따라서 M = 16일 때 MSE 값
plot(test$Apps,pcr.pred, main="PCR Regression", xlab="Y", ylab ="PCR predicted value")

# f
# PLS model
set.seed(1)
pls.fit = plsr(Apps~., data = train, scale = TRUE, validation = "CV")
validationplot(pls.fit, val.type = "MSEP")
summary(pls.fit)          # CV 값이 11comps일 때 1279로 가장 작음
pls.pred = predict(pls.fit, x[-trainid,], ncomp=11)
pls.mse = mean( (pls.pred - y.test)^2)
pls.fit = plsr(Apps~., data = test, scale = TRUE, ncomp =11)
```

```
summary(pls.fit)
plot(test$Apps,pls.pred, main="PLS Regression", xlab="Y", ylab ="PLS predicted value")
# 비교
# PCR :  설명변수에서 설명되는 분산의 양만 최대로 하려고 함
# PLS : 설명변수, 반응변수 둘 다의 분산을 설명하는 방향을 찾으려고 함

# g
err.all =c(lm.mse, ridge.mse,lasso.mse,pcr.mse,pls.mse)
method = c("lm","ridge","lasso","pcr","pls")
table = data.frame(method, err.all)
ggplot(data=table, aes(x=method, y=err.all)) + geom_bar(stat="identity")+ theme_bw()
# BEST : lasso

# 11 번
# train, test split
set.seed(1)
trainid = sample(1:nrow(Boston), nrow(Boston)/2)
train = Boston[trainid,]
test = Boston[-trainid,]
y.train = train$crim
y.test = test$crim
# Best subeset selction 이용한 모델적합
regfit.best = regsubsets(crim ~.,data=train,nvmax=ncol(Boston)-1) # train set 적합한 모델 #nvmax:maximum size of subsets to examine
test.mat = model.matrix(crim~., data = test) # model.matrix : 데이터로부터 X 행렬 구성함
head(test.mat) # 만들어진 X matrix 확인
# Cross- Validation
# regsubsets 함수를 predict 사용하기 위해 predict function 정의
predict.regsubsets = function(object, newdata, id, ...) {
  form = as.formula(object$call[[2]])
  mat = model.matrix(form, newdata)
  coefi = coef(object, id = id)
  mat[, names(coefi)] %*% coefi
}

# forward selection
fit.fwd <- regsubsets(crim~., data=train, nvmax=ncol(Boston)-1, method = "forward")
(fwd.summary <- summary(fit.fwd))
err.fwd <- rep(NA, ncol(Boston)-1)
for(i in 1:(ncol(Boston)-1)) {
  pred.fwd <- predict(fit.fwd, test, id=i)
  err.fwd[i] <- mean((test$crim - pred.fwd)^2)
}
plot(err.fwd, type="b", main="Test MSE for Forward Selection", xlab="Number of Predictors")
which.min(err.fwd)

# backward selection
fit.bwd <- regsubsets(crim~., data=train, nvmax=ncol(Boston)-1, method = "backward")
(bwd.summary <- summary(fit.bwd))
err.bwd <- rep(NA, ncol(Boston)-1)
for(i in 1:(ncol(Boston)-1)) {
  pred.bwd <- predict(fit.bwd, test, id=i)
  err.bwd[i] <- mean((test$crim - pred.bwd)^2)
}
plot(err.bwd, type="b", main="Test MSE for Backward Selection", xlab="Number of Predictors")
which.min(err.bwd)

### plot
par(mfrow= c(4,2))
plot(fwd.summary$rss, xlab = "Number of Variables", main = "Forward RSS",type = "l")
minloc_rss = which.min(fwd.summary$rss)
points(minloc_rss, fwd.summary$rss[minloc_rss], col = "red", cex = 2, pch = 20)
plot(bwd.summary$rss, xlab = "Number of Variables", main = "Backward RSS",type = "l")
minloc_rss = which.min(bwd.summary$rss)
points(minloc_rss, bwd.summary$rss[minloc_rss], col = "red", cex = 2, pch = 20)
plot(fwd.summary$adjr2, xlab = "Number of Variables", main = "Forward Adjusted R-square",type = "l")
maxloc_adjr2 = which.max(fwd.summary$adjr2)
points(maxloc_adjr2, fwd.summary$adjr2[maxloc_adjr2], col = "red", cex = 2, pch = 20)
plot(bwd.summary$adjr2, xlab = "Number of Variables", main = "Backward Adjusted R-square",type = "l")
```

```
maxloc_adjr2 = which.max(bwd.summary$adjr2)
points(maxloc_adjr2, bwd.summary$adjr2[maxloc_adjr2], col = "red", cex = 2, pch = 20)
plot(fwd.summary$cp, xlab = "Number of Variables", main = "Forward Cp",type = "l")
minloc_cp = which.min(fwd.summary$cp)
points(minloc_cp, fwd.summary$cp[minloc_cp], col = "red", cex = 2, pch = 20)
plot(bwd.summary$cp, xlab = "Number of Variables", main = "Backward Cp",type = "l")
minloc_cp = which.min(bwd.summary$cp)
points(minloc_cp, bwd.summary$cp[minloc_cp], col = "red", cex = 2, pch = 20)
plot(fwd.summary$bic, xlab = "Number of Variables", main = "Forward Bic",type = "l")
minloc_bic = which.min(fwd.summary$bic)
points(minloc_bic, fwd.summary$bic[minloc_bic], col = "red", cex = 2, pch = 20)
plot(bwd.summary$bic, xlab = "Number of Variables", main = "Backward Bic",type = "l")
minloc_bic = which.min(bwd.summary$bic)
points(minloc_bic, bwd.summary$bic[minloc_bic], col = "red", cex = 2, pch = 20)
par(mfrow= c(1,1))
# ------------------------------------------------------------------------
# Least Squar Model
lm.fit = lm(crim~., data=train)
lm.pred = predict(lm.fit,test)
lm.mse = mean((lm.pred - y.test)^2)
plot(test$crim,lm.pred)
# Ridge Regression Model
x = model.matrix(crim ~., Boston)[,-1]
y = Boston$crim
set.seed(1)
ridge.fit = cv.glmnet(x[trainid,], y.train, alpha = 0)
plot(ridge.fit)
ridge.bestlam = ridge.fit$lambda.min
ridge.pred = predict(ridge.fit, s = ridge.bestlam, newx = x[-trainid,])
ridge.mse = mean((ridge.pred - y.test)^2)
# Lasso Model
set.seed(1)
lasso.fit = cv.glmnet(x[trainid,], y.train, alpha = 1)
plot(lasso.fit)
lasso.bestlam = lasso.fit$lambda.min
lasso.pred = predict(lasso.fit, s = lasso.bestlam, newx = x[-trainid,])
lasso.mse = mean((lasso.pred - y.test)^2)
lasso.coef = predict(lasso.fit, type = "coefficients", s =lasso.bestlam)[1:ncol(Boston),]
length(lasso.coef[lasso.coef != 0])
length(lasso.coef)
# PCR Model
set.seed(1)
pcr.fit = pcr(crim ~., data =train, scale = TRUE, validation = "CV")
validationplot(pcr.fit, val.type = "MSEP")
summary(pcr.fit)          # CV 값이 13comps 일 때 7.025 가장 작음
pcr.pred = predict(pcr.fit, x[-trainid,], ncomp = 13)
pcr.mse = mean((pcr.pred - y.test)^2)     # 따라서 M = 13일 때 MSE 값
# PLS model
set.seed(1)
pls.fit = plsr(crim~., data = train, scale = TRUE, validation = "CV")
validationplot(pls.fit, val.type = "MSEP")
summary(pls.fit)          # CV 값이 10comps일 때 7.025 가장 작음
pls.pred = predict(pls.fit, x[-trainid,], ncomp=10)
pls.mse = mean( (pls.pred - y.test)^2)
# b
err.all =c(min(err.fwd), min(err.bwd), lm.mse, ridge.mse,lasso.mse,pcr.mse,pls.mse)
method = c("forward","backward","lm","ridge","lasso","pcr","pls")
table = data.frame(method, err.all)
ggplot(data=table, aes(x=method, y=err.all)) + geom_bar(stat="identity", ylim =c(20,40)) + theme_bw()+ coord_cartesian(ylim=c(30,40))
# BEST : lass
# c
lasso.coef # age, tax 변수는 0 , 모두 사용되지 않음
```