Model Selection and Regularization

Consider the wine dataset. We will take Quality as the quantitative response, the remaining 6 variables as predictors, and all the data as training data. For all the models below, use leave-one-out cross-validation (LOOCV) to compute the estimated test error rates.

```
library(car)
library(lmtest)
library(ggplot2)
library(ISLR)
library(MASS)
library(leaps)
library(glmnet)
```

```
wine<-read.table("wine.txt",header = TRUE)
wine$Region<-as.factor(wine$Region)
totpred <- ncol(wine)
k<-nrow(wine)
```

For parts (a)-(f) Summary of the parameter estimates and test MSE using LOOCV are presented in Table 1.

a) Multiple linear regression model using all predictors was performed.

```
full.model<-lm(Quality~.,data=wine)
summary(full.model)
```

```
Call:
lm(formula = Quality ~ ., data = wine)

Residuals:
     Min       1Q   Median       3Q      Max 
-1.80824 -0.58413 -0.02081  0.48627  1.70909 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept)  7.81437    1.96944   3.968 0.000417 ***
Clarity      0.01705    1.45627   0.012 0.990736    
Aroma        0.08901    0.25250   0.353 0.726908    
Body         0.07967    0.26772   0.298 0.768062    
Flavor       1.11723    0.24026   4.650 6.25e-05 ***
Oakiness    -0.34644    0.23301  -1.487 0.147503    
Region2     -1.51285    0.39227  -3.857 0.000565 ***
Region3      0.97259    0.51017   1.906 0.066218 .  
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.9154 on 30 degrees of freedom
Multiple R-squared:  0.8376,    Adjusted R-squared:  0.7997 
F-statistic:  22.1 on 7 and 30 DF,  p-value: 3.295e-10
```

```
a.coeff<-full.model$coefficients
```

```
library(caret)

#specify the cross-validation method
ctrl <- trainControl(method = "LOOCV")

#fit a regression model and use LOOCV to evaluate performance
model <- train(Quality~., data = wine, method = "lm", trControl = ctrl)

#view summary of LOOCV
a.mse<-as.numeric(model$results[2])^2
```

b) Best subset selection was performed and **Figure** 1 shows plot of adjusted R^2 for each posiible model containing a subset of 6 predictors in wine data set. According to the plot adjusted R^2 increase upto 4 predictors including 2 dummy variables for `Region` and then decrease. Highest adjusted R^2 value of 0.8164 obtained for model with predictors `Flavor`, `Oakiness` and `Region`.

```
fit.best <- regsubsets(Quality ~ ., wine, nvmax = totpred)
best.summary <- summary(fit.best)
best.summary
```

```
Subset selection object
Call: regsubsets.formula(Quality ~ ., wine, nvmax = totpred)
7 Variables  (and intercept)
         Forced in Forced out
Clarity      FALSE      FALSE
Aroma        FALSE      FALSE
Body         FALSE      FALSE
Flavor       FALSE      FALSE
Oakiness     FALSE      FALSE
Region2      FALSE      FALSE
Region3      FALSE      FALSE
1 subsets of each size up to 7
Selection Algorithm: exhaustive
         Clarity Aroma Body Flavor Oakiness Region2 Region3
1  ( 1 ) " "     " "   " "  "*"    " "      " "     " "    
2  ( 1 ) " "     " "   " "  "*"    " "      "*"     " "    
3  ( 1 ) " "     " "   " "  "*"    " "      "*"     "*"    
4  ( 1 ) " "     " "   " "  "*"    "*"      "*"     "*"    
5  ( 1 ) " "     "*"   " "  "*"    "*"      "*"     "*"    
6  ( 1 ) " "     "*"   "*"  "*"    "*"      "*"     "*"    
7  ( 1 ) "*"     "*"   "*"  "*"    "*"      "*"     "*"    
```

```
b.adjr2<-best.summary$adjr2
which.max(best.summary$adjr2)
```

```
[1] 4
```

```
b.coeff<-coef(fit.best, 4)
```

```
# Write a function to easily get predictions for a model
# from a regsubsets object
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
}
```

```
# Create a k x totpred matrix to store test errors
best.errors <- matrix(NA, k, totpred, dimnames = list(NULL, paste(1:totpred)))
# use LOOCV to calculate MSE using best subset selection
set.seed(1)
for (j in 1:k) {
    # Best subset selection on the training folds
    best.fit <- regsubsets(Quality~., data = wine[-j,], nvmax = totpred)
        # Prediction on the test fold
    for (i in 1:totpred) {
        # Using the predict.regsubsets function written above
        best.pred <- predict(best.fit, wine[j,], id = i)
        best.errors[j, i] = mean((wine$Quality[j] - best.pred)^2)
    }
}

mean.best.errors <- apply(best.errors, 2, mean)
mean.best.errors
```

```
        1         2         3         4         5         6         7 
1.6830659 1.0633115 0.8945649 0.8705717 1.0657420 1.1337885 1.1351581 
```

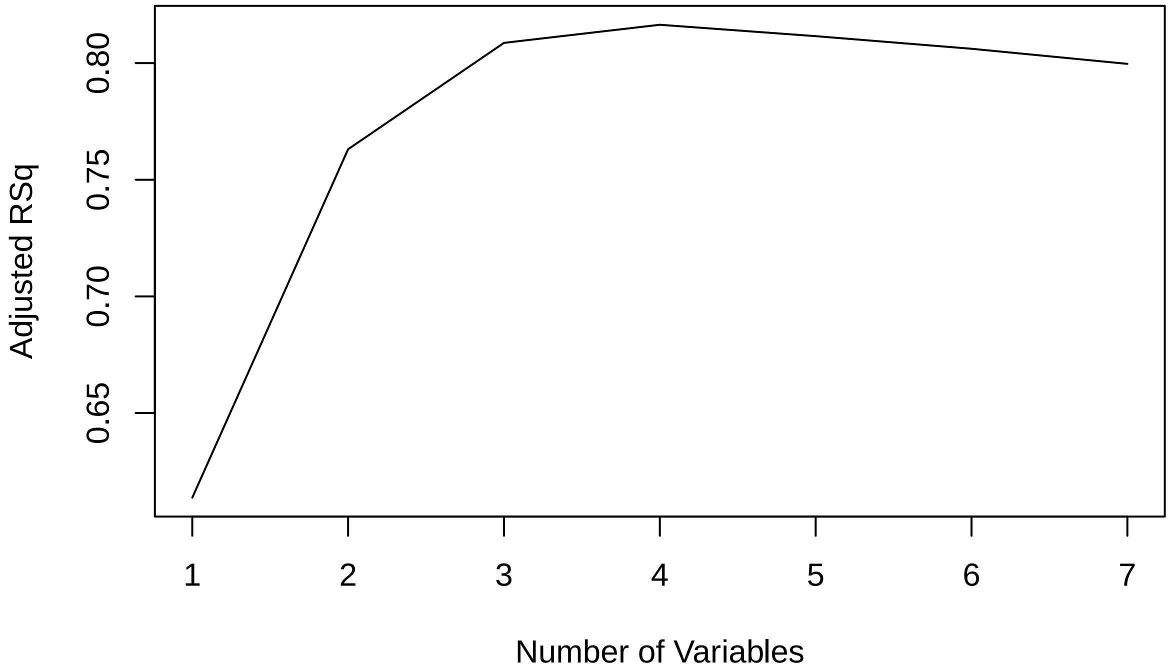
```
b.mse<-mean.best.errors[4]
```

c) Forward stepwise selection was performed **Figure** 1 shows plot of adjusted R^2 for each posiible model containing a subset of 6 predictors in wine data set. Highset adjusted R^2 value of 0.8164 obtained for model with predictors `Flavor`, `Oakiness` and `Region`.

```
fit.forward <- regsubsets(Quality ~ ., wine, nvmax = totpred ,method = "forward")
forward.summary <- summary(fit.forward)
forward.summary
```

```
Subset selection object
Call: regsubsets.formula(Quality ~ ., wine, nvmax = totpred, method = "forward")
7 Variables  (and intercept)
         Forced in Forced out
Clarity      FALSE      FALSE
Aroma        FALSE      FALSE
Body         FALSE      FALSE
Flavor       FALSE      FALSE
Oakiness     FALSE      FALSE
Region2      FALSE      FALSE
Region3      FALSE      FALSE
1 subsets of each size up to 7
Selection Algorithm: forward
         Clarity Aroma Body Flavor Oakiness Region2 Region3
1  ( 1 ) " "     " "   " "  "*"    " "      " "     " "    
2  ( 1 ) " "     " "   " "  "*"    " "      "*"     " "    
3  ( 1 ) " "     " "   " "  "*"    " "      "*"     "*"    
4  ( 1 ) " "     " "   " "  "*"    "*"      "*"     "*"    
5  ( 1 ) " "     "*"   " "  "*"    "*"      "*"     "*"    
6  ( 1 ) " "     "*"   "*"  "*"    "*"      "*"     "*"    
7  ( 1 ) "*"     "*"   "*"  "*"    "*"      "*"     "*"    
```

```
c.adjr2<-forward.summary$adjr2
plot(forward.summary$adjr2, xlab = "Number of Variables", ylab = "Adjusted RSq",
    type = "l")
```



```
which.max(forward.summary$adjr2)
```

```
[1] 4
```

```
c.coeff<-coef(fit.forward, 4)
```

```
# Create a k x totpred matrix to store test errors
forward.errors <- matrix(NA, k, totpred, dimnames = list(NULL, paste(1:totpred)))
# use LOOCV to calculate MSE using forward subset selection
set.seed(1)
for (j in 1:k) {
    # Best subset selection on the training folds
    forward.fit <- regsubsets(Quality~., data = wine[-j,], nvmax = totpred, method = "forward")
        # Prediction on the test fold
    for (i in 1:totpred) {
        # Using the predict.regsubsets function written above
        forward.pred <- predict(forward.fit, wine[j,], id = i)
        forward.errors[j, i] = mean((wine$Quality[j] - forward.pred)^2)
    }
}

mean.forward.errors <- apply(best.errors, 2, mean)
mean.forward.errors
```

```
        1         2         3         4         5         6         7
1.6830659 1.0633115 0.8945649 0.8705717 1.0657420 1.1337885 1.1351581
```

```
c.mse<-mean.forward.errors[4]
```

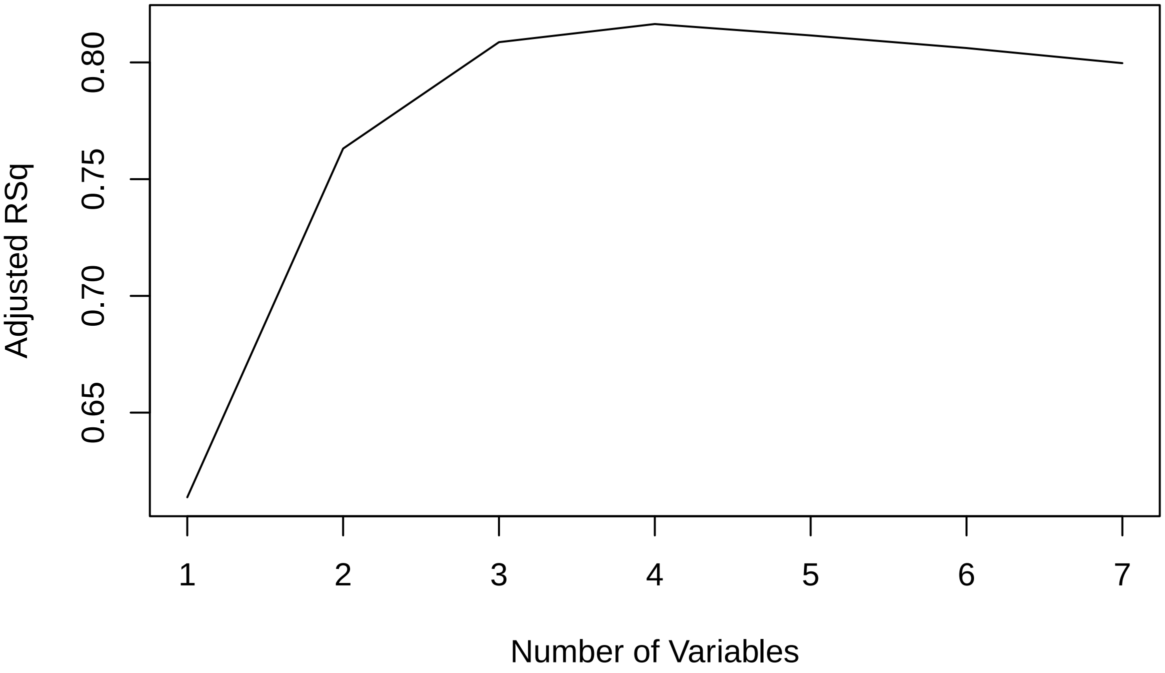
d) Backward stepwise selection was performed **Figure** 1 shows plot of adjusted R^2 for each posiible model containing a subset of 6 predictors in wine data set. Highset adjusted R^2 value of 0.8164 obtained for model with predictors `Flavor`, `Oakiness` and `Region`.

```
fit.backward <- regsubsets(Quality ~ ., wine, nvmax = totpred ,method = "backward")
backward.summary <- summary(fit.backward)
backward.summary
```

```
Subset selection object
Call: regsubsets.formula(Quality ~ ., wine, nvmax = totpred, method = "backward")
```

```
7 Variables  (and intercept)
         Forced in Forced out
Clarity      FALSE      FALSE
Aroma        FALSE      FALSE
Body         FALSE      FALSE
Flavor       FALSE      FALSE
Oakiness     FALSE      FALSE
Region2      FALSE      FALSE
Region3      FALSE      FALSE
1 subsets of each size up to 7
Selection Algorithm: backward
         Clarity Aroma Body Flavor Oakiness Region2 Region3
1  ( 1 ) " "     " "   " "  "*"    " "      " "     " "    
2  ( 1 ) " "     " "   " "  "*"    " "      "*"     " "    
3  ( 1 ) " "     " "   " "  "*"    " "      "*"     "*"    
4  ( 1 ) " "     " "   " "  "*"    "*"      "*"     "*"    
5  ( 1 ) " "     "*"   " "  "*"    "*"      "*"     "*"    
6  ( 1 ) " "     "*"   "*"  "*"    "*"      "*"     "*"    
7  ( 1 ) "*"     "*"   "*"  "*"    "*"      "*"     "*"    
```

```
d.adjr2<-backward.summary$adjr2
plot(backward.summary$adjr2, xlab = "Number of Variables", ylab = "Adjusted RSq",
    type = "l")
```



```
which.max(backward.summary$adjr2)
```

```
[1] 4
```

```
d.coeff<-coef(fit.backward, 4)
```

```
# Create a k x totpred matrix to store test errors
backward.errors <- matrix(NA, k, totpred, dimnames = list(NULL, paste(1:totpred)))
# use LOOCV to calculate MSE using backward subset selection
set.seed(1)
for (j in 1:k) {
    # Best subset selection on the training folds
    backward.fit <- regsubsets(Quality~., data = wine[-j,], nvmax = totpred, method = "backward")
        # Prediction on the test fold
    for (i in 1:totpred) {
        # Using the predict.regsubsets function written above
        backward.pred <- predict(backward.fit, wine[j,], id = i)
        backward.errors[j, i] = mean((wine$Quality[j] - backward.pred)^2)
    }
}

mean.backward.errors <- apply(best.errors, 2, mean)
mean.backward.errors
```

```
        1         2         3         4         5         6         7 
1.6830659 1.0633115 0.8945649 0.8705717 1.0657420 1.1337885 1.1351581 
```

```
d.mse<-mean.backward.errors[4]
```

```
par(mfrow=c(1,3))
par(mar = c(3.8, 3.8,0.5,1))

plot(best.summary$adjr2, xlab = "Number of Variables", ylab = "Adjusted RSq",
    type = "l")
plot(forward.summary$adjr2, xlab = "Number of Variables", ylab = "Adjusted RSq",
    type = "l")
plot(backward.summary$adjr2, xlab = "Number of Variables", ylab = "Adjusted RSq",
    type = "l")
```

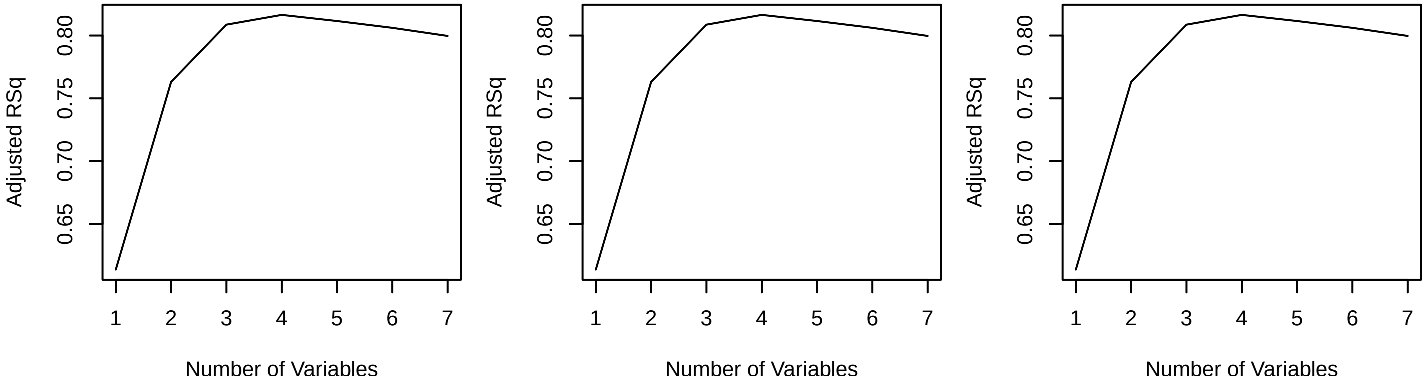


Figure 1: *For each posiible model containing a subset of 6 predictors in wine data set, the adjusted R-squared is desplayed. Left: Using best subset selection, Center : using forward stepwise selection, Right: using backward stepwise selection*

e) Ridge regression was performed and penalty parameter chosen optimally via LOOCV. Best λ value is 0.3356315. **Figure** 2 shows plot of test MSE vs log(λ).

```
# Create response vector and the design matrix (without the first column of 1s)
y <- wine$Quality
x <- model.matrix(Quality ~ ., wine)[, -1]
n<-nrow(wine)
grid <- 10^seq(10, -2, length = 100)
```

```
# Use cross-validation to estimate test MSE from training data
set.seed(1)
cv.out <- cv.glmnet(x,y, alpha = 0,nfolds=38,grouped = FALSE)

# Find the best value of lambda
bestlam <- cv.out$lambda.min
bestlam
```

```
[1] 0.3356315
```

```r
# Test MSE for the best value of lambda
ridge.mod <- glmnet(x, y, alpha = 0, lambda = grid)
ridge.pred <- predict(ridge.mod, s = bestlam, newx =x)
e.mse<-mean((ridge.pred - y)^2)

# Refit the model on the full dataset
out <- glmnet(x, y, alpha = 0)

# Get estimates for the best value of lambda
e.coeff<-predict(out, type = "coefficients", s = bestlam)[1:8, ]
```

f) Lasso was performed and penalty parameter chosen optimally via LOOCV. Best λ value is 0.1293366. **Figure** 2 shows plot of test MSE vs $\log(\lambda)$.

```r
# Use cross-validation to estimate test MSE using training data
set.seed(1)
cv.out1 <- cv.glmnet(x, y, alpha = 1,nfolds=38,grouped = FALSE)

bestlam <- cv.out1$lambda.min
bestlam
```

```
[1] 0.1293366
```

```r
lasso.mod <- glmnet(x, y, alpha = 1, lambda = grid)
lasso.pred <- predict(lasso.mod, s = bestlam, newx = x)
f.mse<-mean((lasso.pred - y)^2)

# Refit the model on the full dataset
out <- glmnet(x, y, alpha = 1)

# Estimates for the best value of lambda
f.coeff <- predict(out, type = "coefficients", s = bestlam)[1:8, ]
```

g) According to the following table ridge regression gives the smallest test MSE and the linear regression model with all the predictors has the highest test MSE. Moreover best subset selection, forward stepwise and backward stepwise selection methods gives the same test MSE and parameter estimates. Therefore we select model obtained from Ridge regression as the best model.

	(a)	(b)	(c)	(d)	(e)	(f)
(Intercept)	7.81437	8.1208167	8.1208167	8.1208167	7.5981854	7.835215148
Clarity	0.01705				0.1128379	0.000000000
Aroma	0.08901				0.2343928	0.002247145
Body	0.07967				0.1999401	0.000000000
Flavor	1.11723	1.1920393	1.1920393	1.1920393	0.8329586	1.060845222
Oakiness	-0.34644	-0.3183165	-0.3183165	-0.3183165	-0.3015529	-0.116897883
Region2	-1.51285	-1.5154840	-1.5154840	-1.5154840	-1.3236383	-1.305641743
Region3	0.97259	1.0935478	1.0935478	1.0935478	0.9071328	1.072992458
Test MSE	1.135158	0.8705717	0.8705717	0.8705717	0.703477	0.7153056

Table 1: *Summary of the parameter estimates and test MSE. (a) linear regression model with all the predictors, (b) best subset selection, (c) forward stepwise selection, (d) backward stepwise selection, (e) ridge regression, (f) lasso*

.

```
par(mfrow=c(1,2))
par(mar = c(3.8, 3.8,1,1))
plot(cv.out)
plot(cv.out1)
```

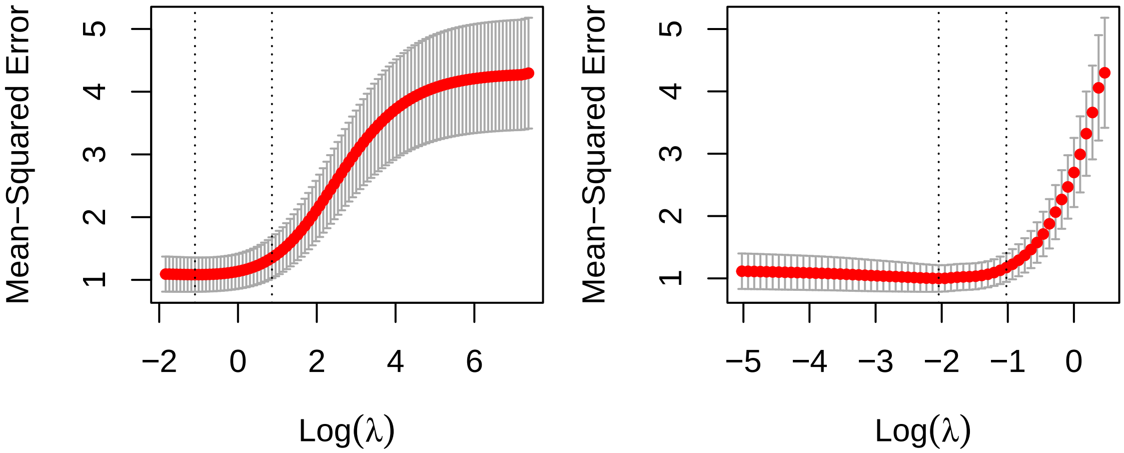


Figure 2: *plot of test MSE vs log(lambda). Left: using ridge regression, Right: using lasso*