Project For Statistic Learning : Boston data and Linear Regression methods

Haoran Duan[1]

170733151, MSc Data Science, School of Computing

Abstract. Fit the model and find the best for Boston data using subset selection and regularisation.

1 Exploratory Data Analysis and Data Preprocess

– Boston data has 506 observations, 14 variables. And this project focus on predicting the natural logarithm of the per capita crime rate(lcrim),so the 'lcrim' is response variable, and others are predictor variables. The three predictor variables named 'chas','disf' and 'rad' are qualitative, and others are quantitative.

1.1 Quantitative Variables

– Using histogram plot(Appendix,Fig.2),the predictor variables 'rm' and 'medv' are look like a normal distribution, and there is some too big and too small values,I use the red line to highlight, but I didn't drop it because I think we can't just drop it without any analysis,because each data has its own meaning.

1.2 Qualitative Variables

– Using box plot(Appendix,Fig.3),the predictor variables 'chas' has the smallest influence to the the response predictor, but the other two predictor variables 'rad' and 'disf', will influence the response variables much more.

1.3 Relationship

– Plot the correlation matrix(Appendix,Fig.4), it shows that the first three strongest positive relationship to the response variables 'lcrim'are predictor variable 'rad', 'tax' and 'nox'. Also, the strongest negative relationship to 'lcrim' is 'ages', 'disf' and 'zn'. It seems that the bad enviroment and low income or low profit will leads to bad life and increase the crime rate.

– Then plot a scatterplot for all variables(Appendix,Fig.5), it shows that there is no obviouse linear relationship among most of variables.But we can see there is a weak linear relationship between 'lcrim' and 'nox', between 'medv' and 'lstat', between 'age' and 'lcrim', between 'lcrim' and 'lstat'.

– In this project, I will present the linear regression,but there are some categorical variables. In scatterplot(Appendix,Fig.5), it shows that these qualitative variables are have some pattern or some relationship with 'lcrim',also in the boxplot(Appendix,Fig.3), with the increasing of the **level** of each qualitative variables, SOME response variables tend to be higher versus 'rad' and 'chas', but tend to be lower versus 'disf', so in this project I just make them as quantitative, use as other variables.

2 Subset Selection For Model Performance

Use all the predictor variables to fit the model using the best subset to find the appropriate exploratory variables for getting the best model, And use two different regularision method to get the two best model.

Plot the different numbers of predictors versus the different estimations such as Adjusted R-square, Mallow's Cp and Bayes Information Criterion(BIC).Also use the the cross validation for each.

2.1 Best Subset Selection

After using the best subset selection methods,as it is shown in plot(Fig.1)

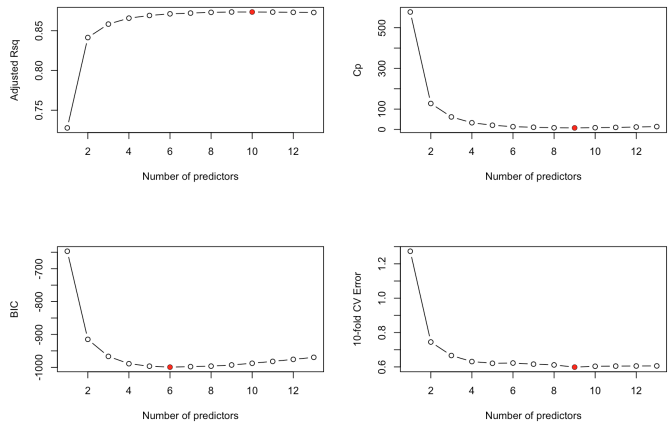

Fig. 1. Different Estimation vs Number of predictors

– Unfortunately,different methods select the different models(Red Point). If I examine the plot for each method,adjusted R-square suggests there is little difference between $M_6...M_{12}$, Mallow's Cp suggests there is little difference between $M_6...M_{12}$,and BIC suggests the M_6.Therefore, I might regard the best model as M_6. Also, if we check the plot of 10-folds cross validation,it seems that a model with 6 explanatory variables can be a good

choice(Fig.1).But if we use the cross validation and MSE to compare different model,I think I should consider to not just use PLOTS to decide.After calculate the MSE of M_6 and M_9 below,so finally, I prefer to choose the M_9.

```
> (bss_mse6 = bss_mse_b[6])
[1] 0.6226157
> (bss_mse9 = bss_mse_b[9])
[1] 0.5986935
```

```
Selection Algorithm: exhaustive
         zn  indus chas nox r m age disf rad tax pto black lstat medv
13  ( 1 ) "*" "*"   "*"  "*" "*" "*"  "*"  "*" "*" "*"   "*"   "*"   "*"
 6  ( 1 ) "*" " "   " "  "*" " " " "  "*"  "*" " " " "   "*"   "*"   " "
9  ( 1 )  "*" "*"   " "  "*" " " "*" "*"  "*"  " " "*"    "*"   "*"   " "
(Intercept) -0.7804363
zn          -0.2557605
indus        0.1005082
nox          0.3950757
age         -0.1351499
disf        -0.1320510
rad          1.2399358
ptratio     -0.1085424
black       -0.1209850
lstat        0.1965521
```

– The results above shows that the variables 'chas','rm','tax',and 'mdev' were dropped out of model.

– Consider the coefficient,if just focus on one coefficient,and keep others unchanged.
The four positive predictor variables indicate that the higher proportion of non-retail business acres per town(indus),higher nitrogen oxides concentration(nox),the higher level of index of accessibility to radial highways(rad) and larger lstat will leads the higher crime rate.And the biggest influence is from the accessibility to radial highways(rad).

– On the contrary, I think we can infer that these negative predictor variables indicate that higher proportion of owner-occupied units built prior to 1940, higher Pupil-teacher ratio, more black people(black), longer distance to the employment centres(disf) and larger proportion of residential land zone(zn) will lead the lower crime rate.

3 Regularisation For Model Performance

For the goal of this project which is to build and interpret linear regression models, and another method of making a model better is regularisation methods.And

when we calculate the loss to estimate the model,it can be equivalent to finding the values which minimize the residual sum of squares.And the grid of tuning parameter set as:

$$grid = 10^{seq(5,-3,length=100)} \tag{1}$$

3.1 Ridge Regression

- First,plot how the effect of varying the tuning parameter,we can see that the time log λ is around 6 (red line)(i.e. λ is around e^6),all regression coefficients are essentially equal to zero(Appendix Fig.6).
- When $\lambda = 10^5$, the regression coefficient for each explanatory variable has been shrunk all the way to zero, too much penalty. When $\lambda = 0.1047616$ the estimates have been shrunk a little.When $\lambda = 0.001$,it seems the penalty is very small and the result is a bit same as the original model use least squares.

```
> beta_hat = coef(ridge_model)
> length(grid)
[1] 100
> dim(beta_hat)
[1]  14 100
> grid[1]
[1] 1e+05
> beta_hat[,1]
  (Intercept)          zn        indus         chas          nox
      rm
-7.804363e-01 -2.409910e-05 3.405973e-05 1.328013e-06 3.675269e-05
    -1.430442e-05
          age         disf          rad          tax      ptratio
        black
-3.067795e-05 -3.181692e-05 3.977134e-05 3.859712e-05 1.815333e-05
    -2.231052e-05
        lstat         medv
 2.920020e-05 -2.116971e-05
> grid[75]
[1] 0.1047616
> beta_hat[,75]
 (Intercept)          zn       indus        chas         nox          rm
            age
-0.780436262 -0.249676389 0.078650629 0.001768562 0.397141823
    -0.023442395 -0.142821776
       disf         rad         tax     ptratio       black       lstat
                  medv
-0.131353673 0.995930573 0.204401723 -0.060602892 -0.143614613
    0.218270122 0.073924130
> grid[100]
```

```
[1] 0.001
> beta_hat[,100]
 (Intercept)          zn       indus        chas         nox          rm
             age
-0.780436262 -0.253808629 0.106692018 -0.009157113 0.410342728
    -0.036908970 -0.137483301
       disf         rad         tax     ptratio       black       lstat
                  medv
-0.117350910 1.245600968 -0.017975337 -0.090538268 -0.131195644
    0.220809226 0.076990673
```

– Also,plot the cross validation error varies with λ,for each value of λ, the mean MSE across the k folds is plotted along with error bars which cover the mean plus or minus one standard error.(Appendix Fig.7)

– Finally, get the coefficients below.Compare to the coefficients using subset selection methods, four of the variables 'chas','rm' , 'tax' and 'medv',which were dropped by subset selection methods,have been shrunk towards zero.

```
> #regression coefficients with chosen lambda
> coef(ridge_model, s = lambda_min)
(Intercept) -0.780436262
zn          -0.254690358
indus        0.100716880
chas        -0.007794949
nox          0.409975446
r m          -0.035364898
age         -0.137485959
disf        -0.118370351
rad          1.213963512
tax          0.012845066
ptratio     -0.087232567
black       -0.132634562
lstat        0.221705050
medv         0.078129955
```

– I think the fitted model using ridge regularisation will always include all explanatory variables. This is because the coefficients which were shrinked, are just towards zero but not exactly equal to zero.

3.2 The LASSO

First,plot the effect of varying the tuning parameter(Appendix,Fig.8), As the value of the tuning parameter λ increases, the third variable 'chas' is the first to drop out of the model, followed by the ninth variable 'tax', and so on.Also plot the cross validation scores(Appendix Fig.9).It cannot directly be compared

the cross-validation error for the optimal LASSO fit with that obtained for the optimal ridge regression fit. This is because we did not use the same subsets (or folds) of variables during cross-validation.(foldid will be considered later)

```
> coef(lasso_model, s = lambda_min)
(Intercept) -0.780436262
zn          -0.242034979
indus        0.094252196
chas         .
nox          0.415140618
r m          -0.008286218
age         -0.131674392
disf        -0.128418902
rad          1.221865015
tax          .
ptratio     -0.075504633
black       -0.119448193
lstat        0.200425485
medv         0.027035744
```

As we can see above, the predictor variables 'chas' and 'tax' were dropped. Compared to the coefficients of original fitted model,the positive and negative of the relationship between the response variable and predictor variables are same for ridge and lasso fitted model.So I think the meaning of each predictor variables are not change,and also the coefficients give the expected increase or decrease in the response variable when increases by one whilst all other predictors remain unchanged.The changed point is that how the model focus on each predictor variables.

```
               Original    ridge         lasso
(Intercept) -0.780436262 -0.780436262 -0.780436262
zn          -0.253298005 -0.254926894 -0.242034979
indus        0.108367541 0.098268719 0.094252196
chas        -0.009447935 -0.007177487 .
nox          0.409870712 0.409636690 0.415140618
rm          -0.037400310 -0.034680019 -0.008286218
age         -0.137771399 -0.137603251 -0.131674392
disf        -0.117401574 -0.118919810 -0.128418902
rad          1.252885339 1.199698170 1.221865015
tax         -0.024989580 0.026560298 .
ptratio     -0.091349810 -0.085717024 -0.075504633
black       -0.130822597 -0.133287287 -0.119448193
lstat        0.220121875 0.221911105 0.200425485
medv         0.076417050 0.078472498 0.027035744
```

– As shown below, a bit same as ridge regression.When $\lambda = 10^5$, the regression coefficient for most explanatory variable has been shrunk to be exactly zero, too much penalty. When $\lambda = 0.1047616$ the estimates have been shrunk a

little.When $\lambda = 0.001$,it seems the penalty is very small and the result is a bit same as the original model use least squares.

```
> beta_hat = coef(lasso_model)
> length(grid)
[1] 100
> dim(beta_hat)
[1]  14 100
> grid[1]
[1] 1e+05
> beta_hat[,1]
(Intercept)        zn      indus       chas        nox         rm
    age
 -0.7804363  0.0000000  0.0000000  0.0000000  0.0000000  0.0000000
     0.0000000
      disf        rad        tax    ptratio      black      lstat
          medv
  0.0000000  0.0000000  0.0000000  0.0000000  0.0000000  0.0000000
      0.0000000
> grid[75]
[1] 0.1047616
> beta_hat[,75]
(Intercept)        zn      indus       chas        nox         rm
    age
-0.78043626 -0.16004650 0.07127118 0.00000000 0.46496438 0.00000000
    -0.10969541
      disf        rad        tax    ptratio      black      lstat
          medv
-0.13384434 1.15037455 0.00000000 0.00000000 -0.06186513 0.14669615
    0.00000000
> grid[100]
[1] 0.001
> beta_hat[,100]
 (Intercept)         zn       indus        chas         nox          rm
             age
-0.780436262 -0.254605408 0.101052051 -0.006923766 0.409801268
    -0.034067647 -0.136571427
       disf         rad         tax     ptratio       black       lstat
                  medv
-0.118674239 1.235801323 -0.003622775 -0.089815026 -0.129725109
    0.218528221 0.072205113
```

– According to the different penalty and different coefficients,they shows the application and main difference between two kinds of regularisation methods.For lasso,we can use to select the predictor variables,because the variables in lasso will be punished to zero one by one.But for ridge, the coefficients of predictor variable will go down and become to zero almost to-

gether,so its application is to find a good tuning parameter and punish the model to increase the performance.

4 Choose Best Model by Cross Validation

– For comparison fair,the point is to make the same partition of the data into k segments for my three 'best' model.So I use 'set.seed' and keep the foldid of Ridge and Lasso same with best subset selection.And the estimation MSE, it is about the total mean error between the model output and real value, so it should be chosen as the least, the smaller the MSE is,the better the model performance.

```
set.seed(1)
nfolds = 10
folds_index = sample(nfolds, nrow(df) ,replace = TRUE)
folds_sizes = numeric(nfolds)
############Best subset selection#################################
> (best_cv_b = which.min(bss_mse_b))
[1] 9
#MSE
> (bss_mse = bss_mse_b[best_cv_b])
[1] 0.5986935
####################Ridge#########################################
> ridge_cv_model = cv.glmnet(X, Y, alpha = 0, standardize = FALSE,
    lambda = grid, foldid = folds_index)
> (lambda_min1 = ridge_cv_model$lambda.min)
[1] 0.01353048
> #Index of which tuning parameter was minimum
> (index = which(ridge_cv_model$lambda == ridge_cv_model$lambda.min))
[1] 86
> #Corresponding mse
> (ridge_mse = ridge_cv_model$cvm[index])
[1] 0.6053586
############################Lasso################################
> (lambda_min2 = lasso_cv_model$lambda.min)
[1] 0.009326033
> #Index of which tuning parameter was minimum
> (index = which(lasso_cv_model$lambda == lasso_cv_model$lambda.min))
[1] 88
> #Corresponding mse
> (lasso_cv_model$cvm[index])
[1] 0.6017774
```

– So from the MSE calculated above, I choose the model from best subset selection M_9.

5 Best Model

For the model we choose—M_9 from best subset selection. And there are three predictor variables were dropped. We can easily check the correlation matrix in Fig.2,the dropped predictor variables has very weak correlation to response variable.And the model directly try many times and get the best model by a estimation function.And I think dropping the uncorrelated variables not only can reduce the dimension but also make the fitness easier.

- The best subset selection methods is very easy and directly.Because of the small number of observations and a bit small number of predictor variables, also a bit simple model,so this method didn't cost lots of time,and if the p is very large,this method can't be used.And I think the method use a very direct way, the predictor variables just add or cut without any trick.So I think it is easy to be over fitting and high variance.With the dropping the data, there are some necessary but very small thing are also dropped.We can check the MSE between lasso and M_9 are very small, and I think it is possible to try lasso.

- About the ridge regression, I think some predictor variables influence very very to the model,I think we can drop them and avoid some wrong things.

- Future work,I think not only check from scatterplot but also check the MSE results,there is little weak linear relationship in our data,so these three methods are not good enough,we can make it complicate or add some non-linear function.

Appendix

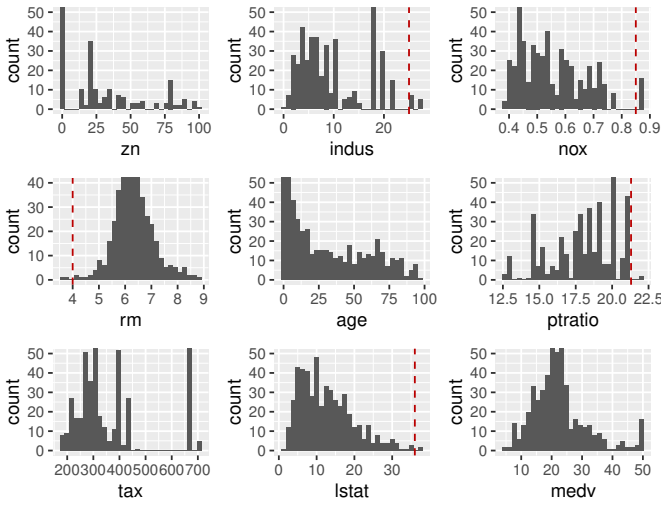

Fig. 2. Histogram for Quantitative

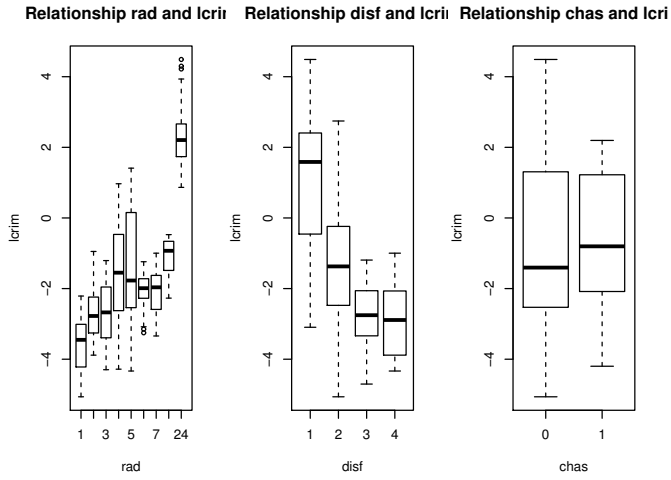

Fig. 3. Box plot for Qualitative

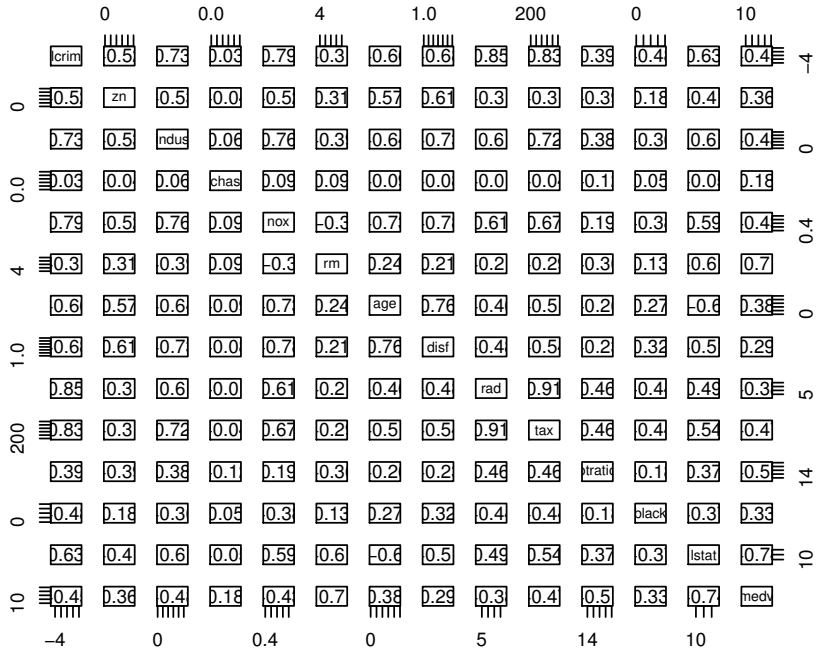

Fig. 4. Correlation matrix for all

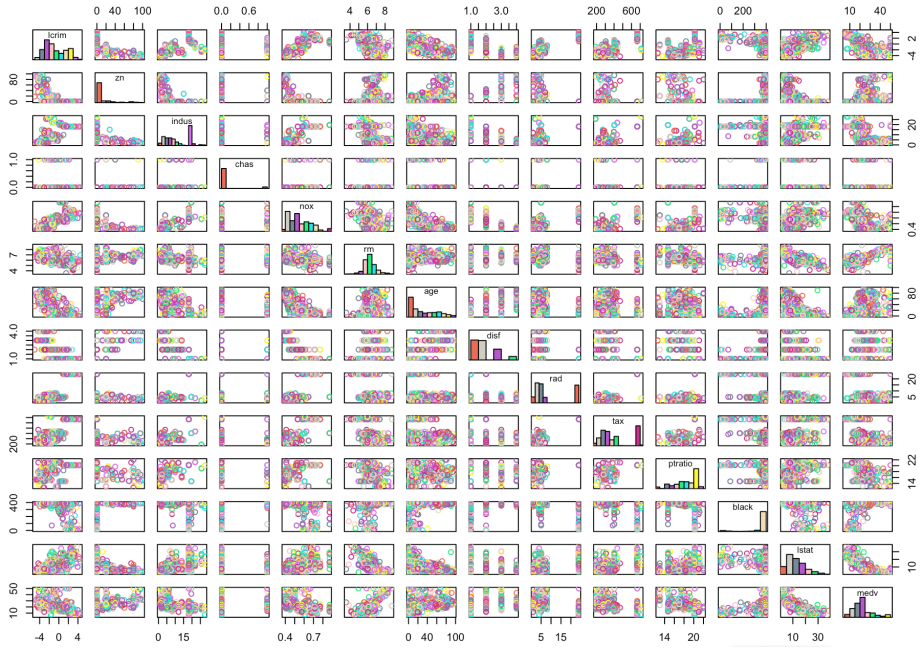

Fig. 5. ScatterPlot for all

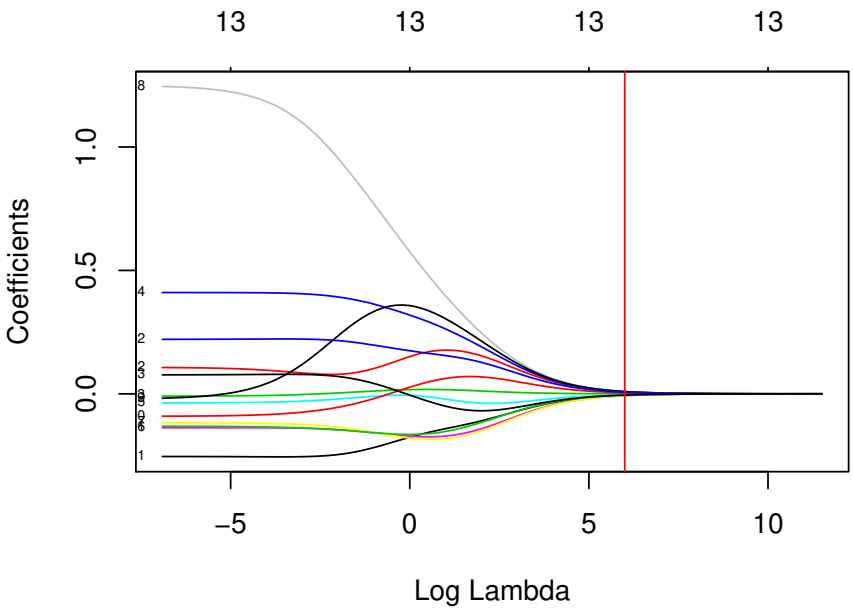

Fig. 6. Graphical illustration of the effect of varying the tuning parameter when applying ridge regression

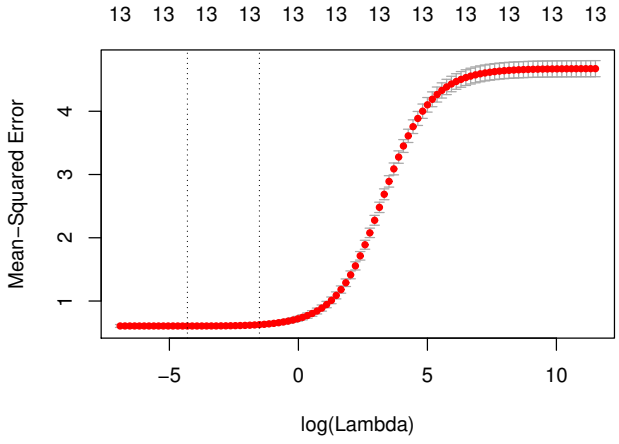

Fig. 7. Cross-validation scores

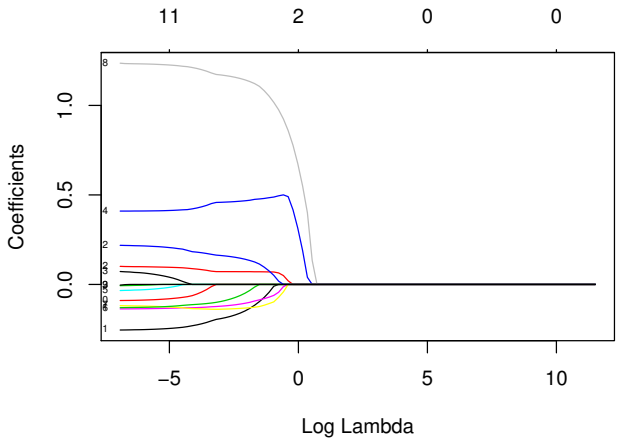

Fig. 8. Graphical illustration of the effect of varying the tuning parameter when applying Lasso regression

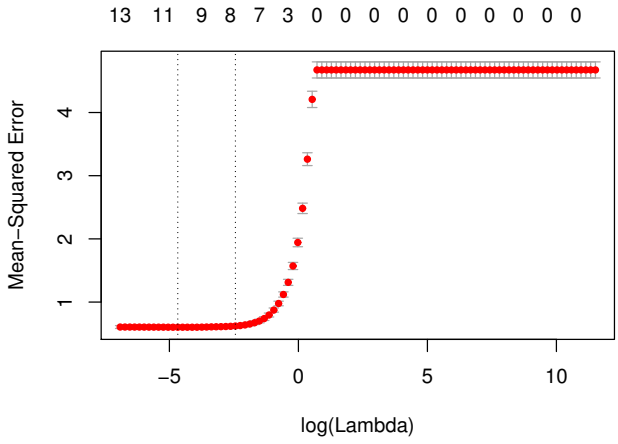

Fig. 9. Cross-validation scores

```
library(gghighlight)
library(tidyverse)
library(ggplot2)
library(xtable)
library(plyr)
library(patchwork)
library(nclSLR)
library(leaps)
library(glmnet)
########################Load Data#######################################
data(Boston, package = 'nclSLR', head =FALSE)
boston_data = Boston
boston_matrix = as.matrix(boston_data)
boston_sdr = scale(boston_data, center = TRUE, scale = TRUE)


#EDA#############################################################################
dim(boston_matrix)
plot(boston_data$lcrim)
plot(as.data.frame(boston_sdr)$lcrim)

##correlatino plot
panel.cor <- function(x, y){
  usr <- par("usr"); on.exit(par(usr))
  par(usr = c(0, 1, 0, 1))
  r <- round(cor(x, y), digits=2)
  txt <- paste0(r)
  cex.cor <- strwidth(txt)
  text(0.5, 0.5, txt)
}
pairs(boston_data,
      lower.panel = panel.cor,
      upper.panel = panel.cor)

##scatterplot
par(mfcol=c(1,1))
c = c('coral2', 'ivory3', 'lightslategray', 'mediumorchid3',
    'rosybrown1', 'springgreen2', 'turquoise2', 'wheat', 'yellow1',
    'mediumorchid1', 'maroon3')
panel.hist <- function(x, ...)
{
  usr <- par("usr"); on.exit(par(usr))
  par(usr = c(usr[1:2], 0, 1.5) )
  h <- hist(x, plot = FALSE)
  breaks <- h$breaks; nB <- length(breaks)
  y <- h$counts; y <- y/max(y)
  rect(breaks[-nB], 0, breaks[-1], y, ...)
}
pairs(boston_matrix, col = c,
```

```
     diag.panel = panel.hist)

##boxplot for qualitative variables
par(mfcol=c(1,3))
boxplot(lcrim~rad, data = Boston, main = 'Relationship rad and lcrim',
    xlab= 'rad', ylab = 'lcrim')
boxplot(lcrim~disf, data = Boston, main = 'Relationship disf and lcrim',
    xlab= 'disf', ylab = 'lcrim')
boxplot(lcrim~chas, data = Boston, main = 'Relationship chas and lcrim
    ', xlab= 'chas', ylab = 'lcrim')

##histogram for quantitative variables
boston = boston_data[-1]
boston_ctn = boston[-7]
boston_ctn = boston_ctn[-7]
boston_ctn = boston_ctn[-3]
p2 = ggplot(boston_ctn) + geom_histogram(aes(boston_ctn$zn)) +
    coord_cartesian(ylim = c(0, 50)) + xlab('zn')
p3 = ggplot(boston_ctn) + geom_histogram(aes(x = boston_ctn$indus)) +
    coord_cartesian(ylim = c(0, 50))+ geom_vline(aes(xintercept=25),
    colour="#BB0000", linetype="dashed") + xlab('indus')
p5 = ggplot(boston_ctn) + geom_histogram(aes(boston_ctn$nox)) +
    coord_cartesian(ylim = c(0, 50))+ geom_vline(aes(xintercept=0.85),
    colour="#BB0000", linetype="dashed") + xlab('nox')
p6 = ggplot(boston_ctn) + geom_histogram(aes(boston_ctn$rm)) +
    coord_cartesian(ylim = c(0, 40)) + geom_vline(aes(xintercept=4),
    colour="#BB0000", linetype="dashed") + xlab('rm')
p7 = ggplot(boston_ctn) + geom_histogram(aes(boston_ctn$age)) +
    coord_cartesian(ylim = c(0, 50))+ xlab('age')
p11 = ggplot(boston_ctn) + geom_histogram(aes(boston_ctn$ptratio)) +
    coord_cartesian(ylim = c(0, 50))+ geom_vline(aes(xintercept=21.3),
    colour="#BB0000", linetype="dashed") +xlab('ptratio')
p12 = ggplot(boston_ctn) + geom_histogram(aes(boston_ctn$tax)) +
    coord_cartesian(ylim = c(0, 50))+ xlab('tax')
p13 = ggplot(boston_ctn) + geom_histogram(aes(boston_ctn$lstat)) +
    coord_cartesian(ylim = c(0, 50))+ geom_vline(aes(xintercept=36),
    colour="#BB0000", linetype="dashed") + xlab('lstat')
p14= ggplot(boston_ctn) + geom_histogram(aes(boston_ctn$medv)) +
    coord_cartesian(ylim = c(0, 50))+ xlab('medv')

p2 + p3 + p5 + p6 + p7 + p11 + p12 + p13 + p14

######################################################################

##response and predictors variables
x = as.matrix(boston_data[,-1])
y = as.matrix(boston_data[,1])

x = scale(x)
```

```
#######Original full Linear Model

class(x)
class(y)
df = data.frame(y,x)

########subset selection method
p = ncol(x)

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)
  return(mat[,xvars] %*% coefi)
}

####Bset subset selection:
bss1 = regsubsets(y ~ ., data = df, method = 'exhaustive', nvmax = p)
Summary = summary(bss1)
adjr2 = Summary$adjr2
cp = Summary$cp
bic = Summary$bic
coef(bss1,13)
best_adjr2 = which.max(adjr2)
best_adjr2
best_cp = which.min(cp)
best_cp
best_bic = which.min(bic)
best_bic
as.matrix(coef(bss1,6))
#numerical describle
summary(bss1)

bss2 = regsubsets(y ~ ., data = df, method = 'forward', nvmax = p)
Summary2 = summary(bss2)
adjr22 = Summary2$adjr2
cp2 = Summary2$cp
bic2 = Summary2$bic
coef(bss2,13)
best_adjr22 = which.max(adjr22)
best_adjr22
best_cp2 = which.min(cp2)
best_cp2
best_bic2 = which.min(bic2)
best_bic2
as.matrix(coef(bss2,6))
#numerical describle
summary(bss2)
```

```
######################subset selection using cross validation#######
set.seed(1)
nfolds = 10
folds_index = sample(nfolds, nrow(df) ,replace = TRUE)
folds_sizes = numeric(nfolds)
cv_bss_errors_b = matrix(NA, p, nfolds)
cv_bss_errors_f = matrix(NA, p, nfolds)

#how many observations are assigned to each of the 10 folds
for(k in 1:nfolds){
  folds_sizes[k] = length(which(folds_index == k))
}

for(u in 1:nfolds) {
  # Fit models M_1,...,M_p by best-subset selection using all but the
      k-th fold
  bss_tmp_fit_b = regsubsets(y ~ ., data=df[folds_index!=u,],
      method='exhaustive',nvmax=p)
  # For each model M_m where m=1,...,p:
  for(m in 1:p) {
    # Compute fitted values for the k-th fold
    bss_tmp_predict_b = predict(bss_tmp_fit_b, df[folds_index==u,], m)
    # Work out MSE for the k-th fold
    cv_bss_errors_b[m, u] = mean((df[folds_index==u,]$y -
        bss_tmp_predict_b)^2)
  }
}
bss_mse_b = numeric(p)

# For models M_1,...,M_p:
for(j in 1:p) {
  bss_mse_b[j] = weighted.mean(cv_bss_errors_b[j,], w=folds_sizes)
}
(best_cv_b = which.min(bss_mse_b))
(bss_mse = bss_mse_b[best_cv_b])


par(mfrow=c(2,2))
## Produce plots, highlighting optimal value of k:
plot(1:p, adjr2, xlab="Number of predictors", ylab="Adjusted Rsq",
    type="b")
points(best_adjr2, Summary$adjr2[best_adjr2], col="red", pch=16)

plot(1:p, cp, xlab="Number of predictors", ylab="Cp", type="b")
points(best_cp, Summary$cp[best_cp], col="red", pch=16)

plot(1:p, bic, xlab="Number of predictors", ylab="BIC", type="b")
points(best_bic, Summary$bic[best_bic], col="red", pch=16)
```

```
plot(1:p, bss_mse_b, xlab="Number of predictors", ylab="10-fold CV
    Error", type="b")
points(best_cv_b, bss_mse_b[best_cv_b], col="red", pch=16)

####################Cross validation
    Test#######################################



#######Regression
grid = 10^seq(5, -3, length = 100)

X = as.matrix(df[,-1])

Y = as.matrix(df[,1])


####ridge regression
ridge_model = glmnet(X, Y, alpha = 0, standardize = FALSE, lambda = grid)

ridge_cv_model = cv.glmnet(X, Y, alpha = 0, standardize = FALSE, lambda
    = grid, foldid = folds_index)

par(mfcol=c(1,1))
plot(ridge_model, xvar = 'lambda', col=1:12,label=TRUE)
abline(v = 6,col= 'red')
plot(ridge_cv_model)

beta_hat = coef(ridge_model)
length(grid)
dim(beta_hat)
grid[1]
beta_hat[,1]
grid[75]
beta_hat[,75]
grid[100]
beta_hat[,100]

(lambda_min1 = ridge_cv_model$lambda.min)
#Index of which tuning parameter was minimum
(index = which(ridge_cv_model$lambda == ridge_cv_model$lambda.min))
#Corresponding mse
(ridge_mse = ridge_cv_model$cvm[index])
#regression coefficients with chosen lambda
crdg = coef(ridge_model, s = lambda_min1)

####Lasso regression
lasso_model = glmnet(X, Y, alpha = 1, standardize = FALSE, lambda = grid)
```

```
lasso_cv_model = cv.glmnet(X, Y, alpha = 1, standardize = FALSE, lambda
    = grid, foldid = folds_index)

lasso_mse = lasso_cv_model$cvm
par(mfcol=c(1,1))
plot(lasso_model, xvar = 'lambda', col=1:12,label=TRUE)
plot(lasso_cv_model)



(lambda_min2 = lasso_cv_model$lambda.min)
#Index of which tuning parameter was minimum
(index = which(lasso_cv_model$lambda == lasso_cv_model$lambda.min))
#Corresponding mse
(lasso_cv_model$cvm[index])
#regression coefficients with chosen lambda
cl = coef(lasso_model, s = lambda_min2)

beta_hat = coef(lasso_model)
length(grid)
dim(beta_hat)
grid[1]
beta_hat[,1]
grid[75]
beta_hat[,75]
grid[100]
beta_hat[,100]
```