CS6140 Assignment 2

Chengbo Gu

Problem 1

HengFang Deng, 1st year PhD student in Civil Engineering
Zexi Han, 1st Year MS student in Data Science
Chengbo Gu, 2nd Year MS student in Bioinformatics

Problem 2

How to classify Alzheimer's disease and identify those presymptomatic individuals with mild cognitive impairment (MCI) who will eventually convert to Alzheimer's is the issue addressed in Sandip Ray's manuscript. Statistical methods including significance analysis of microarrays (SAM), unsupervised clustering algorithm, predictive analysis of microarrays (PAM, 10-fold cross-validation regularization and logistic regression) were used in Ray's work. His conclusion is that 18 signaling proteins in blood plasma are found that can be used to classify blinded samples from Alzheimer's and control subjects with close to 90% accuracy. Also, these signaling proteins can help identify patients who had MCI that progressed to Alzheimer's disease 2-6 years later. Biological analysis of the 18 proteins points to systemic dysregulation of hematopoiesis, immune responses, apoptosis and neuronal support in presymptomatic Alzheimer's disease, which makes the conclusion more convincing.

The first and most interesting part of Ray's work for me is that whenever he draws a conclusion, he wants to verify it from other perspectives. For instance, after SAM identified 19 proteins with highly significant differences in expression between Alzheimer's and NDC samples, he applied an unsupervised clustering algorithm based on the similarity in abundance of these 19 markers which produced two main clusters that contained mostly Alzheimer's or NDC samples respectively. Similarly, after PAM identified 18 predictors and classified Alzheimer's and NDC samples with very high accuracy, the unsupervised clustering based on these 18 markers was able to separate Alzheimer's and NDC samples at the same time. From high level, this manuscript is not merely about statistical analysis but combined with the biological one. Moreover, they are consistent with each other.

Another impressive part is that these 18 predictors also perform well in distinguishing Alzheimer's samples from other neurological diseases and rheumatoid arthritis. Though this may not be Ray's initial intention, the result does support his statement that these 18 biomarkers certainly form an Alzheimer's-specific signature.

The result and accuracy of Ray's work seems to be plausible, but it lacks model competing part from the view of machine learning. There are lots of models and algorithms that can be applied regarding classification issue. For example, Fisher's linear discriminant, naive bayes classifier, least squares support vector machines, etc. In my opinion, it is better to compare logistic regression with some other models first, yield the best model next.

To go further, we may want to figure it out why there exists a discrepancy between SAM and PAM – the missing biomarker CCL22. Why it shows significant difference but is discarded by PAM? What will the regression model and prediction error be with 19 predictors? What will the regulatory networks be connecting the 19 signaling proteins? What's the correlation between CCL22 and M-CSF?

If possible, We also want to collect more plasma samples and test the model against them. Size of only 259 plasma samples seems to be too small. Plus, if computational resource allowed, we could fit different models instead of barely Logistic regression. To sum up, the aforementioned work would make the research more persuasive and richer in content if finished.

Problem 3

project number: 212

URL:http://cs229.stanford.edu/proj2015/212_report.pdf

To make predictions regarding post-college earnings and debt of alumni by machine learning models is the issue discussed in Monica's project. Techniques like single regression imputation, sequential forward-based feature selection, linear/locally weighted linear/K-nearest neighbors/support vector regression and neural networks were applied during the process. The incremental model selection process indicated that regression imputation of privacy-suppressed values improved performance of models. Besides, weighted linear regression outperformed other models with below 10% and 18% average error for earning and debt data respectively. Moreover, this model performed well towards law schools though law schools made up only little of the data.

One of the most attractive highlights in Monica's project is privacy-suppressed values handling. Instead of simply removing all features with any privacy-suppressed entries or merely setting missing values to the mean of observed values, single regression imputation was applied. To make the regression model statistically meaningful, they imposed a requirement that imputed features must have missing data for less than 30% of schools. Since privacy-suppressed values occurred in potentially useful metrics, the aforementioned implementation does increase the credibility of the model.

Model competing is a fascinating part in this project as well. There are totally 5 classes of optimized models among which best weighted linear regression model outperformed others. The authors discussed the superiority of the best weighted linear regression by exploring whether the model generalized well outside their dataset and got a positive result. Besides, they also analyzed the reason why support vector regression did much worse than all other models. Due to the reasonable steps that they gone through model refining, one could safely draw the conclusion that the best model among the 5 was indeed the best weighted linear regression.

Although there is no doubt that this project is amazing with huge workload, there are still some deficiencies as far as I am concerned. The very first one is the representativeness of the dataset the authors selected. Their goal was to make prediction of post-collegiate earnings and debt while the data they collected was from post-college earnings and debt of alumni who were on federal financial aid. Students who received federal financial aid may share some unique features that potentially influenced their earnings and debt. Without eliminating this possibility, the authors couldn't simply assume that their data is representative. I do know the fact that there are some cases when it is extreme hard to retrieve data, but I strongly recommend obtaining data from all kinds of alumni.

The second one could be the poor performance of their best model given data of trade schools. As the author stated, the current algorithm has trouble extending to such schools, which indicates that future earnings and debt might be best characterized by a different set of features.

For future work, partnering with the U.S. Department of Education to gain access to all kinds of data (from all kinds of alumni and maybe some privacy-suppressed data) would be a big plus. More generalized models for imputation can be involved to handle the remaining privacy-suppressed data better if given enough computational resource. Finally, improving the performance towards trade schools by possibly selecting new features could form the project an organic whole.

Problem 4

(a) Select the training set

First, we read the data, retrieve trainSet and testSet. Also build test model which will be used to calculate mean sqaure prediction error later.

```
mydata <- as.data.frame(read.table("prostate.txt"))
trainSet <- mydata[mydata$train == "TRUE", -c(10)]
testSet <- mydata[mydata$train == "FALSE", -c(10)]
x.test <- model.matrix(lpsa ~ ., data = testSet)
y.test <- testSet$lpsa
```

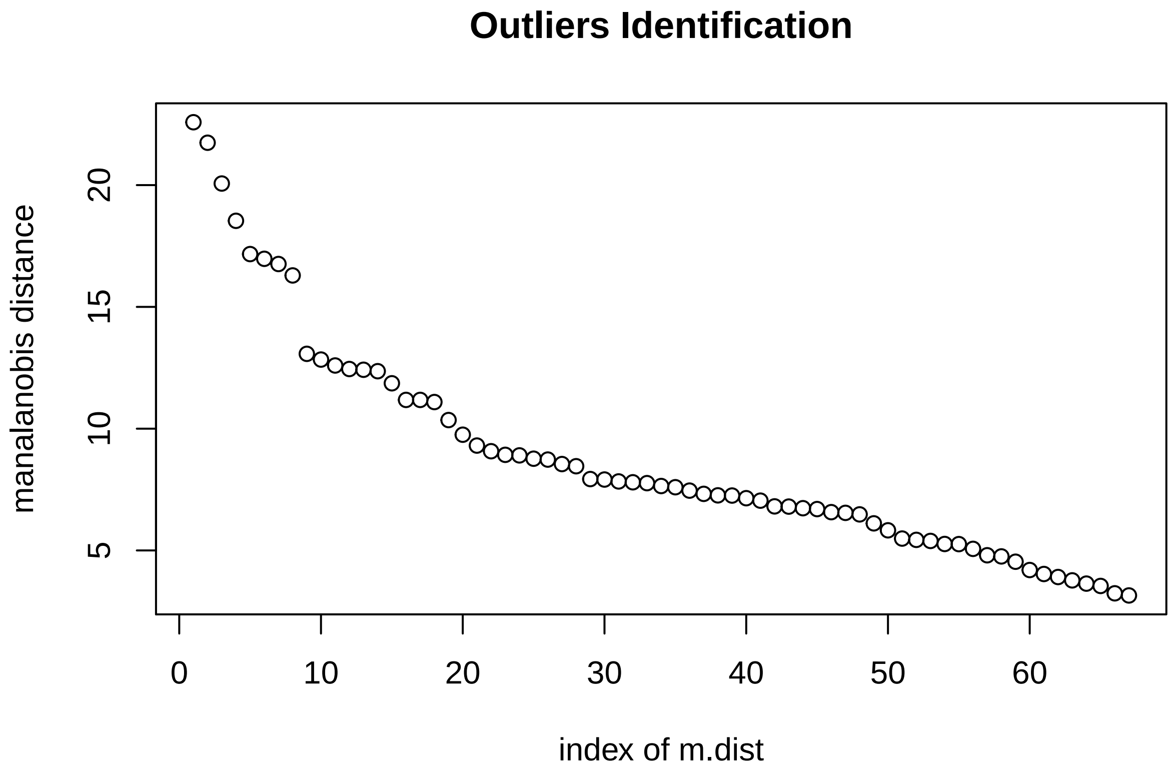
(b) Data exploration

In this section, we will try to identify outliers from training set and perform one-variable and two-variable summary statistics.

```
sum(is.na.data.frame(mydata))
```

```
## [1] 0
```

```
# identify outliers
m.dist <- sort(mahalanobis(trainSet, colMeans(trainSet), cov(trainSet)), decreasing=TRUE)
plot(m.dist, ylab="manalanobis distance", xlab="index of m.dist",
     main = "Outliers Identification")
```



Discussion

As we can see above, there is no missing value in our dataset. Besides, the plot indicates that the first 8 patients are seems to be outliers. However, different from the rest doesn't necessarily mean wrong as we discussed in class. Since the number of observations is small, we decided not to remove the first 8 patients with large mahalanobis distance. After this, we can apply one-variable and two-variable summary.

one variable summary

simple one variable summary

```
# one variable summary
one.variable.summary <- summary(trainSet, digits=2)
one.variable.summary
```

```
##      lcavol         lweight         age           lbph
##  Min.   :-1.35   Min.   :2.4   Min.   :41   Min.   :-1.386
##  1st Qu.: 0.49   1st Qu.:3.3   1st Qu.:61   1st Qu.:-1.386
##  Median : 1.47   Median :3.6   Median :65   Median :-0.051
##  Mean   : 1.31   Mean   :3.6   Mean   :65   Mean   : 0.071
##  3rd Qu.: 2.35   3rd Qu.:3.9   3rd Qu.:69   3rd Qu.: 1.548
##  Max.   : 3.82   Max.   :4.8   Max.   :79   Max.   : 2.326
##       svi             lcp           gleason        pgg45
##  Min.   :0.00   Min.   :-1.39   Min.   :6.0   Min.   :  0
##  1st Qu.:0.00   1st Qu.:-1.39   1st Qu.:6.0   1st Qu.:  0
##  Median :0.00   Median :-0.80   Median :7.0   Median : 15
##  Mean   :0.22   Mean   :-0.21   Mean   :6.7   Mean   : 26
##  3rd Qu.:0.00   3rd Qu.: 0.99   3rd Qu.:7.0   3rd Qu.: 50
##  Max.   :1.00   Max.   : 2.66   Max.   :9.0   Max.   :100
##       lpsa
##  Min.   :-0.43
##  1st Qu.: 1.67
##  Median : 2.57
##  Mean   : 2.45
##  3rd Qu.: 3.37
##  Max.   : 5.48
```

Of course we can explore some more details with following commands

```
format(round(stat.desc(trainSet, basic=F), 2), nsmall = 2)
```

```
##              lcavol lweight   age  lbph  svi   lcp gleason  pgg45 lpsa
## median         1.47    3.60 65.00 -0.05 0.00 -0.80    7.00  15.00 2.57
## mean           1.31    3.63 64.75  0.07 0.22 -0.21    6.73  26.27 2.45
## SE.mean        0.15    0.06  0.92  0.18 0.05  0.17    0.09   3.58 0.15
## CI.mean.0.95   0.30    0.12  1.83  0.36 0.10  0.34    0.17   7.15 0.29
## var            1.54    0.23 56.28  2.14 0.18  1.96    0.50 858.59 1.46
## std.dev        1.24    0.48  7.50  1.46 0.42  1.40    0.71  29.30 1.21
## coef.var       0.95    0.13  0.12 20.49 1.88 -6.54    0.11   1.12 0.49
```

```
format(round(stat.desc(trainSet, desc=F), 2), nsmall=2)
```

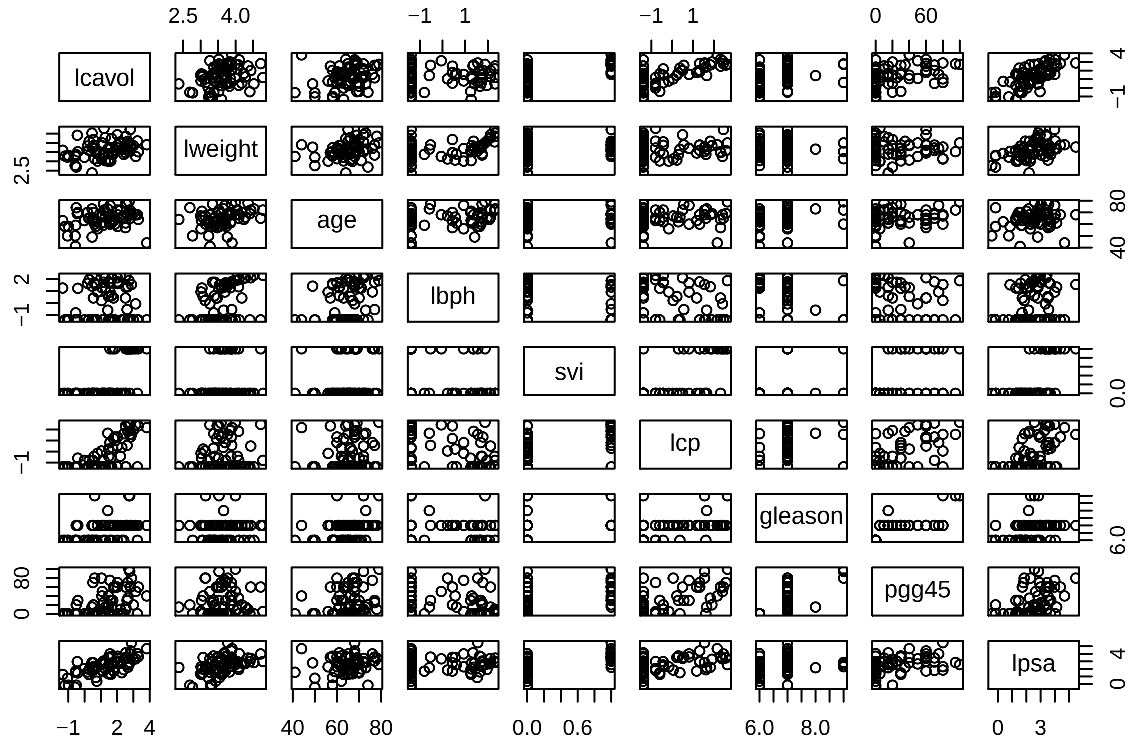
```
##          lcavol lweight    age  lbph   svi    lcp gleason   pgg45   lpsa
## nbr.val   67.00   67.00  67.00 67.00 67.00  67.00   67.00   67.00  67.00
## nbr.null   0.00    0.00   0.00  0.00 52.00   0.00    0.00   25.00   0.00
## nbr.na     0.00    0.00   0.00  0.00  0.00   0.00    0.00    0.00   0.00
## min       -1.35    2.37  41.00 -1.39  0.00  -1.39    6.00    0.00  -0.43
## max          3.82    4.78   79.00  2.33  1.00   2.66    9.00  100.00   5.48
## range        5.17    2.41   38.00  3.71  1.00   4.04    3.00  100.00   5.91
## sum         88.00  242.95 4338.00  4.79 15.00 -14.35  451.00 1760.00 164.31
```

two variable summary

```
# two variable summary
round(cor(trainSet), digits=2)
```

```
##          lcavol lweight  age  lbph   svi   lcp gleason pgg45 lpsa
## lcavol     1.00    0.30 0.29  0.06  0.59  0.69    0.43  0.48 0.73
## lweight    0.30    1.00 0.32  0.44  0.18  0.16    0.02  0.07 0.49
## age        0.29    0.32 1.00  0.29  0.13  0.17    0.37  0.28 0.23
## lbph       0.06    0.44 0.29  1.00 -0.14 -0.09    0.03 -0.03 0.26
## svi        0.59    0.18 0.13 -0.14  1.00  0.67    0.31  0.48 0.56
## lcp        0.69    0.16 0.17 -0.09  0.67  1.00    0.48  0.66 0.49
## gleason    0.43    0.02 0.37  0.03  0.31  0.48    1.00  0.76 0.34
## pgg45      0.48    0.07 0.28 -0.03  0.48  0.66    0.76  1.00 0.45
## lpsa       0.73    0.49 0.23  0.26  0.56  0.49    0.34  0.45 1.00
```

```
pairs(trainSet)
```



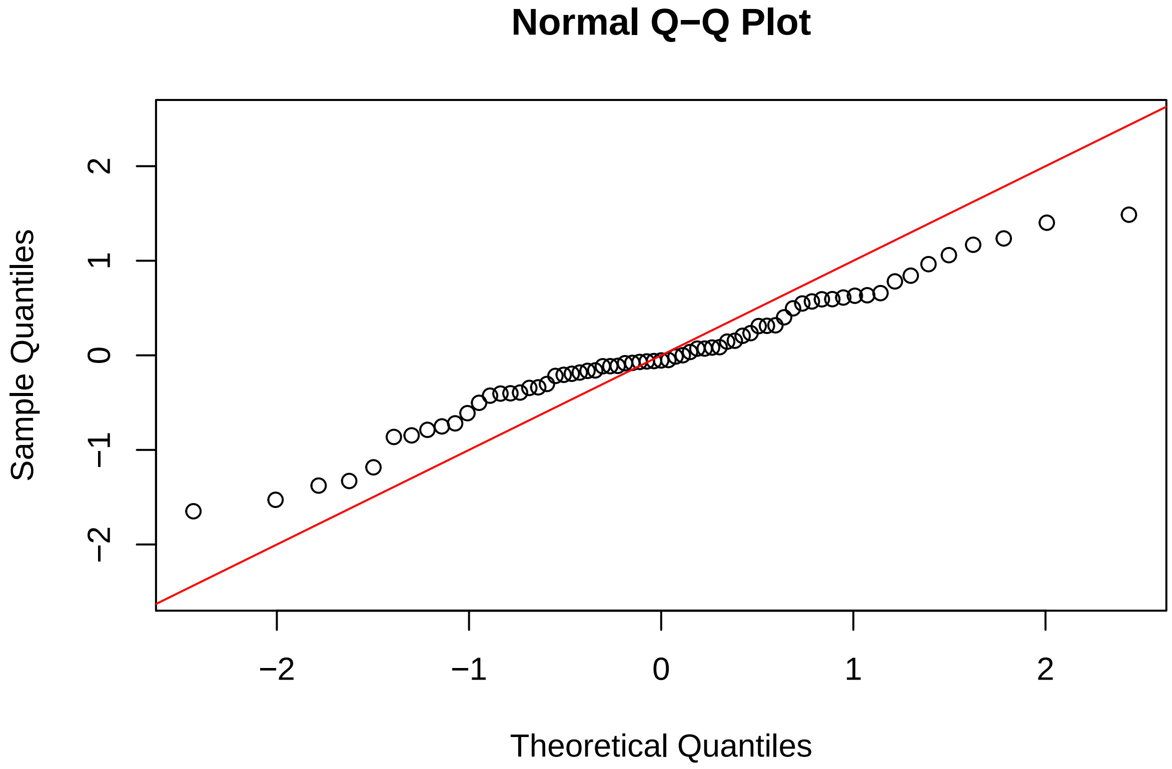
Discussion

svi is the only categorical predictor in our dataset which has the value of either 0 or 1. lcp vs lcavol, pgg45 vs lcp, pgg45 vs gleason are the paris which are highly correlated predictors.

(c) Assumption of Normality

In this section, we will validate the assumption of normality of residuals.

```r
# Assumption of Normality
fit <- lm(lpsa ~ ., data=trainSet)
y <- fit$residuals
fit.summary <- summary(fit)
qqnorm(y, ylim=c(-2.5,2.5))
abline(a=0,b=1,col="red")
```



Discussion

The red line in the qq-plot is y=x which indicates that the closer the data is to the redline, the more likely the data is normally distributed. Since we don't have many observations, this qq-plot is not that bad. We could draw the conclusion that the linearity of the points suggests that the residuals are normally distributed using all predictors.

(d) Variable selection and Performance evaluation

In this section, we will use 2 methods to select variables. They are BIC and Lasso. The reason why we won't apply ridge selection is that ridge regression shrinks all regression coeffients towards 0 at the same time which indicates that ridge doesn't really select variables.

For BIC, we first use exhaustive subset seartch to select the variables and then build interaction models based on it trying to catch the nonlinearship between variables. We report the prediction error of them at last.

For Lasso, we realize that the process is not deterministic but probabilistic because of the cross-validation. Thus, we repeat the process 1000 times, count the number of variables that lasso help us eliminate and store the mean sqaure prediction error every time. We last report the average of mean sqaure prediction error of lasso models when the number of variables eliminated is 0,1,3 respectively.

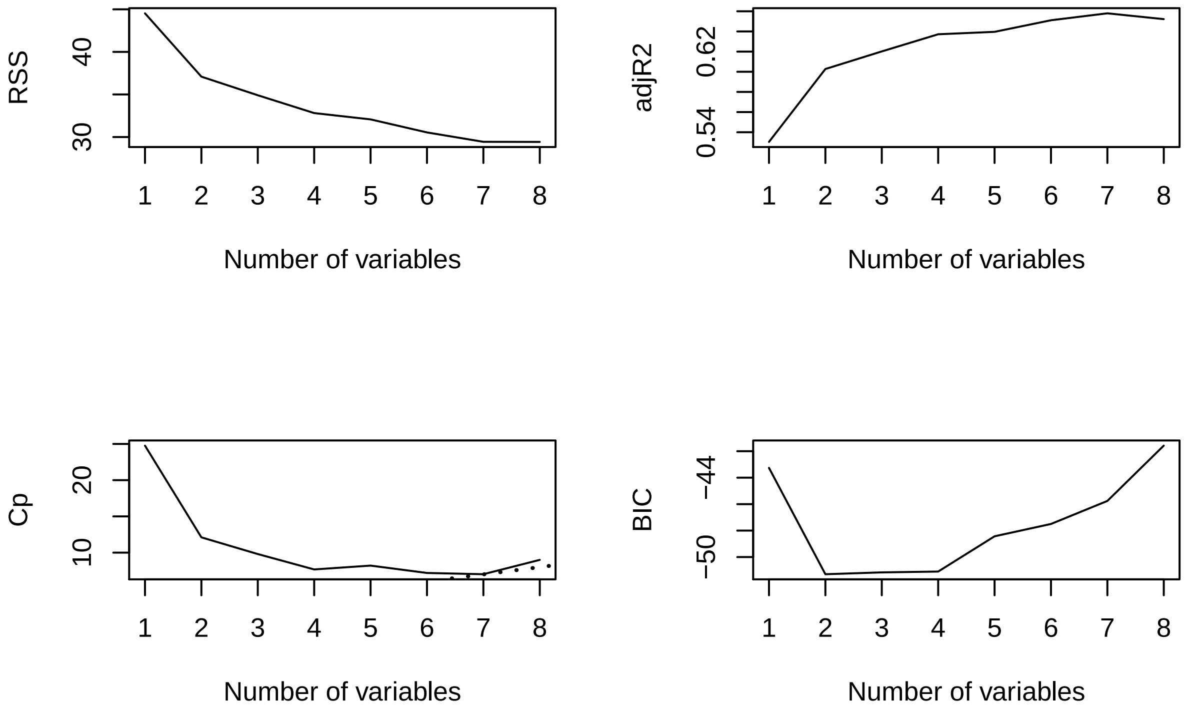
Finally, we compare the models of the two classes by their performance.

BIC

```
# variable selection
# all subset
regfit.full <- regsubsets(lpsa ~ ., data=trainSet)
reg.summary <- summary(regfit.full)
reg.summary
```

```
## Subset selection object
## Call: regsubsets.formula(lpsa ~ ., data = trainSet)
## 8 Variables  (and intercept)
##         Forced in Forced out
## lcavol      FALSE      FALSE
## lweight     FALSE      FALSE
## age         FALSE      FALSE
## lbph        FALSE      FALSE
## svi         FALSE      FALSE
## lcp         FALSE      FALSE
## gleason     FALSE      FALSE
## pgg45       FALSE      FALSE
## 1 subsets of each size up to 8
## Selection Algorithm: exhaustive
##          lcavol lweight age lbph svi lcp gleason pgg45
## 1  ( 1 ) "*"    " "     " " " "  " " " " " "     " "  
## 2  ( 1 ) "*"    "*"     " " " "  " " " " " "     " "  
## 3  ( 1 ) "*"    "*"     " " " "  "*" " " " "     " "  
## 4  ( 1 ) "*"    "*"     " " "*"  "*" " " " "     " "  
## 5  ( 1 ) "*"    "*"     " " "*"  "*" " " " "     "*"  
## 6  ( 1 ) "*"    "*"     " " "*"  "*" "*" " "     "*"  
## 7  ( 1 ) "*"    "*"     "*" "*"  "*" "*" " "     "*"  
## 8  ( 1 ) "*"    "*"     "*" "*"  "*" "*" "*"     "*"
```

```
par(mfrow=c(2,2))
plot(reg.summary$rss, xlab="Number of variables", ylab="RSS", type='l')
plot(reg.summary$adjr2, xlab="Number of variables", ylab="adjR2", type='l')
plot(reg.summary$cp, xlab="Number of variables", ylab="Cp", type='l')
abline(a=0, b=1, lty=3, lwd=2)
plot(reg.summary$bic, xlab="Number of variables", ylab="BIC", type='l')
```



```r
coefi <- coef(regfit.full, which.min(reg.summary$bic))
y.pred.full2 <- x.test[, names(coefi)] %*% coefi
predict.full2 <- mean((y.pred.full2 - y.test)^2)
coefi
```

```
## (Intercept)      lcavol     lweight
##  -1.0494396   0.6276074   0.7383751
```

```r
coefi <- coef(regfit.full, 3)
y.pred.full3 <- x.test[, names(coefi)] %*% coefi
predict.full3 <- mean((y.pred.full3 - y.test)^2)
coefi
```

```
## (Intercept)      lcavol     lweight         svi
##  -1.0227780   0.5199861   0.7367954   0.5379032
```

model refinement based on result of all subset

```r
# add new predictors
x.interaction.train <- cbind(trainSet[,1], trainSet[,2], trainSet[,9],
                             trainSet[,1]^2, trainSet[,2]^2,
                             trainSet$lcavol*trainSet$lweight)
colnames(x.interaction.train)<- c( 'lcavol', 'lweight', 'lpsa',
                                   'lcavol^2', 'lweight^2', 'lcavol*lweight')

x.interaction.test <- cbind(testSet[,1], testSet[,2], testSet[,9],
                            testSet[,1]^2, testSet[,2]^2, testSet$lcavol*testSet$lweight)
```

```r
colnames(x.interaction.test)<- c( 'lcavol', 'lweight', 'lpsa',
                                  'lcavol^2', 'lweight^2', 'lcavol*lweight')

# build model that has lc*lw predictor
x.intermodel1.test <- model.matrix(lpsa ~ .,
                                   data = as.data.frame(x.interaction.test[,-c(4:5)]))
inter1.fit <- lm(lpsa ~ ., data=as.data.frame(x.interaction.train[, -c(4:5)]))
coefi <- coef(inter1.fit)
y.inter1.pred <- x.intermodel1.test[, names(coefi)] %*% coefi
predict.inter1 <- mean((y.inter1.pred - y.test)^2)

# build model that has lc^2 and lw^2 predictors
x.quadratic.test <- model.matrix(lpsa ~ .,
                                 data = as.data.frame(x.interaction.test[,-c(6)]))
quadratic.fit <- lm(lpsa ~ ., data=as.data.frame(x.interaction.train[, -c(6)]))
coefi <- coef(quadratic.fit)
y.quadratic.pred <- x.quadratic.test[, names(coefi)] %*% coefi
predict.quadratic <- mean((y.quadratic.pred - y.test)^2)

# build model that has lc*lw, lc^2 and lw^2 predictors
x.intermodel2.test <- model.matrix(lpsa ~ ., data = as.data.frame(x.interaction.test))
inter2.fit <- lm(lpsa ~ ., data=as.data.frame(x.interaction.train))
coefi <- coef(inter2.fit)
y.inter2.pred <- x.intermodel2.test[, names(coefi)] %*% coefi
predict.inter2 <- mean((y.inter2.pred - y.test)^2)

# report mean sqaure prediction error
result <- t(as.matrix(c(predict.full2, predict.full3, predict.inter1,
                        predict.quadratic, predict.inter2)))
colnames(result) <- c('lc+lw', 'lc+lw+svi', 'lc+lw+lc*lw',
                      'lc+lw+lc^2^+lw^2^', 'lc+lw+lc*lw+lc^2^+lw^2^')
kable(result,
      caption = 'mean sqaure prediction error of BIC models')
```

Table 1: mean sqaure prediction error of BIC models

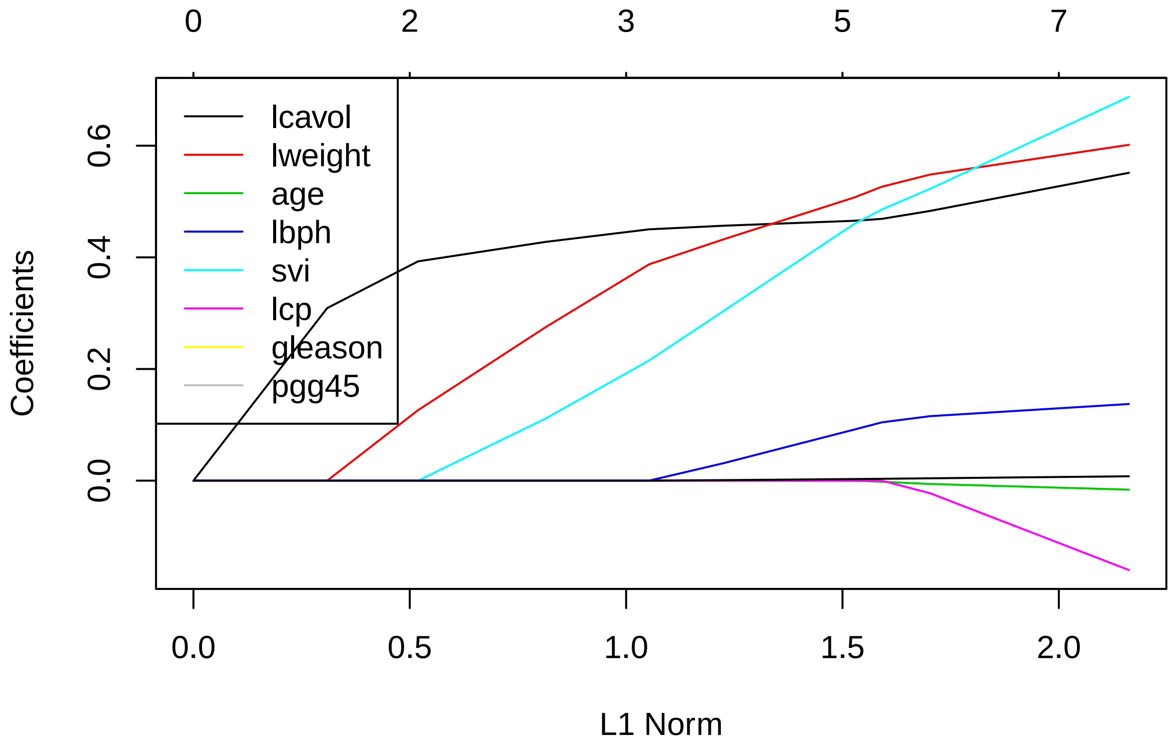
lc+lw	lc+lw+svi	lc+lw+lc*lw	lc+lw+lc^2+lw^2	lc+lw+lc*lw+lc^2+lw^2
0.4924823	0.4005308	0.5013104	0.4988285	0.4725811

Discussion

For simplicity, we only build the models up to quadratic level. The minimum BIC indicates that only lc and lw are needed to build the model. However, we also build a model on lc, lw and svi because the line of BIC is almost flat between 2 and 3. While adding lc*lw or lc^2 and lw^2 alone doesn't improve the performance of the model, combining them together yield the best model among these models using lc, lw and the combination of them as predictors. Interestingly, the model built with lc, lw and svi is the best model with minimum mean sqaure prediction error, which indicates that sometimes BIC is not that accurate because it is just a statistical criterion which uses in-sample error to infer extra-sample error.

Lasso

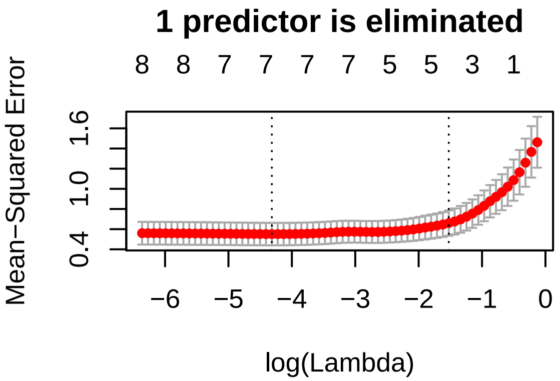
```
# lasso
grid=10^seq(10, -2, length=100)
lasso.mod <- glmnet(x=as.matrix(trainSet[,-c(9)]), y=trainSet[,9], alpha=1, lambda=grid)
plot(lasso.mod)
legend('topleft', legend = names(trainSet[,-c(9)]), col=1:8, lty=1)
```

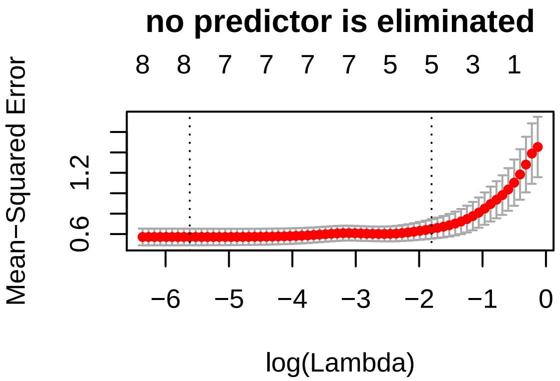


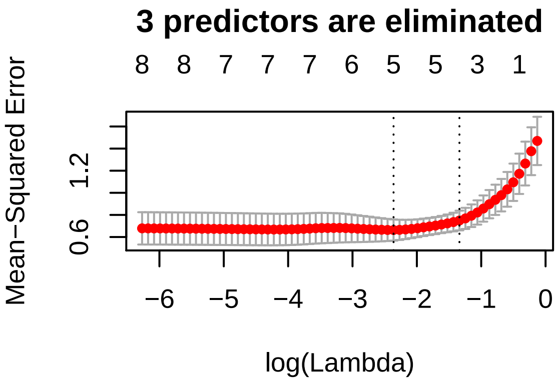
```
# 1000 times experiments to get lassoe0, lassoe1 and lassoe3

# variables initialization
counter1 <- 0; counter3 <- 0; counter0 <- 0
predict.lasso.0 <- 0; flag0 <- FALSE
predict.lasso.1 <- 0; flag1 <- FALSE
predict.lasso.3 <- 0; flag3 <- FALSE; lambda <- 0
par(mfrow=c(2,2))

# 1000 times iteration
for (i in 1:1000){
  cv.out <- cv.glmnet(x=as.matrix(trainSet[,-c(9)]), y=trainSet[,9], alpha=1)
  cv.out$lambda.min
  coefi <- coef(cv.out, s="lambda.min")
  y.pred.lasso <- predict(cv.out, newx=as.matrix(testSet[,-c(9)]), s="lambda.min")
  predict.lasso <- mean((y.pred.lasso - y.test)^2)
  # deviding into 3 cases regarding the number of eliminated variables
  if ( sum(coefi==0) == 1 ) {
    if (flag1 == FALSE) {
      plot(cv.out)
      title("1 predictor is eliminated", line = 2.5)
      flag1 <- TRUE
    }
    counter1 <- counter1 + 1
    predict.lasso.1 <- predict.lasso.1 + predict.lasso
  } else if (sum(coefi==0)==3) {
    if (flag3 == FALSE) {
      plot(cv.out)
      title("3 predictors are eliminated", line = 2.5)
      flag3 <- TRUE
      y.best.pred <- y.pred.lasso
    }
    lambda <- lambda + cv.out$lambda.min
    counter3 <- counter3 + 1
    predict.lasso.3 <- predict.lasso.3 + predict.lasso
  } else {
    if (flag0 == FALSE) {
      plot(cv.out)
      title("no predictor is eliminated", line = 2.5)
      flag0 <- TRUE
    }
    counter0 <- counter0 + 1
    predict.lasso.0 <- predict.lasso.0 + predict.lasso
  }
}
predict.lasso.1 <- predict.lasso.1/counter1
predict.lasso.3 <- predict.lasso.3/counter3
predict.lasso.0 <- predict.lasso.0/counter0
bestlambda <- lambda/counter3

# report the result
result <- t(as.matrix(c(predict.lasso.0, predict.lasso.1, predict.lasso.3)))
colnames(result) <- c('lassoe0', 'lassoe1', 'lassoe3')
kable(result, caption = 'mean sqaure prediction error of lasso models')
```

Table 2: mean sqaure prediction error of lasso models

lassoe0	lassoe1	lassoe3
0.5124439	0.4950146	0.4554594







Discussion

Let lassoe1 denote the situation when 1 predictor was eliminated (similarly for lassoe0 and lassoe3). Among 1000 times experiment, lassoe0, lassoe1 and lassoe3 appears 90, 894 and 16 times respectively. By comparing the average of mean sqaure prediction error or these three models, we draw the conclusion that the best model is built when in lassoe3 with lambda 0.0776673.

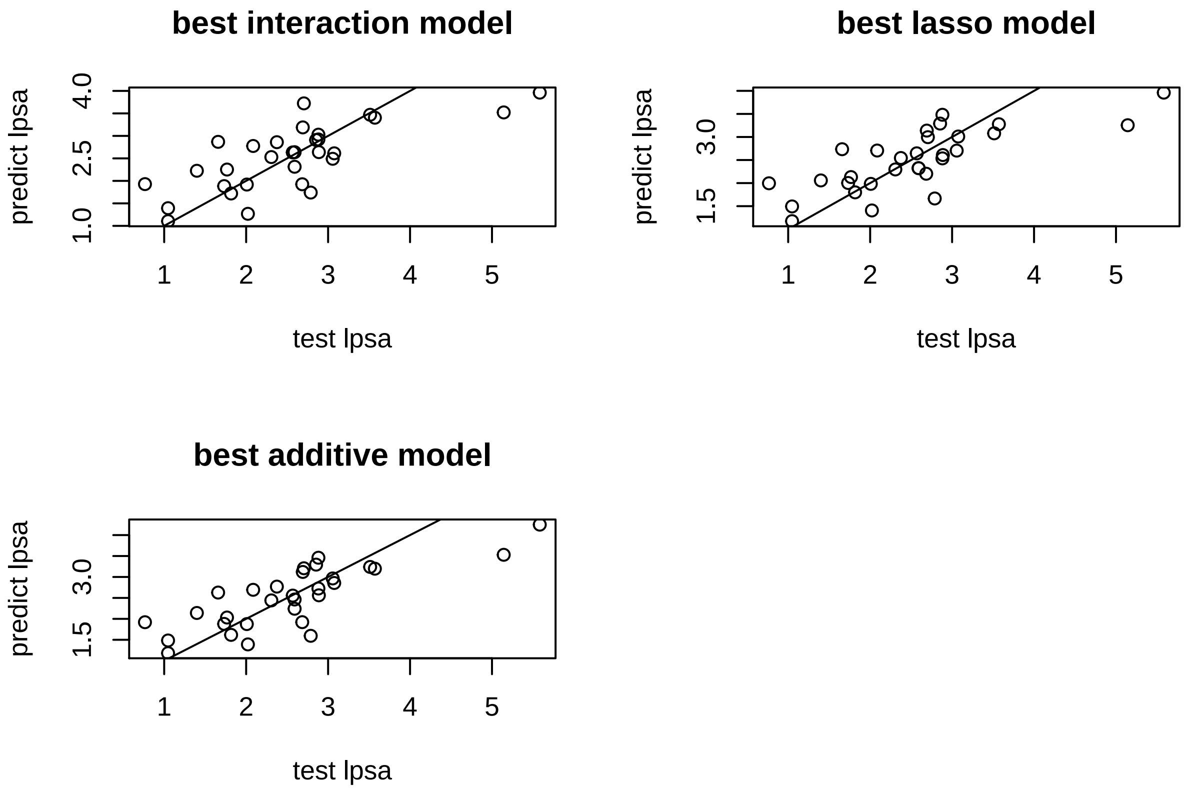
Performance evaluation and Interpretation

```
# performance evaluation
result <- t(as.matrix(c(predict.full2, predict.full3, predict.inter1,
                        predict.quadratic, predict.inter2,
                        predict.lasso.0, predict.lasso.1, predict.lasso.3)))
colnames(result) <- c('lc+lw', 'lc+lw+svi', 'lc+lw+lc*lw',
                      'lc+lw+lc^2^+lw^2^', 'lc+lw+lc*lw+lc^2^+lw^2^',
                      'lassoe0', 'lassoe1', 'lassoe3')
kable(result, caption = 'mean sqaure prediction error of all models')
```

Table 3: mean sqaure prediction error of all models

lc+lw	lc+lw+svi	lc+lw+lc*lw	lc+lw+lc^2+lw^2	lc+lw+lc*lw+lc^2+lw^2	lassoe0	lassoe1	lassoe3
0.4924823	0.4005308	0.5013104	0.4988285	0.4725811	0.5124439	0.4950146	0.4554594

Predict against test



Discussion

Finally we can evaluate the performance of models in two classes. From my point of view, for this certain dataset, there is no significant difference between the models we built. They all have mean sqaure prediction error around 0.4-0.5. But we can certainly pick the additive one with lc, lw and svi as predictors to be the best model regarding this dataset. In this model, svi is a qualitive predictor while the other two are quantative. It holds the minimum mean square prediction error which could be inferred both numerically and graphically. Compared with others, this model holds the minimum mean square prediction error. In addition, points from this model are closer to the line y = x compared with best interaction model and best lasso model. The reason it outperforms is that the number of predictors here is relatively small thus using OLS won't result in a large MSE because of variance.

However, based on bias-variance tradeoff, it would be better to introduce bias rather than having a large variance in the cases when there are only small observations but large number of features. Lasso fit is more likely to ourperform in this situation. Unlike ridge regression which does not really select features, lasso does both parameter shrinkage and variable selection automatically because it zero out the co-efficients of collinear variables. To draw a conclusion, model selection and competition is truly an important part in machine learning. We should select the correct model both accordingly and wisely.