Data-Mining Midterm

20152950 강민석

To solve the midterm questions, I used R to find the answer.

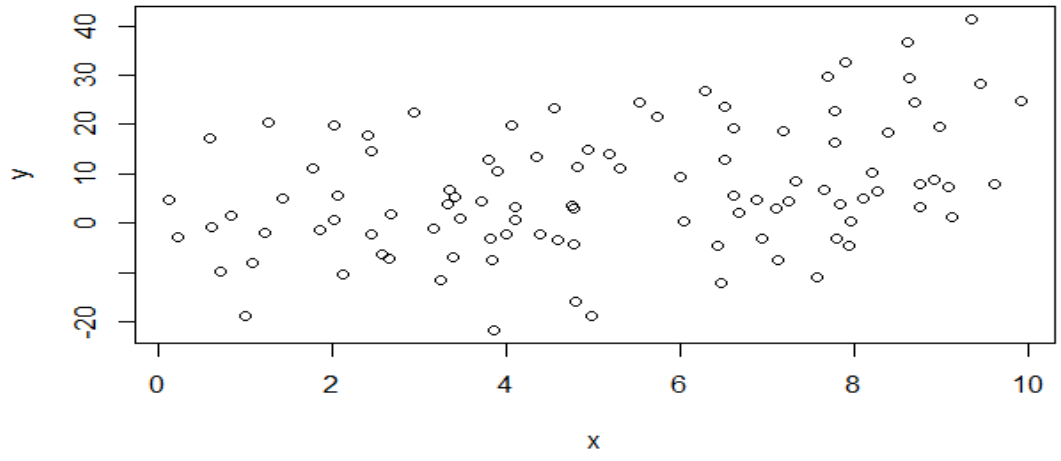
1. Use the attached "MID2020Sdata.txt" to answer the following questions.

Data

In MID2020Sdata.txt, there are two variables, 'x' and 'y'. Both variables are continuous variable. The number of samples for each variable is 100.

Before I calculate the correlation coefficient of x and y, I made scatter plot to see the relationship between two variables.

Figure 1. Scatter plot



It shows a weak positive correlation 0.3731081 between x and y.

a) Calculate the correlation coefficient of x and y with a bootstrap bias estimate and a bootstrap standard error.

Method

I used 'boot' function in 'boot' library. I used 1000 bootstrap replicates using a seed value 20152950(my CAU ID) to get the correlation coefficient of x and y with a bootstrap bias estimate and a bootstrap standard error.

Result

In my result, the correlation coefficient was 0.3731081 with bias -0.0002662021 and standard error 0.08371664.

The correlation coefficient using bootstrap method was exactly same as correlation coefficient with out using bootstrap method.

Further more, I calculated Spearman's correlation coefficient and Kendal's Tau using same bootstrap method. Spearman's correlation coefficient was 0.3549567 with bias -0.001483646 and standard error 0.0867011. Kendal's Tau was 0.2378499 with bias -0.000730947 and standard error 0.06077889.

Method	Value	Bias	Standard error
Pearson correlation	0.3731081	-0.0002662021	0.08371664.
Spearman correlation	0.3549567	-0.001483646	0.0867011
Kendal's Tau	0.2378499	-0.000730947	0.06077889

B) Construct 95% bootstrap confidence intervals.

Method

In 'boot' library, boot.ci function returns bootstrap confidence intervals. I constructed 95% confidence intervals using boot.ci function. For Pearson correlation coefficient, I made 4 types of confidence Interval, normal, basic, perc and bca. Normal confidence interval assumes that the parameter follows normal distribution. Basic confidence interval is confidence interval that uses the difference between the estimator from all sample and the mean of estimators made in each bootstrap samples .Perc is percentile confidence interval. Since I made 1000 bootstrap replicants, the lower bound of percentile C.I is 25^{th} smallest value of all sorted correlation coefficients and upper bound is 25^{th} largest value of all sorted correlation coefficients. Bca is Bias-corrected and accelerated confidence interval, which corrects bias and skewness of the distribution to give more accurate result.

The following table shows the result

Method	Lower bound	Upper bound
Normal C.I	0.2117	0.5399
Basic(residual) C.I	0.2205	0.5410
Percentile C.I	0.2052	0.5257
Bca C.I	0.2048	0.5254

In this example. Normal C.I and Basic C.I looks similar and Percentile C.I and Bca C.I looks similar.

2. Use "Hitters" data in ISLR package to answer the following questions. Consider a binary response whether a player's salary is greater than or equal to 750.

Data

In 'Hitter' dataset, there are 20 variable with 322 observations. The 'Salary' variable is the variable that we are going to predict and other variables are predictors. 'League' and 'NewLeague' predictor are binary data and 'Division' predictor is categorical data. Since some of variable contains NA value, I drop the observation that has NA value. After I removed NA values, there are 263 observations remain.

To make 'Salary' a binary data, I made a dummy variable. If the salary is greater or equal to 750, the dummy value is 1 and if it's not the value is 0.

Before making 10 fold CV models, I gave seed 20152950.

a) Use all predictors and construct a naive Bayes classifier. Report a 10-fold CV classification error.

I used 'caret' package to construct 10-fold naïve Bayes classifier. The accuracy without any kernel was 0.8099715. So the classification error is 0.1900285(1-accuracy).

	Real: under 750	Real: greater or equal 750
Prediction: under 750	170	10
Prediction: greater or equal 750	23	60

(b) Use all predictors and construct a logit model classifier. Report a 10-fold CV classification error.

I used same 'caret' package to construct 10-fold logit model. The accuracy was 0.8022792. The classification error is 1 minus accuracy, so it is 0.1977208

	Real: under 750	Real: greater or equal 750
Prediction: under 750	177	30
Prediction: greater or equal 750	16	40

So in this case, naïve Bayes classifier seems to be better than logit classifier since 10 fold CV classification error is smaller.

3. Suppose a categorical response Y can take on 1, 2, or 3. We have two categorical predictors X1 and X2. X1 has two levels, denoted by "A" and "B". X2 has three levels, denoted by "a", "b", and "c". A training data set consists of 50 observations as follow:

$Y = 1$

X_1	X_2 = a	b	c
A	4	1	0
B	0	1	4

$Y = 2$

X_1	X_2 = a	b	c
A	2	1	7
B	1	8	1

$Y = 3$

X_1	X_2 = a	b	c
A	2	4	2
B	9	1	2

Data

In training data, all 3 variables are categorical. Response Y can take on 1, 2, or 3. X1 has two levels, denoted by "A" and "B". X2 has three levels, denoted by "a", "b", and "c". The number of samples is 50. I made same data in text file to use it in R.

a) Construct Bayes classifier table for Y hat.

Method

I'm going to make LDA and QDA classification.

Method-1) LDA

If X1 and X2 has same covariance matrix, we can use LDA. I use lda function in MASS package, and made confusion matrix to get accuracy.

Confustion Matrix		Real Class of Y		
		1	2	3
Y hat (predicted class)	1	0	0	0
	2	6	17	9
	3	4	3	11

The accuracy is only 0.56 and none of observations are predicted as class 1. The performance is

very poor. The difference of covariance matrix between X1 and X2 might be the reason. Also when we see the data there are only 10 observations with Y is 1. So to solve this problem, we need to collect more data.

Method-2) QDA

If we cannot assume same covariance matrix for X1 and X2, we should use QDA. I use qda function in MASS package and made confusion matrix.

Confustion Matrix		Real Class of Y		
		1	2	3
Y hat (predicted class)	1	8	3	4
	2	1	15	3
	3	1	2	13

The accuracy is 0.72 and 95% C.I is (0.5751,0.8377). Performance in QDA is much better than performance of LDA.

b) Calculate the LOOCV classification error rate for Y hat.

Method

Method-1) LDA

Even the LDA performs poor, I'm going to calculate the LOOCV classification error rate. The error rate is calculated by number of misclassification divided by the number of observation. So it is 22/50=0.44

Method-2) QDA

With same logic, the error rate is calculated by number of misclassification divided by the number of observation. It is 14/50=0.28. It is much lower than error rate in LDA so again, we can say that QDA is much better than LDA in this case.

Appendix-Code

```
data=read.table('C:/Users/Administrator/Desktop/datamining/midterm/MID2020Sdata.txt',header = TRUE)

attach(data)

plot(data) #plot



cor(y,x) # correlation


# a) Calculate the correlation coefficient of x and y with a bootstrap bias estimate and a bootstrap standard error.

library(boot)

set.seed(20152950)


# make ftn to find correlation coefficient


bootscorr <- function(data, indices, cor.type){
   dt<-data[indices,]
   c(
      cor(dt$y, dt$x, method=cor.type)


   )
}
#The number of bootstrap replicates is 1000

myBootstrap_p <- boot(data, bootscorr, R=1000, cor.type='p')

myBootstrap_s <- boot(data, bootscorr, R=1000, cor.type='s')

myBootstrap_k <- boot(data, bootscorr, R=1000, cor.type='k')

myBootstrap_p

myBootstrap_s

myBootstrap_k
```

```
#B) Construct 95% bootstrap confidence intervals.
boot.ci(myBootstrap_p,conf=0.95,type=(c('norm','basic','perc','bca')))

boot.ci(myBootstrap_s,conf=0.95,type=(c('norm','basic','perc','bca')))
boot.ci(myBootstrap_k,conf=0.95,type=(c('norm','basic','perc','bca')))

##############################################################################
```

#2) Use “Hitters” data in ISLR package to answer the following questions. Consider a binary response whether a player’s salary is greater than or equal to 750.

```
library(klaR)
library(caret)
library(e1071)
library(ISLR)
dim(Hitters)
Hitter=na.omit(Hitters)

Hitter$y[Hitter[,19]<750]=0
Hitter$y[Hitter[,19]>=750]=1
Hitter$y=as.factor(Hitter$y)
Hitter <- subset(Hitter, select = -c(Salary))
dim(Hitter)

attach(Hitter)
```

#a)Use all predictors and construct a naive Bayes classifier. Report a 10-fold CV classification error.

```
set.seed(20152950)

naivebayes_model = train(
```

```
  form = y ~ .,
  data = Hitter,
  trControl = trainControl(method = "cv", number = 10),
  method = "nb"
)
naivebayes_model
naivebayes_predictions=predict(naivebayes_model, newdata = Hitter)
confusionMatrix(naivebayes_predictions, Hitter$y)$table

#b)Use all predictors and construct a logit model classifier. Report a 10-fold CV classification error.

set.seed(20152950)

glm_model = train(
  form = y ~ .,
  data = Hitter,
  trControl = trainControl(method = "cv", number = 10),
  method = "glm"
)
glm_model

glm_predictions=predict(glm_model, newdata = Hitter)
confusionMatrix(glm_predictions, Hitter$y)$table

################################################################################
```

#3)Suppose a categorical response Y can take on 1, 2, or 3. We have two categorical predictors X1 and X2. X1 has two levels, denoted by “A” and “B”. X2 has three levels, denoted by “a”, “b”, and “c”. A training data set consists of 50 observations as follow:

#a)Construct Bayes classifier table for Yhat.

```
data3=read.table('C:/Users/Administrator/Desktop/datamining/midterm/Midterm_question3.txt',header= TRUE)
```

```
data3$Y=as.factor(data3$Y)
str(data3)

library(MASS)

# LDA

lda_classifier=lda(Y~.,data=data3)
predl=predict(lda_classifier,data3)
predl
confusionMatrix(predl$class, data3$Y)$table

#QDA
qda_classifier=qda(Y~., data= data3)
predq=predict(qda_classifier,data3)
predq
confusionMatrix(predq$class, data3$Y)

#b)  Calculate the LOOCV classification error rate for Y hat

#LDA
lda_loocv=lda(Y~.,data= data3,CV=T)
lda_loocv
mean(lda_loocv$class != data3$Y)/50

#QDA
qda_loocv=qda(Y~.,data= data3,CV=T)
qda_loocv
mean(qda_loocv$class != data3$Y)
```