Zhangsheng Lai
1002554

Statistics: Homework 4

1. (a) Given p_i and q_i denote the probability of choosing box 1 and 2 respectively if the ball color chosen is i where $i = \{B, W, G\}$, denoting the three different colors. With the given information of the number of different color balls in the different boxes,

$$\begin{array}{rrr} \mathbb{P}(B|1) = 4/10 & \mathbb{P}(B|2) = 3/10 & \mathbb{P}(B|3) = 2/10 \\ \mathbb{P}(W|1) = 2/10 & \mathbb{P}(W|2) = 6/10 & \mathbb{P}(W|3) = 0 \\ \mathbb{P}(G|1) = 4/10 & \mathbb{P}(G|2) = 1/10 & \mathbb{P}(G|3) = 8/10 \end{array}$$

The risk function is represented by,

$$\begin{aligned} R(\theta, \hat{\theta}_{p,q}) &= \mathbb{E}_\theta(|\theta^2 - \hat{\theta}_{p,q}|^2) \\ &= \mathbb{E}_\theta\left(\sum_{i\in\{B,W,G\}} L(\theta, \hat{\theta}_{p,q}(i))\mathbb{P}(i|\theta)\right) \end{aligned}$$

where $L(\theta, \hat{\theta}_{p,q}(i)) = L(\theta, 1)p_i + L(\theta, 2)q_i + L(\theta, 3)(1 - p_i - q_i)$.Therefore,

$$\begin{aligned} R(1, \hat{\theta}_{p,q}) &= [q_B + 4(1 - p_B - q_B)]\frac{4}{10} + [q_W + 4(1 - p_W - q_W)]\frac{2}{10} + [q_G + 4(1 - p_G - q_G)]\frac{4}{10} \\ R(2, \hat{\theta}_{p,q}) &= [9p_B + 4q_B + 49(1 - p_B - q_B)]\frac{3}{10} + [9p_W + 4q_W + 49(1 - p_W - q_W)]\frac{6}{10} \\ &\quad + [9p_G + 4q_G + 49(1 - p_G - q_G)]\frac{1}{10} \end{aligned}$$

(b) Bayes risk is given by

$$r(f, \theta) = \int R(\theta, \hat{\theta}_{p,q})f(\theta)\,d\theta$$

but since our scenario is discrete, we instead have

$$\begin{aligned} r(f, \theta) &= \sum_{\theta=1,2} R(\theta, \hat{\theta}_{p,q})\mathbb{P}(\theta) \\ &= \lambda R(1, \hat{\theta}_{p,q}) + (1 - \lambda)R(2, \hat{\theta}_{p,q}) \end{aligned}$$

where $R(1, \hat{\theta}_{p,q})$ and $R(2, \hat{\theta}_{p,q})$ are the values are from (a).

(c) Given $\lambda = 1/2$, we have

$$r(f, \theta) = \frac{1}{2}\left(R(1, \hat{\theta}_{p,q}) + R(2, \hat{\theta}_{p,q})\right) = \frac{1}{2}(16.3 - 13.6p_B - 14.7q_B - 24.8p_W - 27.6q_W - 5.6p_G - 5.6q_G)$$

thus to the infimum of Bayes risk is when $q_B = q_W = q_G = 1$.

2.

```
library(survey)
library(dplyr)

# read csv file into dataframe car
car <- read.csv('carmpgdat.csv')

# (a) Fitting a mulitple linear regression model

# generate a linear model with normally distributed noise with the model MPG~VOL+HP+SP+WT
# covariates <-cbind('VOL','HP','SP','WT')
lm_car <- lm(MPG~VOL+HP+SP+WT, data = car)



# estimated regression function and residual sum of squares
print(lm_car$coefficients)

# MPG = 192.43775332 - 0.01564501 * VOL + 0.39221231 * HP - 1.29481848 * SP - 1.85980373 * WT
RSS = sum(lm_car$residuals^2)
print(RSS)

# (b) Mallows's Cp
# Since the AIC criterion is equivalent ot Mallows's Cp,

# (i) Forward
base <- lm(MPG~WT, data = car)
step(base, scope = list(upper = lm_car, lower=~1), direction = 'forward', trace = TRUE)

# Start: AIC=240.45
# MPG ~ WT
#
# Df Sum of Sq  RSS   AIC
# + SP   1    82.981 1383.0 237.68
# + HP   1    35.380 1430.6 240.45
# <none>            1466.0 240.45
# + VOL  1     3.883 1462.1 242.24
#
# Step: AIC=237.68
# MPG ~ WT + SP
#
# Df Sum of Sq  RSS   AIC
# + HP   1    349.37 1033.7 215.80
# + VOL  1     45.97 1337.0 236.90
# <none>            1383.0 237.68
#
# Step: AIC=215.8
# MPG ~ WT + SP + HP
#
# Df Sum of Sq  RSS  AIC
# <none>            1033.7 215.8
# + VOL  1    6.2685 1027.4 217.3
#
# Call:
#   lm(formula = MPG ~ WT + SP + HP, data = car)
#
# Coefficients:
#   (Intercept)         WT          SP          HP
# 194.1296    -1.9221     -1.3200      0.4052

# (ii) Backward
step(lm_car,direction = 'backward', trace = TRUE)

# Start: AIC=217.3
# MPG ~ VOL + HP + SP + WT
#
```

```
# Df Sum of Sq  RSS   AIC
# - VOL  1      6.27 1033.7 215.80
# <none>             1027.4 217.30
# - HP   1    309.67 1337.0 236.90
# - SP   1    373.36 1400.7 240.72
# - WT   1   1013.76 2041.2 271.59
#
# Step: AIC=215.8
# MPG ~ HP + SP + WT
#
# Df Sum of Sq  RSS   AIC
# <none>             1033.7 215.80
# - HP   1    349.37 1383.0 237.68
# - SP   1    396.97 1430.6 240.45
# - WT   1   1322.87 2356.5 281.37
#
# Call:
#   lm(formula = MPG ~ HP + SP + WT, data = car)
#
# Coefficients:
#   (Intercept)         HP          SP          WT
# 194.1296     0.4052     -1.3200     -1.9221
#
# For the forward stepwise approach, the base model is important
# as it will change the outcome. For example for this, if we were
# to start with VOL instead of any of the other covariates, we
# will end up with also the VOL covariate. By not starting with with
# the VOL covariate we will end up with a model without VOL which
# corresponds to the backward stepwise approach and also the
# Zheng-Loh model selection. As for the backward stepwise approach
# we do not have such a problem as we start the model with all the
# covariates and reduce it down by computing the AIC of the model
# with different covariate missing then remove the covariate that
# gives the smallest AIC when removed.



# (c) Zheng-Loh Model Selection
# Wald test for the covariates

regTermTest(lm_car, 'VOL', null=NULL,df=Inf, method = "Wald")
regTermTest(lm_car, 'HP', null=NULL,df=Inf, method = "Wald")
regTermTest(lm_car, 'SP', null=NULL,df=Inf, method = "Wald")
regTermTest(lm_car, 'WT', null=NULL,df=Inf, method = "Wald")

# Wald test for VOL
# in lm(formula = MPG ~ VOL + HP + SP + WT, data = car)
# Chisq = 0.4698075 on 1  df: p= 0.49308
#
# Wald test for HP
# in lm(formula = MPG ~ VOL + HP + SP + WT, data = car)
# Chisq = 23.20929 on 1  df: p= 1.4529e-06
#
# Wald test for SP
# in lm(formula = MPG ~ VOL + HP + SP + WT, data = car)
# Chisq = 27.98266 on 1  df: p= 1.2241e-07
#
# Wald test for WT
# in lm(formula = MPG ~ VOL + HP + SP + WT, data = car)
# Chisq = 75.97941 on 1  df: p= < 2.22e-16
#
# Arranging in descending order we have:
#   WT > SP > HP > VOL
# We can do this as the Chisq test statistic is just the square
# of the test statistic of the Wald test.
#
```

```
# Let lm_j be the linear model with the jth largest Wald test statistic

n <- nrow(car)

lm_1 <- lm(MPG ~ WT, data = car)
jhat_1 = sum(lm_1$residuals^2) + (1 * RSS/(n-4) * log(n))
print (jhat_1)
# jhat_1 = 1524.045

lm_2 <- lm(MPG ~ WT + SP, data = car)
jhat_2 = sum(lm_2$residuals^2) + (2 * RSS/(n-4) * log(n))
print (jhat_2)
# jhat_2 = 1499.108

lm_3 <- lm(MPG ~ WT + SP + HP, data = car)
jhat_3 = sum(lm_3$residuals^2) + (3 * RSS/(n-4) * log(n))
print (jhat_3)
# jhat_3 = 1207.78

jhat_4 = RSS + (4 * RSS/(n-4) * log(n))
print (jhat_4)
# jhat_4 = 1259.555

# Thus the Zheng-Loh model selection method will select WT, SP and HP
# as the covariates for predicting the MPG. Comparing it to (b), we see
# that the Zheng-Loh model selection method gives similar outcome to using
# Mallows's Cp model forward and backward stepwise to selecting a model.
```

3.

```
library(dplyr)
library(magrittr)


# Reading data with separator tab
raw_riasec <- read.csv('RIASEC.csv',sep = '\t')

# (a)CLEANING UP
# List of realistic traits
realistic_trait <- c('R1','R2','R3','R4','R5','R6','R7','R8')

# Extracting out the realistic traits
raw_realistic <- raw_riasec[,realistic_trait]

# Removing rows with -1 from the dataframe
realistic <- raw_realistic %>%
  filter(R1* R2 * R3 * R4 * R5 * R6 * R7 * R8 > 0)


# (b)MODEL SELECTION

# Computing the score for the R trait
realistic <- realistic %>%
  rowwise() %>%
  mutate(Rscore = mean(c(R1, R2, R3, R4, R5, R6, R7, R8)))

# Generating the training and validation set
tr_realistic <- realistic[1:6500,]
val_realistic <- realistic[-(1:6500),]

# Building the linear model
lm_riasec <- lm(Rscore~R1, data = tr_realistic)
avg_RSS_tr = mean(lm_riasec$residuals^2)

# estimated regression function and residual sum of squares
print(lm$coefficients)
# R1 = 1.0011862 + 0.4282061 * R1


# (c)VALIDATION
reg.fn <- function(x) 1.0011862 + 0.4282061 * x

val_realistic %<>%
  mutate(pred_Rscore = reg.fn(R1),
         residuals = reg.fn(R1) - Rscore )

avg_RSS_val = mean(val_realistic$residuals^2)

print (avg_RSS_tr) # 0.4540176
print (avg_RSS_val) # 0.5376852

# The residual sum of squares for the validation set using the regression function
# is larger than the residual sum of square for the training set but are of the
# same order. Thus the model generalizes well.
```

4. (a)

```
# (a) Using Monte Carlo method with N samples

N <- 100000
I <- function(u){X = rgamma(1.5,1/2.3,n = N); return(sum(1<X & X<2)/N)}
I_hat <- I

# I_hat = 0.20146

# Estimated standard error by resampling Monte Carlo 10000 times.
se_mc <- sqrt(var(sapply(1:N,I)))

# se_mc =  0.001272072

# Estimated standard error using 10000 bootstrap samples.
B <- 10000
base_sample <- rgamma(1.5,1/2.3,n = N)
I_bootstrap <- function(u){X = sample(base_sample, N, replace = TRUE);return(sum(1<X & X<2)/N)}
se_bootstrap <- sqrt(var(sapply(1:B,I_bootstrap)))

# se_bootstrap = 0.001273952
```

(b) We see that we can rewrite I as the following,

$$I = \int_1^2 f(x|1.5, 2.3)\, dx = \int_0^\infty h(x) f(x|1.5, 2.3)\, dx$$

where $h(x) = \mathbb{1}(1 \leq x \leq 2)$, an indicator function that is 1 when $1 \leq x \leq 2$ and 0 otherwise. Thus we can use the following estimator $\hat{I}$ for I where $X_i \sim \text{Gamma}(1.5, 2.3)$,

$$\hat{I} = \frac{1}{N} \sum_{i=1}^{N} h(X_i)$$

Hence by viewing $h(X) \sim \text{Bernoulli}(p)$, where $p = \mathbb{P}(1 \leq X \leq 2)$ for $X \sim \text{Gamma}(1.5, 2.3)$ the standard error is given by

$$Var(\hat{I}) = \frac{p(1-p)}{N}$$

where we sourced for the value of p from the website thus getting

$$\mathbb{P}(1 \leq X \leq 2) = \int_0^2 f(x|1.5, 2.3)\, dx - \int_0^1 f(x|1.5, 2.3)\, dx = 0.37173 - 0.16723$$

Evaluating the value of the standard error we get $\mathsf{se} = 0.00127545972$