S2208 MATH8050 Data Analysis - Section 001: Homework 5 Due on 10/13/22

Adithya Ravi, C09059838

2022-10-12

Solutions

Question1

1a

```
gen_lm <- glm(formula = Direction ~ Lag1+Lag2+Lag3+Lag4+Lag5+Volume,
data=Smarket,family = binomial(link=logit))

summary(gen_lm)
```

```
##
## Call:
## glm(formula = Direction ~ Lag1 + Lag2 + Lag3 + Lag4 + Lag5 +
##     Volume, family = binomial(link = logit), data = Smarket)
##
## Deviance Residuals:
##    Min      1Q  Median      3Q     Max
## -1.446  -1.203   1.065   1.145   1.326
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -0.126000   0.240736  -0.523    0.601
## Lag1        -0.073074   0.050167  -1.457    0.145
## Lag2        -0.042301   0.050086  -0.845    0.398
## Lag3         0.011085   0.049939   0.222    0.824
## Lag4         0.009359   0.049974   0.187    0.851
## Lag5         0.010313   0.049511   0.208    0.835
## Volume       0.135441   0.158360   0.855    0.392
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1731.2  on 1249  degrees of freedom
## Residual deviance: 1727.6  on 1243  degrees of freedom
## AIC: 1741.6
##
## Number of Fisher Scoring iterations: 3
```

1b

```r
logres = function(X,y,threshold = 1e-10, max_iter = 100) {
  func1 = function(X,beta){
    beta = as.vector(beta)
    return(exp(X%*%beta) / (1+ exp(X%*%beta)))
  }

  beta = rep(0,ncol(X))
  diff = 10000
  count = 0
  while(diff > threshold){
    p = as.vector(func1(X,beta))
    W = diag(p*(1-p))
    beta_change = solve(t(X)%*%W%*%X) %*% t(X)%*%(y - p)
    beta = beta + beta_change
    diff = sum(beta_change^2)
    count = count + 1
    if(count > max_iter) {
      stop("This isn't converging")
    }
  }
  return(beta)
}

dataframe1<-Smarket
dataframe1<-transform(dataframe1, new_d=ifelse(Direction=="Up",1,0))
X.temp1<-Smarket$Lag1
X.temp2<-Smarket$Lag2
X.temp3<-Smarket$Lag3
X.temp4<-Smarket$Lag4
X.temp5<-Smarket$Lag5
X.Volume<-Smarket$Volume

y <- as.numeric(dataframe1$new_d)
n <- nrow(Smarket)
X<-cbind(c(rep(1,n)),X.temp1,X.temp2,X.temp3,X.temp4,X.temp5,X.Volume)
colnames(X) <- c("Intercept","Lag1", "Lag2", "Lag3","Lag4","Lag5","Volume")

logres(X,y)
```

```
##                   [,1]
## Intercept -0.126000259
## Lag1      -0.073073747
## Lag2      -0.042301345
## Lag3       0.011085108
## Lag4       0.009358938
## Lag5       0.010313069
## Volume     0.135440661
```

1c

```
set.seed(12345)
smp_siz = floor(0.85*nrow(Smarket))
train_ind = sample(seq_len(nrow(Smarket)),size = smp_siz)

train =Smarket[train_ind,]
test=Smarket[-train_ind,]

fit.sm_train= glm(formula = Direction ~ Lag1+Lag2+Lag3+Lag4+Lag5+Volume,
              data=train,family = binomial)
beta_values<-as.data.frame(coef(fit.sm_train))
betas<-c(rep(0,7))
c=1
for (i in beta_values$`coef(fit.sm_train)`) {
  betas[c]=i
  c=c+1
}
p= function(x1,x2,x3,x4,x5,x6,betas,n) {
  c=1
  val<-c(rep(n),0)
  ans<-c(rep(n),0)
  i=1
  while ( i <=n){
    X2<-c(x1[c],x2[c],x3[c],x4[c],x5[c],x6[c])
    val[c]<-betas[1]+sum(X2*betas[2:7])
    ans[c]<-1/(1+exp(-val[c]))
    c=c+1
    i=i+1
  }
  return (ans)
}

x1<-test$Lag1
x2<-test$Lag2
x3<-test$Lag3
x4<-test$Lag4
x5<-test$Lag5
x6<-test$Volume

n=nrow(test)
values_p1<-p(x1,x2,x3,x4,x5,x6,betas,n)

test_direction<-test$Direction
train_direction<-train$Direction

my.predict= function(test_direction,values_p){
  test_data_direction<-c(rep(length(test_direction),0))
  c=1
  for (i in values_p){
    if (i>0.5){
      test_data_direction[c]<-"Up"
    }
```

```
    else{
      test_data_direction[c]<-"Down"
    }
    c=c+1
  }
  c=1
  true_down=0
  true_up=0
  true_down_butup=0
  true_up_butdown=0
  for(i in test_data_direction) {
    if(test_direction[c] == "Down" && test_data_direction[c] =="Down") {
      true_down=true_down+1
    }
  else if(test_direction[c] == "Up" && test_data_direction[c] == "Up"){
    true_up=true_up+1
  }
  else if(test_direction[c] == "Up" && test_data_direction[c] == "Down"){
    true_up_butdown=true_up_butdown+1
  }
  else if(test_direction[c] == "Down" && test_data_direction[c] == "Up"){
    true_down_butup=true_down_butup+1
  }
  c=c+1
  }
  final_val<-data.frame(true_down=c(true_down),true_up=c(true_up),
                         true_down_butup=c(true_down_butup),
                         true_up_butdown=c(true_up_butdown))

  return(final_val)
}

my.predict(test_direction,values_p1)
```

```
##   true_down true_up true_down_butup true_up_butdown
## 1        16      85              75              12
```

Therefore, the false positive and false negative are 75 and 12 respectively, for the test data.

```
x1<-train$Lag1
x2<-train$Lag2
x3<-train$Lag3
x4<-train$Lag4
x5<-train$Lag5
x6<-train$Volume
n=nrow(train)
values_p<-p(x1,x2,x3,x4,x5,x6,betas,n)
my.predict(train_direction,values_p)
```

```
##   true_down true_up true_down_butup true_up_butdown
## 1        81     476             430              75
```

Therefore we can see that false positive and false negative are 75 and 430 respectively. We can verify that using predict function in r

```
glm.probs <- predict(fit.sm_train,newdata = train, type="response")
glm.pred <- rep("Down", nrow(train))
glm.pred[glm.probs > 0.5] <- "Up"

glm.pred <- ifelse(glm.probs > 0.5,"Up","Down")
table(glm.pred, train$Direction)
```

```
##
## glm.pred Down  Up
##     Down   81  75
##     Up    430 476
```

```
glm.probs <- predict(fit.sm_train,newdata = test, type="response")
glm.pred <- rep("Down", nrow(test))
glm.pred[glm.probs > 0.5] <- "Up"

glm.pred <- ifelse(glm.probs > 0.5,"Up","Down")
table(glm.pred, test$Direction)
```

```
##
## glm.pred Down Up
##     Down   16 12
##     Up     75 85
```

Question2

2a

The given equation can be written as

$$\int_{-\infty}^{\infty} exp(-x^4) = 2\int_{0}^{\infty} exp(-x^4)$$

Now replace the $-x^4 = t$ thereby, $-t^{1/4} = x$

$$dx = \frac{1}{4}t^{-3/4}dt$$

Now writing the equation w.r.t t, we get:

$$2\int_{0}^{\infty} exp(-x^4) = 2/4\int_{0}^{\infty} t^{\frac{1}{4}-1}exp(-t)dt$$

Which is in the gamma function form,

$$\Gamma(z) = \int_{0}^{\infty} exp(-t)t^{z-1}dt$$

where

$$z = 1/4\Gamma(1/4) = 3.6356$$

thereby,

$$0.5\int_{0}^{\infty} t^{\frac{1}{4}-1}exp(-t)dt = \Gamma(1/4)0.5 = 1.8128$$

2b

```r
h = function(x){
  return(2*exp(-(x^4))/dgamma(x,shape = 1.2,rate=0.95))
}

val<- function(x) 2*exp(-(x^4))
integrate(val, 0, Inf)
```

```
## 1.812805 with absolute error < 6.5e-05
```

```r
I = function(n, x){
  hx = h(x)
  hbar = mean(hx)
  v = sum((hx-hbar)^2) / n^2
  L = hbar - 2*sqrt(v)
  U = hbar + 2*sqrt(v)
  return(list(hbar=hbar, v=v, L=L, U=U))
}

n=10^4
set.seed(29)
x = rgamma(10^4,shape=1.2,rate=0.95)

ans<-I(n, x)$hbar
c(mean(h(x)),var(h(x)))
```

```
## [1] 1.804332 2.398438
```

As the above integration limits follow the limits of Gamma Distribution so we have taken the Gamma Distribution with shape and rate as 1.2 and 0.95.

2c

```r
set.seed(29)
w<-function(x)
dgamma(x,shape=1.2,rate = 0.95)/dexp(x,2.2)
X<-rexp(10^5,2.2)
Y<-w(X)*h(X)
c(mean(Y),var(Y))
```

```
## [1] 1.8128060 0.8133906
```

With the help of sampling we can see the variance of the value is reduced and also the precision of the value is increased at rate = 2.2.