MAT 443: LOGISTIC REGRESSION

Name: Eric Agyemang

**QUESTION 1**

Logistic regression

Logistic regression is a generalized linear model, with a binominal distribution and logit link function. The outcome Y is either 1 or 0. What we are interested in is the expected values of Y, E(Y). In this case, they can also be thought as probability of getting 1, p. However, because pis bounded between 0 and 1, it’s hard to implement the method similar to what we did for linear regression.

#Calculate the first derivative of likelihood function given output (y), input (x) and pi (estimated probability)

calculateder <- function(y,x,pi) {

derv <- y\*x - pi\*x

derv\_sum <- sum(derv)

return(derv\_sum)

}

#Calculate the value of pi (predictions on each observation) given x\_new(input) and estimated betas

findpi <- function(x\_new,beta){

pi <- 1:nrow(x\_new)

expon <- 1:nrow(x\_new)

for (i in 1:nrow(x\_new)){

expon[i] <- 0

for (j in 1:ncol(x\_new)){

expo <- x\_new[i,j] \* beta[j]

expon[i] <- expo + expon[i]}

pi[i] <- exp(expon[i])/(1+exp(expon[i]))

}

return(pi)

}

#Calculate the matrix W with all diagonal values as pi

findW <- function(pi){

W <- matrix(0,length(pi),length(pi))

for (i in 1:length(pi)){

W[i,i] <- pi[i]\*(1-pi[i])

}

return(W)

}

# Lets now make the logistic function given list of required inputs

logistic <- function(x,y,vars,obs,learningrate,dif) {

beta <- rep(0, (vars+1))

bias <- rep(1, obs)

x\_new <- cbind(bias,x)

derivative <- 1:(vars+1)

diff <- 10000

while(diff > dif) {

pi <- findpi(x\_new,beta)

pi <- as.vector(pi)

W <- findW(pi)

derivative <- (solve(t(x\_new)%\*%W%\*%as.matrix(x\_new))) %\*% (t(x\_new)%\*%(y - pi))

beta = beta + derivative

diff <- sum(derivative^2)

ll <- calculatell(y,pi)

print(ll)

}

return(beta)

}

# Time to test our algorithm with the values we mentioned at the start of the article

x <- 1:10

y <- c(rep(0, 4),1,0,1,0,1,1)

a <- logistic(x,y,1,10,0.01,0.000000001)

calculatell(y,findpi(x\_new,a))

#Log Likelihood = 0.01343191

data <- cbind(x,y)

data <- as.data.frame(data)

mylogit <- glm(y ~ x, data = data, family = "binomial")

mylogit

preds <- predict(mylogit, newdata = data,type ="response")