Homework 2

Part 1

The error rate was 0.405.

The zeros affect the predictions by setting the numerator in the native bayes equation to 0. Which is to say, that such an occurrence will cause our native bayes predictor to predict 0 irrespective of whether or not other strong predictors are present. This typically happens when our predictor can take on a large range of values (as the range of our predictor grows, the larger the joint distribution of the predictor and the outcome and hence, for a fixed set of observations, the more likely there will be a combination of outcome and predictor that has yet to be observed). This means that the zeros will cause our native bayes model to yield false-negatives.

2.

Error rates were still 0.405 for the first 1000 data points for the altered training function. There were no improvements (code found in the appendix attached)

3.

a) False

b) True

4

False negative rate

592/(592+520) = 0.532

False positve Rate

345/(345+271) = 0.560

Total error rate:

345+592/1728 = 0.542

5.

false positive

= p0 \* P(X1 = 0|Y = 0) + p1\* P(X1 = 1|Y = 0) = p0\*0.4+p1\*0.6

false negative

= (1-p0) \* P(X1 = 0|Y = 1) + (1-p1)\* P(X1 = 1|Y = 1) = (1-p0)\*0.7+(1-p1)\*0.3

Appendix A

Altered nb.train

nb.train <- function(D = NULL){

#======================================================================

# Function: nb.train

# from nbc-r, available on Google Code

# GPL-2 license

# last updated:2005.11.10

# by TSYo

#----------------------------------------------------------------------

# Description:

# training a naive Bayes model from a given data matrix.

#----------------------------------------------------------------------

# Input arguments:

# D

# A data matrix containing the training data. Each row of D

# contains one training example, with the class at the last

# column.

#----------------------------------------------------------------------

# Return objects:

# class.dist

# A vector contains the relative frequencies of the class labels.

# attr.dist

# A list of matrices which contain the conditional distribution

# of the corresponding attribute given the class label.

#----------------------------------------------------------------------

# Examples:

# nbWSBC <- nb.train(wsbc.dat) # Train a naive Bayes classifier

# nbWSBC$class.dist # Show the prob. of each class label

#======================================================================

# Checking arguments

if(missing(D))

stop("Please specify the matrix containing training data.")

#

#----------------------------------------------------------------------

# Probability calculation

dimD = dim(D)

nRow = dimD[1] # size of training sample

nAttr = dimD[2] -1 # number of attributes

cRow = dimD[2] # row number of the class label

adist = NULL

baysianIncrease = 1000

# Calculate Pr(C)

cdist = table(D[,cRow])/nRow

# Calculate Pr(A\_j|C)=Pr(A\_j,C)/Pr(C)

for (j in 1:nAttr){

if( min(table(D[,cRow],D[,j])) == 0){

print(table(D[,cRow],D[,j]))

tmpAdist = table(D[,cRow],D[,j]) + baysianIncrease #adding 1

print(tmpAdist)

#nRowNew = nRow+baysianIncrease\*dim(tmpAdist)[1]\*dim(tmpAdist)[2]

}

else{

tmpAdist = table(D[,cRow],D[,j])

#tmpAdist = table(D[,cRow],D[,j])/nRow # Pr(A\_i,C)

}

tmpAdist = t(t(tmpAdist)/rowSums(tmpAdist))

adist = c(adist,list(tmpAdist))

}

# End of probability calculation

#----------------------------------------------------------------------

# Return results

return(list(class.dist=cdist,attr.dist=adist))

}