exam_1_classification

2023-03-05

BUAN 6357 2023 Spring (Johnston)

Exam 1: section 2 - classification

A run log of this code is provided as a PDF file. You may run this code, explore its actions, and add comments as you wish.

Based on class discussions and homework assignments, You should extend this code as needed in preparation for answering questions about the process presented here.

```
require(data.table)
```

```
## Loading required package: data.table
```

```
require(partykit)
```

```
## Loading required package: partykit
```

```
## Loading required package: grid
```

```
## Loading required package: libcoin
```

```
## Loading required package: mvtnorm
```

Adding flags for the code

```
classif    <- c(1, 2, 3)
byCols     <- 2
byRows     <- 1
demo <- T

if(demo) {setwd("/Users/harikrishnadev/Library/Mobile Documents/com~apple~CloudDocs/Scho
ol Work/Sem 2/BUAN 6357/BUAN6357/exam_materials/exam_1")}
```

Reading training data

```
in1        <- fread(file="classif.dat")
if(demo) {str(in1)}
```

```
## Classes 'data.table' and 'data.frame':   150 obs. of  5 variables:
##  $ V1 : num  5.2 5.1 6.7 5.4 5.7 4.8 4.4 6.7 4.8 5.9 ...
##  $ V2 : num  2.7 3.5 3.1 3.4 2.6 3.4 3.2 3 3.4 3.2 ...
##  $ V3 : num  3.9 1.4 5.6 1.7 3.5 1.9 1.3 5.2 1.6 4.8 ...
##  $ V4 : num  1.4 0.2 2.4 0.2 1 0.2 0.2 2.3 0.2 1.8 ...
##  $ grp: int  2 1 3 1 2 1 1 3 1 2 ...
##  - attr(*, ".internal.selfref")=<externalptr>
```

Creating a function to return the fitted values for the glm function (logistic regression)

```
fitLogit   <- function(df,i) {
              df$y            <- 0
              df$y[df$grp==i] <- 1
              t               <- glm(y~.-grp,family=binomial(),data=df)
              return(t$fitted.values)
              }
```

Getting the probabilty distribution

```
t         <- data.table(idx=1:3, i=1:3)
tLogit     <- t[,.(fitted=fitLogit(in1,i)), by=.(idx)]
```

```
## Warning: glm.fit: algorithm did not converge
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
mLogit    <- matrix(tLogit$fitted, ncol=length(classif),byrow=F)
idxLogit  <- apply(mLogit,byRows,which.max)
classLogit<- classif[idxLogit]
rMargin   <- mLogit[,1]+mLogit[,2]+mLogit[,3]
t1         <- apply(mLogit,byRows,max)
pLogit    <- t1/rMargin
brLogit   <- 1-pLogit
(tbl1      <- table(in1$grp,classLogit,dnn=c("grp","class")) )
```

```
##    class
## grp  1  2  3
##   1 50  0  0
##   2  0 48  2
##   3  0  1 49
```

Creating a probabilty function to calculate bayers risk

```
probabilty_function <- function(tbl) {
(d           <- diag(tbl) )

rSums        <- apply(tbl, byRows, sum) # classification outcomes
cSums        <- apply(tbl, byCols, sum) # actual underlying values

(rPC          <- d/rSums )                     # classification "pc" by digit
(cPC          <- d/cSums )                     # classification "pc" by actual underlyin
g

(rBR          <- 1-rPC )                       # Bayes Risk by digit
(cBR          <- 1-cPC )
pc.correct <- sum(diag(tbl))/sum(tbl)
results_BR <- list(assigned.bayes.risk=rBR,actual.bayes.risk = cBR,pc.correct = pc.corre
ct)
return(results_BR)
}                    # Bayes Risk by actual underlying
```

Getting Bayers risk and percent correct from tbl1

```
probabilty_function(tbl1)
```

```
## $assigned.bayes.risk
##    1    2    3
## 0.00 0.04 0.02
##
## $actual.bayes.risk
##          1          2          3
## 0.00000000 0.02040816 0.03921569
##
## $pc.correct
## [1] 0.98
```

Same exercise for Tree classification

```
nmDat      <- in1
nmDat$fac  <- as.factor(nmDat$grp)
mn        <- ctree(fac~.-grp,data=nmDat)
mnTree     <- as.matrix(predict(mn,type="prob"))
attr(mnTree,"dimnames") <- NULL
idxTr      <- apply(mnTree,byRows,which.max)
classTree <- classif[idxTr]
pTree     <- apply(mnTree,byRows,max)
brTree    <- 1-pTree
(tbl2      <- table(in1$grp,classTree,dnn=c("grp","class")) )
```

```
##    class
## grp  1  2  3
##   1 50  0  0
##   2  0 49  1
##   3  0  5 45
```

Getting Bayers risk and percent correct from tbl2

```
probabilty_function(tbl2)
```

```
## $assigned.bayes.risk
##    1    2    3
## 0.00 0.02 0.10
##
## $actual.bayes.risk
##          1          2          3
## 0.00000000 0.09259259 0.02173913
##
## $pc.correct
## [1] 0.96
```