Naive Bayes Classifier

Zhoutao Pei

Read in data

```
library(caret)
mydata <- read.table(params$file, sep=",")
head(mydata)
```

```
##   V1  V2 V3 V4  V5   V6    V7 V8 V9
## 1  6 148 72 35   0 33.6 0.627 50  1
## 2  1  85 66 29   0 26.6 0.351 31  0
## 3  8 183 64  0   0 23.3 0.672 32  1
## 4  1  89 66 23  94 28.1 0.167 21  0
## 5  0 137 40 35 168 43.1 2.288 33  1
## 6  5 116 74  0   0 25.6 0.201 30  0
```

```
X <- mydata[, 1:8]
y <- mydata[, 9]
partition <- createDataPartition(y, p=0.8, list=FALSE)
trainX <- X[partition,]
trainy <- y[partition]
testX <- X[-partition,]
testy <- y[-partition]
```

Function to train a Naiva Bayes classifier, assume each attribute has a gaussian distribution.

```
naiveBayesTrain <- function(trainX, trainy) {
  trainX.pos <- trainX[trainy == 1,]
  trainX.neg <- trainX[trainy == 0,]
  mean.pos <- colMeans(trainX.pos, na.rm = T)
  mean.neg <- colMeans(trainX.neg, na.rm = T)
  sd.pos <- apply(trainX.pos, 2, sd, na.rm = T)
  sd.neg <- apply(trainX.neg, 2, sd, na.rm = T)
  p.pos <- sum(trainy == 1) / length(trainy)
  return(list("meanPos" = mean.pos,
              "meanNeg" = mean.neg,
              "sdPos" = sd.pos,
              "sdNeg" = sd.neg,
              "pPos" = p.pos))
}

NBmodel <- naiveBayesTrain(trainX, trainy)
```

Function to make predictions on test data.

```
naiveBayesPred <- function(NBmodel, testX) {
  testX.centered.pos <- t(t(testX) - NBmodel$meanPos)
  testX.scaled.pos <- t(t(testX.centered.pos) / NBmodel$sdPos)
  logLik.pos <- -(1/2)*rowSums(testX.scaled.pos^2) - sum(log(NBmodel$sdPos)) + log(NBmodel$pPos)
```

```
  testX.centered.neg <- t(t(testX) - NBmodel$meanNeg)
  testX.scaled.neg <- t(t(testX.centered.neg) / NBmodel$sdNeg)
  logLik.neg <- -(1/2)*rowSums(testX.scaled.neg^2) - sum(log(NBmodel$sdNeg)) + log(1-NBmodel$pPos)
  pred <- as.numeric(logLik.pos > logLik.neg)
  return(pred)
}
```

Evaluate model on test data.

```
pred <- naiveBayesPred(NBmodel, testX)
error <- sum(pred != testy) / length(testy)
print(paste("Test error rate is", error))
```

```
## [1] "Test error rate is 0.209150326797386"
```