project6_new

Yan Lin

November 6, 2017

```
##referring from the book Machine Learning for Hackers, chapter 3 : http://pdf.th7.cn/down/files/1312/ma

library(tm)
```

```
## Loading required package: NLP
```

```
library(ggplot2)
```

```
##
## Attaching package: 'ggplot2'
```

```
## The following object is masked from 'package:NLP':
##
##     annotate
```

```
library(stringi)

spam.path <- "C:/Users/Yan/Documents/spam/"
spam2.path <- "C:/Users/Yan/Documents/spam_2/"
easyham.path <- "C:/Users/Yan/Documents/easy_ham/"
hardham.path <- "C:/Users/Yan/Documents/hard_ham/"


## get the email massage
spam.docs <- dir(spam.path)

## build tdm function to transform to tdm
get.tdm <- function(doc.vec){
  doc.corpus <- Corpus(VectorSource(doc.vec))
  control <- list(stopwords=TRUE,removePunctuation=TRUE,removeNumbers=TRUE,stripWhitespace=TRUE)
  doc.dtm <- TermDocumentMatrix(doc.corpus,control)
  return(doc.dtm)
}

spam.tdm <- get.tdm(spam.docs)

## usd TDM to build training data for spam
spam.matrix <- as.matrix(spam.tdm)
spam.counts <- rowSums(spam.matrix)
spam.df <- data.frame(cbind(names(spam.counts),
      as.numeric(spam.counts)),stringAsFactors=FALSE)
names(spam.df) <- c("term","frequency")
spam.df$frequency <- as.numeric(spam.df$frequency)

spam.occurence <- sapply(1:nrow(spam.matrix),
      function(i){length(which(spam.matrix[i,]>0))/ncol(spam.matrix)})
spam.density <- spam.df$frequency/sum(spam.df$frequency)

## add density and occurence rate to the spam data frame
```

```r
spam.df <- transform(spam.df,density=spam.density,occurence=spam.occurence)

head(spam.df[with(spam.df,order(-occurence)),])
```

```
##              term frequency   NA.     density   occurence
## 1     ddeaaaecfcf         1 FALSE 0.001996008 0.001996008
## 2     dfbeedbbdea         1 FALSE 0.001996008 0.001996008
## 3 eebceacdbfdcbac         1 FALSE 0.001996008 0.001996008
## 4       eacdedbef         1 FALSE 0.001996008 0.001996008
## 5        addcebfd         1 FALSE 0.001996008 0.001996008
## 6     abddccdbafc         1 FALSE 0.001996008 0.001996008
```

```r
## build TDM for ham emails
## get the email massage
easyham.docs <- dir(easyham.path)

## build tdm
easyham.tdm <- get.tdm(easyham.docs)

## usd TDM to build training data for easyham
easyham.matrix <- as.matrix(easyham.tdm)
easyham.counts <- rowSums(easyham.matrix)
easyham.df <- data.frame(cbind(names(easyham.counts),
      as.numeric(easyham.counts)),stringAsFactors=FALSE)
names(easyham.df) <- c("term","frequency")
easyham.df$frequency <- as.numeric(easyham.df$frequency)

easyham.occurence <- sapply(1:nrow(easyham.matrix),
      function(i){length(which(easyham.matrix[i,]>0))/ncol(easyham.matrix)})
easyham.density <- easyham.df$frequency/sum(easyham.df$frequency)

## add density and occurence rate to the easyham data frame
easyham.df <- transform(easyham.df,density=easyham.density,occurence=easyham.occurence)

head(easyham.df[with(easyham.df,order(-occurence)),])
```

```
##                 term frequency   NA.      density    occurence
## 1400         eaefcdf         2 FALSE 0.0007840063 0.0007840063
## 1      eaedeeaceafae         1 FALSE 0.0003920031 0.0003920031
## 2    bcbcbfeeeefcbbf         1 FALSE 0.0003920031 0.0003920031
## 3    acfcadbbdaddcdd         1 FALSE 0.0003920031 0.0003920031
## 4       eddddecbedff         1 FALSE 0.0003920031 0.0003920031
## 5       cbecffddafab         1 FALSE 0.0003920031 0.0003920031
```

```r
## classifier function
classify.email <- function(path,training.df,prior=0.5,c=1e-06){
  docs <- dir(path)
  msg.tdm <- get.tdm(docs)
  msg.freq <- rowSums(as.matrix(msg.tdm))
  msg.match <- intersect (names(msg.freq),training.df$term)
  if(length(msg.match)<1){
    return(prior * c^(length(msg.freq)))
  }  else {
      match.probs <- training.df$occurence[match(msg.match,training,df$term)]
      return(prior*prod(match.probs)*c^(length(msg.freq)-length(msg.match)))
  }
}

## use classifier and spam and ham training model
hardham.docs <- dir(hardham.path)
hardham.docs <- hardham.docs[hardham.docs != "cmds"]
hardham.spamtest <- sapply(hardham.docs, function(p) classify.email(file.path(hardham.path,
    p), training.df = spam.df))
hardham.hamtest <- sapply(hardham.docs, function(p) classify.email(file.path(hardham.path,
    p), training.df = easyham.df))
hardham.res <- ifelse(hardham.spamtest > hardham.hamtest, FALSE, TRUE)
summary(hardham.res)
```

```
##    Mode    TRUE
## logical     250
```

```
## the test result doesn't make sense. I haven't found anyway to resolve it yet.
```