NB\_Co

yt2547

2017/4/11

setwd("C:/Users/yftang/Documents/GitHub/Spr2017-proj4-team-5/lib/")  
library(stringr)

data.lib="../data/nameset"  
data.files=list.files(path=data.lib, "\*.txt")  
  
#data.files  
  
## remove "\*.txt"  
query.list=substring(data.files,   
 1, nchar(data.files)-4)  
  
#query.list  
  
## add a space  
query.list=paste(substring(query.list, 1, 1),   
 " ",   
 substring(query.list,   
 2, nchar(query.list)),  
 sep=""  
 )  
  
# query.list

f.line.proc=function(lin, nam.query="."){  
  
 # remove unwanted characters  
 char\_notallowed <- "\\@#$%^&?" # characters to be removed  
 lin.str=str\_replace(lin, char\_notallowed, "")  
  
 # get author id  
 lin.str=strsplit(lin.str, "\_")[[1]]  
 author\_id=as.numeric(lin.str[1])  
   
 # get paper id  
 lin.str=lin.str[2]  
 paper\_id=strsplit(lin.str, " ")[[1]][1]  
 lin.str=substring(lin.str, nchar(paper\_id)+1, nchar(lin.str))  
 paper\_id=as.numeric(paper\_id)  
   
 # get coauthor list  
 lin.str=strsplit(lin.str, "<>")[[1]]  
 coauthor\_list=strsplit(lin.str[1], ";")[[1]]  
  
 #print(lin.str)  
 for(j in 1:length(coauthor\_list)){  
 if(nchar(coauthor\_list[j])>0){  
 nam = strsplit(coauthor\_list[j], " ")[[1]]  
 if(nchar(nam[1])>0){  
 first.ini=substring(nam[1], 1, 1)  
 }else{  
 first.ini=substring(nam[2], 1, 1)  
 }  
 }  
 last.name=nam[length(nam)]  
 nam.str = paste(first.ini, last.name)  
 coauthor\_list[j]=nam.str  
 }  
   
 match\_ind = charmatch(nam.query, coauthor\_list, nomatch=-1)  
   
 # print(nam.query)  
 # print(coauthor\_list)  
 # print(match\_ind)  
   
 if(match\_ind>0){  
   
 coauthor\_list=coauthor\_list[-match\_ind]  
 }  
   
 paper\_title=lin.str[2]  
 journal\_name=lin.str[3]  
   
 list(author\_id,   
 paper\_id,   
 coauthor\_list,   
 paper\_title,   
 journal\_name)  
}

data\_list=list(1:length(data.files))  
  
for(i in 1:length(data.files)){  
   
 ## Step 0 scan in one line at a time.  
   
 dat=as.list(readLines(paste(data.lib, data.files[i], sep="/")))  
 data\_list[[i]]=lapply(dat, f.line.proc, nam.query=query.list[i])  
   
}

get.info<-function(n){  
# Input: n-the nameset  
# n = 3  
num.paper<-length(data\_list[[n]])  
auther.id<-NULL  
coauther<-NULL  
for(i in 1:num.paper){  
 auther.id[i]<-data\_list[[n]][[i]][[1]]  
 coauther<-unique(c(coauther,data\_list[[n]][[i]][[3]]))  
}  
a.i<-factor(auther.id)  
uni.auther<-length(levels(a.i))  
levels(a.i)<-seq(uni.auther)  
auther.id<-a.i  
#df<-as.data.frame(a.i)  
df<-data.frame(auther.id)  
  
num.coauther<-length(coauther)  
get.coauther<-function(i){  
 coauthers<-rep(0,num.coauther)  
 for(j in 1:length(data\_list[[n]][[i]][[3]])){  
 coauthers[which(data\_list[[n]][[i]][[3]][j]==coauther)]<- 1  
}  
return(coauthers)   
}  
##########################################################  
#### build dataframe for auther coauther information######  
##########################################################  
id\_co<-NULL  
for(i in 1:num.paper){  
 id\_co<-rbind(id\_co,get.coauther(i))  
}  
colnames(id\_co)<-coauther  
df<-cbind(auther.id,id\_co)   
#df contains the auther coauther information in nameset 1: AGupta  
id.num.co<-apply(df[,-1],1,sum)  
df<-as.data.frame(cbind(id.num.co,df))  
freq.id<-as.data.frame(table(df$auther.id)/length(df$auther.id))   
# numbers of paper for each id  
colnames(freq.id)<-c("auther.id","Freq")  
num.id<-dim(freq.id)[1]  
  
start.time <- Sys.time()  
accuracy<-NULL  
  
for(s in 1:10){  
index<-c(2017,29,666,1024,38,97,10001,88,494,10)  
set.seed(index[s])  
  
  
sample\_split<-function(i){  
   
   
   
   
   
  
Data<- df[df$auther.id==i,]  
bound <- floor(nrow(Data)/2) #define % of training and test set  
Data <- Data[sample(nrow(Data)), ] #sample rows   
df.train <- Data[1:bound, ] #get training set  
df.test <- Data[(bound+1):nrow(Data), ] #get test set  
  
return(list(df.train=df.train,df.test=df.test))  
}  
  
train<-NULL  
test<-NULL  
for(i in seq(num.id)){  
 train[i]<-list(sample\_split(i)[[1]])  
 test[i]<-list(sample\_split(i)[[2]])  
}   
  
TT<-NULL  
for(i in 1:num.id){  
 TT <- rbind(TT,test[[i]])  
}  
test\_x<-TT[,-c(1,2)] # test set as data.frame  
test\_y<-TT[,2] # test lable   
  
NB\_para<-function(data){  
 # Input: df.train   
 # Output: 6 probability used for NB model  
   
 paper.num<-dim(data)[1]  
 p\_0<-sum(data$id.num.co==0)/paper.num   
 #### P(N|X)  
 p\_1<-1-p\_0   
 #### P(Co|x)  
 colsum<-apply(data,2,sum)  
 data<-rbind(data,colsum)  
 p\_s\_cx<-sum(colsum>=2)/sum(colsum>=1)  
 #### P(Seen|Co,X)  
 p\_u\_cx<-1-p\_s\_cx  
 #### P(Unseen|Co,X)  
 total.num.co<-colsum[1]  
 p\_a\_scx<-colsum/total.num.co   
 p\_a\_scx<-p\_a\_scx[-c(1,2)]   
 #### P(A|Seen,Co,X)  
 p\_a\_ucx<-1/(num.coauther -sum(colsum>=1))   
 #### P(A|Unseen,Co,X)  
 outcome<-list(P.N.X = p\_0,  
 P.Co.X = p\_1,  
 P.S.Co.X = p\_s\_cx,  
 P.U.Co.X = p\_u\_cx,  
 P.Ak.S.Co.X = p\_a\_scx,  
 P.Ak.U.Co.X = p\_a\_ucx  
 )  
 return(outcome)  
}  
  
P.N.X<-NULL  
P.Co.X <-NULL  
P.S.Co.X<-NULL  
P.U.Co.X<-NULL  
P.Ak.S.Co.X<-NULL  
P.Ak.U.Co.X<-NULL  
for(i in seq(num.id)){  
 P.N.X[i]<-NB\_para(train[[i]])$P.N.X  
 P.Co.X[i]<-NB\_para(train[[i]])$P.Co.X  
 P.S.Co.X[i]<-NB\_para(train[[i]])$P.S.Co.X  
 P.U.Co.X[i]<-NB\_para(train[[i]])$P.U.Co.X  
 P.Ak.U.Co.X[i]<-NB\_para(train[[i]])$P.Ak.U.Co.X  
 P.Ak.S.Co.X[i]<-list(NB\_para(train[[i]])$P.Ak.S.Co.X)  
}  
  
###############################  
######### Test Model###########  
###############################  
  
freq.x<-NULL  
for(i in seq(num.id)){  
 freq.x[i]<-dim(train[[i]])[1]   
 }  
px<-freq.x/sum(freq.x)  
########### px is the prior for the nameset#####  
  
  
#######################  
##estimation model#####  
#######################  
NB\_model<-function(data){  
 # input: Test datafram  
 # data.df=test\_x  
 # out<-NULL  
 # for(m in seq(dim(data.df)[1])){  
 # data = data.df[m,]  
 if(sum(data)==0){  
 out<-which.max(P.N.X)  
 }  
 else{  
 condition<-function(id){  
 i=id  
 # i=id, k = k-th coauther  
 P\_Ak\_X<-NULL  
   
 significant<-length(P.Ak.S.Co.X[[i]][data>0])  
   
 for(k in seq(significant)){  
 P\_Ak\_X[k]<-P.Co.X[i] \* ( P.S.Co.X[i] \*   
 P.Ak.S.Co.X[[i]][data>0][k] +   
 P.Ak.U.Co.X[i] \* P.U.Co.X[i])  
 }  
 return(prod(P\_Ak\_X))  
}  
 target<-NULL  
 for(i in seq(num.id)){   
 target[i]<-condition(i)\*px[i]  
 }  
   
 out <-which.max(target)  
 }  
 return(out)  
}  
  
est\_y<-apply(test\_x,1,NB\_model) ## test estimate  
  
accuracy[s]<-sum(est\_y==test\_y)/length(test\_y)  
}  
  
  
end.time <- Sys.time()   
time\_sclust <- end.time - start.time  
time\_sclust  
  
  
mean(accuracy)  
sd(accuracy)  
conclusion<-cat("Mean of Accuracy for Nameset ",n," is:",mean(accuracy),"\nSrdDev of Accuracy for Nameset ",n," is:",sd(accuracy),"\nTime for Nameset ",n," to process is",time\_sclust)  
return(conclusion)  
}  
  
conclusion<-NULL  
  
for(i in 1:length(data.files)){  
conclusion[i]<-get.info(i)  
}

## Mean of Accuracy for Nameset 1 is: 0.9077703   
## SrdDev of Accuracy for Nameset 1 is: 0.01788024   
## Time for Nameset 1 to process is 26.01298Mean of Accuracy for Nameset 2 is: 0.7919355   
## SrdDev of Accuracy for Nameset 2 is: 0.02242651   
## Time for Nameset 2 to process is 5.014568Mean of Accuracy for Nameset 3 is: 0.8081928   
## SrdDev of Accuracy for Nameset 3 is: 0.02125709   
## Time for Nameset 3 to process is 1.236477Mean of Accuracy for Nameset 4 is: 0.8053476   
## SrdDev of Accuracy for Nameset 4 is: 0.04941182   
## Time for Nameset 4 to process is 6.747859Mean of Accuracy for Nameset 5 is: 0.7986486   
## SrdDev of Accuracy for Nameset 5 is: 0.01655061   
## Time for Nameset 5 to process is 3.026377Mean of Accuracy for Nameset 6 is: 0.7844828   
## SrdDev of Accuracy for Nameset 6 is: 0.06514818   
## Time for Nameset 6 to process is 3.665123Mean of Accuracy for Nameset 7 is: 0.8534091   
## SrdDev of Accuracy for Nameset 7 is: 0.03278212   
## Time for Nameset 7 to process is 3.417467Mean of Accuracy for Nameset 8 is: 0.7951064   
## SrdDev of Accuracy for Nameset 8 is: 0.0108513   
## Time for Nameset 8 to process is 45.70861Mean of Accuracy for Nameset 9 is: 0.9062937   
## SrdDev of Accuracy for Nameset 9 is: 0.01899445   
## Time for Nameset 9 to process is 3.966833Mean of Accuracy for Nameset 10 is: 0.83375   
## SrdDev of Accuracy for Nameset 10 is: 0.03283481   
## Time for Nameset 10 to process is 2.933592Mean of Accuracy for Nameset 11 is: 0.7526316   
## SrdDev of Accuracy for Nameset 11 is: 0.02309303   
## Time for Nameset 11 to process is 4.958675Mean of Accuracy for Nameset 12 is: 0.9290476   
## SrdDev of Accuracy for Nameset 12 is: 0.01317556   
## Time for Nameset 12 to process is 8.310383Mean of Accuracy for Nameset 13 is: 0.793484   
## SrdDev of Accuracy for Nameset 13 is: 0.01377755   
## Time for Nameset 13 to process is 2.289227Mean of Accuracy for Nameset 14 is: 0.8664625   
## SrdDev of Accuracy for Nameset 14 is: 0.01350176   
## Time for Nameset 14 to process is 2.238624