Proj4-NB

Yifei Tang yt2547

2017/4/13/

In this file, we implement methods for Author Disambiguation problem in paper2 (Naive Bayes Model). We will first give an overall instruction about our process of Naive Bayes Method based on Nameset "AGupta", and show the results of all the Namesets at the end.

## Step 0: Load the packages, specify directories

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

## Step 1: Load and process the data

source("DataCleaningTZ.R")  
n=1 # Nameset "AGupta"  
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<-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))  
 head.matrix(df)

## id.num.co auther.id B Kvande I Levinstein K Maly M Olson R Mukkamala  
## 1 8 10 1 1 1 1 1  
## 2 7 10 0 0 1 0 1  
## 3 8 10 1 1 1 1 1  
## 4 6 10 0 0 0 0 0  
## 5 6 10 0 0 1 0 1  
## 6 8 10 0 0 1 0 0

## R Chambers R Whitney S Nanjangud C Vemuru H Syed H Abdel-Wahab M Kholief  
## 1 1 1 1 0 0 0 0  
## 2 0 0 0 1 1 1 1  
## 3 1 1 1 0 0 0 0  
## 4 0 0 0 0 0 1 0  
## 5 0 0 0 1 1 1 0  
## 6 0 0 0 0 0 1 0

## Step 2: Naive Bayes Model

We assume that each author’s citation data is generated by the naive Bayes model, and use his/her past citations as the training data to estimate the model parameters. Based on the parameter estimates, we use the Bayes rule to calculate the probability that each name entry (, where is the total number of candidate name entries in the citation database) would have generated the input citation.

Given an input test citation C with the omission of the query author, the target function is to find a name entry in the citation database with the maximal posterior probability of producing the citation , i.e.

(1)

Using the Bayes rule, the problem becomes finding:

(2)

Since does not depend on Then Function (2) becomes：

(3)

Function (3) is the target of our method.

First, we use coauthor as the attribute to do the Naive Bayes Model.

### 2.1 Attribute - Coauthor

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]

Split the Nameset into Train set and Test set

set.seed(29)  
sample\_split<-function(i){  
 Data<- df[df$auther.id==i,]  
 bound <- floor(nrow(Data)/2) #define 50 % 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

Then, we use the train set to train the required probabilities

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)  
 }  
 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

Build the Naive Bayes Model and use test set to do the prediction and calculate the accuracy rate.

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<-sum(est\_y==test\_y)/length(test\_y)  
 accuracy

## [1] 0.902027

## Step 3 Outcomes for all nameset

source("NB\_Co.R")  
  
summary.all

## Nameset mean.accuracy StdDev.accuracy  
## 1 A Gupta 0.9077703 0.01788024  
## 2 A Kumar 0.7919355 0.02242651  
## 3 C Chen 0.8081928 0.02125709  
## 4 D Johnson 0.8053476 0.04941182  
## 5 J Lee 0.7986486 0.01655061  
## 6 J Martin 0.7844828 0.06514818  
## 7 J Robinson 0.8534091 0.03278212  
## 8 J Smith 0.7951064 0.01085130  
## 9 K Tanaka 0.9062937 0.01899445  
## 10 M Brown 0.8337500 0.03283481  
## 11 M Jones 0.7526316 0.02309303  
## 12 M Miller 0.9290476 0.01317556  
## 13 S Lee 0.7934840 0.01377755  
## 14 Y Chen 0.8664625 0.01350176