SVM_Final

He Zhu

4/13/2017

Load library

```
library(stringr)
```

```
## Warning: package 'stringr' was built under R version 3.3.2
```

```
library(e1071)
```

```
## Warning: package 'e1071' was built under R version 3.3.2
```

```
library(text2vec)
```

Set working directory

```
#set working directory
setwd("~/Desktop/Spr2017-proj4-team1-master/doc")
#confusion matrix source
source('../lib/evaluation_measures.R')
```

Data cleaning

```
#data cleaning
data.lib="~/Desktop/Spr2017-proj4-team1-master/data/nameset"
data.files=list.files(path=data.lib, "*.txt")


## remove "*.txt"
query.list=substring(data.files,
                     1, nchar(data.files)-4)

## add a space
query.list=paste(substring(query.list, 1, 1),
                 " ",
                 substring(query.list,
                           2, nchar(query.list)),
                 sep=""
)


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=author_id,
       paper_id=paper_id,
       coauthor_list=coauthor_list,
       paper_title=paper_title,
       journal_name=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])
```

```
}
```

Add an extra list for coauthor

```
for (k in 1: length(query.list)){
  for (j in 1:length(data_list[[k]])){
    data_list[[k]][[j]]$coauthor<-paste(data_list[[k]][[j]][[3]], collapse = ' ')
  }
}
```

```
#paper title
it_train_list1 <- list(1:length(data.files))
#journal name
it_train_list2 <- list(1:length(data.files))
#coauthor
it_train_list3 <- list(1:length(data.files))
PaperID_list<- list(1:length(data.files))
AuthorID_list<- list(1:length(data.files))


vocab1 <- list(1:length(data.files))
vocab2 <- list(1:length(data.files))
vocab3 <- list(1:length(data.files))
for (j in 1:length(data.files)) {
  data_unlist <- unlist(data_list[[j]])
  paper_title<- as.vector(data_unlist[which(names(data_unlist)=="paper_title")])
  journal_name<- as.vector(data_unlist[which(names(data_unlist)=="journal_name")])
  coauthor_name<- as.vector(data_unlist[which(names(data_unlist)=="coauthor")])
  #paper_id<- as.vector(data_unlist[which(names(data_unlist)=="paper_id")])
  PaperID_list[[j]]<- 1:length(data_list[[j]])
  AuthorID_list[[j]]<- as.numeric(as.vector(data_unlist[which(names(data_unlist)=="author_id")]))
  #for paper title
  it_train_list1[[j]] <- itoken(paper_title,
                              preprocessor = tolower,
                              tokenizer = word_tokenizer,
                              #ids =paper_id,
                              ids =PaperID_list[[j]],
                              progressbar = FALSE)
  #for journal name
  it_train_list2[[j]] <- itoken(journal_name,
                              preprocessor = tolower,
                              tokenizer = word_tokenizer,
                              #ids =paper_id,
                              ids =PaperID_list[[j]],
                              progressbar = FALSE)
  #for coauthor name
  it_train_list3[[j]] <- itoken(coauthor_name,
                              preprocessor = tolower,
                              tokenizer = word_tokenizer,
                              #ids =paper_id,
                              ids =PaperID_list[[j]],
                              progressbar = FALSE)

  vocab1[[j]] <- create_vocabulary(it_train_list1[[j]], stopwords = c("a", "an", "the", "in", "on",
                                                        "at", "of", "above", "under"))
  vocab2[[j]] <- create_vocabulary(it_train_list2[[j]], stopwords = c("a", "an", "the", "in", "on",
                                                                      "at", "of", "above", "under"))

  vocab3[[j]] <- create_vocabulary(it_train_list3[[j]], stopwords = c("a", "an", "the", "in", "on",
                                                                      "at", "of", "above", "under"))

}
```

Deal with the issue with author8 [J Smith]

```
AuthorID_list[[8]][AuthorID_list[[8]]==1] <- 2
AuthorID_list[[8]] <- AuthorID_list[[8]]-1
```

Get features from Coauthor, Journal title and Paper title

```
vectorizer<-list()
dtm_train<-list()
for ( i in 1:3){
vectorizer[[i]]<-list(1:length(data.files))
dtm_train[[i]] <- list(1:length(data.files))
}


  for (i in 1:length(data.files)){
    vectorizer[[1]][[i]] <- vocab_vectorizer(vocab1[[i]])
    dtm_train[[1]][[i]] <- create_dtm(it_train_list1[[i]], vectorizer[[1]][[i]])
    vectorizer[[2]][[i]] <- vocab_vectorizer(vocab2[[i]])
    dtm_train[[2]][[i]] <- create_dtm(it_train_list2[[i]], vectorizer[[2]][[i]])
    vectorizer[[3]][[i]] <- vocab_vectorizer(vocab3[[i]])
    dtm_train[[3]][[i]] <- create_dtm(it_train_list3[[i]], vectorizer[[3]][[i]])
  }

dtm_train_tfidf<-list()
for (i in 1:3){
  dtm_train_tfidf[[i]]  <- list(1:length(data.files))
}

for (j in 1:3){
for(i in 1:length(data.files)){
  tfidf <- TfIdf$new()
  dtm_train_tfidf[[j]][[i]] <- fit_transform(dtm_train[[j]][[i]], tfidf)
 }
}
```

Perfrom Hybrid I, cbind dtm_train_tfidf[[1~3]] —-> dtm_train_tfidf[[4]]

```
dtm_train_tfidf[[4]]<-list()
for (j in 1:14){
  dtm_train_tfidf[[4]][[j]]<- cbind(dtm_train_tfidf[[1]][[j]],dtm_train_tfidf[[2]][[j]],dtm_train_tfidf
}
```

Choose index, choose 50% from the whole data set as training sample.

```
authorid<-list()
samplesize<-list()
index_list<-list()
```

```
for (i in 1:length(data.files)){
  # numbers of AuthorIDs
  authorid[[i]]<- length(table(AuthorID_list[[i]]))
  #training size of each AuthorID, take around 50%
  samplesize[[i]]<-ceiling(table(AuthorID_list[[i]])/2)
  #index for the training
  index<-NULL
  for (j in 1:authorid[[i]]){
    index<-c(index,sample(PaperID_list[[i]][AuthorID_list[[i]] == j], size = samplesize[[i]][j]))
  }
  index_list[[i]]<-index
}
```

Factor y variable

```
for (i in 1:length(AuthorID_list)){
  AuthorID_list[[i]]<-factor(AuthorID_list[[i]])
}
```

Gain train and test data

```
#get train and test data
#Note: tm_train_tfidf[[i]][[j]] :
#i= 1: paper title. i = 2: journal name. i = 3 : coauthor. i = 4 : Hybrid I  j: 1~14 authors

x.train<-list();x.test<-list();y.train<-list();y.test<-list()
for (i in 1:4){
  x.train[[i]]  <- list(1:length(data.files))
  x.test[[i]]  <- list(1:length(data.files))
  y.train[[i]]  <- list(1:length(data.files))
  y.test[[i]]  <- list(1:length(data.files))

}
for ( i in 1:4){
  for (j in 1:length(data_list)){
    x.train[[i]][[j]]<-dtm_train_tfidf[[i]][[j]][index_list[[j]],]
    x.test[[i]][[j]]<-dtm_train_tfidf[[i]][[j]][-index_list[[j]],]
    y.train[[i]][[j]]<-AuthorID_list[[j]][index_list[[j]]]
    y.test[[i]][[j]]<-AuthorID_list[[j]][-index_list[[j]]]
  }
}
```

Choose best parameter

```
svm_tune<-list()
for (i in 1:4){
  svm_tune[[i]]  <- list(1:length(data.files))
}

############# Warning, this step takes forever###############
a<-Sys.time()
for (i in 1:4){
  for (j in 1:length(data_list)){
svm_tune[[i]][[j]] <- tune(svm, train.x=x.train[[i]][[j]], train.y=y.train[[i]][[j]], kernel="radial",
                 ranges=list(cost =c(10,20,30,40,60,80,100,150,200,300),
                             gamma=c(0,0.01,0.05,seq(0.1,1,0.2),2)))
  }
}
Sys.time()-a
```

```
## Time difference of 2.174511 hours
```

```
############# Warning, this step takes forever###############

best_mar<-matrix(NA,nrow = 14, ncol = 4)
best_gam<-matrix(NA,nrow = 14, ncol = 4)
for (i in 1:4){
  for (j in 1:14){
    best_mar[j,i]<-svm_tune[[i]][[j]]$performance$cost[which.min(svm_tune[[i]][[j]]$performance$error)]
    best_gam[j,i]<-svm_tune[[i]][[j]]$performance$gamma[which.min(svm_tune[[i]][[j]]$performance$error)
  }
}
```

Predict y value

```
pred<-list()
for (i in 1:4){
  pred[[i]]  <- list(1:length(data.files))
}

for (i in 1:4){
  for (j in 1:14){
    pred[[i]][[j]]<-predict(svm_tune[[i]][[j]]$best.model,x.test[[i]][[j]])
  }
}
```

accuracy matrix

```
accuracy1<-matrix(NA,nrow = 14, ncol = 4)
for (i in 1:4){
  for (j in 1:14){
    accuracy1[j,i]<-mean(pred[[i]][[j]]==y.test[[i]][[j]])
  }
}

colnames(accuracy1)<-c("Paper Title","Journal Title","Coauthor","Hybrid")
accuracy1<-rbind(accuracy1,apply(accuracy1,2,mean),apply(accuracy1,2,sd))
rownames(accuracy1)<-c(query.list,"Mean","StdDev")
```

Shows accuracy

```
accuracy1
```

```
##             Paper Title Journal Title  Coauthor    Hybrid
## A Gupta       0.6975089     0.5693950 0.8327402 0.8434164
## A Kumar       0.6250000     0.6583333 0.5750000 0.7500000
## C Chen        0.5751295     0.4766839 0.7098446 0.7253886
## D Johnson     0.7458564     0.7237569 0.7955801 0.8287293
## J Lee         0.6362297     0.5169367 0.6804124 0.7717231
## J Martin      0.4259259     0.5740741 0.7222222 0.6481481
## J Robinson    0.6626506     0.6867470 0.7469880 0.9036145
## J Smith       0.7374179     0.7111597 0.7024070 0.8205689
## K Tanaka      0.8832117     0.7664234 0.8467153 0.9124088
## M Brown      0.7808219     0.6027397 0.8082192 0.6986301
## M Jones      0.6299213     0.6614173 0.6141732 0.7795276
## M Miller     0.8960396     0.8663366 0.9306931 0.9207921
## S Lee        0.6418539     0.5716292 0.6685393 0.7556180
## Y Chen       0.6323529     0.5735294 0.7875817 0.7794118
## Mean         0.6835657     0.6399402 0.7443654 0.7955698
## StdDev       0.1218202     0.1052102 0.0962749 0.0812570
```

Build confusion matrix

```r
confusion<-list()
for (i in 1:4){
  confusion[[i]]  <- list(1:length(data.files))
}

b<-Sys.time()
for (i in 1:4){
  for (j in 1: 14){
    confusion[[i]][[j]]<-performance_statistics(matching_matrix(pred[[i]][[j]], y.test[[i]][[j]]))
  }
}
Sys.time()-b
```

```
## Time difference of 1.54136 hours
```

```r
accuracy_matrix<-matrix(NA,nrow = 14, ncol = 4)
percision_matrix<-matrix(NA,nrow = 14, ncol = 4)
recall_matrix<-matrix(NA,nrow = 14, ncol = 4)
f1_matrix<-matrix(NA,nrow = 14, ncol = 4)

for (i in 1:4){
  for (j in 1:14){
    accuracy_matrix[j,i]<-confusion[[i]][[j]]$accuracy
    percision_matrix[j,i]<-confusion[[i]][[j]]$precision
    recall_matrix[j,i]<-confusion[[i]][[j]]$recall
    f1_matrix[j,i]<-confusion[[i]][[j]]$f1
  }
}
```

Rename the matrix

```r
colnames(accuracy_matrix)<-c("Paper Title","Journal Title","Coauthor","Hybrid")
accuracy_matrix<-rbind(accuracy_matrix,apply(accuracy_matrix,2,mean),apply(accuracy_matrix,2,sd))
rownames(accuracy_matrix)<-c(query.list,"Mean","StdDev")

colnames(percision_matrix)<-c("Paper Title","Journal Title","Coauthor","Hybrid")
percision_matrix<-rbind(percision_matrix,apply(percision_matrix,2,mean),apply(percision_matrix,2,sd))
rownames(percision_matrix)<-c(query.list,"Mean","StdDev")

colnames(recall_matrix)<-c("Paper Title","Journal Title","Coauthor","Hybrid")
recall_matrix<-rbind(recall_matrix,apply(recall_matrix,2,mean),apply(recall_matrix,2,sd))
rownames(recall_matrix)<-c(query.list,"Mean","StdDev")

colnames(f1_matrix)<-c("Paper Title","Journal Title","Coauthor","Hybrid")
f1_matrix<-rbind(f1_matrix,apply(f1_matrix,2,mean),apply(f1_matrix,2,sd))
rownames(f1_matrix)<-c(query.list,"Mean","StdDev")
```

Accuracy

```
accuracy_matrix
```

```
##            Paper Title Journal Title   Coauthor    Hybrid
## A Gupta      0.8593798    0.85149975 0.93708693 0.9288002
## A Kumar      0.5735294    0.70504202 0.66652661 0.7211485
## C Chen       0.8549492    0.89617119 0.93548213 0.9250521
## D Johnson    0.6862492    0.71454880 0.81184776 0.8218539
## J Lee        0.9290862    0.95722497 0.95859780 0.9652838
## J Martin     0.5828092    0.80153739 0.84346611 0.7763802
## J Robinson   0.7716721    0.83573318 0.88774611 0.9562151
## J Smith      0.9037967    0.88865216 0.89259664 0.9461400
## K Tanaka     0.8884714    0.80410047 0.83576642 0.9156290
## M Brown      0.8352359    0.76217656 0.90258752 0.7990868
## M Jones      0.7722785    0.80414948 0.76315461 0.8743907
## M Miller     0.8826659    0.84237230 0.92162948 0.9066548
## S Lee        0.8796520    0.94137075 0.94129964 0.9491616
## Y Chen       0.7698405    0.89425350 0.93957725 0.9036670
## Mean         0.7992583    0.83563089 0.87409750 0.8849617
## StdDev       0.1143887    0.07667804 0.08272095 0.0758423
```

Percision

```
percision_matrix
```

```
##            Paper Title Journal Title  Coauthor     Hybrid
## A Gupta      0.7340102     0.5598985 0.8030457 0.89238579
## A Kumar      0.8526316     0.7078947 0.5368421 0.87500000
## C Chen       0.6787592     0.4145636 0.6635121 0.83149317
## D Johnson    0.8793028     0.8183007 0.9455338 0.92723312
## J Lee        0.6313964     0.4225499 0.5809193 0.78699046
## J Martin     0.6511628     0.6899225 0.7441860 0.70542636
## J Robinson   0.6727642     0.5284553 0.7825203 0.92479675
## J Smith      0.8092100     0.6958633 0.7008065 0.88066950
## K Tanaka     0.9298643     0.6755656 0.8515837 0.93303167
## M Brown      0.8119891     0.6403270 0.8555858 0.83378747
## M Jones      0.6275739     0.6401074 0.6150403 0.79409132
## M Miller     0.9406323     0.8728427 0.9505730 0.94601684
## S Lee        0.7744405     0.6180801 0.6389870 0.84962701
## Y Chen       0.8127732     0.6019913 0.8282338 0.86134046
## Mean         0.7718936     0.6347402 0.7498121 0.86013499
## StdDev       0.1078603     0.1288998 0.1313886 0.06698377
```

Recall

```
recall_matrix
```

```
##            Paper Title Journal Title  Coauthor    Hybrid
## A Gupta      0.3920824     0.3493823 0.6506272 0.5966401
## A Kumar      0.3147923     0.3929876 0.3273165 0.4247844
## C Chen       0.2127204     0.2232130 0.4180192 0.3909282
## D Johnson    0.4696846     0.4960380 0.6065688 0.6236811
## J Lee        0.2040588     0.2720876 0.3200803 0.4015044
## J Martin     0.1320755     0.2672673 0.3344948 0.2439678
## J Robinson   0.3495248     0.4429302 0.5833333 0.8024691
## J Smith      0.5439231     0.4977976 0.5107445 0.7056494
## K Tanaka      0.6992174     0.5740100 0.6102464 0.7637037
## M Brown       0.4501511     0.3228022 0.6073501 0.3958603
## M Jones       0.3327005     0.3803191 0.3192379 0.5336943
## M Miller      0.7772961     0.7350308 0.8481153 0.8200096
## S Lee         0.2188384     0.3651281 0.3680045 0.4329732
## Y Chen        0.1978132     0.3336324 0.5272324 0.3950035
## Mean          0.3782056     0.4037590 0.5022408 0.5379192
## StdDev        0.1934838     0.1366551 0.1600578 0.1819840
```

F1

```
f1_matrix
```

```
##            Paper Title Journal Title  Coauthor    Hybrid
## A Gupta      0.5111347     0.4302711 0.7188458 0.7151429
## A Kumar      0.4598191     0.5054016 0.4066783 0.5719200
## C Chen       0.3239242     0.2901831 0.5129039 0.5318201
## D Johnson    0.6123037     0.6176616 0.7390379 0.7457508
## J Lee        0.3084354     0.3310232 0.4127434 0.5317316
## J Martin     0.2196078     0.3852814 0.4615385 0.3625498
## J Robinson   0.4600417     0.4819277 0.6684028 0.8593012
## J Smith      0.6505612     0.5803978 0.5908675 0.7835044
## K Tanaka     0.7982132     0.6206610 0.7109936 0.8399185
## M Brown      0.5792031     0.4292237 0.7104072 0.5368421
## M Jones      0.4348635     0.4771438 0.4203120 0.6383591
## M Miller     0.8511994     0.7980308 0.8964260 0.8785179
## S Lee        0.3412482     0.4590654 0.4670349 0.5736249
## Y Chen       0.3181861     0.4293260 0.6443122 0.5416232
## Mean         0.4906244     0.4882570 0.5971789 0.6507576
## StdDev       0.1890770     0.1315340 0.1523183 0.1545516
```