Project 4 - Paper 5 Main Script

Xuehan Liu

04/05/2017

In this file, we illustrate our step-by-step procedure on the error driven online training algorithm on the nameset AKumar.txt (step0 - step3). We implement the algorithm on all name sets based on the step-by-step procedure described from step0-step3, and saved it in the "lib" folder. In step4, we report the evaluation to all namesets provided.

Step 0: Load the packages, specify directories

```
setwd("~/Desktop/Spr2017-proj4-team-9")
# here replace it with your own path or manually set it in RStudio
# to where this rmd file is located


if (!require("pacman")) install.packages("pacman")
```

```
## Loading required package: pacman
```

```
pacman::p_load(text2vec, dplyr, qlcMatrix, kernlab, knitr)

#Create useable csv for each name set
source("~/Desktop/Spr2017-proj4-team-9/lib/data cleaner.R")
```

Step 1: Load and process the data

```
AKumar <- data.frame(scan("../data/nameset/AKumar.txt",
                          what = list(Coauthor = "", Paper = "", Journal = ""),
                          sep=">", quiet=TRUE),stringsAsFactors=FALSE)

# extract canonical author id befor "_"
  AKumar$AuthorID <- sub("_.*","", AKumar$Coauthor)
  # extract paper number under same author between "_" and first whitespace
  AKumar$PaperNO <- sub(".*_(\\w*)\\s.*", "\\1",  AKumar$Coauthor)
  # delete "<" in AKumar$Coauthor, you may need to further process the coauthor
  # term depending on the method you are using
  AKumar$Coauthor <- gsub("<","",sub("^.*?\\s","",  AKumar$Coauthor))
  # delete "<" in AKumar$Paper
  AKumar$Paper <- gsub("<","", AKumar$Paper)
  # add PaperID for furthur use, you may want to combine all the nameset files and
  # then assign the unique ID for all the citations
   AKumar$PaperID <- rownames( AKumar)
```

Step 2: Feature Design

As mentioned in the paper, we can use TF-IDF to collect all unique terms in each citation.

```r
it_train <- itoken(AKumar$Paper,
             preprocessor = tolower,
             tokenizer = word_tokenizer,
             ids = AKumar$PaperID,
             # turn off progressbar because it won't look nice in rmd
             progressbar = FALSE)
vocab <- create_vocabulary(it_train, stopwords = c("a", "an", "the", "in", "on",
                                                   "at", "of", "above", "under"))

#vocab

vectorizer <- vocab_vectorizer(vocab)
dtm_train <- create_dtm(it_train, vectorizer)
dim(dtm_train)
```

```
## [1] 244 666
```

```r
tfidf <- TfIdf$new()
dtm_train_tfidf <- fit_transform(dtm_train, tfidf)
```

Step 3: Implementing hierirchical clustering and training parameters

In this section, our goal is to train the lambda on hierirchical clustering on our text file AKumar by the error driven online training method introduced in the paper5. We use the ranking perceptron to update the parameters.

```r
####Initialize Parameter lambda
lambda<-rep(1,nrow(dtm_train_tfidf))

#Add the Author's ID as the label column to the feature matrix for future use
dtm_train_tfidf<-cbind(dtm_train_tfidf,as.numeric(AKumar$AuthorID))


#Given the training set, we are able to generate the true clusters.
#Based on the paper, we define true score S_star as the distance of the sum of clusterwise distance. We


#Compute the true score S_star for the giving training data
element<-list()
S_star<-vector(length=length(unique(AKumar$AuthorID)))
for (i in 1:length(unique(AKumar$AuthorID))){
  element[[i]]<-dtm_train_tfidf[dtm_train_tfidf[,ncol(dtm_train_tfidf)]==i,]
  S_star[i]<-sum(dist(element[[i]]))/2
}
S_star<-mean(S_star)
T_star<-dtm_train_tfidf[,ncol(dtm_train_tfidf)]

K=14
k=1
lambda1 <- matrix(NA, nrow = nrow(AKumar), ncol = K)
S1 <- numeric(K)
acc <- numeric(K)
while (k<(K+1)){
```

```
#Implement Hierirchical Clustering
h<-hclust(dist(dtm_train_tfidf*lambda))
#Check the result for the number of cluster equals to the number of unique authors in the dataset.
h_result<-cutree(h,k=length(unique(AKumar$AuthorID)))

#Compute the our own score function S
S<-vector(length=length(unique(AKumar$AuthorID)))
element_s<-list()
for (i in 1:length(unique(AKumar$AuthorID))){
  element_s[[i]]<-dtm_train_tfidf[which(h_result==i),]
  S[i]<-sum(dist(element_s[[i]]))/2
}
S<-mean(S)



#Identify true author for each cluster generated by hclust() function, and assign it to each element of

label<-dtm_train_tfidf[,ncol(dtm_train_tfidf)]
author.clust <- vector(length=length(unique(AKumar$AuthorID)))
for (i in 1:length(unique(AKumar$AuthorID))){
    author.clust[i]<-as.numeric(names(which.max(table(label[which(h_result==i)]))))
}

for (i in 1:unique(AKumar$AuthorID)){
    h_result[h_result==i]<-author.clust[i]
}
T_hat<-h_result

#Update lambda

for (i in 1:length(T_star)){
  if (T_hat[i]!=T_star[i]){
    lambda[i]<-lambda[i]-((S-S_star)/S) #!!!
  }
  else {
    lambda[i]<-lambda[i]
  }
}
lambda1[,k] <- lambda
S1[k] <- S
acc[k] <- mean(label == h_result)
k=k+1
}

#plot(acc)
#lambda1[,K]
h_new<-hclust(dist(dtm_train_tfidf*lambda))
T_hat_overall<-cutree(h_new,k=unique(AKumar$AuthorID))
```

Step 5: Evaluation on all dataset

```
source('../lib/evaluation_measures.R')
source('../lib/data cleaner.R')

#performance statistics for AKumar
matching_matrix_hclust <- matching_matrix(AKumar$AuthorID,T_hat_overall)
performance_hclust.AK <- performance_statistics(matching_matrix_hclust)
performance_hclust.AK
```

```
## $precision
## [1] 0.785056
##
## $recall
## [1] 0.9940082
##
## $f1
## [1] 0.8772613
##
## $accuracy
## [1] 0.9404979
```

```
source('../lib/Y Chen.R')
performance_hclust
```

```
## $precision
## [1] 0.1215654
##
## $recall
## [1] 0.4054701
##
## $f1
## [1] 0.1870506
##
## $accuracy
## [1] 0.7753102
```

```
source('../lib/KTanaka.R')
performance_hclust.KT
```

```
## $precision
## [1] 0.5623479
##
## $recall
## [1] 0.663977
##
## $f1
## [1] 0.6089513
##
## $accuracy
## [1] 0.8024834
```

```
source('../lib/JSmith.R')
performance_hclust.JS
```

```
## $precision
```

```
## [1] 0.2017767
##
## $recall
## [1] 0.4320772
##
## $f1
## [1] 0.275089
##
## $accuracy
## [1] 0.7509349
```

```r
source('../lib/MMiller.R')
performance_hclust.MM
```

```
## $precision
## [1] 0.4083057
##
## $recall
## [1] 0.3601701
##
## $f1
## [1] 0.3827303
##
## $accuracy
## [1] 0.5965795
```

```r
source('../lib/MBrown.R')
performance_hclust.MB
```

```
## $precision
## [1] 0.2902839
##
## $recall
## [1] 0.8924401
##
## $f1
## [1] 0.4380751
##
## $accuracy
## [1] 0.6796526
```

```r
source('../lib/MJones.R')
performance_hclust.MJ
```

```
## $precision
## [1] 0.3828712
##
## $recall
## [1] 0.692079
##
## $f1
## [1] 0.4930036
##
## $accuracy
## [1] 0.8009207
```

```r
source('../lib/JLee.R')
performance_hclust.JL
```

```
## $precision
## [1] 0.08453005
##
## $recall
## [1] 0.6672021
##
## $f1
## [1] 0.1500498
##
## $accuracy
## [1] 0.8175904
```

```r
source('../lib/JMartin.R')
performance_hclust.JM
```

```
## $precision
## [1] 0.132294
##
## $recall
## [1] 0.9769737
##
## $f1
## [1] 0.2330326
##
## $accuracy
## [1] 0.3709781
```

```r
source('../lib/JRobinson.R')
performance_hclust.JR
```

```
## $precision
## [1] 0.2897944
##
## $recall
## [1] 0.7596567
##
## $f1
## [1] 0.4195417
##
## $accuracy
## [1] 0.696732
```

```r
source('../lib/DJohnson.R')
performance_hclust.DJ
```

```
## $precision
## [1] 0.3393976
##
## $recall
## [1] 0.4149859
##
## $f1
## [1] 0.3734048
```

```
##
## $accuracy
## [1] 0.6124422
```

```r
source('../lib/CChen.R')
performance_hclust.CC
```

```
## $precision
## [1] 0.1275084
##
## $recall
## [1] 0.6198893
##
## $f1
## [1] 0.2115102
##
## $accuracy
## [1] 0.770671
```

```r
source('../lib/AGupta.R')
performance_hclust.AG
```

```
## $precision
## [1] 0.2103288
##
## $recall
## [1] 0.6031183
##
## $f1
## [1] 0.3118903
##
## $accuracy
## [1] 0.7380789
```

```r
#compute average of the all four performance statistics
Precision<-c(performance_hclust.AK$precision,performance_hclust$precision,performance_hclust.MB$precisio

Recall<-c(performance_hclust.AK$recall,performance_hclust$recall,performance_hclust.MB$recall,performanc


F1<-c(performance_hclust.AK$f1,performance_hclust$f1,performance_hclust.MB$f1,performance_hclust.KT$f1,p




Accuracy<-F1<-c(performance_hclust.AK$accuracy,performance_hclust$accuracy,performance_hclust.MB$accurac

#Final Performance Summary combining all 14 namesets.
error.driven<-data.frame(Precision=mean(Precision),Recall=mean(Recall),F1=mean(F1),Accuracy=mean(Accura
error.driven
```

```
##   Precision    Recall        F1  Accuracy
## 1 0.3027739 0.6524652 0.7194517 0.7194517
```