Project 4 - Main Script

Boxuan Zhao, Zixuan Guan, Zheren Tang, Yingxin Zhang, Jihan Wei

3/22/2017

ABSTRACT

In this project, our team was assigned two papers that proposed two algorithms concerning name disambiguations and we implmented these two algorithms in R code and we have also proposed evaluation methods to compare these algorithms. Throughout the process of this project, we have observed several interesting trends. In this file, we will present our data reading, preporcessing, algorithm implements as well as evaluation results.

Step 0: Load the packages, specify directories

```
#####################################################################################################
# Here replace it with your own path or manually set it in RStudio to the lib folder               #
#setwd("D:/Columbia University/Spring2017-Applied Data Science/Project_4_Bz2290/Spr2017-proj4-team13/lib")
#####################################################################################################

#Relevant packages
list.of.packages = c("expm","pacman","text2vec","stringr")

new.packages <- list.of.packages[!(list.of.packages %in% installed.packages()[,"Package"])]

if(length(new.packages))
{
  install.packages(new.packages)
}

library("expm")
```

```
## Loading required package: Matrix

##
## Attaching package: 'expm'

## The following object is masked from 'package:Matrix':
##
##     expm
```

```
library("pacman")
library("text2vec")
library("stringr")
```

Step 1: Load and process the data

For each record in the dataset, there are some information we want to extract and store: canonical author id, coauthors, paper title, publication venue title. In our main.rmd file, you will find our programs for input of each data file which have been proprocessed by our functions stored in "dataclean.R" under the lib folder.

```
#Preprocess our data files
source("../lib/dataclean.R")
#Read in our data files
source("../lib/dataInput.R")
```

Step 2: Feature design

Following the section 3.1 in the paper, we want to use paper titles to design features for citations. As the notation used in the paper, we want to find a m-dimensional citation vector α_i for each citation i, $i = 1, ..., n$. In this dataset, $n = 244$. We study "TF-IDF" (term frequency-inverse document frequency) as suggested in the paper.

TF-IDF is a numerical statistics that is intended to reflect how important a word is to a document in a collection or corpus. It is often used as a weighting factor in information retrieval, text mining, and user modeling. The TF-IDF value increases proportionally to the number of times a word appears in the document, but is offset by the frequency of the word in the corpus, which helps to adjust for the fact that some words appear more frequently in general.

$$\mathrm{TF}(t) = \frac{\text{Number of times term } t \text{ appears in a document}}{\text{Total number of terms in the document}}$$
$$\mathrm{IDF}(t) = \log \frac{\text{Total number of documents}}{\text{Number of documents with term } t \text{ in it}}$$
$$\text{TF-IDF}(t) = \mathrm{TF}(t) \times \mathrm{IDF}(t)$$

For Paper 3 Construct our feature design for paper 3 with resepct to Coauthor, Title and Journal

```
source("../lib/paper3/TFIDF_FeatureDesign.R")
#For paper 3
author_name <- list(AGupta,AKumar,CChen,DJohnson,JLee,JMartin,JRobinson,
                    JSmith,KTanaka,MBrown,MJones,MMiller,SLee,YChen)
coauthor <- list()
paper <- list()
journal <- list()
for(i in 1:14){
  coauthor[[i]] <- Create_Coauthor(author_name[[i]])
  paper[[i]] <- Create_Title(author_name[[i]])
  journal[[i]] <- Create_Journal(author_name[[i]])

}
```

For Paper 6

Firstly, source functions and read data:

Then we can choose the interested author:

Step 3: Clustering

First of all, we perofrom the spectral cluster with QR decomposition on the data sets

```
source("../lib/paper3/Spectral ClusterQR.R")

spec_coauthor <- list()
spec_title <- list()
spec_journal <- list()
for(i in 1:14){
  spec_coauthor[[i]] <- Spectral.Cluster(my.dat = coauthor[[i]],n.cluster = length(unique(author_name[[i]]$Aut
  spec_title[[i]] <- Spectral.Cluster(my.dat = paper[[i]],n.cluster = length(unique(author_name[[i]]$AuthorID)
  spec_journal[[i]] <- Spectral.Cluster(my.dat = journal[[i]],n.cluster = length(unique(author_name[[i]]$Autho
}
```

Second of all, we implememnt the algorithm from paper 6 to analysis our data set.

```
##If you want to rerun our algorithm,please set it as TRUE:
##Otherwise, we will load the pre-saved answers:
retrain<-F ##Basically for shorter time for kniting the pdf

####Get Constrian Matrix:
if (retrain){
  n<-nrow(X)
```

```r
  Constraint<-matrix(NA,n,n)

  for(i in 1:n){
  Constraint[i,]<-sapply(1:n,constraint,paper2=i,Split_coauthor)
  }

  ##Initilization:
  answer<-initialization(data,X)

  ##EM Steps:
  cluster<-answer$cluster
  cluster2<-cluster
  A<-answer$A
  m=0

  a1<-Sys.time()
  while(any(cluster!=cluster2)|(m==0)){
    cluster<-cluster2
    M_step<-mstep(cluster=cluster,X=X,A=A,ita=0.01)
    A<-M_step$A
    centroids<-M_step$centroids
    m=m+1

    cluster2<-estep_fixed_clusters2(cluster=cluster,X=X,centroids=centroids,A=A)
    cluster2<-as.numeric(factor(cluster2))
    }
  a2<-Sys.time()
  cat("The training time is",a2-a1)
  cat("The iteration number is",m)
}
```

Step 4: Evaluation

The evaluation will be two fold, the first part of our evaluation will be base on the perfomrance of our model in paper 3 using different features(i.e. coauthor, paper, and jounral). We applied the evaluation based on the following methods:

Let M be the set of machine-generated clusters, and G the set of gold standard clusters. Then. in the table, for example, a is the number of pairs of entities that are assigned to the same cluster in each of M and G. Hence, a and d are interpreted as agreements, and b and c disagreements. When the table is considered as a confusion matrix for a two-class prediction problem, the standard "Precision", "Recall","F1", and "Accuracy" are defined as follows.

$$\text{Precision} = \frac{a}{a+b}$$
$$\text{Recall} = \frac{a}{a+c}$$
$$\text{F1} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$
$$\text{Accuracy} = \frac{a+d}{a+b+c+d}$$

```r
source('../lib/evaluation_measures.R')

spec_eva <- function(author,result){
  matching <- matching_matrix(author$AuthorID,result)
  perform <- performance_statistics(matching)
  return(as.data.frame(perform))
}

eva_df <- data.frame()
```

```r
for(i in 1:14){
  eva_df <- rbind(eva_df,spec_eva(author_name[[i]],spec_coauthor[[i]]))
  eva_df <- rbind(eva_df,spec_eva(author_name[[i]],spec_title[[i]]))
  eva_df <- rbind(eva_df,spec_eva(author_name[[i]],spec_journal[[i]]))
}


rownames(eva_df) <- c("AGupta_coauthor","AGupta_paper","AGupta_journal",
                      "AKumar_coauthor","AKumar_paper","AKumar_journal",
                      "CChen_coauthor","CChen_paper","CChen_journal",
                      "DJohnson_coauthor","DJohnson_paper","DJohnson_journal",
                      "JLee_coauthor","JLee_paper","JLee_journal",
                      "JMartin_coauthor","JMartin_paper","JMartin_journal",
                      "JRobinson_coauthor","JRobinson_paper","JRobinson_journal",
                      "JSmith_coauthor","JSmith_paper","JSmith_journal",
                      "KTanaka_coauthor","KTanaka_paper","KTanaka_journal",
                      "MBrown_coauthor","MBrown_paper","MBrown_journal",
                      "MJones_coauthor","MJones_paper","MJones_journal",
                      "MMiller_coauthor","MMiller_paper","MMiller_journal",
                      "SLee_coauthor","SLee_paper","SLee_journal",
                      "YChen_coauthor","YChen_paper","YChen_journal")
write.csv(eva_df, file = "../output/paper3/eva.csv")
```

```r
eva_df
```

```
##                    precision     recall        f1  accuracy
## AGupta_coauthor    0.7064466 0.46835830 0.5632767 0.9295032
## AGupta_paper       0.2551846 0.15047386 0.1893150 0.8749065
## AGupta_journal     0.1604241 0.22574136 0.1875587 0.8101668
## AKumar_coauthor    0.2441605 0.19284137 0.2154876 0.6996222
## AKumar_paper       0.3536902 0.21460107 0.2671246 0.7480942
## AKumar_journal     0.2466457 0.64348786 0.3566061 0.5032719
## CChen_coauthor     0.4224443 0.29242672 0.3456120 0.9453266
## CChen_paper        0.2713341 0.11359919 0.1601490 0.9411745
## CChen_journal      0.1251820 0.09196125 0.1060304 0.9234386
## DJohnson_coauthor  0.8301258 0.59682827 0.6944057 0.8538236
## DJohnson_paper     0.4384660 0.20078761 0.2754417 0.7060479
## DJohnson_journal   0.3140730 0.23575116 0.2693337 0.6440588
## JLee_coauthor      0.5898543 0.43341983 0.4996795 0.9791131
## JLee_paper         0.2601685 0.13357222 0.1765186 0.9700092
## JLee_journal       0.1428167 0.12479921 0.1332015 0.9609133
## JMartin_coauthor   0.5373134 0.47368421 0.5034965 0.9086229
## JMartin_paper      0.1697861 0.20888158 0.1873156 0.8227156
## JMartin_journal    0.1764108 0.57072368 0.2695146 0.6973938
## JRobinson_coauthor 0.3422039 0.43538388 0.3832109 0.8024600
## JRobinson_paper    0.2446913 0.35717692 0.2904226 0.7539992
## JRobinson_journal  0.2884268 0.30424416 0.2961244 0.7961420
## JSmith_coauthor    0.8908362 0.60096287 0.7177367 0.9484130
## JSmith_paper       0.3510421 0.24506316 0.2886319 0.8681648
## JSmith_journal     0.1948824 0.38548878 0.2588860 0.7591252
## KTanaka_coauthor   0.6436914 0.37161490 0.4711983 0.8108914
## KTanaka_paper      0.5262416 0.33027523 0.4058404 0.7807433
## KTanaka_journal    0.2946669 0.64187023 0.4039090 0.5704584
## MBrown_coauthor    0.3823147 0.54210203 0.4483986 0.8133815
## MBrown_paper       0.3806356 0.31653350 0.3456376 0.8323013
## MBrown_journal     0.1638942 0.36939152 0.2270495 0.6480908
## MJones_coauthor    0.5888228 0.52580166 0.5555306 0.8823285
## MJones_paper       0.2390791 0.43002761 0.3073071 0.7288684
## MJones_journal     0.2115966 0.43087704 0.2838159 0.6958717
## MMiller_coauthor   0.8739247 0.66346939 0.7542923 0.8499043
```

```
## MMiller_paper      0.5416736 0.55020408 0.5459055 0.6821510
## MMiller_journal    0.3990009 0.51074830 0.4480115 0.5629651
## SLee_coauthor      0.6263633 0.45709398 0.5285061 0.9681622
## SLee_paper         0.3000901 0.11866863 0.1700802 0.9547905
## SLee_journal       0.3001564 0.16872565 0.2160206 0.9521918
## YChen_coauthor     0.6535327 0.39018600 0.4886361 0.9479362
## YChen_paper        0.3013238 0.11611207 0.1676297 0.9264872
## YChen_journal      0.1886077 0.08101162 0.1133407 0.9191950
```

Then we implement the code we written from paper 6 for the evaluation

```
#answer_eva<-evalu(True_Author,cluster2)
```

Paper 6 evaluation results:

		Final Result	Time	Iteration
	Precision	0.423	16.47min	3
Mbrown	Recall	0.633		
	F1	0.507		
	Accuracy	0.885		

		Final Result	Time	Iteration
	Precision	0.167	43.31988 mins	3
Akumar	Recall	0.465		
	F1	0.246		
	Accuracy	0.781		

		Final Result	Time	Iteration
	Precision	0.41	5.97min	3
Jmartin	Recall	0.55		
	F1	0.47		
	Accuracy	0.91		

		Final Result	Time	Iteration
	Precision	0.39	14.37 mins	2
JRobinsion	Recall	0.58		
	F1	0.47		
	Accuracy	0.87		

		Final Result	Time	Iteration
	Precision	0.25	375mins	4
DJohnson	Recall	0.75		
	F1	0.38		
	Accuracy	0.77		

		Final Result	Time	Iteration
	Precision	0.421	44.877	1
KTanaka	Recall	0.794		
	F1	0.551		
	Accuracy	0.801		

Figure 1: Figure 1: Evaluation results

Then we start to compare the two papers using same evaluation methods. Since we do not want to produce a length report, we only use some of the data set to demonstrate what we foud during the development of this project

First of all, we compare two algorithms using each of the three feautres(Coauthor, Paper, Journal) using AKumar data set:

AKumar data set results:

Akumar					
		Paper 3		Paper 6	
Coauthor		Results	Time	Results	Time
	Precision	0.23	~0.1 sec	0.75	1.996min
	Recall	0.2		0.27	
	F1	0.21		0.4	
	Accuracy	0.69		0.52	
Title		Results	Time	Results	Time
	Precision	0.35	~0.1 sec	0.18	13.91min
	Recall	0.21		0.47	
	F1	0.27		0.26	
	Accuracy	0.75		0.78	
Journal		Results	Time	Results	Time
	Precision	0.25	~0.1 sec	0.17	6.14min
	Recall	0.64		0.36	
	F1	0.36		0.23	
	Accuracy	0.5		0.76	

Figure 2: Figure 2: Evaluation results

From the above graph, we can observe that:

Then, we also compare the two algorithm using a different data set KTanaka:

KTanaka data set results:

KTanaka					
		Paper 3		Paper 6	
Coauthor		Results	Time	Results	Time
	Precision	0.6	~0.1 sec	0.45	3.35min
	Recall	0.37		0.36	
	F1	0.46		0.4	
	Accuracy	0.79		0.69	
Title		Results	Time	Results	Time
	Precision	0.47	~0.1 sec	0.4	31.71 min
	Recall	0.38		0.76	
	F1	0.42		0.52	
	Accuracy	0.76		0.83	
Journal		Results	Time	Results	Time
	Precision	0.3	~0.1 sec	0.33	13.90min
	Recall	0.65		0.6	
	F1	0.41		0.43	
	Accuracy	0.56		0.79	

Figure 3: Figure 3: Evaluation results

From the above graph, we can see that:

We also have compared the two algorithms using all features:

JMartin data set results:

JMartin (All)				
	Paper 3		Paper 6	
	Final Result	Time	Final Result	Time
Precision	0.23	0.07sec	0.41	5.97min
Recall	0.53		0.55	
F1	0.32		0.47	
Accuracy	0.79		0.91	

Figure 4: Figure 4: Evaluation results

MBrown data set results:

MBrown (All)				
	Paper 3		Paper 6	
	Final Result	Time	Final Result	Time
Precision	0.33	.11 sec	0.423	16.47min
Recall	0.47		0.633	
F1	0.39		0.507	
Accuracy	0.79		0.885	

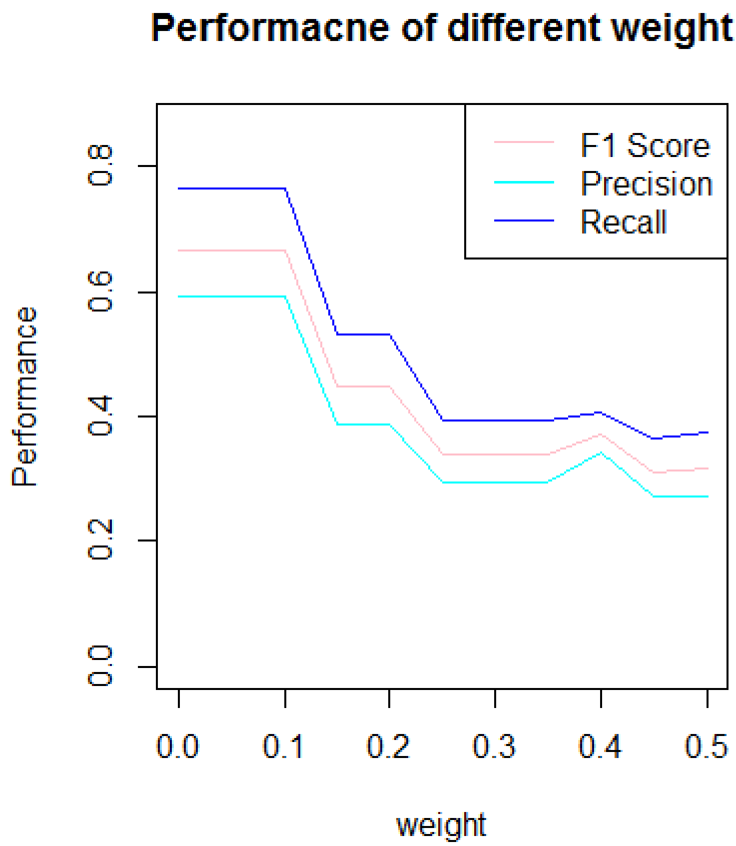
Figure 5: Figure 5: Evaluation results

From the above two graphs, we can see that:

Step 5 Further Observations of methods in paper 6

In addition to the above evaluation, we have also observed several interesting trends for algorithm introduced in paper 6 using c2 constraint.

Interesting results cont'd:



From randomly choose 50 lines for author MBrown for tuning the value for weight, we figured out that the smaller the weight, the better the result.Although we get this conclusion, there should still be further discussions about how to set weight value if we could do deeper optimization.

Interesting results cont'd:

		Scaled X	Sparse X
		After Initilization	After Initilization
MBrown	Precision	0.48	0.64
	Recall	0.66	0.48
	F1	0.57	0.55
	Accuracy	0.89	0.82

		Scaled X	Sparse X
		After Initilization	After Initilization
JRobinson	Precision	0.39	0.62
	Recall	0.58	0.26
	F1	0.47	0.37
	Accuracy	0.87	0.79

After carefully reading and discussing paper6, we first create an initial algorithm. But the result does not have a high accuracy, then we found out that the problem may happens since the X matrix is too sparse. Considering the equation of updating each parameter amm in A, we scaled X matrix, to made the differentiating function more reliable. After this optimization, we finally found that we could get better result as we hope to.

Interesting results cont'd:

Since we found that after EM algorithm, the accuracy does not approve a lot, then we found out that the problem may happens since we initial weight value as 1. Considering the larger the weight, the greater the impact of the constraint is, we

		weight=1	weight=0.01
JMartin	Precision	0.1036184	0.4506579
	Recall	0.1536585	0.5744235
	F1	0.1237721	0.5050691
	Accuracy	0.8564994	0.91361

Figure 6: Figure 8: Interesting results

reset the weight value to 0.1.