Final_Report

Team 3

April 14, 2017

In this project, we implemented the algorithms from **paper1** and **paper6 (with C2 constraint)** to deal with name disambiguation problem. Both of the papers use unsupervised learning algorithm to cluster publications from each of the authors.

Step 0: Load the packages

Step 1: Load and process the data

Paper 1

The algorithm of paper 1 is relatively simple. We only consider explicit coauthor feature. Basically, we pairwisely cluster publications by counting matched coauthors between the two authors.

Experiment Design

- Features: Coauthor matrix Each element xij in the matrix X represents the count of matched coauthors between publications i and j.
- Tuned Parameters: Theta = 1 This means we will stop clustering when none of the author-pair from any two clusters share a coauthor.
- Algorithm: For the initialization, we think of each publications as a single cluster. Then we find the most similar author-pair (the maximum entry in coauthor matrix). Then we merge the corresponding two clusters into a new cluster. We iterate this process until it hits the threshold theta.
- Evaluation: we use matching matrix to evaluate our results (following the function in TA's code)

Due to the imperfect data, we applied three methods trying to improve the result of the clustering algorithm. 1 Leave papers only with the main author or with one unique coauthor as individual cluster 2 Combine papers methioned above to one cluster 3 Delete those papers

The overall performance are shown as the table below

```
start.time <- Sys.time()
table1<-matrix(0, nrow = 2, ncol = 2)
accuracy1<-c()
ratio<-c()
result1<-list()
for(k in 1:length(data_list)){

  result1[[k]]<-paper1_indiv(k)
  table1<-table1+result1[[k]][[1]]
  accuracy1[k]<-result1[[k]][[2]]$accuracy
  ratio[k]<-result1[[k]][[3]]

  result1<-paper1_indiv(k)

  table1<-table1+result1[[1]]
```

```r
  table1<-table+result1[[1]]

  accuracy1[k]<-result1[[2]]$accuracy
  ratio[k]<-result1[[3]]

}
eval1<-performance_statistics(table1)
#plot(ratio, accuracy1, main = "Number of Paper per Author for Each Dataset vs Accuracy")
end.time <- Sys.time()
time1 <- end.time - start.time
#table1
```

```r
start.time <- Sys.time()
table2<-matrix(0, nrow = 2, ncol = 2)
accuracy2<-c()
result2<-list()
for(k in 1:length(data_list)){
  result2[[k]]<-combinecluster(cd_cluster(k))
  table2<-table2+result2[[k]][[1]]
  accuracy2[k]<-result2[[k]][[2]]$accuracy
}
eval2<-performance_statistics(table2)
#plot(ratio, accuracy2, main = "Number of Paper per Author for Each Dataset vs Accuracy")
end.time <- Sys.time()
time2 <- end.time - start.time
#table2
```

```r
start.time <- Sys.time()
table3<-matrix(0, nrow = 2, ncol = 2)
accuracy3<-c()
result3<-list()
for(k in 1:length(data_list)){
  result3[[k]]<-deletecluster(cd_cluster(k))
  table3<-table3+result3[[k]][[1]]
  accuracy3[k]<-result3[[k]][[2]]$accuracy
}
eval3<-performance_statistics(table3)
#plot(ratio, accuracy3, main = "Number of Paper per Author for Each Dataset vs Accuracy")
end.time <- Sys.time()
time3 <- end.time - start.time
#table3
```

```r
compare_df <- data.frame(method=c("Individual Cluster","Combine Cluster","Delete Cluster"),
                         precision=c(eval1$precision, eval2$precision, eval3$precision),
                         recall=c(eval1$recall, eval2$recall, eval3$recall),
                         f1=c(eval1$f1, eval2$f1, eval3$f1),
                         accuracy=c(eval1$accuracy, eval2$accuracy, eval3$accuracy),
                         time=c(time1,time2, time3))
result_table<-kable(compare_df,caption="Comparision of performance for three clustering methods",digits
#result_table
save(eval1, eval2, eval3, ratio, accuracy1, accuracy2, accuracy3, time1, time2, time3, result_table, tal
```

method	precision	recall	f1	accuracy	time
Individual Cluster	0.09	0.72	0.17	0.52	4.613901 mins

method	precision	recall	f1	accuracy	time
Combine Cluster	0.09	0.74	0.17	0.52	43.558536 mins
Delete Cluster	0.09	0.86	0.17	0.43	37.724378 mins

Then we want to find some pattern about the accuracy and ratio (The average number of paper for each author). So change result matrix of each dataset to 14*4 dataframe which can be fed to the "cluster" package

```r
load("../output/paper1.RData")
A <- unlist(result1[[1]][2])
for(i in 2:14){
  t <- unlist(result1[[i]][2])
  t <- as.data.frame(t)
  A <- cbind(A, t)
}
colnames(A) <- c(1:14)
A <- t(A)
print(A)
```

cluster

We use four features(precision, recall, fi, accuracy) of each dataset for feeding the kmeans algorithm. according to the whith "Within groups sum of squares", we choose to have four clusters.

According to the cluster result we calculate the mean accuracy and ratio of each cluster, we found that datasets with higher ratio tend to have higher accuracy which means the algorithm in paper1 is sensitive to the ratio of dataset and improvement needed. (the result shown in Figure 1)

Paper 6

This paper uses a constraint-based probabilistic framework and EM algorithm to do the clustering. We only consider c2 constraint (if two publications share at least one coauthor, then they satisfy the constraint).

Experiment Design

- Features: We use "Paper Name" and "Journal Title" to extract features for each publication. More specifically, we use TF-IDF to construct a dtm matrix as our feature matrix.

- Tuned parameters: step size = 0.01
- Algorithm: objective function we want to minimize: $\sum_i \sum_j \{ \mathbf{D(xi,xj)\ I(li \neq lj)\ w2\ c2(xi,xj)} \} + \sum_{xi} \mathbf{D(xi,yh)}$

The first term measures the distance between two publications which satisfy the constraint but not in the same cluster. The second term measures the distance between xi and its cluster centroid yh. For the initialization, we cluster publications based on c2 constraint. Then we use EM algorithm to re-assign each publication and update corresponding parameters.

- Evaluation: We also use matching matrix to evaluate our results.

The overall performance are shown as the table below (running the whole EM algorithm may take a few hours)

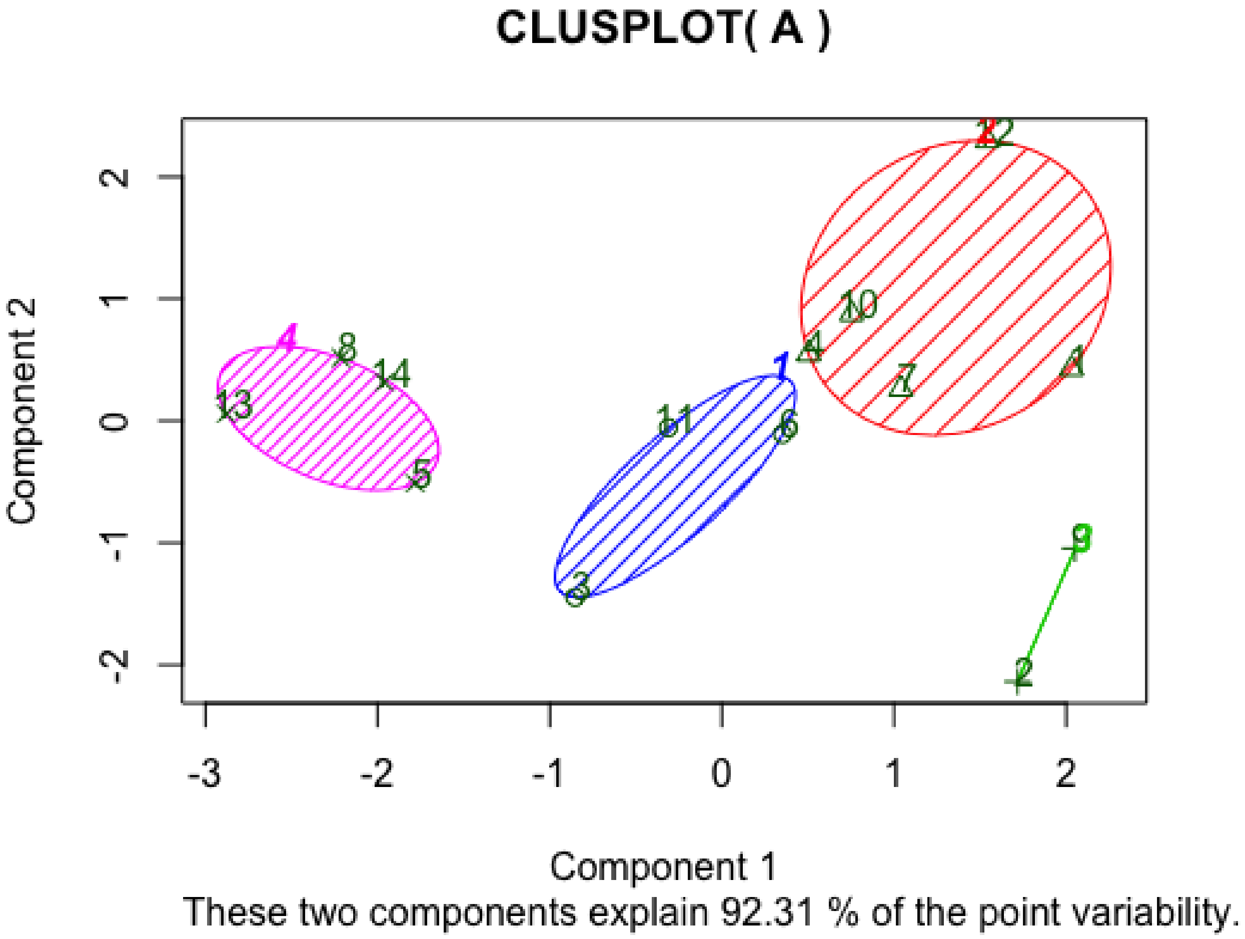
CLUSPLOT(A)
Component 2
Component 1
These two components explain 92.31 % of the point variability.
-3
-2
-1
0
1
2
-2
-1
0
1
2
13
8
14
5
11
6
3
10
4
7
1
12
9
2
4
1
2
3

Figure 1:

```
##        [,1]   [,2]
## [1,] 56004  45106
## [2,] 48870 623545
```

```
##   Precision    Recall        f1  Accuracy
## 1 0.5538918 0.5340122 0.5437704 0.8785094
```

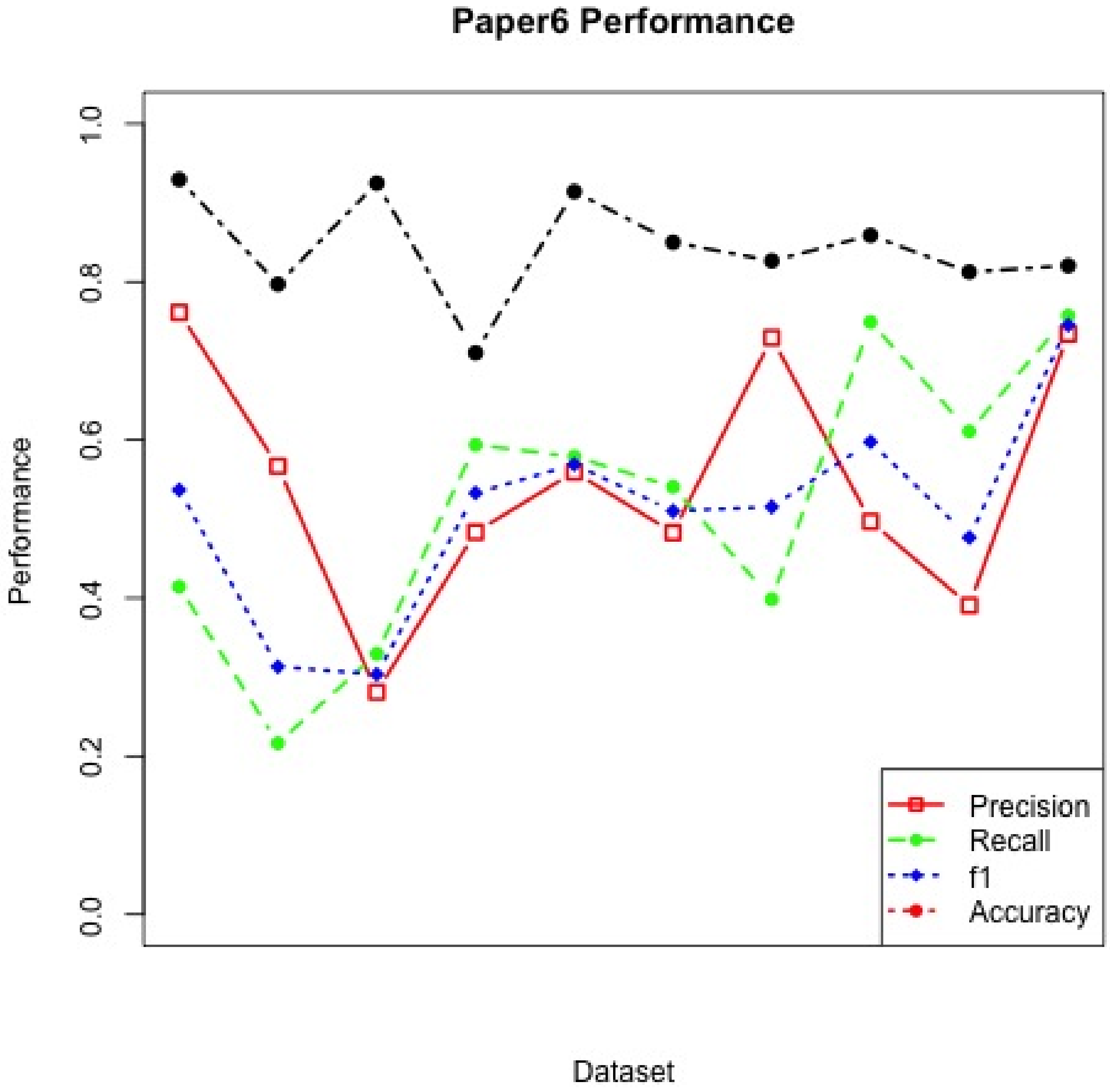

Figure 2:

Comparison and Conclusion

Paper	Precision	Recall	f1	Accuracy	Time
1	0.09	0.72	0.17	0.52	4.613901 mins
6	0.55	0.53	0.54	0.88	A few hours

Based on the overall performance, although it may take longer to run EM algorithm, we can see that algorithm of paper 6 is much better than that of paper 1 in general.