Project 4 - Name disambiguation in author citations using Spectral clustering & EM algorithm with C6 tau-coauthor constraints

Here is the report of the implement of two different algorithms to deal with name disambiguation in author citations,separately used Spectral clustering (paper 3) & EM algorithm with C6 tau-coauthor constraints (paper 6).

Step 0: Load the packages, specify directories

```
## Loading required package: pacman

## Loading required package: nnet

## Loading required package: plyr

## Loading required package: text2vec

## Loading required package: ggplot2

## Loading required package: xlsx

## Loading required package: rJava

## Loading required package: xlsxjars
```

Step 1: Load and process the data

For the data given, there are some information we want to extract and store them in a regular form: canonical author id, coauthors, paper title, publication venue title.

Following the TA's instruction, We read in the records and use regular expressions to replace the punctuations and special symbols. We also split original records to create new variables we need and delete NA rows by writing R function `read_citation()` in `read_citation.R`.

Notice: We saved the data we processed in a Rdata list containing canonical author id, coauthors, paper title, journal title as `text.Rdata`,so we can directly use them while knit pdf without running the above chunk. While our project is evaluated,feel free to check by let `eval=T`.

Step 2: Feature design

Basically,We want to use $Papertitles$, $Coauthor$ and $Journaltitle$ to design features for citations(For paper 6,we also consider these three features' combination like $Papertitles + Coauthor$,$Papertitles + Journaltitle$,$Journaltitle + Coauthor$). 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 is the row number of the dataset.

We study "TF-IDF" (term frequency-inverse document frequency) and NTF (the normalized "TF") as suggested in the paper,which is intended to reflect how important a word is to a document in a collection or corpus,they are defined as the following.

$freq(i, d)$ in the last equation refers to the term frequency of feature i in a citation d. $max(freq(i, d))$ refers to the maximal term frequency of feature i in any citation d.

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

$$TF - IDF(t) = TF(t) * IDF(t)$$
$$ntf(i, d) = freq(i, d) * max(freq(i, d))$$

To compute TF-IDF, we first need to construct a document-term matrix (DTM). In other words, the first step is to vectorize text by creating a map from words to a vector space. Here, we can apply function `vocabulary_Paper`, `vocabulary_Journal` and `vocabulary_Coauthor` to choose different variables in the Dataset we have in `create_dtm.R`.We also remove pre-defined stopwords, the words like a, the, in, I, you, on, etc, which do not provide much useful information and create a DTM. we apply fit_transform function for tfidf DTM and directly compute ntf DTM.

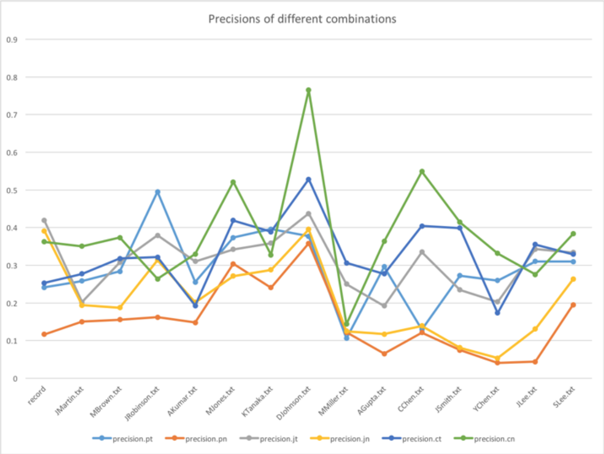
Step 3: Clustering

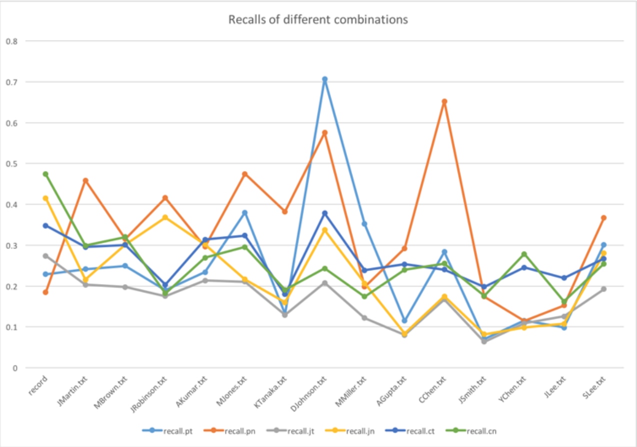
method 1: spectral clustering (paper 3)

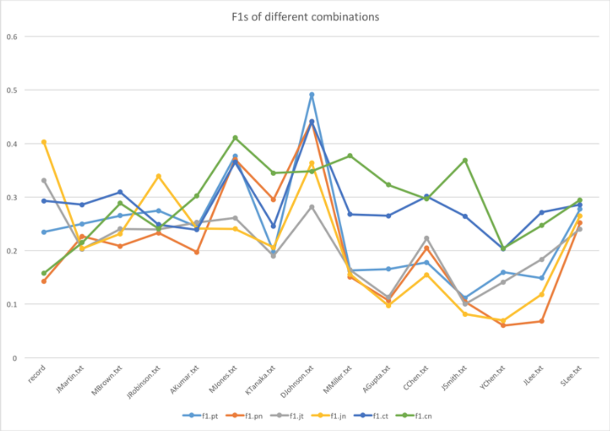
The third part of cluster_citation function above is the spectral cluster executation. Following suggestion in the paper, we carry out spectral clustering on the Gram matrix of the citation vectors sourcing the cluster_citation function in lib. The number of clusters is assumed known as stated in the paper.

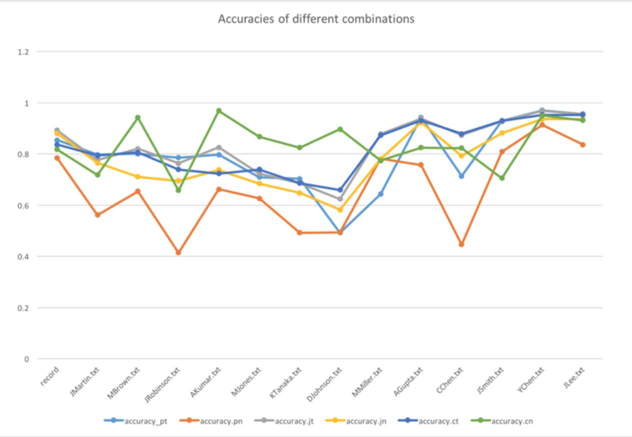
Notice: We saved the result as `spectralc_result_Coauthor.Rdata`,so we can directly use them while knit pdf without running the above chunk.

We tried $Papertitles$, $Coauthor$,$Journaltitle$,$Papertitles + Coauthor$,$Papertitles + Journaltitle$,$Journaltitle + Coauthor$ and $Journaltitle + Coauthor$ separately with tfidf DTM and ntf DTM used.









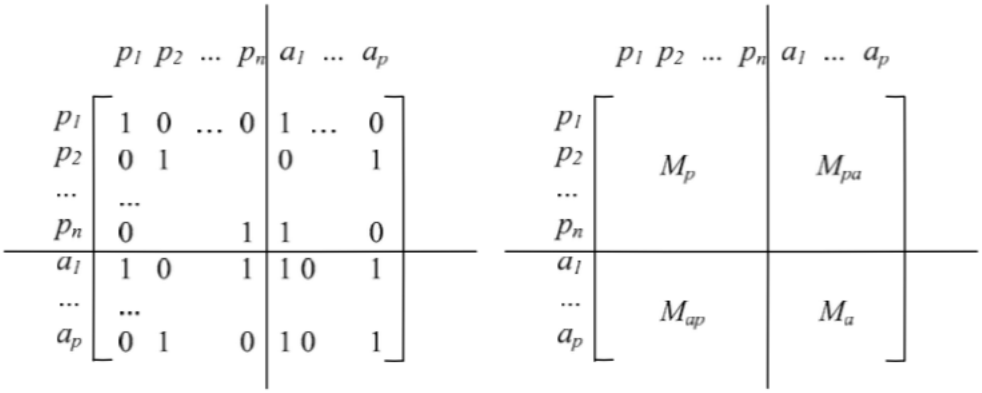
With the matching results, we now do a simple visualization to see the distribution of performance index among all weights of features.four plots are separately demonstrating precision,recall,F1 and accuracy for 14 txt files given.different colors represent different combination of column features chosen($Papertitles$, $Coauthor$,$Journaltitle$,$Papertitles + Coauthor$,$Papertitles + Journaltitle$ and $Journaltitle + Coauthor$).

In the figures,the text files on horizontal axis are in increase orders. Since there is a trade off to do between recall and precision so we chose their harmonic mean f1 as the most essential standard. We can see that when using coauthor names to do the prediction, the stableness and F1 value is higher than other F1 generated by other 2 variables. Considering ntf+coauthor combination provides much better performance in precision, we will apply ntf + coauthor names as our final model.

method 2: EM algorithm with probabilistic model using HMRF incorporating c6 constraint (paper 6)

Based on the unified probabilistic model using Hidden Markov Random Fields (HMRF) in paper 6, we used the model incorporating c6 constraint and a parameterized-distance measure to do the clustering through EM algorithm.

C6 constraint is one of the constraints defined between papers and coauthors. Use papers who got the same name among coauthors, and abstrctact all the unique coauthors from the author information of the paper, exculde the author's name who "wrote" for all the papers. So for matrix M, we define the columns and row as paperID following by unique coauthors' name. The matrix would be like as following:



For M matrix, M_p matrix the initial form is a identify matrix. As for M_{pa} matrix, the element should be 0 or 1. 1 stands for if this author have written the paper(row of the matrix) and 0 vice versa. M_{ap} matrix is the transpose of Mpa. For Maa, the element is also 0 or 1. 1 stands for these two author have coorperated on paper(s). And 0 means no intereaction.

And for the constrain, here is a parameter τ.τ represents the degree for the constrain. When τ=0, $M^{(\tau=0)} = M$, which is the matrix defined above. When τ=1, $M^{(\tau=1)} = M \times M$, and $M^{(1)} \times M$ for τ=2.

What do $M^{(1)}$ matrix stand for when τ=1? For the M_p matrix part, the element on i_{th} row and j_{th} column, means how many coauthor do these two papers share + $I\{i = j\}$. For the M_ap part, element on i_{th} row and h_{th} column: $I\{a_h$ is one of the coauthor of $p_i\}$ + number of coauthors for $paper_i$ author a_h coporated. Also for M_{aa} matrix part, the element stands for number of paper these two authors wrote together, and number of colleage (other authors who published paper with bothe these two authors) they share together.

For τ=2. The colleages dimension is increased. Since we mainly focus on M_p part of the matrix, then explain this part. For element on i_{th} (paper i) row and j_{th} column (paper j): besides how many coauthors do these two paper shares, the element also contains number of the authors that all the author for this paper have directly corporated before.

For building the M^τ matrix, use the function `M_prod()` defined in `customized_function.R`.

Notice: We save the result above as `journal_tfidf.Rdata` and `paper_tfidf.Rdata` for later use,so we can directly load them.

```
load("../output/paper_tfidf.Rdata")
load("../output/journal_tfidf.Rdata")

## initialize
source("../lib/customized_function.R")
source("../lib/em_test5.R")

## construct dtm
dtm<-cbind(tfidf_paper$JMartin.Rdata,tfidf_journal$JMartin.Rdata)   ### change the name here  ###
data.lib="../data/nameset"
file_names=list.files(path=data.lib, "*.txt")
file_names
```

```
##  [1] "AGupta.txt"    "AKumar.txt"    "CChen.txt"     "DJohnson.txt"
##  [5] "JLee.txt"      "JMartin.txt"   "JRobinson.txt" "JSmith.txt"
##  [9] "KTanaka.txt"   "MBrown.txt"    "MJones.txt"    "MMiller.txt"
## [13] "SLee.txt"      "YChen.txt"
```

```
truth<-Data[[6]]$AuthorID  ### change the number here  ###
k <- length(unique(truth))
```

```
n<-nrow(dtm)
M_p<-M_prod(6,tao=1,n)   ### change the number here  ###
```

Here is our author's list, you can change the name and corresponding number above to fit our model.Here take JMartin as an example to illuminate the algorithm.

After calculating the C6 constraint matrix,we followed the EM process to update labels for every paper to minimize the objective function in each iteration until the clusters stop changing.

For initialization of our EM framework, we first cluster publications into disjoint groups based on the constraints over them. Then,we calculate the current number of clusters as lambda and re-adjust the labels of some papers to make the lambda equal to the actual author number.The detail are in the function `ini()` in `em_test5.R`.

```
tag<-ini(M_p)
tag_ini<-tag
```

Before getting into the EM framework, we define two kinds of distance function.

First,distance function $D(x_i, x_j)$.It's used to measure the distance between two papers as the following,A is a parameter matrix(initialized as a diagonal matrix).

$$D(x_i, x_j) = 1 - \frac{x_i^T A x_j}{||x_i||_A ||x_j||_A}, ||x_i||_A = \sqrt{x_i^T A x_j}$$

Second,distance function $D(x_i, y_{l_i})$,l_i is the tag for x_i.It's used to measure the distance between a paper and author y_{l_i}(represented by a set of assigned papers).Actually, we count this distance in this way, for a certain paper x_i,we find the papers with the same current labels with x_i, calculate the distance between the paper x_i and one of these papers respectively,get an average as $D(x_i, y_{l_i})$.

In the E-step,to update a certain paper x_i's label,we changing different label for x_i and try to minimize our objective function as the following.

$$f(y_h, x_i) = D(x_i, y_h) + \sum_{i,j \neq i} D(x_i, x_j) * w * c_6$$

The objective function contains two parts:

The first part is to minimize the distance between paper x_i and its current label, the second part wants to minimize the sum of the distances between x_i and a certain type of papers,whose labels is different from x_i but have a constrain with x_i.

We sequentially update the assignment of each paper until all papers' assignment updated.

In the M-step, we update each cluster's center as the following way:

$$y_h = \frac{\sum_{i:l_i=h} x_i}{|| \sum_{i:l_i=h} x_i||_A}$$

Then, each parameter amm in A is updated by (only parameters on the diagonal):

$$a_{mm} = a_{mm} + \eta \frac{\partial f_{obj}}{\partial a_{mm}}$$

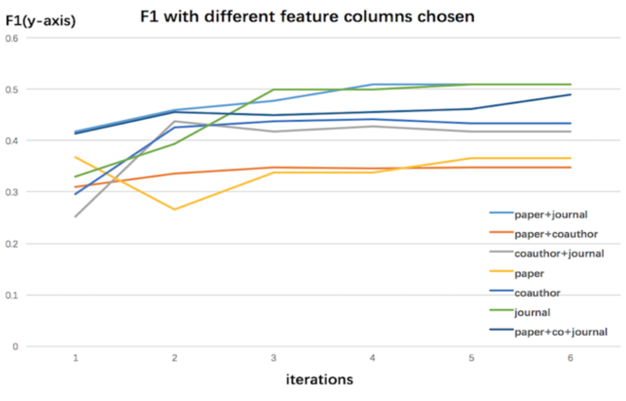
We stop the iterations until the papers' labels do not change.

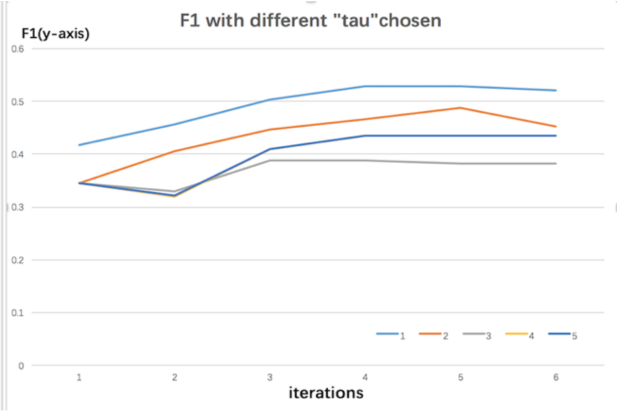
```
## Time difference of 28.23026 secs
```

After the model was basically built,we need to think of which feature columns to use as our candidate to generate DTM which can lead to better performance.

We take JMartin as a test data and try $Papertitles$, $Coauthor$,$Journaltitle$,$Papertitles+Coauthor$,$Papertitles+Journaltitle$,$Journaltitle + Coauthor$ and $Journaltitle + Coauthor + Papertitles$ separately.We use F1 to measure model performance and plot the following F1 curve with iteration as x-axis.

From the plot we can see that,first, there is an increasing trend for F1 as interation times increasing,and $Papertitles + Journaltitle$ lead to best and relatively stable performance of the model.

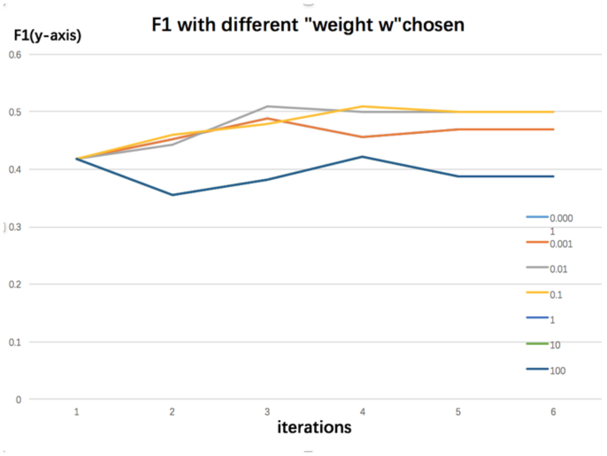


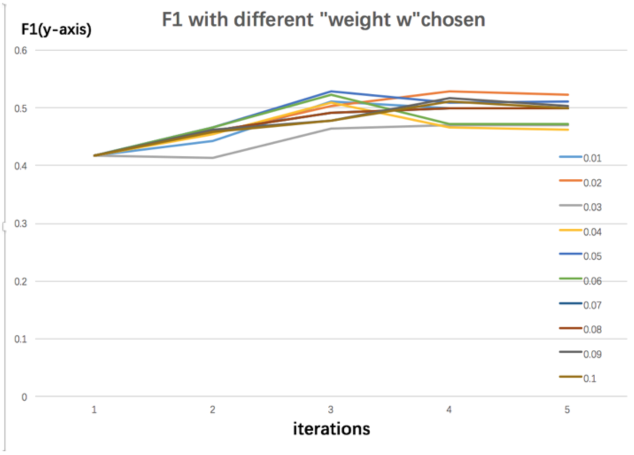


We also need to tune the parameters appeared in the model,like tau which lead to tau-coauthor constrain,weight in the objective function,eta appeared in M step to update A and the initial number of A(a_{mm}).

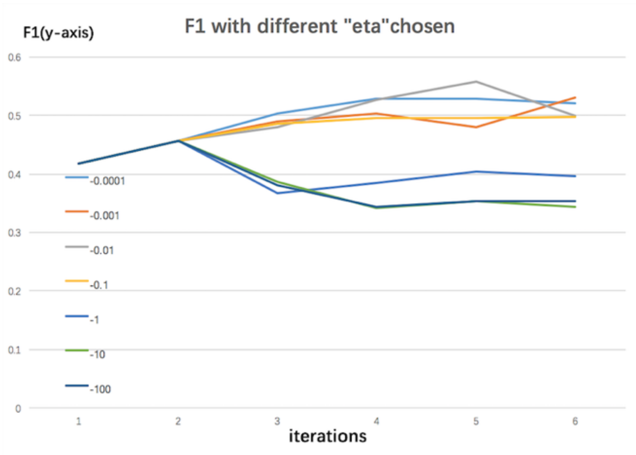
As for tau(the above plot in the right),we tried tau from 1 to 6 and find 1-coauthor constrain better than tau larger than 1,as tau increases which iteration fixed, the model performance become worse and worse.

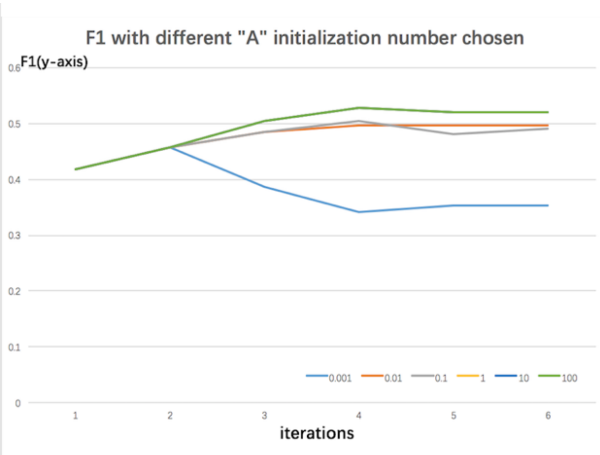
As for weight,we tried from 10ˆ-4 to 10ˆ2 and find 0.01-0.1 being better,then we tried from 0.01 to 0.1.According to the result as following, we determine weight to be 0.02.





We do the similar trying to eta and A and choose eta to be -0.001, a_{mm}(m=1. . . M) to be 1(1,10,100 have the almost same F1).



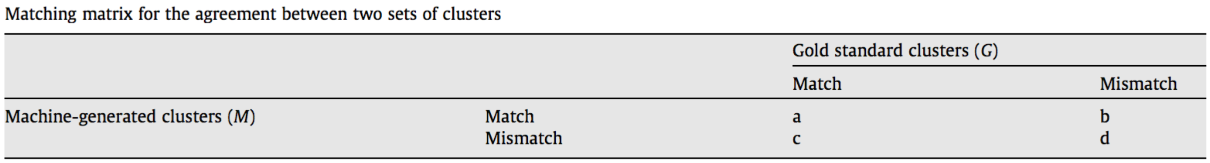


After feature column selection and parameters tuning, we determine our model using $Papertitles+Journaltitle$ as feature data with 1-coauthor constrain and other parameters set.

Step 4: Evaluation

Till now, we use two different methods to deal with name disambiguation problems as a multi-class classification problem. But what we care about while evaluating, is that, for every pair of two papers among the whole dataset(papers), whether they belong to the same cluster or not separately in ground truth and our result given by models,so it becomes a "binary classification" problem and we can use the commen evaluation index like precision,recall,F1,accuracy and also ROC curve to evaluate our methods.

First,to measure the degree of agreement between a set of system-output partitions and a set of true partitions,we need to build the matching matrix as the following for further calculating other indexes(sourcing `evaluation_measures.R` in lib).


Matching matrix for the agreement between two sets of clusters

		Gold standard clusters (*G*)	
		Match	Mismatch
Machine-generated clusters (*M*)	Match	a	b
	Mismatch	c	d

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.

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

1. Precision & Recall

Precision is the ratio of the number of pairs of entities that are assigned to the same cluster in each of M and G over the number of pairs of entities that are assigned to the same cluster in each of M.

Recall the ratio of the number of pairs of entities that are assigned to the same cluster in each of M and G over the number of pairs of entities that are assigned to the same cluster in each of G.The higher they are, the better classification our models perform,saying "better" we mean the consistency of our model and ground truth.

2. F1

Although we want precision and recall both to be high,but there kind of exists conflict between them. So we use $F1$ as a consideration of both precision and recall, defined as the Harmonic Mean of Precision and Recall.The higher it is, the better classification our models perform.

3. accuracy

Accuracy is the ratio that the pair result of 2 papers are correct.The higher it is, the more accurate our models' result are.

The following are our result comparation showing the above four indexes(average of 14 files for paper 3 and 10 files for paper6) side by side.

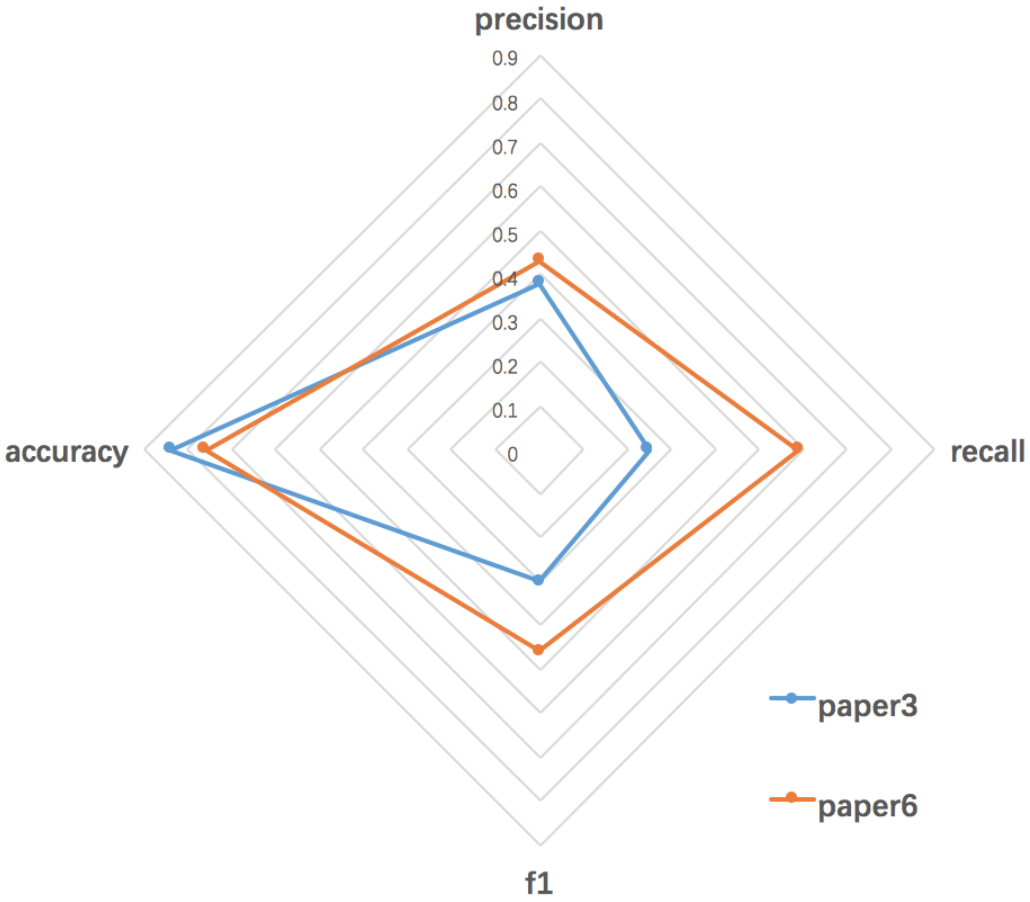
4. time for running the model

*Notice:the actual running time for paper 6 should be longer,since we did not run Jlee.txt, Jsmith.txt, Slee.txt and Ychen.txt for computational difficulty.

```
#0.383556271    0.254224043 0.294654914 0.835649186
#0.42931294 0.586268585 0.45628788  0.75472642
compare_df <- data.frame(method=c("paper3","paper6"),
                         precision=c(0.384, 0.429),
                         recall=c(0.254, 0.586),
                         f1=c(0.295, 0.456),
                         accuracy=c(0.836, 0.755),
                         average_running_time=c("0.495 min",">40.756 min*"))
kable(compare_df,caption="Comparision of performance for two clustering methods",digits = 2)
```

Table 1: Comparision of performance for two clustering methods

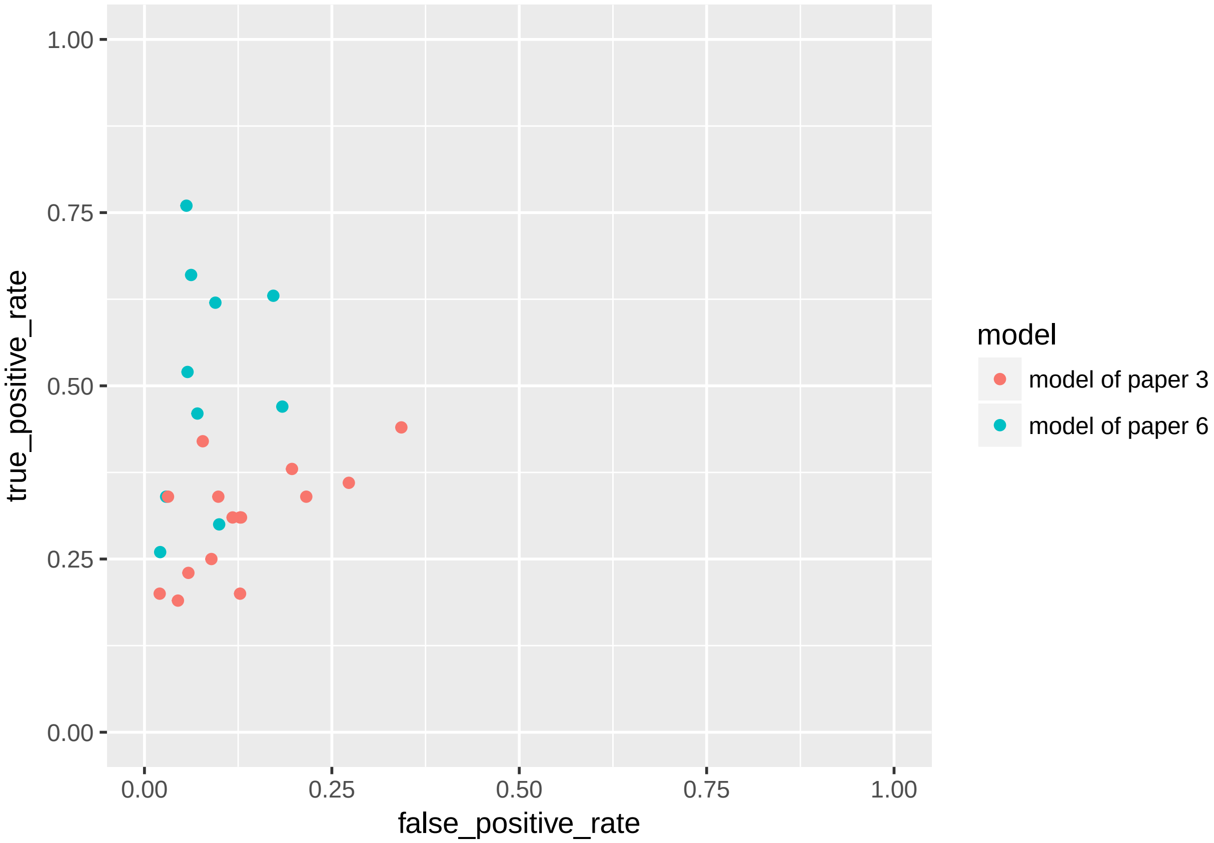
method	precision	recall	f1	accuracy	average_running_time
paper3	0.38	0.25	0.30	0.84	0.495 min
paper6	0.43	0.59	0.46	0.76	>40.756 min*



Generally,we can see from the radar that,paper 6 has a better performance with higher precision,recall,F1 and a little lower accuracy.

5. ROC curve

We also applied ROC (Receiver operating characteristic) to evaluate our data. If the true positive rate is high, it means the model can assign the papers by one author into the same cluster. If the false positive value is low, it means papers of different authors are assigned to different clusters. predict result is good. So if the predict results are close to the up-lift corner, the predict result is the best. From the figure here, we can clearly see model of paper 6 performs better than paper 3.

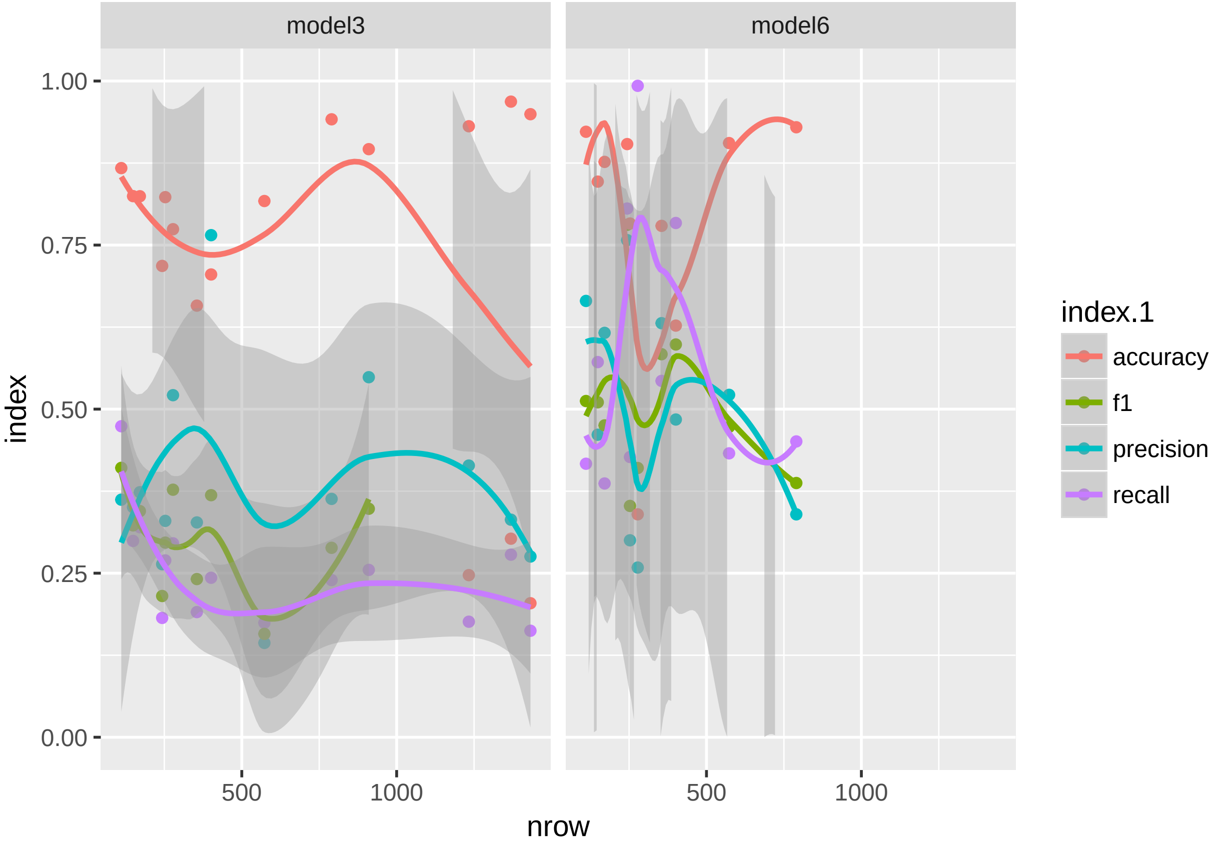


6. robust

Robust, we mean whether our models perform stably with dataset changing and the amount of data correspondingly changing.

```
result<-read.csv("../output/plot.csv",header = T)
result.dat<-as.data.frame(result)
p<-ggplot(data=result.dat, mapping=aes(x=nrow, y=index, colour=index.1))
p+geom_point()+ylim(0,1)+facet_wrap(~ model,nrow=1)+geom_smooth()
```

```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```



From the plot, we cannot find a clear and significant trend that the performance of models will improve when the dataset grows bigger. On the opposite, as the number of rows become larger, the performances of both the two models tend to be more unstable.So in our point of view,the model performance relies more on characteristics of data instead of data scale.

7. time and effort spent for algorithm implement

For paper 3,it's quite clear and easy(due to TA's help) to understand and implement the algorithms,no need to do a lot of debug.

For paper 6, totally the opposite! A lot of uncertainty in the algorithm need to make sure and the time-assuming code need to simplify,which takes a lot of time and effort.

But! The actual performances of two models are not proportional to the time and effort we spent on them.

Conclusion

Paper 6's algorithm performs relatively better than spectral-cluster method in paper 3 with the same dataset provided,though with much more time and effort to study and run the code.