Postgraduate Project 2022

Project II: From raw data to temporal graph structure exploration

Course: Social Network Analysis

Master Program: MSc in Business Analytics

Name: Vioni Konstantina

Student id: p2822107

Supervisor: Katia Papakonstantinopoulou

Teaching Faculty, Researcher,

Department of Informatics,

Athens University of Economics & Business

ATHENS, JULY 2022

Table of Contents

1. Introduction…….……..pg.3

2. Answers…….……pg.3

2.1. Task 1: DBLP co-authorship graph……...……..….pg.3

2.2. Task 2: Average degree over time …….....pg.6

- Number of vertices.……pg.6
- Number of edges…….....pg.6
- Diameter of the graph…….....pg.7
- Average degree (simple, not weighted)…….pg.7

2.3. Task 3: Important nodes…….....pg.8

- Degree (simple, not weighted)……pg.8
- PageRank……..……..….pg.9

2.4. Task 4: Communities……..pg.11

Introduction

In the context of Social Network Analysis course, we were asked to implement our second project. For this project, we used raw data from dblp .[1] Our data were a compressed file with conference proceedings records listed in dblp. The first column indicated the year the paper was published, the second one the title of the paper, and the third one the conference where the paper was presented. Finally, the fourth column of the line was a comma separated list of the paper's authors. To complete our analysis in this assignment we had to manipulate our raw data and ,by the five created csv files, to produce the weighted undirected co-authorship graph for the respective year. After creating these graphs , we had to visualize the 5-year evolution of different metrics for the graph. Furthermore, we had to print data frames for the 5-year evolution of the top-10 authors with regard to degree and pagerank. As a final task we were asked to perform community detection on the mention graphs, detect the evolution of the communities where a random selected user belonged to and visualize the graph using a different color for each community. The process that has been followed to answer these questions is analytically described in the next chapters of this report.

Answers:

Task 1*: DBLP co-authorship graph*

Our first task was to create a weighted undirected graph with igraph, using raw data from dblp. To do so, we downloaded the compressed file "authors.csv.gz" and by the use of Unix Tools we uncompressed the file. The command that was used was ***gzip -d authors.csv.gz.*** It was also needed to filter out the data and keep only the records that referred to conferences "CIKM", "KDD", "ICWSM", "WWW" and "IEEE BigData" and for the last five years. As a matter of fact, we selected these records that their year of publishment was in 2016, 2017 ,2018, 2019 and 2020.The command that was used to select the conferences that we were interested in, was ***cat authors.csv| grep ",CIKM,\|,KDD,\|,ICWSM,\|,WWW,\|,IEEE BigData"|awk '{if($1>=2016)print$0}'> author-sample.csv***. In order to create the files that referred to each year separately, we created the above code in Unix Tools:

- For 2016 "authors_16" csv file:

```
cat authors-sample.csv|awk '{if($1>=2016 && $1<2017)print$0}'> authors-sample16.csv
```

- For 2017 "authors_17" csv file:

```
cat authors-sample.csv|awk '{if($1>=2017 && $1<2018)print$0}'> authors-sample17.csv
```

- For 2018 "authors_18" csv file:

```
cat authors-sample.csv|awk '{if($1>=2018 && $1<2019)print$0}'> authors-sample18.csv
```

- For 2019 "authors_19" csv file:

```
cat authors-sample.csv|awk '{if($1>=2019 && $1<2020)print$0}'> authors-sample19.csv
```

- For 2020 "authors_20" csv file:

```
cat authors-sample.csv|awk '{if($1>=2020 && $1<2021)print$0}'> authors-sample20.csv
```

[1] https://hive.di.uoa.gr/network-analysis/files/authors.csv.gz

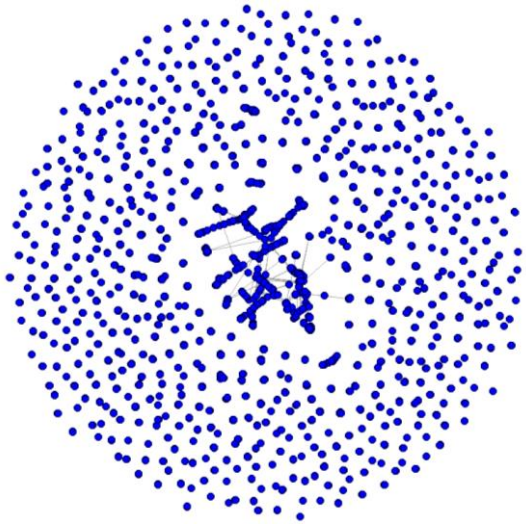
Commands explanation:

- ***cat authors-sample.csv*** : we read the whole file that was created in the previous step
- ***awk '{if($1>=2016 && $1<2017)print$0}'*** : we selected the years that we were interested in
- ***> authors_16.csv*** : we saved the output into a csv file

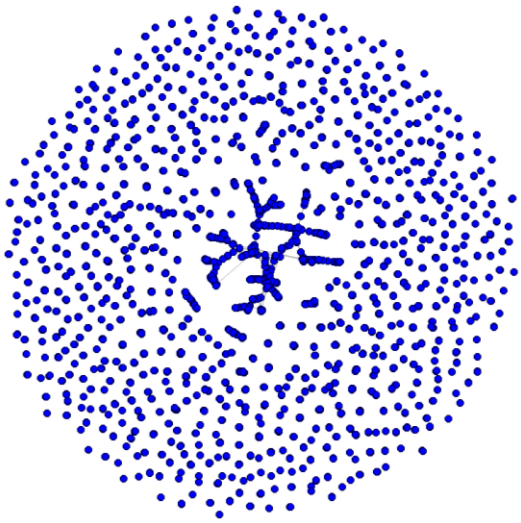
This procedure was followed for every year . After implementing each step, we had finally created the csv files that we wanted. These files consisted of the year, the title of the paper ,the name of the conference and the authors.

Then, we used Python programming language to transform the five csv files in the needed format (from, to, weight). We inserted the data for each year and created the respective dataframes. Since we would not need the columns related to year, title of the paper and conference , we removed them and kept only the column of authors. We also checked for the existence of missing values and dropped them out. To generate our dataset in the wanted format we firstly converted each year's dataframe in a list and then split them separated by comma. In order to calculate the frequency that a pair of authors occurred ,we used libraries 'collections' and 'itertools' and counted the times of each pair existence. The new produced dataframes contained two columns. The first one consisted of the pair of authors separated by comma and the second one by the weight that was calculated before. By the use of a specific code we split the pairs in to two new columns named as 'From' and 'To'. Finally, these dataframes were illustrated in the three columns that we needed and we exported each year's dataframe as a comma separated file.

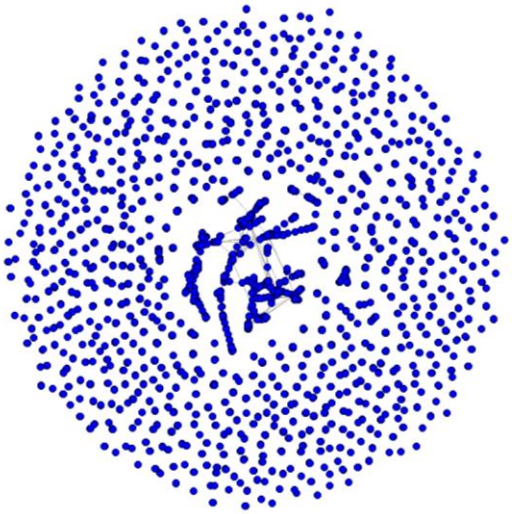
The last step for this task was to insert these datasets in the R programming language and create the igraph graphs that are provided below:



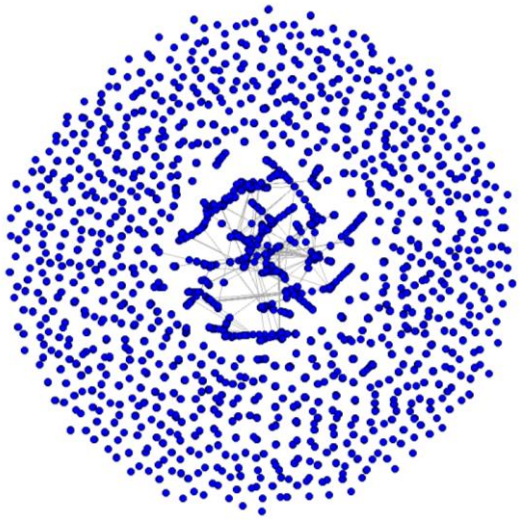
Plot 1.1 : Graph of 2016



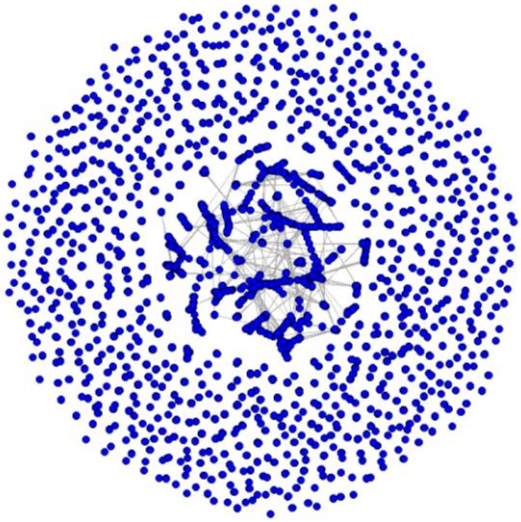
Plot 1.2 : Graph of 2017



Plot 1.3 : Graph of 2018



Plot 1.4 : Graph of 2019

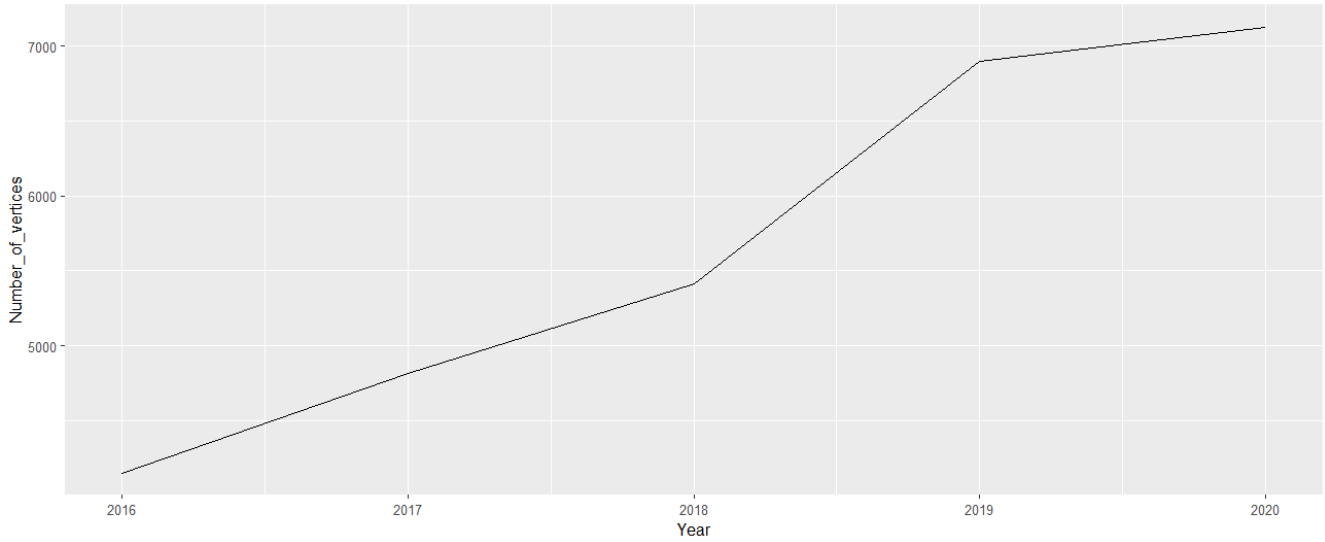


Plot 1.5 : Graph of 2020

Task 2: *Average degree over time*

Our next task was to create plots that visualize the 5-year evolution of different metrics for the graph. More specifically, we created plots for:

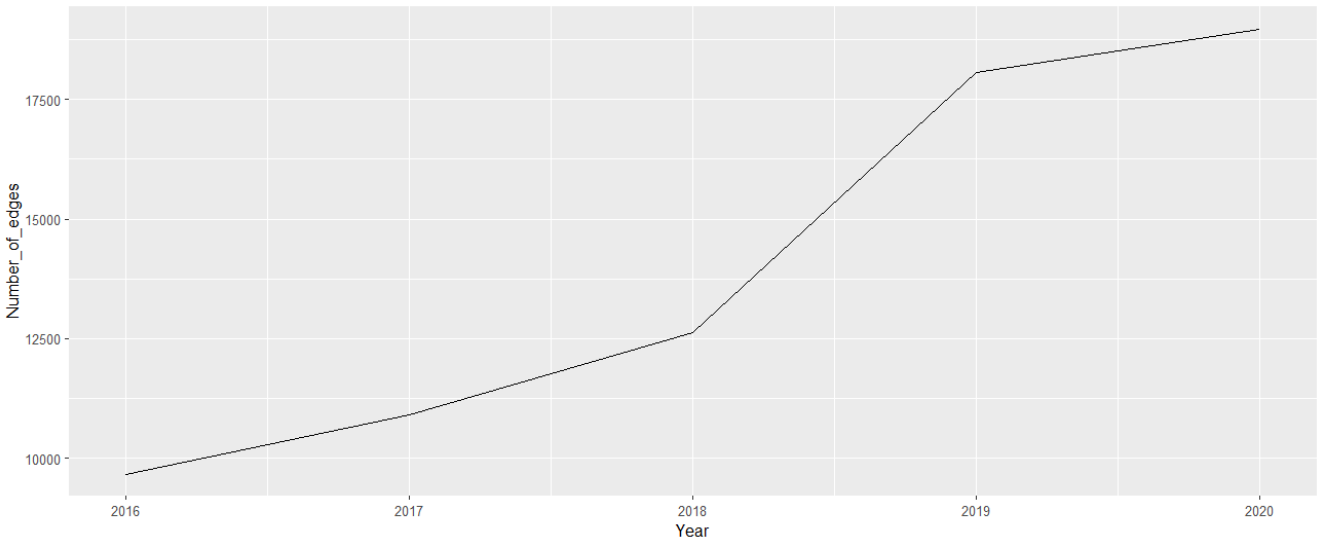
• Number of vertices:



Plot 2.1 : The 5-year evolution of the number of vertices of the graph

As it can be observed, through the years, the number of vertices increased. Since each vertex represented an author, we concluded that in the last five years more authors wrote papers.

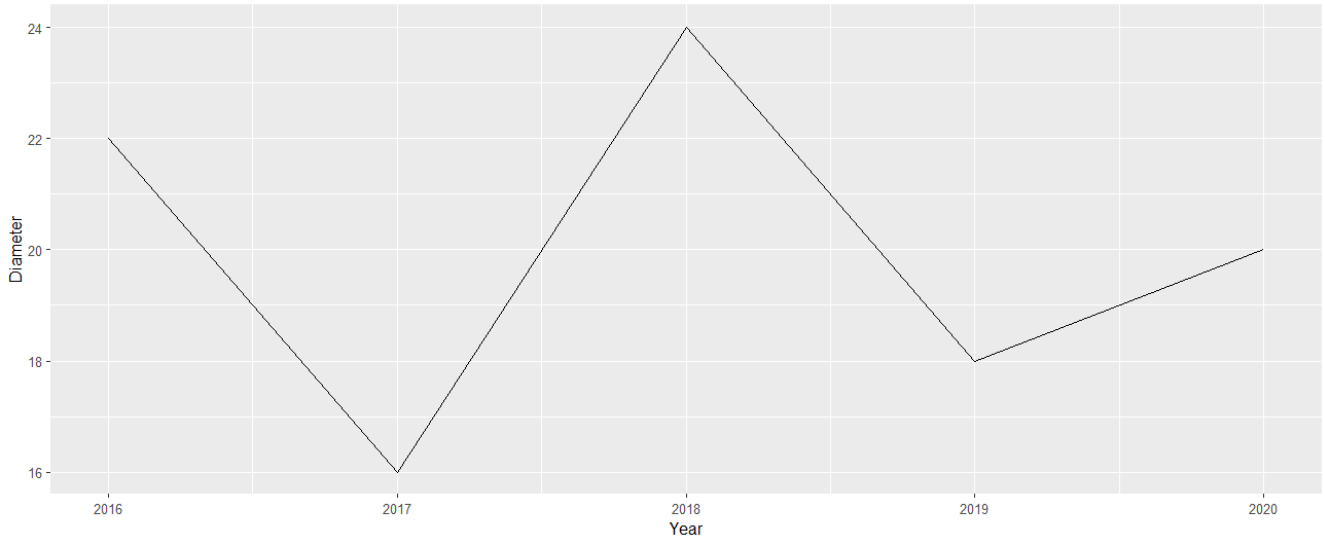
• Number of edges:



Plot 2.2 : The 5-year evolution of the number of edges of the graph

In this plot, we understood that through the years the number of edges increased in nearly same way as the number of vertices did. This is reasonable because the number of authors increased and consequently the cooperation between them would increase.

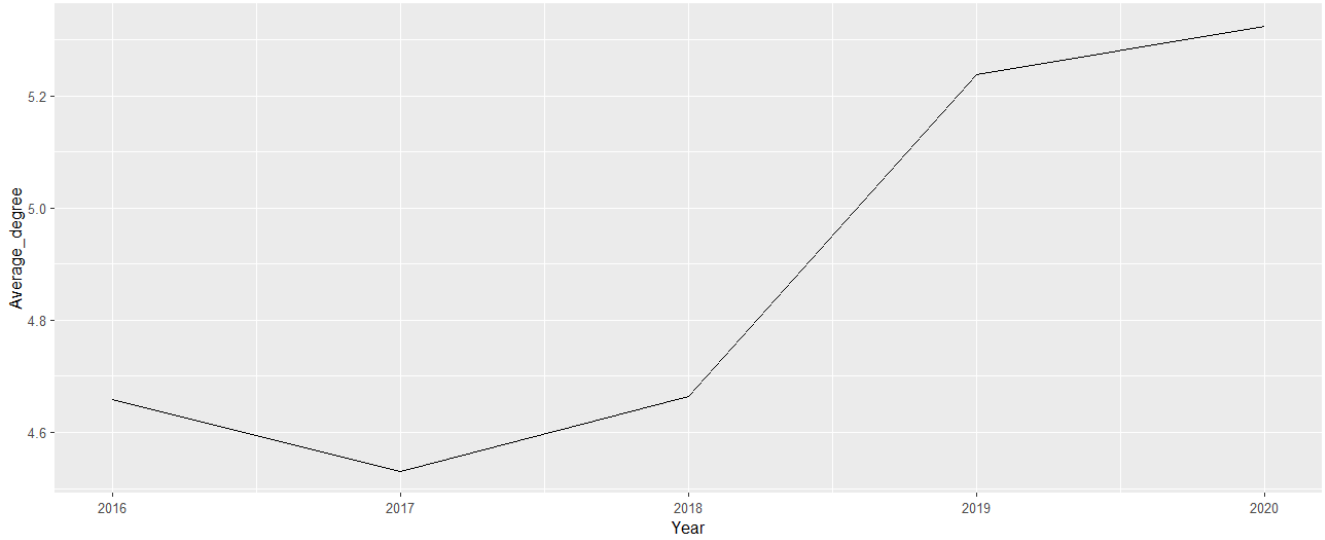
• Diameter of the graph:



Plot 2.3 : The 5-year evolution of the diameter of the graph

From the above plot, we understood that the distance between the two most distanced nodes in the graph was the highest in 2018.High values were in 2016 and 2020 but not such high as in 2018. The lowest distance was observed in 2017.

• Average degree (simple, not weighted):



Plot 2.4 : The 5-year evolution of the average degree of the graph

This plot illustrated the fluctuation of the average degree in the last 5 years. As it can be seen, the peak in this plot is observed in 2020 and the minimum in 2017.

Concluding, from this task we gained some insightful information regarding our project authors. We noticed that in 2020 ,where there was the highest number of authors (vertices), the average degree was close to 5.3. This indicated that most authors cooperated in small groups since their average degree was at a low level. This was also seen in 2019, which was another year with a high number of vertices but lower than that in 2020.

Task 3: *Important nodes*

At this task we were asked to write code to create and print data frames for the 5-year evolution of the top-10 authors with regard to:

• Degree (simple, not weighted):

As it is well known, the degree network metric is considered as the number of edges incident on a node. In our project, the degree can be described as the number of the authors that an author has cooperated in the publication of a paper. To find the top 10 authors for each year regarding their degree, we first calculated their degree and then sorted the result. The output is provided below.

The top-10 authors for 2016:

Philip S. Yu	Jiawei Han 0001	Hui Xiong 0001	Naren Ramakrishnan	Jieping Ye	Yi Chang 0001
46	41	39	32	32	31
Jiebo Luo	Rayid Ghani	Chang-Tien Lu	Yannis Kotidis		
29	28	25	25		

The top-10 authors for 2017:

Philip S. Yu	Jiawei Han 0001	Hui Xiong 0001	Yi Chang 0001	Claudio Rossi 0003	Clemens Mewald
44	42	38	32	32	31
Heng-Tze Cheng	Martin Wicke	Mustafa Ispir	Zakaria Haque		
31	31	31	31		

The top-10 authors for 2018:

Philip S. Yu	Jiawei Han 0001	Kun Gai	Wenwu Zhu 0001	Jing Gao 0004	Chao Zhang 0014	Jure Leskovec
70	37	35	28	27	27	27
Xing Xie 0001	Enhong Chen	Qi Liu 0003				
26	25	25				

The top-10 authors for 2019:

Philip S. Yu	Weinan Zhang 0001	Hui Xiong 0001	Jieping Ye	Jie Tang 0001	Jiawei Han 0001
69	59	49	41	39	37
Yong Li 0008	Enhong Chen	Jingren Zhou	Jian Pei		
36	36	35	35		

The top-10 authors for 2020:

Jiawei Han 0001	Hongxia Yang	Hui Xiong 0001	Xiuqiang He	Ji Zhang	Peng Cui 0001
69	43	42	41	40	39
Christos Faloutsos	Wei Wang 0010	Jieping Ye	Ruiming Tang		
38	38	37	35		

In order to notice variations on the top-10 lists for the different years, we created a dataframe. This dataframe contained all authors' degrees for each year illustrating the evolution that each author had. To do this , we decided to take the output from each year's degree calculation, convert it to a dataframe, and then merge all these five dataframes into one. Since the final dataframe contained the

names of all authors and not the top 10,we selected the authors that we were interested in. The whole process is analytically described in the R file named “2nd Assignment.R”. The final dataframe is the below:

Names	Year_2016	Year_2017	Year_2018	Year_2019	Year_2020
Chang-Tien Lu	25	10	5	4	2
Chao Zhang 0014	5	23	27	9	34
Christos Faloutsos	23	14	16	11	38
Claudio Rossi 0003	0	32	0	0	0
Clemens Mewald	0	31	0	0	0
Enhong Chen	16	15	25	36	26
Heng-Tze Cheng	0	31	0	0	23
Hongxia Yang	0	4	22	28	43
Hui Xiong 0001	39	38	0	49	42
Ji Zhang	0	0	0	4	40
Jian Pei	20	0	17	35	16
Jiawei Han 0001	41	42	37	37	69
Jie Tang 0001	19	9	10	39	13
Jiebo Luo	29	26	17	5	7
Jieping Ye	32	25	24	41	37
Jing Gao 0004	20	14	27	11	3
Jingren Zhou	0	0	7	35	16
Jure Leskovec	12	26	27	21	17
Kun Gai	0	6	35	23	27
Martin Wicke	0	31	0	0	0
Mustafa Ispir	0	31	0	0	0
Naren Ramakrishnan	32	15	9	10	0
Peng Cui 0001	11	10	18	34	39
Philip S. Yu	46	44	70	69	27
Qi Liu 0003	11	15	25	27	13
Rayid Ghani	28	0	20	0	2
Ruiming Tang	0	0	0	5	35
Wei Wang 0010	5	0	9	18	38
Weinan Zhang 0001	12	13	17	59	34
Wenwu Zhu 0001	10	3	28	26	5
Xing Xie 0001	13	10	26	18	7
Xiuqiang He	0	0	0	0	41
Yannis Kotidis	25	0	0	0	18
Yi Chang 0001	31	32	13	8	0
Yong Li 0008	0	13	8	36	32
Zakaria Haque	0	31	0	0	0

From the above results emerged that some authors were the same in the list of top 10 authors for the years 2016 to 2020. These authors were Philip S. Yu, Jiawei Han 0001, and Hui Xiong 0001 reaching the first three places four times in the five years. Moreover, the author that has consistently appeared in the first place was Philip S. Yu except for the last year of our analysis where he was replaced by Jiawei Han 0001.

• PageRank:

The same procedure was followed for the Pagerank. We calculated the Pagerank for each year‘s authors and sorted the result. The top 10 authors for each year of this metric were the below:

The top-10 authors for 2016:

Philip S. Yu	Hui Xiong 0001	Jiawei Han 0001	Jiebo Luo	Jieping Ye
0.0017288334	0.0014581015	0.0014119510	0.0013099364	0.0010027077
Yi Chang 0001	Hanghang Tong	Christos Faloutsos	Maarten de Rijke	Jiliang Tang
0.0009601005	0.0009272920	0.0009216757	0.0009158533	0.0009155034

The top-10 authors for 2017:

Philip S. Yu	Jiawei Han 0001	Hui Xiong 0001	Jure Leskovec	Jiebo Luo	Hanghang Tong
0.0014558956	0.0013585699	0.0010997688	0.0010681579	0.0009454158	0.0009285808
Jiliang Tang	Yi Chang 0001	Chao Zhang 0014	Ingmar Weber		
0.0007750644	0.0007711858	0.0007510406	0.0007208090		

The top-10 authors for 2018:

Philip S. Yu	Jiawei Han 0001	Jure Leskovec	Wenwu Zhu 0001	Chao Zhang 0014	Xing Xie 0001
0.0019816952	0.0009305425	0.0008756725	0.0007845882	0.0006777814	0.0006265688
Jing Gao 0004	Martin Ester	Yiqun Liu 0001	Kun Gai		
0.0006262191	0.0006203928	0.0006145961	0.0006132149		

The top-10 authors for 2019:

Philip S. Yu	Hui Xiong 0001	Weinan Zhang 0001	Jieping Ye	Hanghang Tong
0.0015871036	0.0009633261	0.0008767308	0.0007255196	0.0007021244
Jiawei Han 0001	Peng Cui 0001	Jie Tang 0001	Enhong Chen	Gerhard Weikum
0.0006855583	0.0006574207	0.0006517701	0.0006377621	0.0006257373

The top-10 authors for 2020:

Jiawei Han 0001	Hui Xiong 0001	Hongxia Yang	Elke A. Rundensteiner
0.0010753255	0.0007594661	0.0007284981	0.0006983864
Yong Li 0008	Jieping Ye	Peng Cui 0001	Xiuqiang He
0.0006821198	0.0006800497	0.0006533883	0.0006465968
Ji-Rong Wen	Jiliang Tang		
0.0006450074	0.0006423610		

The 5-year evolution of the top nodes:

Names	Year_2016	Year_2017	Year_2018	Year_2019	Year_2020
Chao Zhang 0014	0.00019	0.00075	0.00068	0.00016	0.00053
Christos Faloutsos	0.00092	0.00057	0.00056	0.00032	0.00052
Elke A. Rundensteiner	0.00024	0.00045	0.00041	0.00041	0.00070
Enhong Chen	0.00060	0.00041	0.00056	0.00064	0.00048
Gerhard Weikum	0.00074	0.00063	0.00034	0.00063	0.00000
Hanghang Tong	0.00093	0.00093	0.00056	0.00070	0.00048
Hongxia Yang	0.00000	0.00023	0.00059	0.00049	0.00073
Hui Xiong 0001	0.00146	0.00110	0.00000	0.00096	0.00076
Ingmar Weber	0.00056	0.00072	0.00032	0.00015	0.00020
Ji-Rong Wen	0.00000	0.00021	0.00015	0.00000	0.00065
Jiawei Han 0001	0.00141	0.00136	0.00093	0.00069	0.00108
Jie Tang 0001	0.00060	0.00030	0.00032	0.00065	0.00024
Jiebo Luo	0.00131	0.00095	0.00059	0.00015	0.00018
Jieping Ye	0.00100	0.00060	0.00060	0.00073	0.00068
Jiliang Tang	0.00092	0.00078	0.00046	0.00032	0.00064
Jing Gao 0004	0.00075	0.00045	0.00063	0.00024	0.00008
Jure Leskovec	0.00071	0.00107	0.00088	0.00045	0.00033
Kun Gai	0.00000	0.00021	0.00061	0.00037	0.00034
Maarten de Rijke	0.00092	0.00032	0.00053	0.00056	0.00032
Martin Ester	0.00042	0.00037	0.00062	0.00014	0.00035
Peng Cui 0001	0.00048	0.00037	0.00060	0.00066	0.00065
Philip S. Yu	0.00173	0.00146	0.00198	0.00159	0.00050
Weinan Zhang 0001	0.00054	0.00046	0.00035	0.00088	0.00045
Wenwu Zhu 0001	0.00041	0.00013	0.00078	0.00054	0.00008
Xing Xie 0001	0.00057	0.00029	0.00063	0.00041	0.00014
Xiuqiang He	0.00000	0.00000	0.00000	0.00000	0.00065
Yi Chang 0001	0.00096	0.00077	0.00033	0.00019	0.00000
Yiqun Liu 0001	0.00037	0.00037	0.00061	0.00057	0.00028
Yong Li 0008	0.00000	0.00042	0.00024	0.00062	0.00068

Concerning the results from Pagerank metric, we concluded that there was not a wide variation between the years. Again, Philip S. Yu reached the first place in the top 10 list apart from 2020 where he was replaced by Jiawei Han 0001. Consequently, we selected information about the importance of each author and how valuable they were.

Task 4: Communities

For this task, we wrote code to perform community detection on the graphs. We applied fast greedy clustering, infomap clustering, and louvain clustering on the 5 undirected co-authorship graphs. During the execution of our code, we did not face any problems. We might have a problem in the execution of fast greedy clustering if in our data we had multiple edges. These edges between two nodes (authors) represent pairs of authors that have written multiple publications together and the fast greedy clustering method cannot work in such type graphs. Due to the cleaning process, we have followed, we might have extracted such records that consisted of multiple edges.

To detect the performance of each algorithm , we used some performance parameters such as number of communities, modularity and execution time. Regarding modularity metric, it is known that it measures the community strength. The best community is the one that obtains the maximal modularity. The results for each method are provided in the below tables.

Year	*Fast Greedy Clustering*	*Infomap Clustering*	*Louvain Clustering*
2016	680	744	680
2017	820	891	820
2018	868	961	872
2019	964	1111	964
2020	943	1108	942

Table 4.1 : The performance of community detection algorithms in terms of number of communities (C)

Year	*Fast Greedy Clustering*	*Infomap Clustering*	*Louvain Clustering*
2016	0.98	0.96	0.98
2017	0.98	0.97	0.99
2018	0.98	0.96	0.98
2019	0.97	0.94	0.98
2020	0.96	0.93	0.97

Table 4.2 : The performance of community detection algorithms in terms of modularity

Fast Greedy Clustering	0.39 secs
Infomap Clustering	7.98 secs
Louvain Clustering	0.14 secs

Table 4.3 : The performance of community detection algorithms in terms of execution time in seconds

Taking into consideration the results from the performance parameters, we decided to use Louvain clustering which calculates the communities rapidly and with a higher performance. This method works in two steps. At first, it evaluates small communities by local modularity optimization and as a second step it aggregates the nodes that belong to same community. It recursively merges communities into a single node until maximum modularity is attained. Furthermore, this algorithm cannot generate overlapping communities.

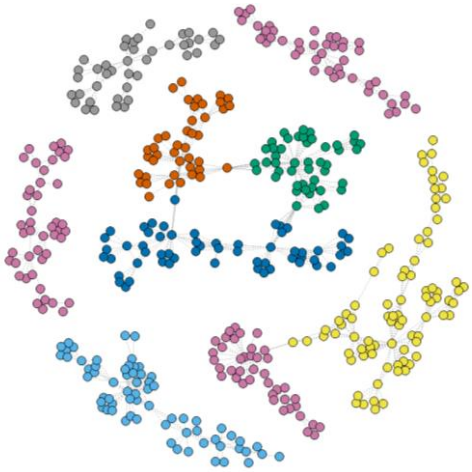
In order to detect the evolution of the community a random author belonged , we executed a code that selected an author who appeared in all five years. Then, in the communities of the random author, we calculated the number of common authors they had through the years. The author that was randomly selected was " Meng Jiang 0001" and the results are presented in the following table.

Years	*Number of Common Authors*	*Common Authors*
2016-2017	8	"Christos Faloutsos", "Bryan Hooi", "Kijung Shin", "Peng Cui 0001", "Shiqiang Yang","Tianyang Zhang", "Wenwu Zhu 0001", "Chengxi Zang"
2016-2018	0	-
2016-2019	0	-
2016-2020	0	-
2017-2018	14	"Chao Zhang 0014" , "Jiawei Han 0001","Jingbo Shang" ,"Quan Yuan 0001","Meng Qu","Heng Ji", "Wenqi He","Xiang Ren 0001,""Dongming Lei", "Timothy P. Hanratty","Carl Yang","Xinhe Geng", "Zeqiu Wu", "Yu Shi"
2017-2019	0	-
2017-2020	4	"Yanfang Ye", "Yiming Zhang 0002","Shifu Hou", "Yujie Fan"
2018-2019	2	"Yiyu Shi" , "Tong Zhao 0003"
2018-2020	2	"Xiusi Chen" , "Tong Zhao 0003"
2019-2020	10	"Chao Huang 0001", "Nitesh V. Chawla", "Suwen Lin" , "Xian Wu", "Chuxu Zhang", "Qingkai Zeng" , "Wenhao Yu","Tianwen Jiang","Daheng Wang", "Tong Zhao 0003"

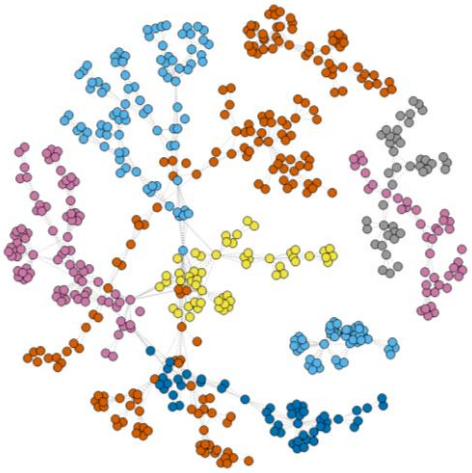
Table 4.4 : Evolution of the communities that the random author belonged

As it can be understood from the above table, the communities contained some common authors. Specifically, the most common nodes were seen in the period 2017 to 2018 with 14 neighbors and between the years 2019-2020 with 10. The third place was reached by the period 2016-2017 with 8 similar authors. In the other years, there were some common vertices but not as high as in the aforementioned years. The periods 2016-2018,2016-2019 and 2016-2020 had no similarity.

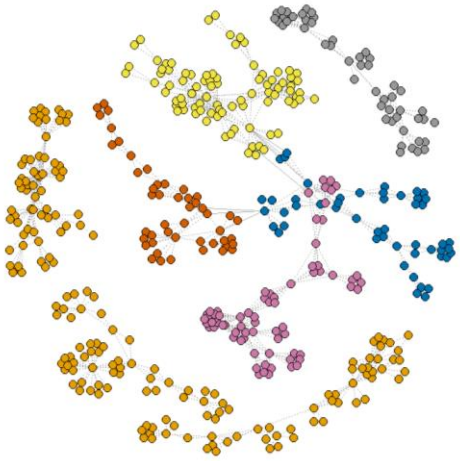
After completing the analysis in the communities , we visualized the graph for each year using different color for each community. In order to create aesthetically pleasing visualizations we kept the communities that their size was higher than 40 and lower than 90.



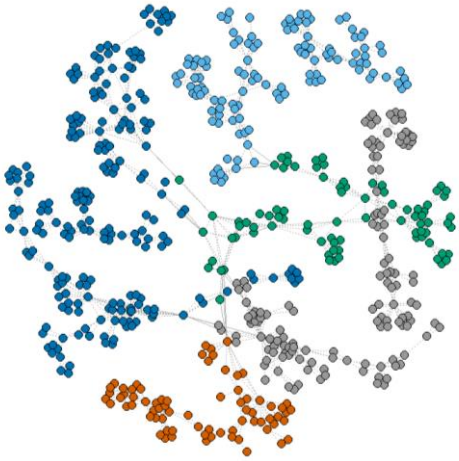
Plot 4.1 : Communities of 2016



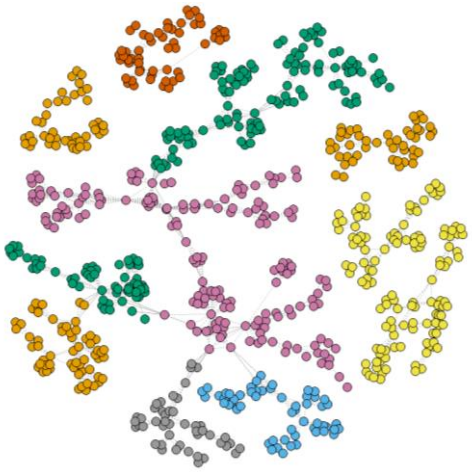
Plot 4.2 : Communities of 2017



Plot 4.3 : Communities of 2018



Plot 4.4 : Communities of 2019



Plot 4.5 : Communities of 2020

References

1. Prajakta Vispute, Shirish Sane,(2020) “Performance Evaluation of Community Detection Algorithms in Social Networks Analysis”, Biosc.Biotech.Res.Comm