Athens University of Economics and Business
M.Sc. Program in Business Analytics
Course: Social Network Analysis – Spring 2024
Instructor: Dr Katia Papakonstantinopoulou

Project 2:

From raw data to temporal graph structure exploration

Due 11:59pm EEST July 28, 2024

General Instructions

Your answers should be as concise as possible.

Submission instructions: You should submit a compressed directory, containing your answers and code, via `https://e-mscba.dmst.aueb.gr`.

Submitting answers: Prepare a report with your answers on this project in a single PDF file named *p2.pdf*

Submitting code: Prepare the source file(s) with your code.

Problem

1 DBLP co-authorship graph

Your first task is create a weighted undirected graph with igraph,[1] using raw data from dblp. You will download a compressed file with conference proceedings records listed in dblp. [2] The format of the file is the following:

```
2003,"The link prediction problem for social networks.",CIKM,"David Liben-Nowell,Jon M. Kleinberg"
2006,"Sampling from large graphs.",KDD,"Jure Leskovec,Christos Faloutsos"
2010,"Measuring User Influence in Twitter.",ICWSM,"M. Cha,H. Haddadi,F. Benevenuto,P. Krishna Gummadi"
2016,"Do Cascades Recur?",WWW,"Justin Cheng,Lada A. Adamic,Jon M. Kleinberg,Jure Leskovec"
2016,"Big data security and privacy.",IEEE BigData,"Elisa Bertino"
```

In each line above, the first column indicates the year the paper was published, the second column is the title of the paper, and the third column is the conference where the paper was presented. Finally, the fourth column of the line is a comma separated list of the paper's authors.

[1] https://igraph.org/r/

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

You will first manipulate the raw data with the programming language of your choice to filter out all records that are not related to the five (5) conferences listed above, e.g, CIKM, or are older than 5 years. Then, you will create a total of 5 .csv files, one for each of the last 5 years, using the following format:

```
from,to,weight
author1,author2,5
author1,author3,2
...
```

Each .csv file should describe the weighted undirected co-authorship graph for the respective year, e.g., in the example above *author1* has co-authored 5 papers with *author2*, and 2 with *author3*.

Having created the .csv files it will be trivial to use them and create the respective igraph graphs.

Your submission should include the code you used to create the .csv files (any programming language), the code you used to create the igraph graphs (R) and the 5 (compressed) .csv files.

2 Average degree over time

Your next task is to create plots that visualize the 5-year evolution of different metrics for the graph. More specifically, you will create plots for:

- Number of vertices
- Number of edges
- Diameter of the graph
- Average degree (simple, not weighted)

What do you notice for each of the 4 above metrics? Are there significant fluctuations during these five years?

3 Important nodes

Next, you will write to code to create and print data frames for the 5-year evolution of the top-10 authors with regard to:

- Degree (simple, not weighted)

- PageRank

Again, provide short comments on your findings. Do you notice variations on the top-10 lists for the different years?

4 Communities

Your final task is to perform community detection on the co-authorship graphs. Try applying *fast greedy* clustering, *infomap* clustering, and *louvain* clustering on the 5 undirected co-authorship graphs. Are you able to get results with all methods? Include a short comment on your report regarding the performance of the 3 algorithms.

Then, pick one of the three methods as well as a random author that appears in all 5 graphs and write code to detect the evolution of the communities this user belongs to. Do you spot similarities in the communities?

Finally, you will create a visualization of the graph using a different color for each community. Make sure to have a look at the sizes of the communities and filter out all nodes that belong to very small or very large communities, in order to create a meaningful and aesthetically pleasing visualization.

5 Hints

Managing large volume data can be troublesome if you do not follow an appropriate approach. Below you can find some tips on:

1. how to process large files, and
2. how to extract the desired information using regular expressions.

Moreover, while developing your solution you could use just a sample of the provided dataset. Such a sample can be easily created using command line utilities. The following example writes the first 1,000 lines of a compressed file named authors.csv.gz that match a particular pattern to a new file named authors-sample.csv:

```
$ zcat authors.csv.gz | grep ",CIKM," | head -n 1000 > authors-sample.csv
```

5.1 Parsing large files

You can parse compressed files directly with R. However, you must be careful not to load the entire content of the file in memory Instead, you should focus on reading the file line by line. Below you can find code written in R that parses one line at a time and examines whether a paper was presented at a particular conference:

```r
inputFile <- "authors.csv.gz"
con  <- file(inputFile, open = "r")
while (length(oneLine <- readLines(con, n = 1)) > 0) {
  columns <- scan(text=oneLine, sep=',', what='character', quiet=TRUE)
  if (!is.na(columns[3]) && columns[3]=="CIKM") {
    print(columns[4])
  }

}
close(con)
```