

INFS7450 Social Media Analytics

Project 1 – Fast Computation of User Centrality Measures

Semester 1, 2020

Marks:	15 marks (15%)
Submission Due:	23 Apr 20 23:59 (Brisbane Time)
Deliverables:	See deliverables part
How to submit:	Electronic submission via Blackboard

Goal: The purpose of this project is to help students gain practical experiences and understand the concepts of various centrality measurements for social networks.

Dataset: In this project, you will be working with the public available Facebook social network data. The Facebook data has been anonymized by replacing the Facebook-internal ids for each user with a new value. The data contains 4039 nodes, 88234 edges in total. Each line of the data represents an undirected link starting from one node to another.

The dataset is available from UQ blackboard. See /Assessment/INFS7450 Project One.

Tasks:

1. Calculate the Betweenness Centrality for nodes in the Facebook dataset. **(8 marks)**
Overview: write code to load the Facebook social network data and construct an undirected and unweighted graph. Based on the constructed graph, you are required to write a program to calculate the betweenness centralities for the graph vertices.
Input: The provided Facebook social network data.
Output: The top-10 nodes with the highest betweenness centralities.
Requirements: You may use third-party libraries, such as NetworkX to read, load and manipulate the Facebook network dataset. However, you must write your own code to implement the function of node centrality calculation rather than using the third-part or built-in functions. (You can use any functions in NetworkX other than the functions for centrality calculation.)
2. Calculate PageRank Centrality for nodes in the Facebook dataset. **(7 marks)**
Overview: write code to load the Facebook social network data and construct an undirected and unweighted graph. Based on the constructed graph, you are required to write a program to calculate the PageRank (with $\alpha = 0.85, \beta = 0.15$) centralities for the graph vertices.
Input: The provided Facebook social network data.
Output: The top-10 nodes with the highest PageRank centralities.
Requirements: You may use third-party libraries, such as NetworkX to read, load and manipulate the Facebook network dataset. However, you must write your own code to implement the function of node centrality calculation rather than using the third-part or built-in functions. (You can use any functions in NetworkX other than the functions for centrality calculation.)

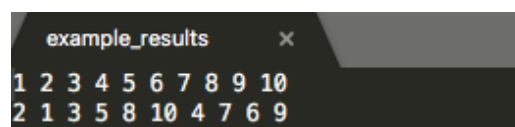
Programming Languages:

Python and NetworkX are recommended. However, you have your own choices of preferred programming languages including, but not limited to, Python, MATLAB, Java, C, C++, etc.

Deliverables:

Your submission must include the following:

1. A report (.pdf). See the given appendix for an example template.
2. A source code file.
3. A results text file. The file must contain two lines of results. The first line is the top-10 nodes based on the calculated betweenness centralities. The second line is the top-10 nodes based on the calculated PageRank centralities. (each item in each line should be separated by a space.) See the following picture as an example.



```
example_results x
1 2 3 4 5 6 7 8 9 10
2 1 3 5 8 10 4 7 6 9
```

Figure 1. the example format of the results

4. Name all the submitted files by using your student ID. For example, 41234567.py for the source code, 41234567.txt for your submitted results, and 41234567.pdf for your report.
5. Submit one archive file with your student number as the file name (e.g. 12345678.zip) with all the files and folders mentioned above.

Marking criteria (Total marks: 15):

- Task 1: 8 marks = 3 marks (code) + 3 marks (results) + 2 marks (report)
- Task 2: 7 marks = 2 marks (code) + 3 marks (results) + 2 marks (report)
- Your results should be reproducible and your codes should be readable. If your codes cannot be executed or generate the results as reported, the corresponding marks for the code and results will be deducted.
- We will evaluate your submitted results via calculating the Jaccard Similarity between the submitted results and the ground truth. That means your mark for each task will be calculated by:

$$\text{Result Mark} = \text{Jaccard Similarity (Submitted Results, Ground Truth)} * 3$$