LINFO1115: Project –The Social Network Analysis

Peter VAN ROY Lucile DIERCKX

Academic Year: 2022 – 2023

Context

How well do the message interactions between students represent the social behavior of university students? Is it possible to retrieve any specific structure and observe some known social evolutions from the message interaction? These are some questions you will investigate in this project.
To do so, you will be working with the CollegeMsg temporal network, which is a dataset comprised of private messages sent on an online social network at the University of California. Users could search the network for others and then initiate a conversation based on profile information. The dataset also provides temporal information about when were the messages sent.

1 Format

Every entry of the dataset represents the sending of a private message. There are three columns in the dataset: `Src`, `Dst`, and `Time`. A line of the dataset `(u, v, t)` means therefore that user `u` sent a private message to user `v` at time `t`.
Integers `u` and `v` represent unique student id while integer `t` represents the Unix-based timestamp of the message.

2 In practice

For this project, you have to use Python language. (Note that if the code does not compile when we run it, you will not succeed in the project). We request you to implement all algorithms seen during the class *by yourself*, without using any library that implements it already. You may of course use the available implementation, but for validation purposes only.
Also, this assignment must be completed by **group of two students or alone (no group of three)**. We deleted the groups made for the previous project, now you don't have to register in a Moodle group anymore, you just have to put both names on the report. If you do not manage to find a partner, please start by requesting a partnership via the Moodle forum. Then, **if and only if** you cannot find anyone else to work with, get in touch with us.
Unless specified otherwise, for each of the items below, we request you to explain your quantitative results. For each algorithm, indicate also the temporal complexity in your

report. This complexity should be expressed using the number of vertices v and/or the number of edges e.
You will have to submit your code on Inginious (`https://inginious.info.ucl.ac.be/course/LINFO1115`) (a code template is available on Moodle with some details about what should be returned for each question) and you will have to write a report to analyze the obtained results.

2.1 Task 1: Required to pass

For this exercise, consider the graph as an undirected graph. If you find multiple transactions between the same two nodes, consider just one of them.

The bare minimum you need to do in order to have a chance at passing this project is to count:

1. The number of different components in the graph (do not hesitate to look at the size of each of them even if you don't have to return that value on Inginious, it might still help your interpretation of the network).
2. The number of bridges in the graph.
3. The number of local bridges in the graph.

How can you interpret those numbers in the context of social networks?

2.2 Task 2: If you aim at 12/20

For this exercise, consider the graph as an undirected graph. If you find multiple transactions between the same two nodes, consider only the oldest one according to the timestamp.

Compute the number of triadic closures that have appeared between the median timestamp and the end. Create a graph of the accumulated number of triadic closures over time. Can you interpret it in the context of social networks?

2.3 Task 3: If you aim at 14/20

For this exercise, you have to consider the graph as a directed graph and do not consider the timestamp. If you find multiple transactions between the same two nodes, consider just one of them.

We would like to check whether the small world phenomenon is also observable in the biggest component of this network. Therefore you have to measure the distance of the shortest path between each pair of nodes that are part of this component. Then generate a graph of the number of paths having a given distance (number vs length), you can find an example of such a graph in the slides of the first course. Interpret the obtained measurements.

2.4 Task 4: If you aim at 17/20

For this exercise, you have to consider the graph as a directed graph and you should not use the timestamp.

We want to measure the importance of each node in the directed graph. Therefore, we want to compute the PageRank score of each person in the network. The common PageRank (PR) score of a node p is usually computed recursively as

$$PR(p) = \frac{(1-d)}{N} + d \sum_{n \in B(p)} \frac{PR(n)}{Nout_n}$$

where N is the total number of nodes in the graph, $B(p)$ is the set of nodes pointing to p, $Nout_n$ is the number of outgoing links of node n and d is the damping factor, having a value of 0.85.

We ask you to compute the PageRank score of each node. Give the id number of the node having the highest weighted PageRank score and indicate its value.

3 Task 5: If you aim at 20/20

For this exercise, you have to consider the graph as an undirected graph and you should not use the timestamp.

We want to measure how close are the friends of a given person with each others, therefore we want to compute the local clustering coefficient of every node. The local clustering coefficient of a node is computed as:

$$LCC(u) = \frac{2 \cdot T(u)}{deg(u) \cdot (deg(u) - 1)}$$

Where $T(u)$ is the number of triangles in which u is present (i.e. the number of connexions between nodes to which u is also connected).

We ask you to then compute the average local clustering coefficient of the graph.

4 Deliverables

Your rapport should be **maximum 4 pages** long, should have the following structure, and should contain the following information:

Name and NOMA of the two students

1. Task 1
    1.1. Results: Indicate the number of components, the number of bridges, and the number of local bridges.
    1.2. Interpretation: Interpret the obtained values in the context of social networks.
    1.3. Complexity: Indicate the time complexity of your algorithms.

2. Task 2

    2.1. Result: Give the total number of triadic closures you found for the defined time interval.

    2.2. Graph: Show the graph of the accumulated number of triadic closures over time. Interpret.

    2.3. Complexity: Indicate the time complexity of your algorithm.

3. Task 3

    3.1. Result: Indicate the longest path you found and the id number of the two nodes concerned.

    3.2. Graph: Show the graph of the number of paths having a given distance. Interpret.

    3.3. Complexity: Indicate the time complexity of your algorithm.

4. Task 4

    4.1. Result: Indicate the highest PageRank score you found and the id number of the related node.

    4.2. Complexity: Indicate the time complexity of your algorithm.

5. Task 5

    5.1. Result: Give the average value of the local clustering coefficient of the graph.

    5.2. Complexity: Indicates the time complexity of your algorithm.

6. Appendix: Give your code in appendix

5 Deadline

The assignment is due by **Friday, May 5th 2023 at 23:59**. The code should be submitted on Inginious (`https://inginious.info.ucl.ac.be/course/LINFO1115`) by the same deadline. The report must be handed in on Moodle as a zip file containing both your report and your complete source code. Only one student per group of two should submit on Inginious and on Moodle (preferably the same student for both).