SINF1115: Project –The Bitcoin analysis

Peter VAN ROY Lucile DIERCKX Jean GILLAIN

Academic Year: 2020 – 2021

Context

With all the disasters that are happening in the world, the value of money has become more and more unstable. To counteract this, many people have started to invest in bitcoin (be careful, the investment is not without risk).

For this work, you will be working with the Bitcoin OTC trust weighted signed network, which is a network of people who trust each other and exchange bitcoins on a platform called Bitcoin OTC. Because Bitcoin users are anonymous, it is necessary to keep a record of users' reputations in order to avoid transactions with fraudulent and risky users. Bitcoin OTC members rate other members on a scale of -10 (completely distrusted) to +10 (completely trusted) in increments of 1.

1 Format

The dataset contains one type of unique entity: User id ("uid"): a unique integer identifying each person.

Beside that, there are two additional metadata:

- Timestamp: denotes the Linux "timestamp" of a specific rating from user x to y.
- Rating: indicates the level of reliability between user x and y.

The dataset contains one "csv" file:

- "data.csv" : row format <uid buyer(source of rating)> , <uid seller(receiver of the rating)>, <rating> , <timestamp> .

2 In practice

For this project, you have to use the 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**. 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 explicit your quantitative results and then explain how you implemented the required algorithm. 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.

2.1 Task 1: Required to pass

For this exercise, do not take into consideration the rating of the links but just the fact that a transaction happened or not. You also have to 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.
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 Bitcoin trading?

2.2 Task 2: If you aim at 12/20

For this exercise, do not take into consideration the rating of the links but just the fact that a transaction happened or not. You also have to 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 Bitcoin trading?

2.3 Task 3: If you aim at 14/20

For this exercise, consider the graph as an undirected graph. If you find multiple transactions between the same two nodes, take into account the most recent one, that happened before the timestamp you are currently computing.

We want to have an overview of the degree of structural balance for our graph. We define therefore a new metric that gives an indication of the global structural balance. To do so, we look at the proportion of balanced triangles, weakly balanced triangles and unbalanced triangles. As weakly balanced triangles represent a weaker equilibrium than balanced triangles, we reduce the weight of the weakly balanced triangles. Our final formula for the degree of structural balance is:

$$balance\ degree = \frac{\#\ balanced\ triangles + 2/3 \cdot \#\ weakly\ balanced\ triangles}{Total\ number\ of\ triangles}$$

Compute the balance degree of the graph over time, starting at the median timestamp until the end. Create the graph of the balance degree over time. Can you interpret it in the context of Bitcoin trading?

2.4 Task 4: If you aim at 17/20

For this exercise, do not take into consideration the rating of the links but just the fact that a transaction happened or not. This time, however, you have to consider the graph as a directed graph. 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.

2.5 Task 5: If you aim at 20/20

For this exercise, if you find multiple transactions between the same two nodes, consider the one having the highest absolute value of rating

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. As we have weighted links, we will use a variation of the usual PageRank algorithm. The common PageRank (PR) score of a node p is usually computed recursively as

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

where $B(p)$ is the set of node 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.
The weighted PageRank (W_PR) is computed recursively as

$$W\_PR(p) = (1-d) + d \sum_{n \in B(p)} \frac{W\_PR(n) * W_{np}}{\sum W_n}$$

where W_{np} is the weight of the link going from node n to node p and $\sum W_n$ is the sum of the weights of all outgoing links of node n.

The pseudo-code to recursively compute the W_PR score is the following:

Initialisation: (assume the network has m nodes)
  ∀p: W_PR(p) = 1/m
Iteration: (repeat until convergence)

- Compute the amount flowing out each edge from node k to node l:
  $W\_PR_{link}(k,l) = W\_PR(k) * W_{kl} / \sum W_k$

- Sum the values of all incoming edges of node p and add damping factor: $W\_PR(p) = (1 - d) + d \sum_{n \in B(p)} W\_PR_{link}(n, p)$

We ask you to compute the weighted PageRank score of each node using the absolute value of the rating as a weight. Give the id number of the node having the highest weighted PageRank score and indicate its value. Discuss also the fact of considering the absolute value of the weight without taking into account the fact that it is a good or bad review. Can you make a parallel to another real-world situation where this applies?

3 Deliverables

Your rapport should be **maximum 4 pages** long (without counting the code in appendix), 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 Bitcoin context.
   - 1.3. Complexity: Indicate the time complexity of your algorithms.
2. Task 2
   - 2.1. Result: Give the 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: Give the highest balance degree you found for the defined time interval. Indicate the related timestamp.
   - 3.2. Graph: Show the graph of the balance degree over time. Interpret.
   - 3.3. Complexity: Indicate the time complexity of your algorithm.
4. Task 4
   - 4.1. Result: Indicate the longest path you found and the id number of the two nodes concerned.
   - 4.2. Graph: Show the graph of the number of paths having a given distance. Interpret.
   - 4.3. Complexity: Indicate the time complexity of your algorithm.
5. Task 5

5.1. Result: Indicate the weighted PageRank score you found and the id number of the related node.

5.2. Discussion: Discuss the choice of absolute value for the weights.

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

6. Appendix: Give your code in appendix

4 Deadline

The assignment is due by Friday, April 29th 2022 at 23:59. It must be handed in on Moodle as a zip file containing both your report and your complete source code.