Programming assignment 1: Network Measures

Saurabh Burewar (B18CSE050)

Contents

Introduction

This assignment includes implementation of basic network measures using a simple dataset from the snap library. I have used the “Bitcoin alpha web of trust network” in this assignment. The network is created from the dataset and a few important network measures are calculated using the NetworkX library in Python.

Dataset

The dataset gives a who-trusts-whom network of people who trade using Bitcoin on a platform called Bitcoin Alpha. Members of Bitcoin Alpha rate other members on a scale of -10 to 10 (in steps of 1) which denotes the amount of trust there is between the members.

The dataset consists of a CSV file compressed to GZ. The data format is as follows -

Source	**Target**	**Rating**	**Time**
7188	1	10	1407470400

- **Source** - Node id of the source (rater)
- **Target** - Node id of the target (one who is being rated)
- **Rating** - The rating (can be on a scale of -10 to 10 in steps of 1)
- **Time** - The time of the rating, measured in seconds since Epoch.

Link to the dataset: http://snap.stanford.edu/data/soc-sign-bitcoin-alpha.html

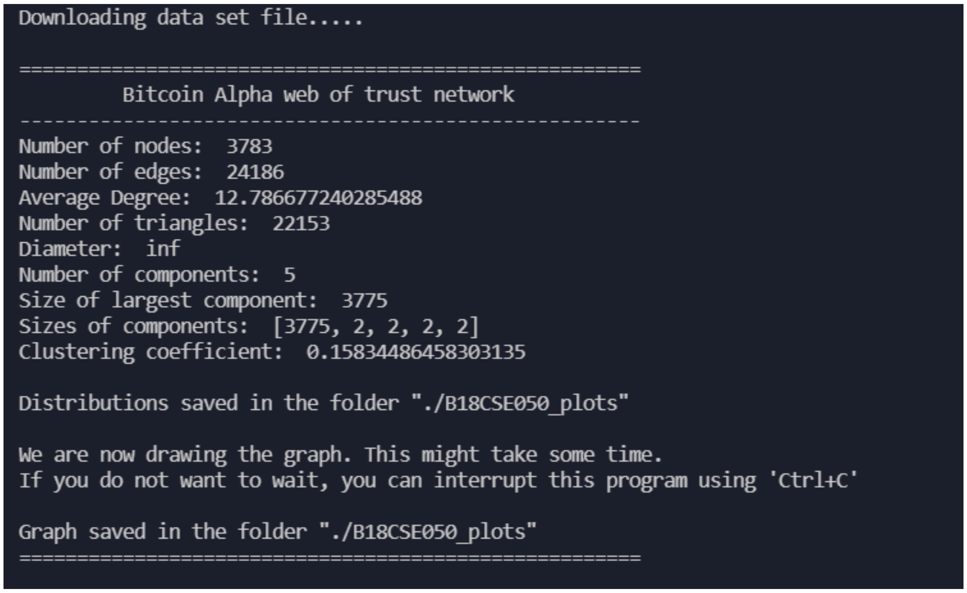
Solution

The solution code for this assignment is written in Python and is uploaded as “B18CSE050.py”. It uses the NetworkX library to build and work with the network and each section of the assignment is solved as a separate function. The code is commented using python docstring to make understanding easier.

Make sure to have the following libraries installed on your system -

- NetworkX
- Matplotlib
- Requests

- Gzip



```
Downloading data set file.....

=====================================================
         Bitcoin Alpha web of trust network
-----------------------------------------------------
Number of nodes:  3783
Number of edges:  24186
Average Degree:  12.786677240285488
Number of triangles:  22153
Diameter:  inf
Number of components:  5
Size of largest component:  3775
Sizes of components:  [3775, 2, 2, 2, 2]
Clustering coefficient:  0.15834486458303135

Distributions saved in the folder "./B18CSE050_plots"

We are now drawing the graph. This might take some time.
If you do not want to wait, you can interrupt this program using 'Ctrl+C'

Graph saved in the folder "./B18CSE050_plots"
=====================================================
```

Solution: Executed code

Data set loading

Downloading the data set

The Python "requests" library is used to request and download the data set file from the snap url.

Preparing the data set

The data set file is compressed in GZ format. So, we extract the file using the Python "gzip" library which gives us the CSV file containing the data.

Creating the graph

The NetwrokX library is used to create a directed graph. The CSV file is read one row at a time. For each row, we add the nodes with the corresponding ids and add the edges with the rating as the weight.

The graph is saved in PNG format in the directory named "B18CSE050_plots" in root.

Node count, Edge count, Average degree

Node count

The NetworkX library maintains a set of nodes. The length of this set gives the number of nodes present in the network.

This data set contains 3,783 nodes.

Edge count

The NetworkX library maintains a set of edges. The length of this set gives the number of edges present in the network.

This data set contains 24,186 edges.

Average degree

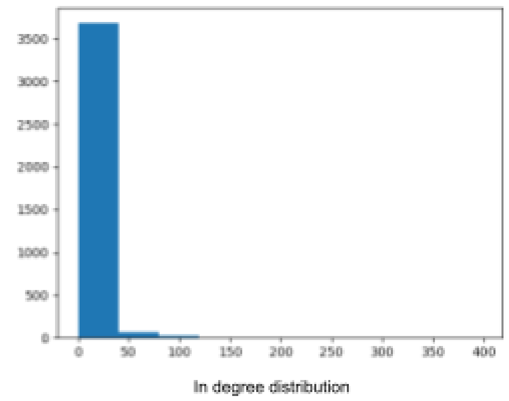
The NetworkX library provides a function that returns the degree of a node. Iterating this for all nodes and adding them gives the sum of the degrees of all nodes. Dividing this by the number of nodes gives the average degree.

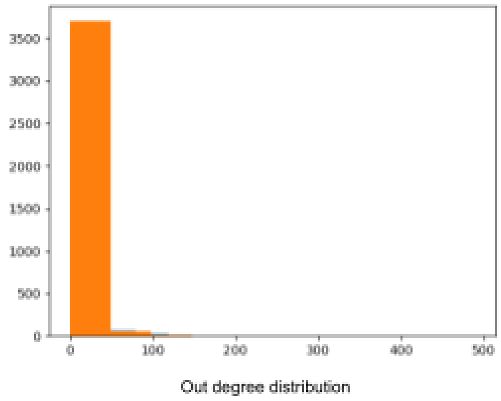
This network has an average degree of 12.787.

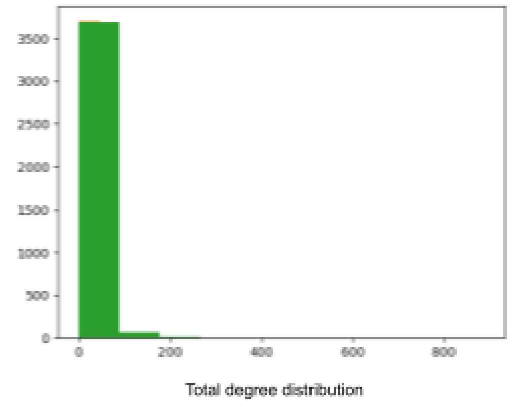
Degree distribution

Since we are working with a directed graph, we can have 3 degree distributions which can be useful, the in-degree, out-degree and total degree distribution. The NetworkX library provides functions to find in-degree, out-degree and degree of a node. We iterate through all nodes and maintain a list of in-degree, out-degree and degree of all nodes. Then we plot distributions of these lists.

The distributions are saved in PNG format in the directory named “B18CSE050_plots” in root.







Number of triangles

The NetworkX library provides a function that returns a dictionary with nodes and keys and number of triangles as values. Sum of values in this dictionary will give us 3x the number of triangles in the network because every triangle is counted thrice for each of its nodes. So, the number of triangles is given by the sum of values divided by 3.

The number of triangles in this network is 22,153.

Diameter

The NetworkX library provides a function that gives the diameter of the graph. In our case, the network is not strongly connected which gives the diameter as infinite.

Number of components

The NetworkX library provides a function which gives the number of components in a network. Since this function is only for undirected graphs, we convert our graph to undirected.

The number of components in this network is 5.

Size of largest component

The NetworkX library provides a function which gives the list of nodes in every component. The length of these lists gives the size of every component, the maximum of which is the size of the largest component.

The size of the largest component in this network is 3,775.

Clustering coefficient

The NetworkX library provides a function that gives the clustering coefficient for the graph.

The clustering coefficient for this network is 0.158.

References

- NetworkX documentation - https://networkx.org/documentation/stable/reference/index.html
- Mathsight degree distribution - https://mathinsight.org/degree_distribution
- Stack Overflow (for some unexpected errors) - https://stackoverflow.com/