Management and Content Delivery for Smart Networks: Algorithms and Modeling

Lab 1: Graph Analysis

Ex 1: Characterizing basic properties of a graph

This is a warm-up exercise. The goal is get used to NetworkX for graph analysis. You will read a file describing a graph and extract basic characteristics of the graph. You will then compare the given graph to classic random graph models.

The input file consists of the activity of Italian influencers on Instagram. For each line you will find the time an influencer has posted something on Instagram, the login name of the influencer and the login name of another person cited by the influencer in her/his post.

For instance, in the following you see five citations in posts of Chiara Ferragni:

```
1549626339 chiaraferragni financialtimesfashion
1549626339 chiaraferragni financialtimes
1549648329 chiaraferragni beautybites
1548607728 chiaraferragni naliannalisa
1548607728 chiaraferragni fedez
```

First assume that this graph is undirected. Study:

- The number of nodes and edges.
- The node degree distribution – plot the CCDF, and calculate the mean degree.
- The clustering coefficient of this graph.
- The size of the giant component if it exists.

Now reload the graph as a directed graph:

- Characterize the strongly connected components of this graph. How many of them do you see? Focus on the largest strongly connected component and characterize it (diameter, radius etc).
- Try the same for the largest weakly connected component. Explain the outcome.

Describe how the above graphs relate to the results seen in the slides for RG models.

Ex 2: Simulating the propagation of information in networks

Lets simulate the propagation of information in the undirected graph used in the previous exercise.

Assume the graph is static, thus connections never change over time. Assume that in the time step $t = 0$ an initial set of nodes becomes *active* – i.e., they have been reached by a particular *message*.

The system evolves in discrete steps. In time $t = n$ a number of neighbors of the nodes active in time $t = n - 1$ can become active too – i.e., they will be reached by the particular message coming from the active neighbors, and will become active spreaders of the message in the future.

Plot the *survival function* over time, defined as the proportion of nodes that is inactive in time $t = n$.

Note that you have some parameters to simulate, in particular the initial percentage of active nodes and the probability that a node will become active when neighboring an active node. For example, you could assume that a node will also become active when at least a percentage p of its neighbors is active.

Study how the survival function is affected by these parameters by running the simulation in different scenarios.

Use SimPy to simulate the evolution of the system. Note that this exercise is a very simple simulation, as time advances in constant steps (defined by you). Implement functions to handle the initial system state, events changing the nodes status and a stop condition. Run the simulation a number of times and calculate confidence intervals for the simulated survival.

Groups and Final Reporting

You are expected to work on groups of up to three students. Each group is required to prepare a **single** report describing results of **all labs in the course**. This report must not exceed 10 pages.

You need to delivery both the written report and your source code before the end of exam session in September.

References

[1] NetworkX Tutorial: https://networkx.github.io/documentation/stable/tutorial.html
[2] NetworkX Algorithms: https://networkx.github.io/documentation/stable/reference/algorithms/
[3] SimPy in 10 Minutes. `https://simpy.readthedocs.io/en/latest/simpy_intro/`
[4] matplotlib. `http://matplotlib.org/`