Assignment 4

Submission deadline: February 1, 2022 (submission via OPAL or alternatively via email to torsten.heinrich@wiwi.tu-chemnitz.de).

Background

Scott Feld's friendship paradox states that in social networks, "our friends have (on average) more friends than we do." More generally, a random person's friends will, on average, have more friends than that random person.

The phenomenon makes it possible to identify more well-connected people with little information about the network: Choose a random person, ask them to name a friend. That firend will, on average be more well-connected than the first random person.

Feld's friendship paradox has applications in marketing (target well-connected people first), in tracking of epidemics (well-connected people are likely to catch it first), as well as immunization.[1].

Problem

1. Please choose one of the standard network generating algorithms implemented in Python's `networkx` module (e.g., `nx.erdos_renyi_graph()`, `(nx.barabasi_albert_graph)`, or `nx.watts_strogatz_graph()`), generate a network with 100 agents at least, and assess whether Feld's friendship paradox holds in this network. Include a brief explanation of your findings and concludions, i.e. why you think Feld's friendship paradox does or does not hold, in the PDF explanation document.

Hint (1): While Feld's friendship paradox holds for many types of networks, it does not hold for all of them.

Hint (2): There are no great easy models that replicate all the properties we observe in real social networks. As such, any generating algorithm from the `networkx` module will be an imperfect model. Do not worry about this. You should solve this problem for the network generating algorithm you have chosen, whether it would be a good model for social networks or not.

Hint (3): Some commands / code snippets that may be useful for this are the following:

- Generate an Erdős-Rényi network with n nodes, any two of which are connected with probability p:
  `nx.erdos_renyi_graph(n, p)`

[1] If vaccine supply is not sufficient for everyone, more well-connected people could be vaccinated because they pose a higher risk of infecting others if they catch the epidemic.

- Generate a Barabási-Albert network with n nodes, with parameter k (minimum degree of the nodes):
  `nx.barabasi_albert_graph(n, k)`
- Generate a Watts-Strogatz small world network with n nodes, each initially connected to k nearest neighbors, but rewired with probability p:
  `nx.watts_strogatz_graph(n, k, p)`
- Obtain number of neighbors of node x in network G:
  `nx.degree(G, x)`
  Note that x is the label of the node. In the network generating functions listed above, the labels of the nodes will be integer numbers 0, 1, 2, 3, ...
- Obtain (the labels of) all the neighbors of node x in network G:
  `list(G.neighbors(x))`
- Obtain the number of neighbors for each one of the neighbors of node x in network G:
  `[nx.degree(G, v) for v in G.neighbors(x)]`

Please submit your solution including the .py script and a very brief explanation in .pdf format. The script should be well-structured and well-commented. The explanation should be brief and should explain how to use the script and show the results.

Please be sure to include your name in the submission.

If you do not receive a confirmation email, assume that the transmission of your solution did not work and resubmit/resend. I will send a confirmation email in response to any submissions made via email, OPAL sends confirmation emails automatically.

Good luck.