Exercise set #3 (24 pts)

- The deadline for handing in your solutions is October 2nd 2023 23:59.
- There are multiple ways for submitting your solutions. Check each question for details.
- Return also one `.zip` file containing all your Python code of the round in MyCourses.
- Check also the course practicalities file in MyCourses for more details on submitting your solutions.

1. Implementing the Barabási-Albert model (6 pt)

The Barabási-Albert (BA) model is a model of network growth, where new nodes continuously enter the network and make links to existing nodes with a probability that is linearly proportional to their degree. The steps required for generating a BA network with N nodes are as follows:

- Create a small seed network that has at least m nodes, where m is the number of links a new node creates to already existing nodes. In this exercise, use a 4-clique as the seed network.
- Add new nodes to the network until your network has N nodes, such that each entering node has m links and connects to existing nodes proportional to their degrees.

In this exercise, we will implement the model and investigate the networks it generates. Submit your solutions by taking a quiz in MyCourses.

a) (4 pt) Implement a Python function for generating Barabási-Albert networks. Then generate a network with ($N = 250$) and ($m = 2$) (starting from a 4-clique seed network).
   - Write down the degree of the node with the highest degree in your generated network.
   - Write down the total number of links in your generated network.
   - Visualize the network with NetworkX using the spring layout i.e., `nx.draw_spring()`. You should be able to spot some nodes that have many connections, while most of the nodes have few connections.

   Hints:
   - In general, the seed network can be anything, e.g., a clique of size $m + 1$, as long as it has enough nodes for the first incoming node to attach to with m links. In this exercise, use a 4-clique.
   - The easiest way of picking nodes with probability proportional to their degree is to use the `p` keyword in function `rng.choice()` to assign the probability that each element is selected and the `size` keyword to indicate how many nodes to sample. Remember to set the `replace` keyword to `False` to make sure you don't select the same node more than once.

b) (2 pt) Generate a new network using parameters $N = 10^4$ with $m = 3$ and plot the logarithmically binned probability density function for degree, $P(k)$ on double logarithmic axes. Compare your result with the theoretical prediction of

$$P(k) = 2m\,(m+1) \,/\, [k\,(k+1)\,(k+2)]\,.$$

To do this, plot both the experimental and theoretical distributions on the same axes.

Hints:

- Generating the results should take less than a minute (this is not a requirement for grading, just a note to let you know what to expect). If your code is taking too long to run, there are probably ways to improve its efficiency. Try to be efficient when picking nodes as link targets.
- The code for plotting the binned degree distribution with logarithmic bins is provided in the code template.

2. Friendship paradox (6 pts)

The friendship paradox states that the average number of friends that your friends have is, in general, greater than the average number of friends that you have. In other words, your friends are likely to be more popular than you are—and this is true even if you have more friends than average. This counterintuitive result arises from the mathematical nature of networks. In this exercise, we will investigate the friendship paradox for simple network models.

Submit your solutions by taking a quiz in MyCourses.

a) (2 pt) First, let us consider network data given by `moreno_highschool.edg`. The network represents the friendship between students in a small high school in Illinois, US in 1957-1958.

- Calculate the fraction of nodes in the network that have a higher average neighbor degree than their own degree. Report up to three digits (e.g., 12.3%).
- Plot for each node the number of neighbors it has (i.e., the degree of the node) and the average number of neighbors that its neighbors have (i.e., the average degree of the neighbors) as a bar chart. For convenience, we will relabel the nodes so that higher-degree nodes have smaller index labels.
- What do you observe? Does the friendship paradox hold in this network? Explain your answer.

b) (2 pt) Now let us check the friendship paradox for larger networks. For that, we consider an Erdős-Rényi random network and a Barabási-Albert scale-free network. For each of these network models, generate a network with $N = 10000$ nodes. For each network, do the following:

- Calculate the fraction of nodes in the network that have a higher average neighbor degree than their own degree. Report up to three digits.
- Plot the degree distribution and the distribution of the average degree of the neighbors.

c) (2 pt) Based on the results you obtained in a) and b), what can you say about the friendship paradox? Answer whether each of the following statements is correct or not. For each statement, explain/justify your answer in roughly one to four sentences.

- "The friendship paradox only holds for social networks. It is not applicable to other kinds of networks, such as metabolic networks, brain networks, and power grids."
- "The effect of the friendship paradox is more pronounced in a scale-free network than in an Erdős-Rényi network."
- "If you randomly select a node from a network, and select a random neighbor of that node, the probability that the second node has degree k is proportional to k."

3. Implementing random walk based network model (4 pts)

The results of the previous section suggest that the friendship paradox is a general property of random networks. Can we leverage this property to design a growing network model that generates scale-free networks? Here, we implement the random walk based growing network model. In this model, the new node that enters the system at each step deploys random walkers on the network and connects to the destinations of the walkers. More precisely, it is formulated as follows:

1. Add a new node to the network and randomly select m target nodes.
2. For each target node, select a random neighbor and move the target to that neighbor.
3. Create a link between the new node and each of the m target nodes after repeating step 2 for t steps.
4. Repeat steps 1-3 until the network reaches the desired size N.

In this exercise, we implement this model and compare its properties to the Barabási-Albert model.

a) (3 pt) Generate a network with $N = 10^4$, $m = 3$, and $t = 8$, and plot the degree distribution.

b) (1 pt) Compare the degree distribution of the network generated by the random walk based model with the degree distribution of the Barabási-Albert model. Select which of the following best describes your observation.

A. The degree distribution for the random walk based model is a Poisson distribution with the same mean as that of the Barabási-Albert model.

B. The degree distribution for the random walk based model follows a power law with the exponent -3, which is the same as the Barabási-Albert model.

C. The degree distribution for the random walk based model follows a power law with exponent -2, while the degree distribution for the Barabási-Albert model follows a power law with exponent -3.

D. The degree distribution for the random walk based model follows a power law with exponent -2.5, while the degree distribution for the Barabási-Albert model follows a power law with exponent -3.

Aalto University
CS-E5740 Complex Networks, Autumn 2023
Hiraoka, Kivelä, Korhonen
Exercise set #3

4. Challenge exercise: Deriving the degree distribution for the BA model (8 pts, pen and paper)

In this exercise, we show that the degree distribution of the Barabási-Albert scale-free model is $P(k) = 2m\,(m+1) \,/\, [k\,(k+1)\,(k+2)]$ in the limit of infinite size, i.e. it becomes a power law for large values of k.

Submit your solution to MyCourses as a pdf file.

As a reminder, the BA scale-free network growth algorithm goes as follows:

1. Start the network growth from a small "seed" network of N_0 fully connected vertices.
2. Pick m different vertices from the existing network so that the probability of picking vertex v_i of degree k_i equals $\pi_i = \frac{k_i}{\sum_j k_j}$, i.e. the degree of that vertex divided by the sum of the degrees of all vertices.
3. Create a new vertex and connect it to the m vertices which were chosen above.
4. Repeat steps 2–3 until the network has grown to the desired size of $N_{\text{final}} = N_0 + I$ vertices, where I denotes the number of iterations.

The exact degree distribution for the Barabási-Albert model in the limit of infinite network size[1] can be derived using the so-called *master equation* approach (see, e.g., [1]). This approach makes use of the fact that the BA model is a model of network growth, i.e. the network is continuously expanding. The key idea of the master equation approach is to write an equation for the changes in the fraction of vertices of degree k, f_k, as function of time and find stationary solutions. In such solutions, f_k does not change anymore when the network grows, corresponding to an infinite network size. This stationary solution for f_k when $N \to \infty$ equals the degree distribution $P(k)$ of the network.

In this exercise, you will derive the distribution step-by-step.

Throughout this exercise, show all intermediate steps and motivate your reasoning either mathematically or verbally.

a) (2 pts) Let $f_{k,N}$ be the density of vertices of degree k in a network that, at time t, has altogether $N(t)$ vertices. Thus, $\mu_{k,N} = N(t) f_{k,N}$ is the number of vertices of degree k in the network. At each time step, one vertex is added, and hence $N(t) = t + N_0$, where $t \in \mathbb{Z}$ denotes the time step of the network growth process. Since N_0 is small, we can approximate $N \approx t$. In the following, N will be used for $N(t)$ for readability.

In the BA model, the probability π_i that a new edge attaches to a *particular* vertex of degree k_i equals:

$$\pi_i = \frac{k_i}{\sum_{j=1}^{N} k_j}. \tag{1}$$

As our first intermediate result, we will need the probability $\Pi(k)$ that a new edge attaches to *any* vertex of degree k in a network of N vertices. This equation reads:

$$\Pi(k) = \frac{k f_{k,N}}{2m}. \tag{2}$$

[1] Some words of explanation might be helpful here. This approach is typical of statistical physics — very large systems are usually well approximated by results derived for infinite systems. This also applies in the case of complex networks. The exact degree distribution can (and has been) derived for finite-sized networks as well, but the calculations are extremely cumbersome.

Your first task is to **derive** (2) from (1).

Hint: Formulate the sum of degrees $\sum_{j=1}^{N} k_j$ in terms of m and N (How much the total degree grows when a new vertex is added? You can approximate $N_0 \approx 0$.) and note that there are $Nf_{k,N}$ vertices of degree k.

b) (2 pts) Next, we will construct the master equations for the changes of the *average* numbers of vertices of degree k. From Eq. (2), the average number[2] of vertices of degree k that gain an edge when a single new vertex with m edges is added is $m \times kf_{k,N}/2m = \frac{1}{2}kf_{k,N}$. This means that the number $\mu_{k,N}$ of vertices with degree k must *decrease* by this amount, since these vertices become vertices of degree $k+1$. Let's mark this as:

$$\mu_k^- = \frac{1}{2}kf_{k,N}. \tag{3}$$

But at the same time, there is an increase μ_k^+ as well. For vertices with degree $k > m$ this is equal to the average number of vertices that used to have degree $k-1$ and became vertices of degree k by gaining an edge. For vertices with $k = m$, $\mu_k^+ = 1$. **Explain why?**

Now, we're ready for the master equation! With the help of the above results, **write down** equations for the *net change* of the number of vertices of degree k as the network grows in size from N to $N+1$,

$$\begin{aligned}(N+1)f_{k,N+1} - Nf_{k,N} = &\quad \mu_k^+ - \mu_k^-. \\ = &\quad ?\end{aligned} \tag{4}$$

Write separate equations for both cases ($k > m$, $k = m$). In the case $k = m$, denote the densities by $f_{m,N+1}$ and $f_{m,N}$ as this will make things easier in what follows. (Note that the left-hand side in the above equation is simply due to $\mu_{k,N+1} = (N+1)f_{k,N+1}$ and $\mu_{k,N} = Nf_{k,N}$.)

c) (2 pts) Now, let the network grow towards the infinite network size limit and consider stationary solutions of the two equations you just wrote. In this case, there are no longer changes in the probability density f_k, and so you can write $f_{k,N+1} = f_{k,N} = f_k$, $f_{k-1,N} = f_{k-1}$ and $f_{m,N+1} = f_{m,N} = f_m$. **Explain why?**

Write down equations for f_k and f_m. The density f_k should now be of the form $G(k) \times f_{k-1}$, where $G(k)$ is some prefactor depending on k alone. f_m should be a function of m only, $f_m = H(m)$.

d) (2 pts) We're almost there! Now things get recursive: f_k depends on f_{k-1}. At the same time, we have a formula for f_m which depends only on m, which is the smallest degree in our network. Your final task is to **derive** a formula for f_k, so that f_k is a function of k and m only.

Hint: First, write a formula for f_{m+1} using the formulas $f_k = G(k) \times f_{k-1}$ and $f_m = H(m)$, then to write a formula for f_{m+2}, etc. Continue until you see which terms cancel out.

Feedback (1 pt)

To earn one bonus point, give feedback on this exercise set and the corresponding lecture latest two days after the exercise round's submission deadline. You can find the feedback form in

[2] This is the so-called *mean-field* approach. Instead of keeping track on what happens to each vertex, we will focus on what happens to vertices of some degree k *on average*.

MyCourses.

References

[1] M. E. J. Newman, "The Structure and Function of Complex Networks," *Siam Review*, vol. 45, pp. 167–256, 2002.