Exercise set #3 (14 pts)

- The deadline for handing in your solutions is October 5th 2020 23:55.
- Return your solutions (one `.pdf` file and one `.zip` file containing Python code) in MyCourses (Assignments tab). Additionally, submit your pdf file also to the Turnitin plagiarism checker in MyCourses.
- Check also the course practicalities page in MyCourses for more details on writing your report.

1. Implementing the Barabási-Albert BA model (6 pts)

The Barabási-Albert (BA) scale-free network 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 Barabási-Albert scale-free network with N nodes are as follows:

- Create a small seed network which 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 3-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. **The provided template file covers most of the plotting and binning required in this exercise, which lets you focus practically only on the actual implementation of the model.** Using the template is **optional**.

a) (3 pts)

- **Implement** a Python function for generating Barabási-Albert networks. Then **generate** a network with $N = 200$ and $m = 1$ (starting from a 3-clique seed network – the network is thus tree-like except for the initial clique).
- *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(G)`. 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 3-clique*.

- The easiest way of picking nodes with probability proportional to their degree is to use the `p` keyword in function `np.random.choice` to assign probability of selection to each element 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 get any node more than once.

b) (3 pts) **Generate** a new network using parameters $N = 10^4$ with $m = 2$ and **plot** the logarithmically binned probability density function for degree, $P(k)$ (on double logarithmic axes, `ax.loglog`).

Compare your result with the theoretical prediction of $P(k) = 2m\,(m+1)\,/\,[k\,(k+1)\,(k+2)]$ (proved in the next exercise). To do this, **plot** both the experimental and theoretical distributions on the same axes.

Hints:

- Generating the results should take a few seconds. If your code is taking too long to run, there are probably ways to improve its efficiency.
- The code for plotting the binned degree PDF with logarithmic bins is provided in the template. If you wish to do the binning and plotting yourself, or are simply interested in how it's properly done, have a look at the materials in the **binning tutorial** in MyCourses. Some considerations for those further interested:
- There is no simple rule of thumb for selecting the number of bins. However, ideally there should be no empty bins, but on the other hand one would like to have as many bins as possible to best present the shape of the true distribution.
- When plotting the degree PDF, you may end up with empty bins with small values of k. (Consider e.g. if you had a bin [3.1, 3.9]: this bin would always have value zero.) To circumvent this, it is often practical to bin degree distributions using "lin-log" bins: [0.5, 1.5, 2.5, . . . , 9.5, 10.5, 12.5, 16.5, . . .] so that one does not end up with empty bins for small values of k. **If you wish**, you may use this more sophisticated approach of "lin-log" bins for plotting the binned degree PDF.

2. 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.

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 $p(k_i) = \frac{k_i}{\sum_j k_j}$, 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.

Aalto University, School of Science
CS-E5740 Complex Networks, Fall 2020
Saramäki, Korhonen, Alakörkkö, Badie Modiri, Heydari, Hiraoka, Nurmi, Triana, Ureña Carrion, Sormunen
Exercise set #3

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, p_k, as function of time and find stationary solutions. In such solutions, p_k does not change anymore when the network grows, corresponding to an infinite network size. This stationary solution for p_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 $p_{k,N}$ be the density of vertices of degree k in a network that, at time t, has altogether $N(t)$ vertices. Thus, $n_{k,N} = N(t)p_{k,N}$ 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:

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

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

Hint: Formulate the sum of degrees $\sum_j^N k_j$ 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 $Np_{k,N}$ 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 $m \times kp_{k,N}/2m = \frac{1}{2}kp_{k,N}$. This means that the number $n_{k,N}$ of vertices with degree k must *decrease* by this amount, since these vertices become vertices of degree $k+1$. Let's mark this as:

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

But at the same time, there is an increase n_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 $k = m$, $n_k^+ = 1$. **Explain why?**

[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.

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

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 $N+1$,

$$\begin{aligned}(N+1)p_{k,N+1} - Np_{k,N} = & \quad n_k^+ - n_k^- . \\ = & \qquad ?\end{aligned} \tag{4}$$

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

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 p_k, and so you can write $p_{k,N+1} = p_{k,N} = p_k$, $p_{k-1,N} = p_{k-1}$ and $p_{m,N+1} = p_{m,N} = p_m$.

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

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

   Hint: First, write a formula for p_{m+1} using the formulas $p_k = F(k) \times p_{k-1}$ and $p_m = G(m)$, then to write a formula for p_{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 report's submission deadline.
Link to the feedback form: `https://forms.gle/YdwDZNnYvf3cGivz8`.

References

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