PHYS 5116 – Network Science I
Assignment 2

Giuliano Porciúncula Guedes

October 31, [Fall] 2025

Friendship Paradox (25 points).
(Textbook HW 4.10.2) The degree distribution of a network, p_k, expresses the probability that a randomly selected node has k neighbors. However, if we randomly select a link, the probability that a node at one of its ends has degree k is $q_k = Akp_k$, where A is a normalization factor.

(a) Find the normalization factor A, assuming that the network has a power law degree distribution with $2 < \gamma < 3$, with minimum degree k_{min} and maximum degree k_{max}.

(b) In the configuration model, q_k is also the probability that a randomly chosen node has a neighbor with degree k. What is the average degree of the neighbors of a randomly chosen node?

(c) Calculate the average degree of the neighbors of a randomly chosen node in a network with $N = 10^4$, $\gamma = 2.3$, $k_{\text{min}} = 1$ and $k_{\text{max}} = 1000$. Compare this value with the average degree of the network, $\langle k \rangle$.

(d) How can you explain the "paradox" of (c), that is a node's friends on average have more friends than the node itself?

Solution

(a) Our assumption is that is that our network has a power-law degree distribution with exponent $2 < \gamma < 3$, thus

$$p_k = Ck^{-\gamma} \tag{1}$$

To find our normalization constant C of the degree distribution, we need to normalize it. By definition, assuming it's continuous without any loss, we have

$$\int_{k_{\text{min}}}^{k_{\text{max}}} p_k dk = 1 \tag{2}$$

Thus we have

$$C^{-1} = \int_{k_{\text{min}}}^{k_{\text{max}}} k^{-\gamma} dk \tag{3}$$

Which is a trivial integral. Our constant is given by

$$C = \frac{(\gamma - 1)}{k_{\text{min}}^{-(\gamma-1)} - k_{\text{max}}^{-(\gamma-1)}} \tag{4}$$

Now we also need to normalize q_k. Again, by definition, k in q_k can go from k_{min} to k_{max}

$$\int_{k_{\text{min}}}^{k_{\text{max}}} q_k dk = 1 \tag{5}$$

Thus

$$A^{-1} = \int_{k_{\text{min}}}^{k_{\text{max}}} k p_k dk \tag{6}$$

Note that the integral is $\langle k \rangle$ by it's definition

$$\langle k \rangle = \int_{k_{\text{min}}}^{k_{\text{max}}} k p_k dk \tag{7}$$

Let's calculate it

$$\langle k \rangle = C \int_{k_{\text{min}}}^{k_{\text{max}}} k^{-(\gamma-1)} dk \tag{8}$$

Again, the same trivial integral

$$\langle k \rangle = C \frac{k_{\text{min}}^{-(\gamma-2)} - k_{\text{max}}^{-(\gamma-2)}}{(\gamma - 2)} \tag{9}$$

So

$$\boxed{A = \frac{(\gamma - 2)}{(\gamma - 1)} \frac{k_{\text{min}}^{-(\gamma-1)} - k_{\text{max}}^{-(\gamma-1)}}{k_{\text{min}}^{-(\gamma-2)} - k_{\text{max}}^{-(\gamma-2)}}} \tag{10}$$

□

(b) Since that, in the configurational model, q_k is the probability that a randomly chosen node has a neighbor with degree k, then the average degree of neighbors of a randomly chosen node is by definition

$$\langle k_{\text{neighbors}} \rangle = \int_{k_{\text{min}}}^{k_{\text{max}}} k q_k dk \tag{11}$$

Substituting we have the integral

$$\langle k_{\text{neighbors}} \rangle = AC \int_{k_{\text{min}}}^{k_{\text{max}}} k^{-(\gamma-2)} dk \tag{12}$$

Which again is our trivial integral

$$\boxed{\langle k_{\text{neighbors}} \rangle = \frac{(\gamma - 2)}{(3 - \gamma)} \frac{k_{\text{max}}^{3-\gamma} - k_{\text{min}}^{3-\gamma}}{k_{\text{min}}^{-(\gamma-2)} - k_{\text{max}}^{-(\gamma-2)}}} \tag{13}$$

□

(c) Given our formulas for $\langle k \rangle = 1/A$ and $\langle k_{\text{neighbors}} \rangle$, we calculate the values $\langle k \rangle \approx 3.788$ and $\langle k_{\textbf{neighbors}} \rangle \approx 61.234$.

□

(d) You can compare q_k and p_k, since they define respectively $\langle k_{\textbf{neighbors}} \rangle$ and $\langle k \rangle$. We have the relation (remember e found $A = 1/\langle k \rangle$)

$$q_k = \frac{k}{\langle k \rangle} p_k \tag{14}$$

So for $k < \langle k \rangle$ we have a lower probability of connecting to the node of degree k and for $k > \langle k \rangle$ we have a higher probability of connecting to a node of degree k. Re-stated: Lower degree nodes have less q_k probability of being chosen compared to p_k and higher degree nodes have more q_k probability of being chosen compared to p_k. A direct result of this is that the first moment using q_k will be higher than p_k. No paradox.

■

Generating Barabási-Albert Networks (25 points).
(Textbook HW 5.12.1) Write code that generates a Barabási-Albert network with $N = 10^4$ and $m = 4$. Start with a fully connected 4-node network as initial condition. (Note: Various network libraries including NetworkX in Python already have implementations of the Barabási-Albert model, but you will need to implement it yourself in order to do parts (a) and (e). Include your source code with your solution.

(a) Plot the degree distribution at intermediate steps in the network's growth. Plot these degree distributions when the network has 10^2, 10^3, and 10^4 nodes.

(b) Compare the distributions at these intermediate steps by plotting them together and fitting each to a power-law with degree exponent γ (include the γ values for each distribution somewhere in the plot). Do the distributions "converge"?

(c) Plot together the cumulative degree distributions at intermediate steps.

(d) Measure and plot the average clustering coefficient as a function of N.

(e) Following Figure 5.6a, measure the degree dynamics of one of the initial nodes and of the nodes added to the network at times $t = 100$, $t = 1000$ and $t = 5000$.

Solution

(a,b) Our algorithm starts by creating a complete graph of size m. Then we add new nodes until we have N nodes. Each added node makes m link. For each new link we scan all nodes j not connected to the new node i. We pick one of these nodes with probability $p_j = \frac{k_j}{\sum\limits_{\ell \in \Omega_i} k_\ell}$ where Ω_i is the set of nodes not connected to i at that instant.

Figure 1 shows us the degree distribution at $N = 10^2$, 10^3, and 10^4 nodes with $m = 4$. Here we took the configurational average of 6 network construction, i.e., we fully build each network with size $N = 10^4$, taking snapshots of the degree distribution at $N = 10^2$, 10^3, and 10^4, and then for each point (p_k, k), we took the average of all 6 configurations. This helps us reduce the noise and better sample the configurational space. If I had more time, I'd crank up the number of configurations to at least 100. The transparent hollow circles are the actual degree distribution, while the opaque solid circles are the log-binned degree distribution with increments of $\times 10^{1/10}$ starting at 1. We also fit a power-law (stright line since we linearized our plot) and foung the power-law exponents.

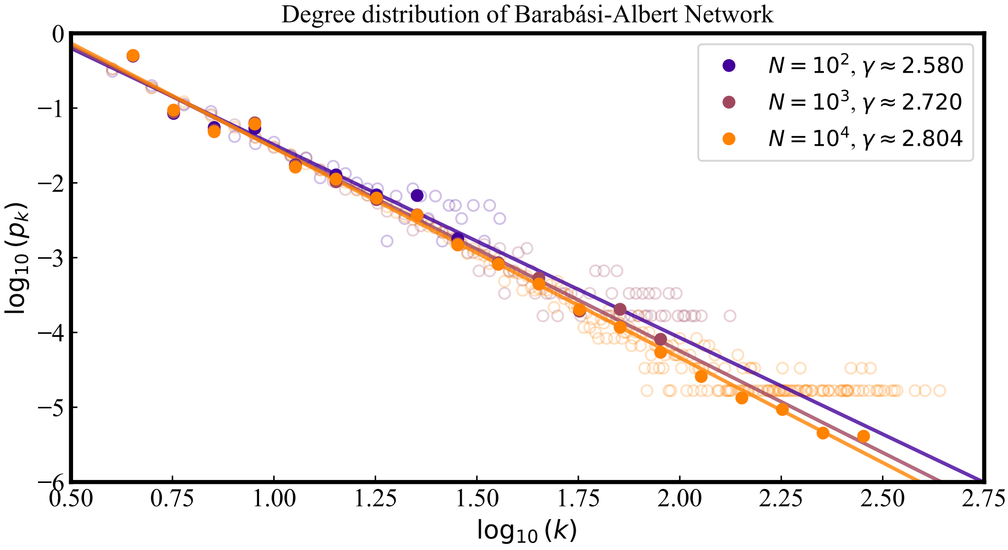

Figure 1: Degree distribution of the Barabási-Albert Network with $m = 4$, $N = 10^4$ with intermediate steps. The data is averaged in 6 configurations and power-law $p_k \sim k^{-\gamma}$ is fitted.

□

(c) Figure 2 shows us the cumulative degree distribution with the same data as before. The cumulative degree distribution is defined as $f_k = \sum_{k'=k_{\min}}^{k} p_{k'}$.

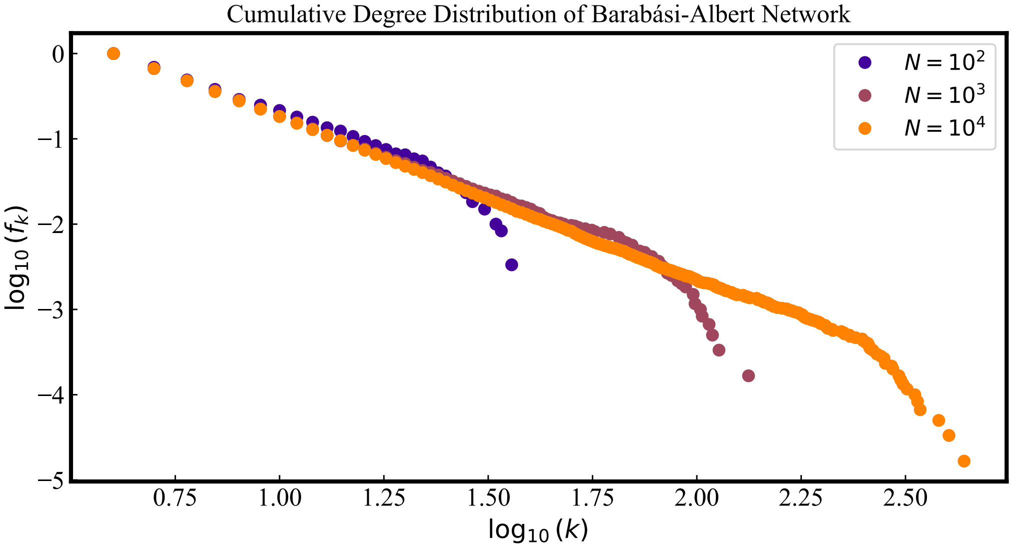

Figure 2: Cumulative degree distribution of the Barabási-Albert Network with $m = 4$, $N = 10^4$ with intermediate steps. The data is averaged in 6 configurations.

□

(d) For this item, after adding each node, we calculate the average clustering coefficient of the network and return a list of the number of nodes (the time) and the average clustering coefficient at that intermediate step. For the average clustering coefficient we used the networkx function average_clustering(). We could've easily created a function to do it but time was slow. Here follows how I would implement it.

Take the average of the clustering coefficient of all nodes in the network. The clustering coefficient of a node i is defined as $C_i = \frac{2L_i^{(n)}}{k_i(k_i-1)}$. The clustering coefficient of a node i measures how it's neighbors are connected with themselves. So the denominators what would be if all of i's neighbors were connected to all the other neighbors of i, thus $k_i(k_i-1)$ is a connected graph. The numerator is how many of these links actually exist, if $C_i = 0$ then none of it's neighbors are connected to it's other neighbors. On the other hand, if $C_i = 1$, then all it's neighbors are connected to all it's other neighbors. The code to calculate C_i is simple. First find i's neighbors and store in a set A. Then for each node j in the set A, find it's neighbors B and calculate how many elements are in the set $(B/\{i\}) \cap (A/\{j\})$ (how many of j's neighbors are also i's neighbors, except both nodes i and j). Sum all of these values then divide by $k_i(k_i-1)$.

Figure 3 show us the Clustering coefficient of the Barabási-Albert Network during it's construction. We observe that $\langle C \rangle \sim \ln(N)^2/N$, indicated by the straight black dashed line in the linearized $\langle C \rangle$ vs $\ln(N)^2/N$. For the inset we have all values from $N = m$ to $N = 10^4$, while the bigger plot we cut off some of $N \leq 100$ (in purple) so we discard the small network size unwanted aberrations. This was also averaged in 6 configurations.

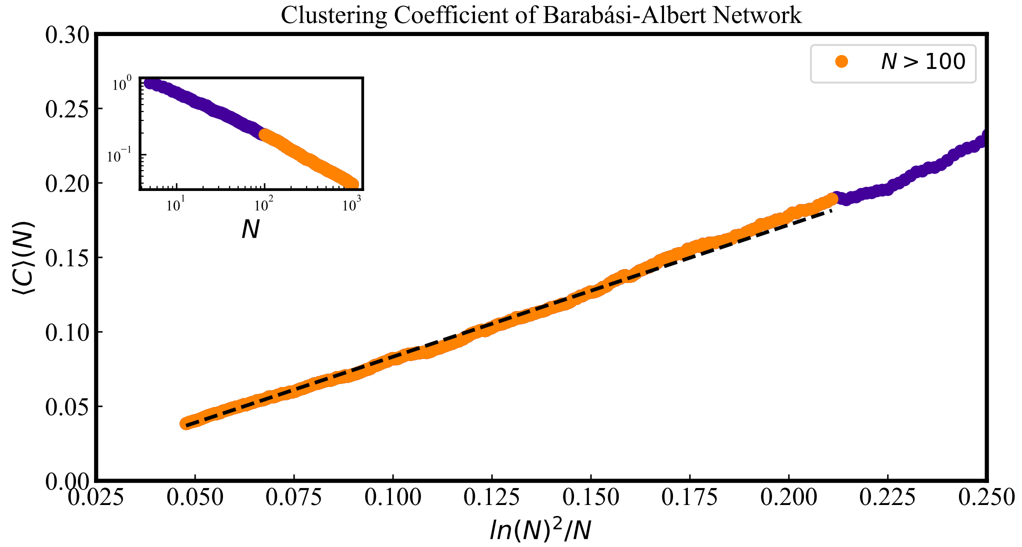

Figure 3: Clustering coefficient of the Barabási-Albert Network with $m = 4$, $N = 10^4$ with all the intermediate steps. The data is averaged in 6 configurations.

□

(e) Now we follow the degree versus time of nodes introduced at the steps $t = 100$, 1000, and 5000. Here we also took 6 configurational averages.

Figure 4 shows us how the degree of added nodes change in time. We have linearized our curves as for it to be the degree of nodes vs t^β, where $\beta = 1/2$ as given by theory. We confirm that these curves follows $\sim t^\beta$ predicted by theory, as shown by the fitted straight black dashed lines in the linearized axis.

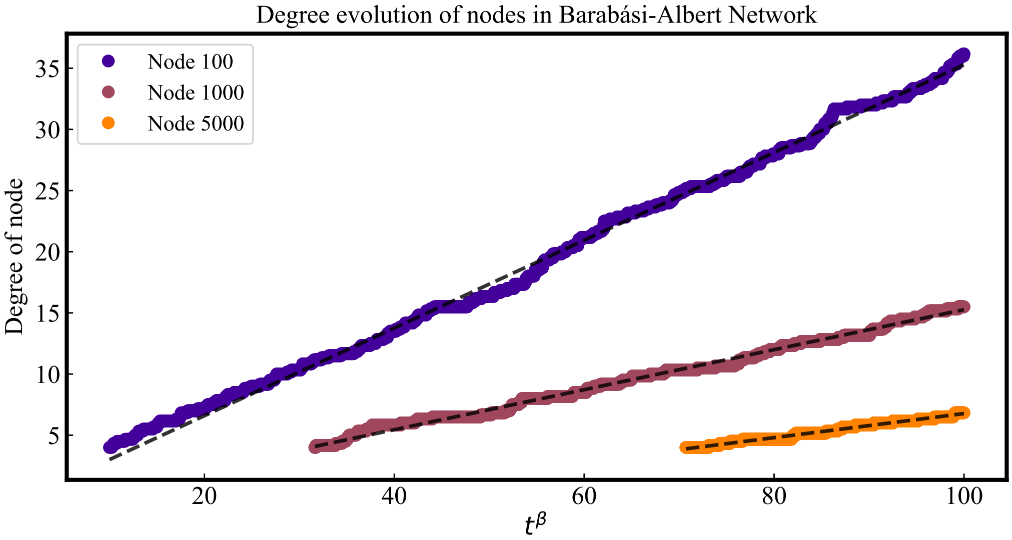

Figure 4: Clustering coefficient of the Barabási-Albert Network with $m = 4$, $N = 10^4$ with all the intermediate steps. The data is averaged in 6 configurations.

■

The Bianconi-Barabási model (25 points).

Following the same procedure as question 2 (start with a fully connected 4 node clique, $N = 10^4$, $m = 4$) modify your code to implement the Bianconi-Barabási model. Create two different networks, drawing node fitnesses from:

- A uniform distribution between 0 and 1
- A delta distribution where nodes have a 90% chance to be assigned $\eta = 0.1$ and a 10% chance to be assigned $\eta = 0.9$

(a) Comment on how you expect the different distributions of fitness to impact the network structure (degree distribution, hubs, clustering, etc).

(b) Plot the degree distributions of the two networks. In the legend, include the size of the largest hub of each network. Comment on any differences between the two.

(c) Plot the average clustering in each network, binned by degree. Again comment on any differences between the two networks.

(d) Repeat part (e) of question 2 for nodes added at timestep 0, 100, and 1000 for each network. Also track the degree dynamics of the three nodes with the highest fitness. (For the second network you can pick any nodes that have $\eta = 0.9$, but it will be more interesting to pick nodes that are not in the first few hundred time steps). Does the fitness impact the degree dynamics?

Solution

(a) Since it's not uniform, we can expect some nodes added later surpassing nodes that were added before. This is not present in the BA network. This breaks the first-mover advantage since fitness is not uniform. Remember that the degree, analytically, evolves with $\sim t^{\beta(\eta)}$, where η is the fitness of that node. In the BA network, $\beta(\eta) = 1/2$, but in the generalized, BB, network it is $\beta(\eta) = \eta/C$ where C is a constant that can be found by

$$C = \int \rho(\eta) \frac{\eta}{1 - \beta(\eta)} d\eta \tag{15}$$

Where $\rho(\eta)$ is the distribution of fitness. After solving the integral, one needs to solve for C, which is not always analytical and needs to be numerically estimated.
What $\sim t^{\beta(\eta)}$ tells us that that the more spread out and extreme the distribution of η, the more it will favor hubs and so the degree distribution will be more fat-tailed. I also expect the network to be more clustered with more spread out η. This comes from the fact that since we will have everyone connecting to the hubs, then inevitably nodes will connect with neighbors of the hub, increasing clustering.

The degree distribution in the BB model is given by

$$p_k = C \int \frac{\rho(\eta)}{\eta} \left(\frac{m}{k}\right)^{\frac{C}{\eta}+1} d\eta \tag{16}$$

Let's do some math to find the degree distributions for both our cases. For an uniform degree distribution, we have $\rho(\eta) = A$ where A is a constant. We have here $\eta \in [0, 1]$, thus our normalization is

$$\int_0^1 \rho(\eta) d\eta = 1 \tag{17}$$

This straightforward gives us $A = 1$, thus $\rho(\nu) = 1$. Now let's find C. We have that

$$C = \int_0^1 \frac{\eta}{1 - \frac{\eta}{C}} d\eta \tag{18}$$

We can change variables $x = \eta/C$

$$C^{-1} = \int_0^{1/C} \frac{x}{1 - x} dx \tag{19}$$

Calling $y = 1 - x$

$$C^{-1} = \int_{1-\frac{1}{C}}^1 \frac{1 - y}{y} dy = \int_{1-\frac{1}{C}}^1 \frac{1}{y} dy - \int_{1-\frac{1}{C}}^1 dy = \left(\ln[1] - \ln\left[1 - \frac{1}{C}\right]\right) - \left[1 - \left(1 - \frac{1}{C}\right)\right] \tag{20}$$

$$\frac{1}{C} = -\ln\left[1 - \frac{1}{C}\right] - \frac{1}{C} \tag{21}$$

$$\mathrm{e}^{-\frac{2}{C}} = 1 - \frac{1}{C} \tag{22}$$

Which cannot be solved analytically. Numerically we have that $C \approx 1.255$.

Now for the other distribtion, we have

$$\rho(\eta) = 0.9\delta(\eta - 0.1) + 0.1\delta(\eta - 0.9) \tag{23}$$

Thus our C integral becomes

$$C = 0.9 \int \delta(\eta - 0.1) \frac{\eta}{1 - \frac{\eta}{C}} d\eta + 0.1 \int \delta(\eta - 0.9) \frac{\eta}{1 - \frac{\eta}{C}} d\eta \tag{24}$$

Applying the δs we have

$$C = 0.9 \frac{0.1}{1 - \frac{0.1}{C}} + 0.1 \frac{0.9}{1 - \frac{0.9}{C}} \tag{25}$$

Which can be written into

$$C^2 - 1.18C + 0.18 = 0 \tag{26}$$

Which has two solutions, 0.18 and 1, since we know that $C > \eta_{\max}$, the only physical solution is $C = 1$ ($\eta_{\max} = 0.9$).

Now with both Cs in hand, we can go for the degree distribution. First let's tackle the uniform degree distribution

$$p_k = C \int_0^1 \frac{1}{\eta} \left(\frac{m}{k}\right)^{\frac{C}{\eta}+1} d\eta \tag{27}$$

Calling $x = \eta/C$

$$p_k = C \int_0^{C^{-1}} x^{-1} \left(\frac{m}{k}\right)^{x^{-1}+1} dx \tag{28}$$

Calling $y = x^{-1}$

$$p_k = C \left(\frac{m}{k}\right) \int_C^\infty \left(\frac{m}{k}\right)^y \frac{1}{y} dy \tag{29}$$

We can massage this into the gamma function

$$p_k = C \left(\frac{m}{k}\right) \int_C^\infty \exp\left[-y \ln\left(\frac{k}{m}\right)\right] y^{-1} dy \tag{30}$$

Now taking $w = y \ln\left(\frac{k}{m}\right)$

$$p_k = C \left(\frac{k}{m}\right)^{-1} \int_{C \ln\left(\frac{k}{m}\right)}^\infty w^{0-1} \mathrm{e}^{-w} dw \tag{31}$$

The incomplete Gamma function is defined as

$$\Gamma(z, w_0) = \int_{w_0}^\infty w^{z-1} \mathrm{e}^{-w} dw \tag{32}$$

Thus we have

$$\boxed{p_k^{(\text{uniform})} = C^{(\text{uniform})} \left(\frac{k}{m}\right)^{-1} \Gamma\left(0, C^{(\text{uniform})} \ln\left(\frac{k}{m}\right)\right)} \tag{33}$$

where we found from before

$$\boxed{\mathrm{e}^{-\frac{2}{C}^{(\text{uniform})}} = 1 - \frac{1}{C^{(\text{uniform})}}} \quad \overset{\text{numerical approximation}}{\Longrightarrow} \quad \boxed{C^{(\text{uniform})} \approx 1.255} \tag{34}$$

Remember that

$$\Gamma(0, w_0) = \int_{w_0}^{\infty} \mathrm{e}^{-w} w^{-1} dw \tag{35}$$

by applying successive integration by parts, one can find the following asymptotic expansion

$$\Gamma(0, w_0) = \mathrm{e}^{-w_0} \sum_{n=0}^{\infty} (-1)^n \frac{n!}{w_0^{n+1}} \tag{36}$$

Thus for very large w_0, we have

$$\Gamma(0, w_0) \approx \frac{\mathrm{e}^{-w_0}}{w_0} \tag{37}$$

Thus for very very very large k we have straightforward by simple mathematical manipulations that

$$\boxed{p_{k\to\infty}^{(\text{uniform})} \sim \frac{k^{-(1+C^{(\text{uniform})})}}{\ln(k)}} \tag{38}$$

Where the exponent is ≈ 2.255.

Now for the other spiked distribution, we have

$$p_k = \int \frac{\rho(\eta)}{\eta} \left(\frac{m}{k}\right)^{\frac{1}{\eta}+1} d\eta \tag{39}$$

Where here we remember that $C = 1$ and $\rho(\eta) = 0.9\delta(\eta - 0.1) + 0.1\delta(\eta - 0.9)$

$$p_k = 0.9 \int \delta(\eta - 0.1) \frac{1}{\eta} \left(\frac{m}{k}\right)^{\frac{1}{\eta}+1} d\eta + 0.1 \int \delta(\eta - 0.9) \frac{1}{\eta} \left(\frac{m}{k}\right)^{\frac{1}{\eta}+1} d\eta \tag{40}$$

Applying the δs we have

$$\boxed{p_k^{(\text{spiked})} = 9 \left(\frac{k}{m}\right)^{-11} + \frac{1}{9} \left(\frac{k}{m}\right)^{-\frac{1.9}{0.9}}} \tag{41}$$

Which is a double power-law for big k, we expect the second term to dominate, thus for big k we have that

$$\boxed{p_{k\to\infty}^{(\text{spiked})} \sim k^{-\frac{1.9}{0.9}}} \tag{42}$$

Where the exponent is ≈ 2.111.

□

(b) Figures 5 and 6 shows the degree distributions of the BB model for our both desired fitness distributions: uniform fitness and spiked fitness. In each plot we have: the standard and log binned generated data for $N = 10^4$, $m = 4$, and 6 configurations; the approximation ($k \to \infty$) fit following the expressions discussed in (a); and the normalized analytical solution also discussed in (a). Here we had to "re-"normalize our distributions since we have a $k_{\min}$ and $k_{\max}$, where our analysis on (a) did not take that into account, and so our analytical solutions were shifted since N is not big enough. All the results follow the expectation discussed in (a).

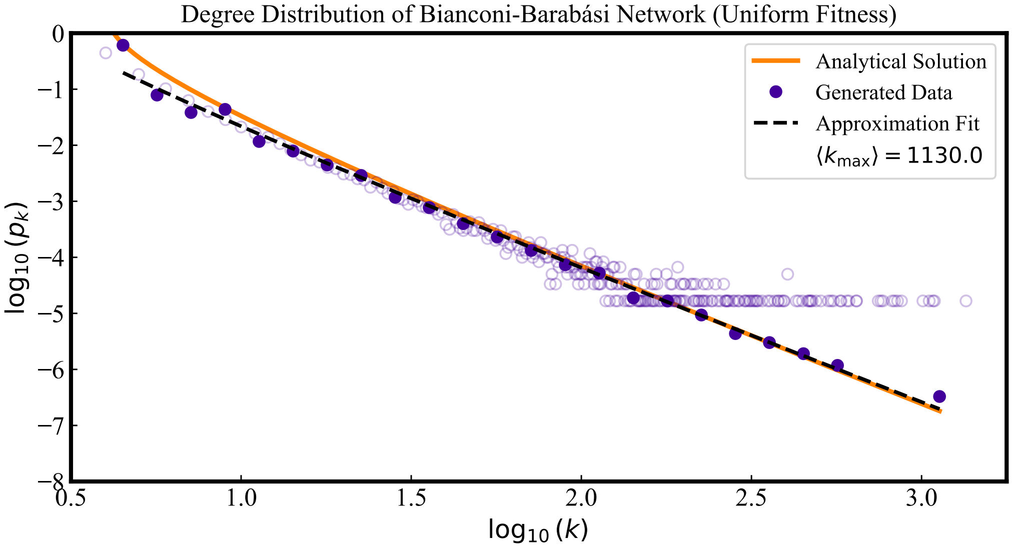

Figure 5: Degree distribution of the BB model with uniform fitness distribution, $N = 10^4$, $m = 4$, and 6 configurations. The corrected power-law $p_k \sim k^{-\gamma}/\ln(k)$ exponent was fitted to be $\gamma \approx 2.221$, close to the expected 2.255. Analytical solution agrees with the generated data.

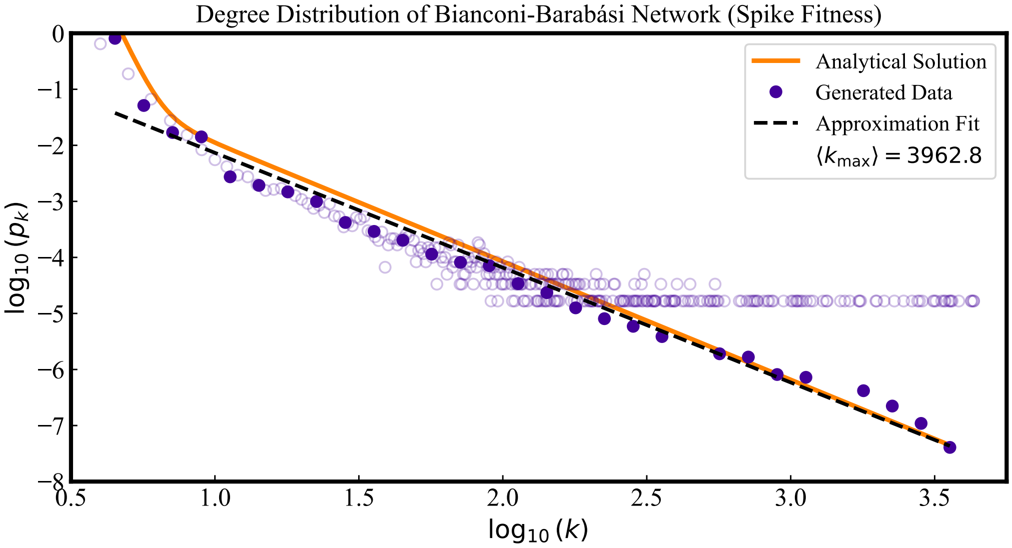

Figure 6: Degree distribution of the BB model with spiked fitness distribution, $N = 10^4$, $m = 4$, and 6 configurations. The power-law $p_k \sim k^{-\gamma}$ exponent was fitted to be ≈ 2.049, close to the expected 2.111. Analytical solution agrees with the generated data.

□

(c) Figure 7 show the Clustering Coefficient of the BB model versus network size for all the intermediate steps from $N = m$ to $N = 10^4$ with 6 configurations. From $N = m$ until around $N = 30$ both curves have the same power-law tendency. In the figure they don't look like they have the same slope but I'm pretty sure it is the same slope, we can't see it due to fluctuations and noise. After around $N = 30$, the uniform fitness keeps with it's power-law tendency $\langle C\rangle(N) \sim N^{-\gamma}$ with exponent $\gamma \approx 0.428$, while the spike fitness stagnates horizontally becoming constant. All this is very interesting for two reasons. First for our uniform distribution, it doesn't follow the BA expected $ln(N)^2/N$, it is a power-law instead. For the spiked distribution is even more interesting since it stagnates horizontally. All this confirms our suspicions discussed in (a), mainly that a disperse fitness distribution as seen in our spiked one generates more clustering since after your network is pretty much formed (30 is very small) our average clustering coefficient keeps constant at ≈ 0.4. This is due to the fact that all the nodes link to the clusters, and then they link to each-other increasing clustering.

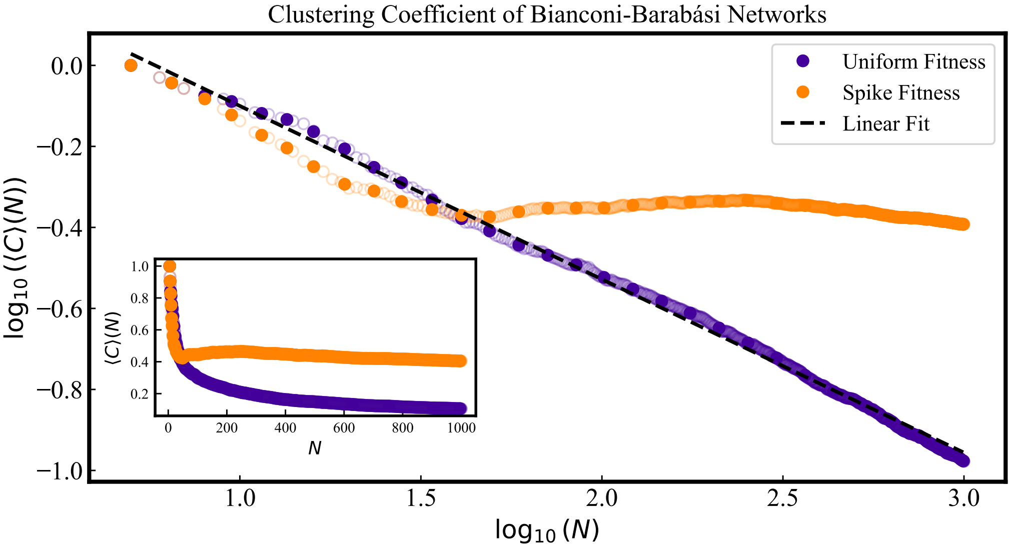

Figure 7: Clustering coefficient of the BB model for uniform and spiked fitness distribution with $m = 4$, $N = 10^4$ and 6 configurations. On the uniform distribution we observe a power-law $\langle C\rangle(N) \sim N^{-\gamma}$ with exponent $\gamma \approx 0.428$. On the spiked distribution we stagnate to around 0.4 after adding approximately 30 nodes.

□

(d) For this one we took the nodes with highest degree from $t = 100$ to 1000. We did not take any configurational averages since we would not have the same nodes, thus the trajectories are quite jaggedy in comparison to problem 2. Figures 8 and 9 show that the higher fitness nodes increases much faster and dominates over the ones with lower fitness, given enough time. This is due to the evolution of the degree $k \sim t^{\beta(\eta)}$ where $\beta(\eta) = \eta/C$. Since C are all the same for a given fitness distribution, the nodes with higher fitness will eventually surpass the ones with lower fitness, given enough time.

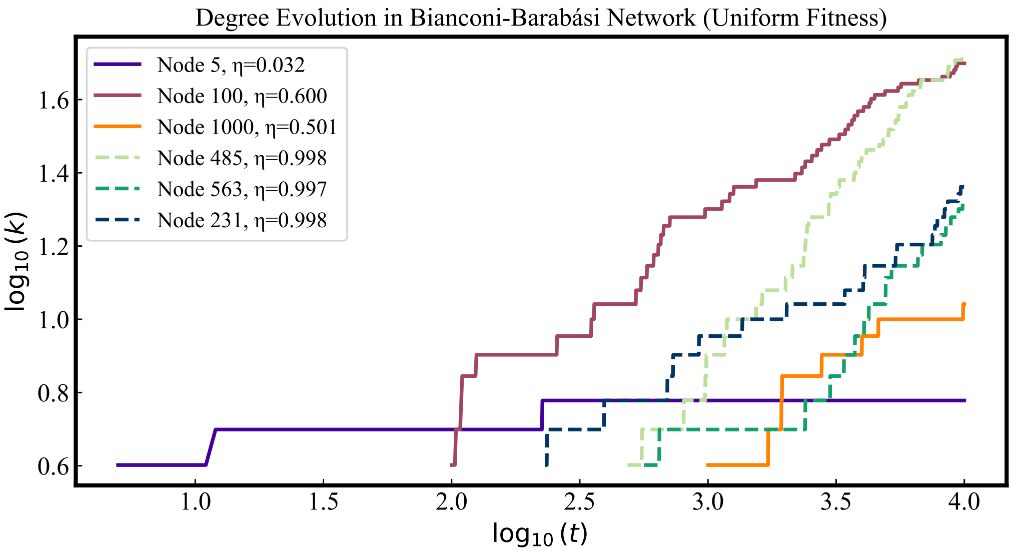

Figure 8: Degree evolution of some nodes of the uniform fitness distribution with $m = 4$, $N = 10^4$ and 1 configuration. We observe steeper increase of the degree of higher fitness nodes since $k \sim t^{\beta(\eta)}$ where $\beta(\eta) = \eta/C$.

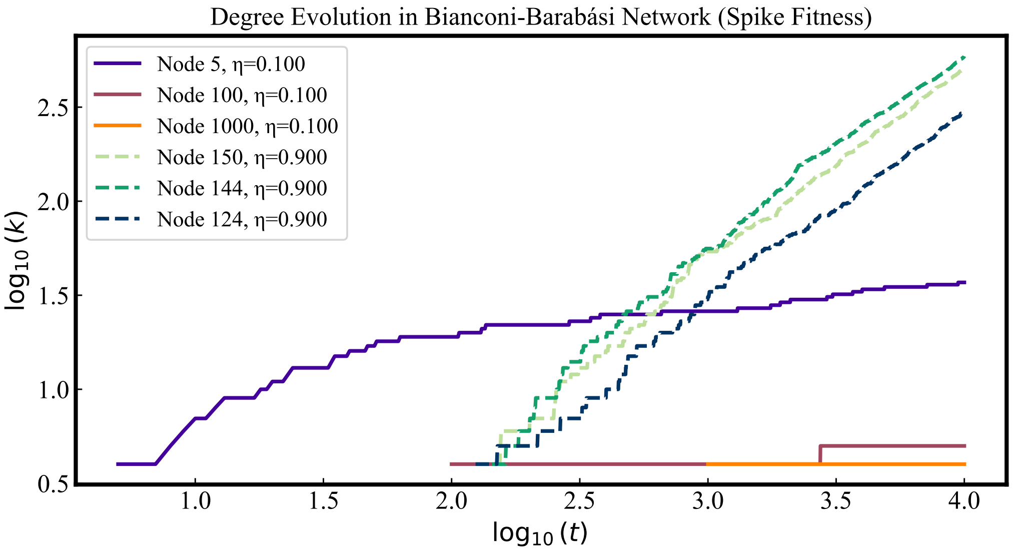

Figure 9: Degree evolution of some nodes of the spiked fitness distribution with $m = 4$, $N = 10^4$ and 1 configuration. We observe steeper increase of the degree of higher fitness nodes since $k \sim t^{\beta(\eta)}$ where $\beta(\eta) = \eta/C$.

■

Scale-free networks via copying and optimization (25 points).

In Section 5.9 of the textbook, we are introduced to multiple ways of generating networks with heavy-tail (scale-free) degree distributions: the Link-Selection Model, the Copying Model, and Optimization Model.

(a) Implement the Optimization Model.

(b) Using your implementation of the Optimization Model, reproduce Figure 5.16b (degree distributions when $\delta = 0.1$, $\delta = 10$, and $\delta = 1000$) and Figure 5.16c (visualizations of the networks at each δ value; Note: your visualizations do not need to look exactly like the networks from the textbook, but they should capture similar information).

(c) Implement the Link-Selection Model and the Copying Model.

(d) Create a figure with four subplots—inspired by the middle panel of Figure 5.16d that plot the preferential attachment measure $\Pi(k)$ of networks sampled from each of the following network generating models: 1) the Barabási-Albert model (with $N = 10000$, $m = 1$), 2) the Copying Model, 3) the Link-Selection Model, 4) the Optimization Model (with $\delta = 10$). Note the caption of Figure 5.16: "We used the method described in Section 5.6 to measure the preferential attachment function. Starting from a network with $N = 10000$ nodes we added a new node and measured the degree of the node that it connected to. We repeated this procedure 10000 times, obtaining $\Pi(k)$."

(e) Describe what you observe in the figure created above.

Solution

(a) The implementation is pretty straightforward. We start with an empty network and add the first node. All added nodes have a random position in the unit square. The initial node has resources cost 0. Then we add new nodes until we have N nodes. Each new node i we calculate the the cost $C_j = \delta d_{i,j} + h_j$ to connect to every node j, where $d_{i,j}$ is the distance between i and j and h_j is the resource cost of j. To realize the connection the new node i chooses j with the smallest cost.

□

(b) Figure 10 show the degree distribution of the Optimization Model for $N = 10^4$ and values of δ equal to 0.1, 10, and 1000. We did not take configurational averages due to time constraints, but you can easily do as we did before and have very smooth curves. All three subplots have the same horizontal and vertical axis to really show the difference between the distributions.

Figure 11 shows the visualization for the same deltas, but for $N = 10^3$ for performance reasons. Each node sizes is proportional to it's degree.

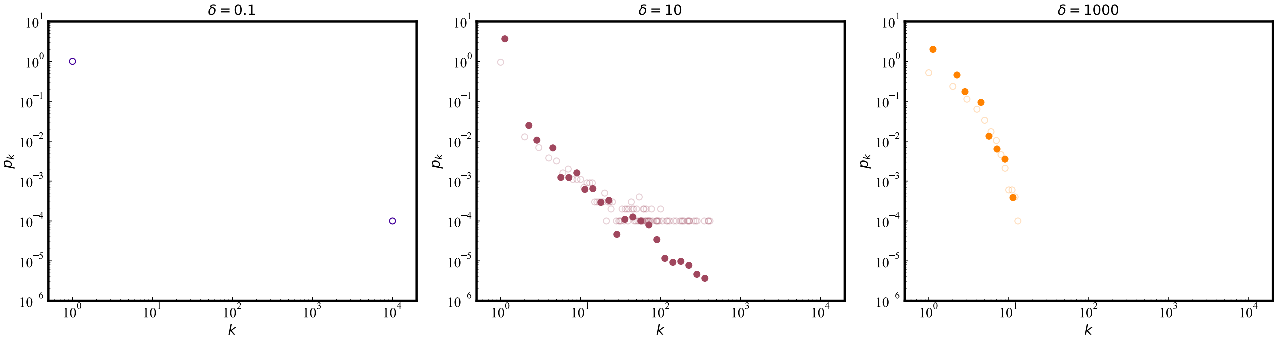

Figure 10: Degree distribution of the Optimization Model for $N = 10^4$ and different δ values. Hollow circles represent the raw data and the solid circles are the log-binned data.

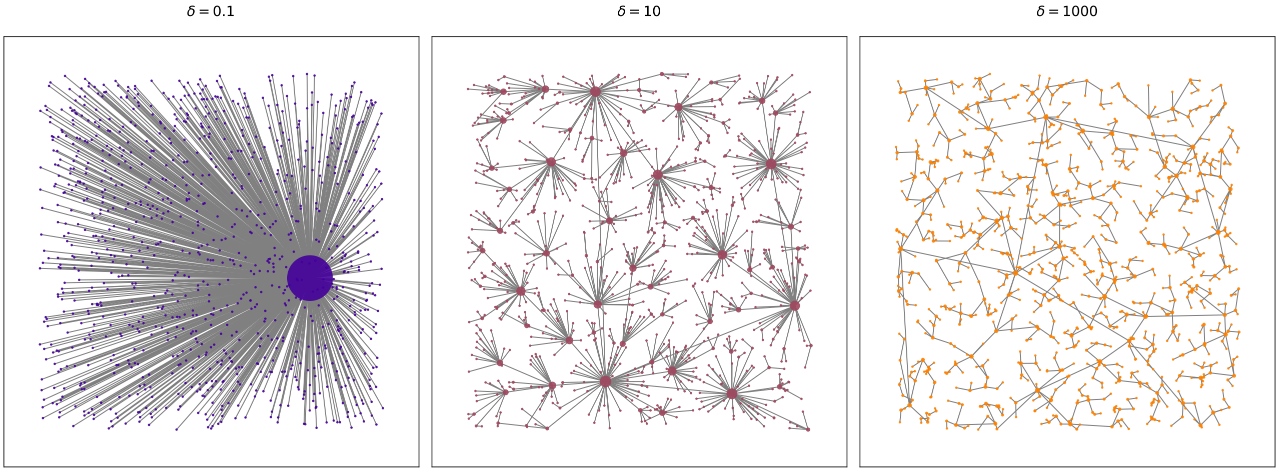

Figure 11: Visualization of the Optimization Model for $N = 10^4$ and different δ values. Node size is proportional to it's degree.

□

(c) For the Link-Selection we start with two nodes connected to each other. Then we add nodes until we have N new nodes. Each new node uniformly randomly selects one of the links in the network, then it also uniformly randomly selects one of the two nodes in that link and links itself to it. For the Copying model we start with a directed graph of two nodes linking to eachother. Then we add new nodes until we have N nodes. Each new node will randomly select another node in the network. It will link with probability p to that node or, with probability $1 - p$, to one of it's neighbors uniformly randomly chosen.

□

(d) We used a different method than the one discussed in Section 5.7. For each network, we repeated $N_{\text{iter}} = 10^4$ trials. For each trial, we do the same process of adding a new node, but instead of adding it we store which node was selected. After all the trials are done, we compute the fraction of selections of each node, that's our $\Pi(k_{i]})$, then we export these probabilities and their respective degrees and plot them. Be aware that $\Pi(k_i)$ is very much different than $\Pi(k)$! $\Pi(k_i)$ is the probability of a node i with degree k_i be chosen, while $\Pi(k)$ is the probability of any node with degree k be chosen. Thus we have to be cautious on how we group (or in this case we don't group) together our counts (this gave me a headache...).

□

(e) Figure 12 presents the preferential attachment function $\Pi(k_i)$ for all the discussed methods in this chapter + the standard BA model. We can see the so desired and beautiful linearity $\Pi(k_i) \sim k$ in all of the methods. This evidences that there are many ways that the linear preferential attachment can rise naturally by other mechanisms. It can be branded in, as in the BA network, or can be a consequence of other mechanisms such as the Optimization Network ($\delta = 10$), Link Selection, or Copying.

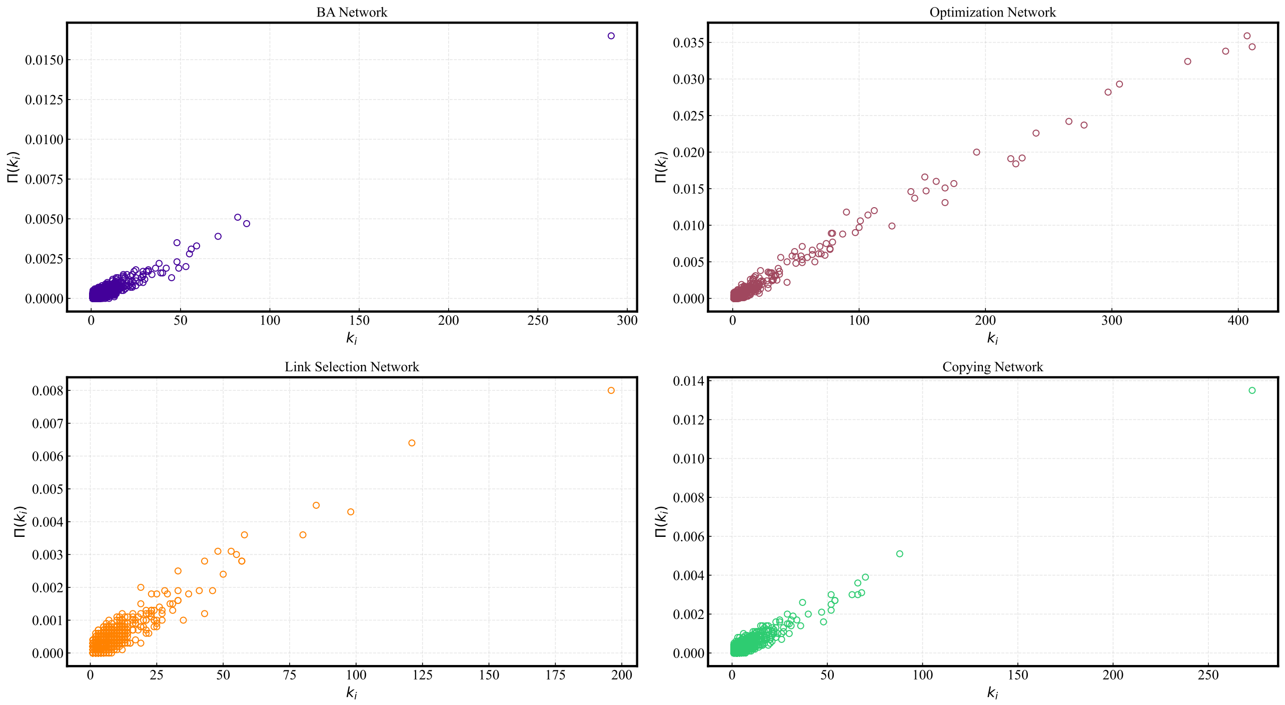

Figure 12: Preferential attachment $\Pi(k_i)$ versus k_i for different network creation mechanisms. For all networks we chose $N = 10^4$ and did $N_{\text{iter}} = 10^4$ to sample $\Pi(k_i)$. For the BA network, we used $m = 1$. For the Optimization model we used $\delta = 10$. For the Copying model we used $p = 0.5$. All four methods show linear preferential attachment, even with different network creation mechanisms.

■

All the code from this assignment can be found: Here.