Final Exam

Sriram Balasubramanian

December 18, 2021

For the questions with a coding component, please refer to the Jupyter notebook in this Github repository
.

1. Downloaded, see accompanying code.
2. After running DFS on the graph, we find that there is only 1 connected component with 7624 nodes in the graph.
3. The degree distribution is plotted using linear binning and log binning in Figure 1. The exact values of p_k for each k can be found in the Jupyter notebook
4. We fit the log-binned data to the form $p_k = Ce^{-\alpha k}k^{-\tau}$.

   Taking log on both sides and rearranging, we have $\log p_k = \log C - \alpha k - \tau \log k$. Using a linear least squares solver to solve for C, α, and τ, we get $C = 0.3147, \alpha = 0.0447, \tau = 1.0006$.

   The estimated p_k vs k curve is plotted in Figure 1
5. Using the `average_shortest_path_length` from the `networkx` package in Python, we can compute the average shortest path length in the actual network as 5.232 .

   From [1], we obtain the expression of the average shortest path length ℓ as

$$\ell = \frac{\ln[(N-1)(z_2 - z_1) + z_1^2] - \ln z_1^2}{\ln z_2/z_1} \approx 1 + \frac{\ln N - \ln z_1}{\ln z_2 - \ln z_1}$$

   where z_1 is the average number of nearest neighbors (or the average degree), and z_2 is the average number of second nearest neighbors. Using generating functions, we obtain $z_1 = G_0'(1)$ and $z_2 = G_0'(G_1(1))G_1'(1) = G_0'(1)G_1'(1)$ where G_0 and G_1 are the generating functions for the degree distribution and excess degree distribution respectively.

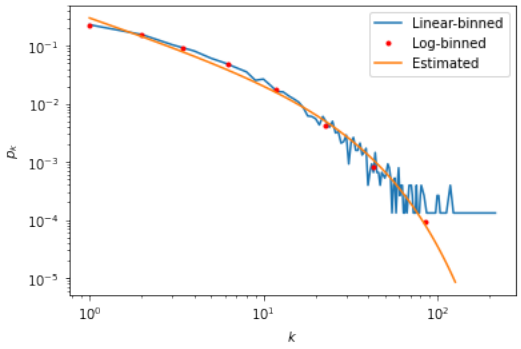

Figure 1: Degree distribution p_k vs k using linear binning, log binning and estimating parameters using least squares

For the exponential power cut-off random graph, we have the following equations from [1]

$$z_1 = \frac{\mathrm{Li}_{\tau-1}(e^{-\alpha})}{\mathrm{Li}_{\tau}(e^{-\alpha})} \tag{1}$$

$$z_2 = \frac{\mathrm{Li}_{\tau-2}(e^{-\alpha}) - \mathrm{Li}_{\tau-1}(e^{-\alpha})}{\mathrm{Li}_{\tau}(e^{-\alpha})} \tag{2}$$

$$\tag{3}$$

where $\mathrm{Li}_s(z) = \sum_{k=1}^{\infty} \frac{z^k}{k^s}$ is the polylogarithm function.

Using the estimated values of α and τ for computing z_1 and z_2 and plugging it into the equation for ℓ, we get $\ell \approx 3.252$

6. The clustering coefficient for the actual network is estimated to be ≈ 0.1786

   We can calculate the clustering coefficient for the random graph with same degree distribution as follows.

   Let v be an arbitrary node of degree 2 or more. We randomly pick two of its first neighbors i and j with excess degrees k_i and k_j . The probability that there is a link between i and j is

   $$\frac{k_i k_j}{nz}$$

   where n is the total number of nodes and $z = \langle k \rangle$ is the average node degree.

   Then assuming that k_i and k_j are independent,

   $$C_{random} = E_{k_i, k_j \sim Q(k)} \left[\frac{k_i k_j}{nz} \right] = \frac{E_{k \sim Q(k)}[k]^2}{n\langle k \rangle}$$

   where $Q(k)$ is the excess degree distribution. Now,

   $$E_{k \sim Q(k)}[k] = \sum_{k=0}^{\infty} k q_k = \frac{1}{z} \sum_{k=0}^{\infty} k(k+1) p_{k+1} = \frac{1}{z} \sum_{k=0}^{\infty} k^2 p_k - k p_k = \frac{\langle k^2 \rangle - \langle k \rangle}{\langle k \rangle}$$

   Plugging this into the formula for C_{random}, we get

   $$C_{random} = \frac{[\langle k^2 \rangle - \langle k \rangle]^2}{n\langle k \rangle^3} = \frac{z_2^2}{n z_1^3}$$

   Using the values of z_1 and z_2 from the previous question, we get $C_{random} \approx 0.0089$

7. The clustering coefficient for the random graph is much lower than that of the actual graph. This is because the parameters of the probability distribution we estimated from the log-binned data predict much lower high degree nodes than are actually found in the model, as can be seen in 1 where the actual curve is much above the curve of the estimated probability distribution. These high degree nodes contribute disproportionately to the clustering coefficient. Since the model predicts lower number of high degree nodes, it also predicts low clustering coefficient.

   The average shortest path length for the random graph is also lower that that of the actual graph. While the clustering coefficient of the random graph is indeed much lower than the clustering coefficient of the actual network, there exist quite a few more nodes with only 1 neighbour as compared to what the model predicts. This means that distance between these nodes and the rest are higher than what the model predicts, which implies that average shortest path length is higher than the prediction by the model.

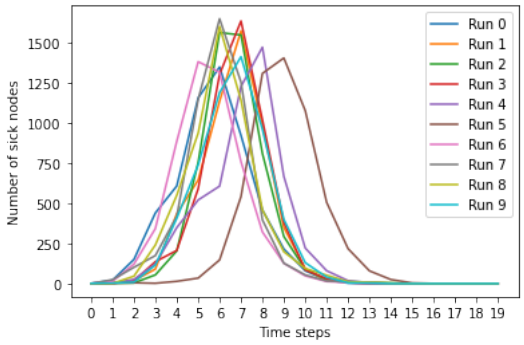

Figure 2: 10 random runs of the SIR model

8. We know that

$$T_c = \frac{G_0'(1)}{G_0''(1)} = \frac{\langle k \rangle}{\langle k^2 \rangle - \langle k \rangle}$$

Using empirical estimates for $\langle k^2 \rangle \approx 185.437$ and $\langle k \rangle \approx 7.294$, we get $T_c \approx 0.041$

9. We run the SIR model for 20 timesteps 10 times with $T = 0.4$, and plot the number of sick nodes at each time step for all 10 runs in Figure 2

We observe an epidemic in all 10 runs. Whenever an epidemic is observed, around 25 - 29% of nodes are infected at the peak of the epidemic. The exact fractions for each of the 10 runs are $0.927, 0.9214, 0.921, 0.9273, 0.9101, 0.9292, 0.9242, 0.9284, 0.9206, 0.9194$. The mean and standard deviation of the fraction of affected nodes during epidemics is 0.9229 and 0.0054

10. (a) The critical transmissibility $T_c = \frac{G_0'(1)}{G_0''(1)} = \frac{z_1}{z_2} \approx 0.0457$

(b) We solve the following equations to get the fraction of nodes infected as a function of transmissibility $S(T)$ (fraction of nodes in the giant component in the transmissibility graph)

$$H_1(1,T) = G_1(H_1(1,T),T) \tag{4}$$
$$H_0(1,T) = G_0(H_1(1,T),T) \tag{5}$$
$$S(T) = 1 - H_0(1,T) \tag{6}$$
$$\tag{7}$$

We first solve $h = G_1(h,T) = G_1(1 + (h-1)T)$. From [1], we know that $G_1(x) = \frac{\mathrm{Li}_{\tau-1}(xe^{-\alpha})}{x\mathrm{Li}_{\tau-1}(e^{-\alpha})}$. Therefore, we solve the following equation:

$$h = \frac{\mathrm{Li}_{\tau-1}((1+(h-1)T)e^{-\alpha})}{(1+(h-1)T)\mathrm{Li}_{\tau-1}(e^{-\alpha})}$$

Since we only need to know $S(0.4)$, we substitute $T = 0.4$ and solve the equation numerically to get $h \approx 0.1144$

Now, $S(0.4) = 1 - G_0(h, 0.4) = 1 - G_0(1 + 0.4(h-1))$. We have $G_0(x) = \frac{\mathrm{Li}_\tau(xe^{-\alpha})}{\mathrm{Li}_\tau(e^{-\alpha})}$. Substituting the value of h, we get $S(0.4) = 1 - G_0(1 + 0.4(0.11)) \approx 0.6928$.

(c) Suppose that a fraction of nodes p is randomly removed from the transmission graph. From [2], we know that $G^*{}_0(x,p,T) = G_0(1+(1-p)(x-1),T)$ where $G^*{}_0(x)$ is the new generating function after removing the nodes. But $G_0(x,T) = G_0(1+(x-1)T)$, therefore, $G^*{}_0(x,p,T) = G_0(1+(1-p)T(x-1))$. The criterion for this new graph to have a giant component is $(1-p)T \geq \frac{G_0'(1)}{G_0''(1)} \approx 0.0457$. Therefore, $1 - p \geq 0.0457/0.4 = 0.11425$, or $p \leq 0.88575 = p_c$ If more than p_c fraction of nodes

are removed (in this case vaccinated), the graph no longer has a giant component and there will not be an epidemic.

The critical transmissibility of the actual graph is 0.041 which is slightly lower than the theoretical estimate 0.0457. Also, the theoretical estimate of the fraction of nodes affected by the epidemic which is around 0.6928 is much smaller than the fractions observed in the 10 runs in Figure 2. This is because the theoretical model predicts much fewer high degree nodes as compared to the actual graph. This is also reflected in the fact that the clustering coefficient predicted by the theoretical model is much lower than the actual clustering coefficient. These high degree nodes increase the connectivity of the graph and thus much more nodes are affected. Since these nodes are few in number, they only increase the critical transmissibility by a small amount, but the presence of even a few such nodes increase the number of affected nodes drastically.

References

[1] M. E. J. Newman, S. H. Strogatz, and D. J. Watts, 2001, Random graphs with arbitrary degree distributions and their applications

[2] Reuven Cohen, Keren Erez, Daniel ben-Avraham, and Shlomo Havlin, 2000, Resilience of the Internet to Random Breakdowns