Assignment 4: Modeling Epidemics

Summer 2023

Overview

The objective of this assignment is to experiment with the concepts we covered in Module-4 about network epidemics, and see how the theoretical results that were derived in class compare to simulation results.

Submission

Please submit your Jupyter Notebook **A4-YOURLASTNAME.ipynb** with **requirements.txt** so that we may be able to replicate your Python dependencies to run your code as needed.

With Anaconda, you can do this by running:
conda list -e > requirements.txt

Ensure all graphs and plots are properly labeled with unit labels and titles for x & y axes. Producing readable, interpretable graphics is part of the grade as it indicates understanding of the content – **there may be point deductions if plots are not properly labeled.**

Getting Started

Assignment 4 requires Epidemics on Network python module available here. You can download this module and run the examples given in the documentation to become familiar with it.

You can install the library using: `pip install EoN`

This homework, especially parts 2 and 3, might take several minutes to run. Be aware of this and plan to complete it accordingly.

****IMPORTANT** As with assignment 3 the structure has been modified to have several subsections in each part. The first few subsections are meant to just define useful functions and the final subsection of each part is where the functions are called and the analysis is done. If you are confused about how a function is meant to be used, check the final subsection in each part to see how they are being called. This should clear up a lot of potential points of confusion early on.**

Part 1: Outbreak Modeling [40 Points]

The file **"fludata.txt"** has the list of students, teachers and staff at a school. The interaction between them was measured based on the proximity of the sensors that people were carrying on them. The data file has three columns, the first two columns are the IDs of the people and the third column is the number of interactions.

Construct an undirected graph from that text file using functions in the networkX module. For the purpose of this assignment, we consider only the **unweighted** graph (i.e., you can ignore the third column). That is to say, even though the graph we provide does have weights, you will simply ignore them.

1. [10 points] Suppose there is a pathogen with **transmission rate of 0.01** and **recovery rate of 0.5**. Suppose that an outbreak started at node 325 ("patient-0"). Complete the ***simulate_outbreak*** function to simulate an outbreak under the SIS model using the provided parameters. The function should return a list of length n_iter containing simulation runs where n_iter is an argument to the function.

   Important: *When running your simulations, you will want to discard the outbreaks that died out stochastically. To do this, check whether the number of infected nodes at the last time step is 0 and ignore them.*

   Additionally, complete the ***plot_outbreaks*** function to visualize the results of the ***simulate_outbreak*** function. Show the results for each of the simulations on a single plot and break each simulation into 2 lines, one for the number of infected and the other for number of susceptible over time. Make sure to properly label these lines and to create a legend identifying which lines are which.

2. [10 points] In the lecture we modeled the initial exponential increase of the number of infected nodes as $I(t) \approx I_0 e^{t/\tau}$, where τ is a time constant. Note that here $I_0 = 1$ as only one node was infected initially. Now, complete the ***get_exponent*** function to fit an exponent to the curve of the number of infections. Choose only the initial portion of the outbreak, say for $I(t) \leq 100$ (the exponential region of the outbreak, where the *number of infected* is less than or equal to 100) and return the estimated time constant τ.

   Hint*: scipy.optimize.curve_fit is a helpful function to fit the exponent to a curve.*

   Additionally, complete the ***plot_curve_fit*** function to plot both the actual number of infected and the theoretical curve given a value of τ (for values of Infected < 100). This function should also compute the r-squared between the two curves and print the value for τ and r-squared in the title of the plot. Again, make sure to label both curves and create a legend identifying which is which.

3. [5 points] In the lecture and textbook we discussed theoretical values for τ that can be calculated from properties of the graph and the dynamics of the infection spread. Complete the ***calculate_theoretical_taus*** function to compute:

- The **random distribution** shown in the Lesson 9 Canvas lecture "*SIS Model*".
- The **arbitrary distribution from the Canvas lectures** shown in the Lesson 9 Canvas lecture "*Summary of SI, SIS, SIR Models with Arbitrary Degree Distribution*".
- The **arbitrary distribution from the textbook** found in Ch. 10, Equation 10.21.

Additionally, complete the ***compare_taus*** function to show a boxplot of the distribution of sample τ's calculated from simulation runs (see 1.5 to understand where these come from). Visualize the theoretical calculations as dots on the box plot. Again, label each of these dots with the calculation used to generate them.

4. [10 points] Complete the ***calculate_theoretical_endemic_size*** function to compute the percentage of the population that remains infected at the endemic state.

   Then, complete the ***compare_endemic_sizes*** function to plot the distribution of endemic sizes across several simulation runs as a boxplot, and compare it with the theoretical calculation for endemic size as a single dot, similarly to the previous subsection.

5. [5 points] Run the code provided in cell 1.5 and look at the resulting figures. How good of a fit is the exponential curve in section 1.2? Explain how the theoretical estimates in 1.3 & 1.4 compare to the empirical distribution and indicate which you would consider a reasonable fit for the data.

Part 2: Transmission Rate [25 Points]

Next, let us vary the transmission rate and see how it affects the spread of infection. Since we know that only the ratio of the transmission rate and the recovery rate matters, let us **keep the recovery rate constant at 0.5** and vary only the transmission rate.

1. [10 points] Complete the ***simulate_beta_sweep*** function to vary the transmission rate over a range of beta values between beta_min, beta_max with beta_samples number of points. For each value of the transmission rate, compute 5 simulations to avoid outliers. You can reuse your ***simulate_outbreak*** function from Part 1 in this function.

   Next, complete the ***extract_average_tau*** function to return a list of the average τ value calculated over the five simulation runs for EACH beta value.

   Finally, complete the ***plot_beta_tau_curves*** function to show the exponential curve given by the τ values for each beta value. Use a log scale on the y-axis and make sure that each line has its own color. This function should be similar to the ***plot_curve_fit*** function in part 1.2, but you will be showing a series of exponentials instead of comparing an experimental with a theoretical curve.

2. [10 points] Complete the ***extract_average_end*** function to return a list of the average endemic size calculated over the five simulation runs for EACH beta value.

Next, complete the ***calculate_theoretical_endemic*** function to find the minimum theoretical beta values of the transmission rate for an epidemic to occur. Calculate this minimum based on the equations derived in lecture for both the random distribution and the arbitrary distribution. Also, calculate the theoretical endemic size for each value of beta.

Finally, complete the ***compare_endemic_sizes_vs_beta*** function to plot the average endemic sizes and theoretical endemic sizes as a curve vs beta. Additionally, plot the minimum values for beta to start an endemic as vertical lines. Make sure to label each line and provide a legend.

3. [5 points] Run the code provided in cell 2.3 and look at the resulting figures. How similar is the theoretical to experimental endemic sizes? How closely do the minimum beta values provide a reasonable lower bound for the start of an endemic?

Part 3: Patient-0 Centrality & τ [30 Points]

Now, let us see how the choice of “patient-0” affects the spread of an outbreak. Consider every node of the network as patient-0, and run the SIS model using the parameters in Part 1 to compute . Run the simulation with each node in the simulation as patient-0. **Hint**: You can skip cases where the infection quickly diminishes to 0.

1. [10 points] Complete the ***sweep_initial_infected*** function to complete a single simulation run for each node in the graph as the initial infected. Check for runs that stochastically die out and do not save those. Return the list of simulation run results and a list of nodes (integer IDs) where the simulation was successful.

   Additionally, complete the ***compute_centrality*** function to calculate the: degree centrality, closeness centrality (with wf_improved=false), betweenness centrality, and eigenvector centrality of the graph. Remember to use the **unweighted** centrality metrics. Return the centralities for each node where the simulation was kept in the previous function.

   Hint: *We provide “nodes” as an argument which is meant to represent the second output of the previous function. You can use this to filter for centralities of only these nodes before you return them. Check the cell for 3.3 to see exactly how this is used.*

2. [15 points] Complete the ***calculate_pearson_correlation*** to compute the Pearson correlation coefficient between each centrality metric and τ, along with a confidence level for that correlation.

   Additionally, complete the ***plot_centrality_vs_tau*** function to plot a scatter plot between the τ value that corresponds to each node, and different centrality metrics of that node: degree centrality, closeness centrality, betweenness centrality, and eigenvector centrality. Do this all as one figure with four subfigures. Include the Pearson correlation

values in the title for each scatter plot. Remember to use the **unweighted** centrality metrics.

3. [5 points] Rank these centrality metrics based on Pearson's correlation coefficient, and determine which metrics can be a better predictor of how fast an outbreak will spread from the initial node. Analyze your results. That is, do the results match your intuition? If they differ, why might that be?

Part 4: Knowledge Question [5 Points]

Answer the following food for thought question from Lesson 10 – Submodularity of Objective Function:

Prove that a non-negative linear combination of a set of submodular functions is also a submodular function.

Hint: *Make sure you understand the definition of linearity.*