MENG 21200 Project - Analyzing Network Models to Understand Localized Effects

University of Chicago, February 21, 2025

This year, your midterm project will cover network modeling. You will investigate a new concept you may not have been exposed to before using the tools you have gained in class. Be sure to show us your thinking: create any plots or graphics that you find useful, and use markdown cells to explain your thoughts, neatly write any equations or symbols, and interpret your results.

Network Modeling

Systems of differential equations that model the behavior of an entire population often assume that every individual interacts uniformly with the rest of the population. However, there are many systems that scientists wish to model where this is not a good assumption. One example is the transmission of disease. People interact with their friends, neighbors, and co-workers rather than the entire population. Therefore, a good disease model requires an emphasis on modeling *local* interactions and responses.

Network modeling is a powerful approach for understanding and simulating complex systems, including the spread of diseases. It represents a system as a collection of interconnected entities. The key components are nodes and edges. **Nodes** represent individual entities within the system. **Edges** represent the relationships or interactions between individual nodes. When visualizing a network, nodes are represented as points and edges as lines connecting points. The **degree** of a node refers to the number of edges connected to that node. Here is an article on common types of networks for more background. In a disease network model, interactions are limited to direct neighbors (*i.e.*, connected by an edge), making for a much more realistic model of disease transmission. Virologists therefore commonly model diseases by creating a network model representing a population to capture more nuanced local effects arising from social networks.

Your goal is to create network models to simulate the spread of a disease through a population. A node represents a person, and an edge between two nodes signifies that two people have contact that could lead to disease transmission. The model you will investigate is called SEIR, consisting of **S**usceptible, **E**xposed, **I**nfected, or **R**ecovered populations[1][2][3], and it was used to model COVID during the height of the pandemic.

To create the disease network, each node i with degree k_i is assigned a state vector [S, E, I, R] describing the likelihood of an individual being in each state at a given time (kind of like a quantum state). Because they represent probabilities, $S + E + I + R = 1$. As examples, a node in the state [1, 0, 0, 0] means the person is 100% likely to be susceptible to disease. In contrast, a node in the state [0, 0.5, 0.5, 0] represents a person who is 50% likely to be exposed and 50% likely to be infected. For a node i, these states evolve as

$$\begin{aligned} \frac{dS_i}{dt} &= -\beta S_i P_{I,i} \\ \frac{dE_i}{dt} &= \beta S_i P_{I,i} - \sigma E_i \\ \frac{dI_i}{dt} &= \sigma E_i - \gamma I_i \\ \frac{dR_i}{dt} &= \gamma I_i \end{aligned} \tag{1}$$

where β is the infection rate, σ is the rate at which exposed individuals become infectious, γ is the recovery rate, and N is the total population size. $P_{I,i}$ is the infection pressure, or average "infectiousness" of the node i's neighbors. $P_{I,i}$ can be represented as

$$P_{I,i} = \frac{\sum_{j \in \text{neighbors}(i)} I_j}{k_i} \tag{2}$$

In programming terms, for each neighbor j of node i, the value of I_j (the infectious state of neighbor j) is summed. This sum is then normalized by dividing by the degree of node i. Therefore, the infection pressure gives a normalized measure of how much "infectiousness" i experiences based on its local network connections. Because the number of neighbors and the probability that a neighbor is infected varies from node to node, the infection pressure needs to be calculated for each node and evolves with time.

1 Creating and Analyzing a Simple Disease Network

Prepare a Jupyter notebook that answers the following questions. You are recommended to review Appendix A on page 4 of this document for useful functions in the NetworkX library before you get started.

(a) Using the NetworkX python package, create a 2-D lattice graph, a small-world network, and a scale-free network with $N = 100$ nodes. These networks represent a small group of people with various connections between them. Visualize each network.

(b) Compare each network by calculating and interpreting the degree distribution (*i.e.*, the distribution of number of nodes with a certain degree).

(c) The disease progresses with the rates $\beta = 0.8$ days^{-1}, $\sigma = 0.3$ days^{-1}, and $\gamma = 0.3$ days^{-1}. Assign an initial state [S, E, I, R] to each node in the networks that you have created, where every node is susceptible except for 10 randomly selected nodes (10% of the 100 nodes in each network) that begin as exposed. Solve the system of ODEs over time. Visualize the spread of the disease through your population by plotting the total numbers of S, E, I, and R for the population as a function of time.

(d) Plot the time-dependent behavior of individual nodes in the system, presenting the behavior of 5 to 10 distinct nodes. Compare and discuss your observations of the behavior of the individual nodes to that of the overall population.

(e) For the lattice network, create several contour plots at various time steps to visualize the propagation of disease spatially through the network.

(f) Integrate the generalized SEIR model that describes the behavior of an entire population that uniformly interacts (*i.e.*, the system without the network model and limited connectivity). Plot the results, and assess how the general population model compares to the network models that you have already analyzed. Then estimate effective β, σ, and γ parameters for the general population model that enables it to most closely match the behavior of the network models. Can you rationalize the influence of the network's connectivity on these rates?

(g) Find the cumulative populations of S, E, I, and R over time for each network. Differentiate the global S, E, I, and R populations with respect to time. Plot and discuss the results.

(h) Perform a linear regression between the average degree of the networks and the maximum number of infected people. Are they linearly related?

Potential opportunities for you to explore further:

- Explore the β, σ, and γ parameter space, and investigate if the disease spreads differently.
- Create an animation or movie of contour plots indicating the evolution of the disease through the lattice network with time.

2 Adding Sophistication to the Model

(a) The disease now evolves to become capable of reinfection, which means that recovered individuals now become susceptible to the disease again at a rate δ. Update your system of equations and explain the change you made. How does adding a reinfection rate increase or decrease the spread of the disease? Does it vary between network structures?

(b) Plot the maximum number of individuals infected as a function of beta and gamma for $\delta = 0.5$ days^{-1}.

(c) Issue positions to the nodes in each network and add a geographic distance effect where the infection pressure decreases exponentially with the distance between nodes. How does this change increase or decrease the speed and depth of the spread of the disease?

3 Extending your Knowledge of Network Modeling to Other Systems

Networks are also useful for modeling gene regulatory systems. For the final part of the project, you will create a basic Boolean gene regulatory network model. Each node represents a gene that can be either active or inactive (*i.e.*, a binary ON or OFF), and edges represent interaction pathways between genes [4]. Only the nodes connected to a node contribute to the activity of the gene.

(a) As in the previous sections, create three different gene networks using NetworkX, each with 200 nodes. You may use the same networks from the previous part if you wish. Randomly assign half of the genes to be initially active and the rest to be inactive. In each time step, activate a gene if more than half of its neighboring genes are active.

(b) For each network, plot the number of active genes over time, and record the final number of active genes and the time step at which the system stabilizes (if it does). Change the fraction of genes starting as active. Does this change the final number of active genes? You may find it valuable in this discussion to plot the steady-state number of active genes versus the fraction of genes that were initially active.

(c) Compare the three networks based on their plots and the recorded metrics. How do their behaviors differ, and what might this suggest about the structure of gene regulatory networks in nature?

Potential opportunities for you to explore further:

- Plot and interpret the global rates of gene activation over time for each network.
- Choose a network type and create a contour plot of final number of active genes as a function of initial fraction of active genes and activation threshold.
- Make the Boolean gene network a bit more complicated[5] by specifying different kinds of genes which turn each other on and off (*e.g.*, gene A turns on gene B turns off gene C).

A Appendix of Functions in NetworkX

Here are a few functions in NetworkX you may find helpful, using the example of a ring-shaped network. Refer to the NetworkX documentation to find more functions and examples, but these should cover the minimum of what you need to complete the project.

```
import networkx as nx

G = nx.cycle_graph(N) # generates a ring network with N nodes

dir(G) # displays the attributes of object G

G.number_of_nodes() # returns number of nodes of ring network G

G.number_of_edges() # returns number of edges of ring network G

G.nodes() # returns an array of nodes of ring network G

G.degree(node) # returns the degree of a node of ring network G

G.neighbors(node) #returns the neighbors of a node of ring network G

# issues circular positions to nodes for plotting (or geographic effects)
pos = nx.circular_layout(G)

# visualizes a ring network
nx.draw(G, pos, ax=ax,
        node_color='lightblue',
        node_size=100,
        with_labels=False,
        edge_color='gray',
        alpha=0.7)
```

Below is an example of creating a class of networks and an example of issuing positions to each node. It is not required to create a class of networks, but it may save time.

```
    def create_networks(N=100):
    return {
        "Ring": nx.cycle_graph(N),
        "Lattice": nx.grid_2d_graph(int(np.sqrt(N)), int(np.sqrt(N))),
        "Random": nx.erdos_renyi_graph(N, 0.1),
    }

    networks = create_networks()
    # issue positions for each network in networks
    for idx, (name, G) in enumerate(networks.items()):
        if name == "Ring":
            pos = nx.circular_layout(G)
        if name == "Lattice":
            pos = {(x,y):(y,-x) for x,y in G.nodes()}
        else:
            pos = nx.spring_layout(G, seed=42)
            # specify a seed for reproducibility
```

References

[1] M. E. J. Newman. "Spread of epidemic disease on networks". In: *Phys. Rev. E* 66 (1 2002), p. 016128. DOI: 10.1103/PhysRevE.66.016128. URL: https://link.aps.org/doi/10.1103/PhysRevE.66.016128.

[2] Romualdo Pastor-Satorras et al. "Epidemic processes in complex networks". In: *Rev. Mod. Phys.* 87 (3 2015), pp. 925–979. DOI: 10.1103/RevModPhys.87.925. URL: https://link.aps.org/doi/10.1103/RevModPhys.87.925.

[3] Jianquan Liu, Jianguo Wu, and Zuliang Gan. "Modeling the epidemic SEIR model with the finite incubation period of COVID-19". In: *Journal of Applied Analysis & Computation* 10.5 (2020), pp. 1963–1986. DOI: 10.11948/20200326001.

[4] Claus Kadelka et al. "A meta-analysis of Boolean network models reveals design principles of gene regulatory networks". In: *Science Advances* 10.2 (2024), eadj0822. DOI: 10.1126/sciadv.adj0822. eprint: https://www.science.org/doi/pdf/10.1126/sciadv.adj0822. URL: https://www.science.org/doi/abs/10.1126/sciadv.adj0822.

[5] Yinghao Wu. *Gene Regulatory Networks (GRN)*. https://einsteinmed.edu/uploadedFiles/labs/Yaohao-Wu/Lecture2.pdf. Accessed on February 06, 2025. 2014.