Modeling Complex Systems (CS/CSYS 6020), Fall 2023

Assignment #3 of 3

Done in teams of 2 or 3. Due on Blackboard by midnight on Friday, November 10th. Your write-up should contain everything but the codes, such as any necessary figures as well as your answers to the questions. A brief justification should accompany your answers to each question. Codes should be submitted separately.

1. The *cascade model* is a simple toy model to help us think about energy flow in directed acyclic graph such as food webs and some power grids. Imagine N nodes with indices $i = 1, \ldots, N$ and place an undirected edge between each unique pair with probability p just as in the Erdős-Rényi random graph seen in class. Now add a direction to every edge such that each edge goes from node i to node $i' < i$. The index i then informs us of the position of a node in a hierarchy (e.g. trophic level in food webs, or distance from source in power grids) and the model can be used to help us understand the distribution of energy flow through the hierarchy.

    a) Explain why this network is acyclic (i.e. contains no cycle).

    b) What is the average in-degree (edges going in) of vertex i? What is the average out-degree (edges going out) of vertex i?

    c) Show that the expected number of edges that run from nodes $i' > i$ to nodes $i' \leq i$ is $(ni - i^2)p$.

    d) Assuming N is even, what are the largest and smallest values of the quantity calculated in *c)* and where do they occur (in i).

2. *Epidemic spreading* on the configuration model allows us to test simple targeted vaccination strategies since not all individuals (nodes) are considered equal. Let us consider a simple example on an endemic Influenza epidemic, and test the impact of vaccination strategies knowing that around 40% of adults tend to get vaccinated.

    a) Write a heterogeneous mean-field system for an SIS epidemic model on a configuration model network. You should obtain a system of equations for $\dot{I}_1, \ldots, \dot{I}_{k_{\max}}$, where $k_{\max}$ is the largest degree in the network. Feel free to nondimensionalize your system if you wish.

    b) Modify the above system of equations by adding new compartments for vaccinated nodes. Assume that around vaccinated nodes, the transmission rate (both to and from these nodes) is reduced by a multiplicative factor (1-ρ). For example, each vaccinated and susceptible individual with degree k should move from compartment S_k^V to I_k^V with rate $(1-\rho)\beta k\theta$ (where θ is defined in Eq. (10) of "Mean-field Models on Networks" handout, but needs to be modified).

c) Using the integrator of your choice, test how strong ρ needs to be to eradicate an outbreak with the following parameters: transmission rate $\beta = 0.3$ (week^{-1}) and recovery rate $\alpha = 1$ (week^{-1}) in a population with a geometric degree distribution $p_k = p(1-p)^k$ with $p = 1/4$. Assume that a random 40% of the population gets vaccinated.

d) Repeat *c)* but now assume that the vaccination campaign convinced the 40% of nodes with highest degree to get vaccinated.

3. Pick a network of your choice on the Colorado Index of Complex Networks. Play around with network visualizations, share one of them in your write-up and note of what information if any you can glean from visualizing the network. Describe the network: What are the nodes? What are the edges? What is the average degree? What are some other static network measures, and why are they significant to this system? Tip: Avoid taking a network larger than a thousand of nodes to help you with the visualization and the next problem.

4. The *voter model* is the simplest computational analogy to social debate and allows us to study the ability of a network to reach a consensus. Nodes are typically in one of two states (say red and blue), and at every time step, a random node adopts the state of one of its neighbors, also chosen randomly.

   a) Using the network you chose in the previous problem, implement the voter model and run it on your network using some initial conditions (i.e. 60% red and 40% blue randomly distributed). Averaged over a few runs, does the network tend reach consensus? If so, how fast? Try a few different initial conditions.

   b) Imagine running the same model on a version of your network randomized with the configuration model (or actually do it). How do you expect the dynamics to compare to the real network? Explain with in words or simulation results.