

Department of Bioengineering

BE3-HMIB – Modelling in Biology (MiB), Dr Guy-Bart Stan & Dr Tom Ouldridge

Training coursework 9

Disease propagation on networks

In this exercise, we will construct some simple networks and analyse the propagation of diseases on those networks. We will use the file ‘SocialNetwork.mat’, available from blackboard, to define a realistic example.

1. Construct an adjacency matrix for an undirected Erdős-Rényi network. The network should have 276 nodes, and each possible link should exist with a probability 0.05 (don’t add self-links). Use the `spy` command to visualise the adjacency matrix, and calculate the total number of connections.
2. The command `load('SocialNetwork.mat')`; will create a matrix `A` of 276 nodes, corresponding to a social network. Visualize it using `spy`, calculate the total number of connections and comment on the differences between the social network and your random graph.
3. A disease spreads through a population by infecting susceptible population members that are adjacent to an infected member. Each day, susceptible members that are adjacent to a single infected member become infected with a probability $p = 0.2$ (if a susceptible member is adjacent to two infected members, each independently infects it with probability 0.2). Infected population members are infectious for one day, then die (if this is too gruesome, you could call them “recovered and immune”). Define a vector “susceptible” of length 276; `susceptible(n)` is 1 if the n th population member is currently susceptible, and 0 otherwise. Initialise all entries to 1. Similarly, define “infected” and “dead” vectors initialised to 0.
 - Randomly infect one individual “ i ” (set `susceptible(i)=0` and `infected(i)=1`). Evolve the system for 100 days, infecting susceptible neighbours of infected nodes with a probability of 0.2, using your Erdős-Rényi adjacency matrix. Plot the total number of victims (as a fraction of the total population) over this period. Repeat 100 times and plot on the same graph. What tends to happen?
 - Repeat for the social network represented by adjacency matrix `A`. Do these networks behave similarly? Is there any evidence of a difference?

Hint: I find it useful to define a function to implement each step; see help box overleaf.

4. The Government implements a vaccination programme, which reaches 75% of people. Restart the simulation for matrix `A`, but this time set `susceptible(i)=0` with a probability of 75% for each individual i . Pick an initially susceptible victim, infect them, and simulate for 100 days. Repeat for 100 runs and plot the results. What fraction of the initially susceptible people are now typically infected? In what way could this be described as “herd immunity”?
5. Certain communities in the social network described by `A` oppose the vaccination. Rather than setting people to be immune randomly, set the final 75% of individuals in the list to be immune (70 to 276). You can confirm, using `spy`, that the first 69 individuals are indeed well connected within their community, but poorly connected outside. Repeat the above test; how are the results different? Why?

In this coursework you may need to use the following Matlab commands: `rand` and `randi`. You will also find it helpful to define a separate “.m” file defining a function that stochastically generates next step of the disease propagation. The first line of the file (in my case named `infect_new.m`) should look something like: `function [susceptible_new,infected_new,dead_new] = infect_new(adjacency,susceptible,infected,dead, ptrans)`. In other words, you should feed in an adjacency matrix, and vectors listing the current set of susceptible, infected and dead population members, and the infection probability. The function should stochastically infect those that are connected to infected population members, and then update previously infected members to the dead list. You can check the Matlab help by using `help COMMAND`.