Network Science
Spring 2024
Project 2

Due: 20th March 1pm
CID: (insert)

In lectures we were studying epidemics on networks, in particular we produced a naive network SI model to determine the expected spread of disease on a network. In this project, the SIR model will be considered, where another state is added to the SI model, namely a recovery state, whereby someone recovers from the disease. In the SIR model a person can be infected, susceptible and recovered, where the latter is defined as a person who had the disease and recovered, and are therefore no longer infected or susceptible.

We will consider the model in terms of networks and again consider what happens on average for a network of N nodes. Define A_{ij} to be the adjacency matrix for the network under consideration and s_i, x_i, r_i to be the probabilities that node i is susceptible, infected or recovered, respectively. It can be shown that

$$\frac{ds_i}{dt} = -\beta s_i \sum_{j=0}^{N-1} A_{ij} x_j \tag{1}$$

$$\frac{dx_i}{dt} = \beta s_i \sum_{j=0}^{N-1} A_{ij} x_j - \gamma x_i \tag{2}$$

$$\frac{dr_i}{dt} = \gamma x_i \tag{3}$$

where

$$s_i(t) + r_i(t) + x_i(t) = 1. \tag{4}$$

Here, γ and β are model parameters where γ is the recovery rate, i.e. the probability per unit time that an infected individual will recover, and β is the infection rate, i.e. the probability per unit time that an infected individual will be infected. Assume that deaths do not occur. Note that we number the graph nodes from 0 to $N-1$, which is why the sums in the equations are over the same range. For the purpose of this project, we will suppose that at time $t = 0$ there is only one node infected with $x(0) = x0$, and for all other nodes $x(0) = 0$. Moreover, assume $r_i(0) = 0$ for all i, then $s_i(0)$ can be inferred from the governing equations for each i.

1 Part 1

1. (6 marks) Following a similar method used in lectures for the SI model, consider a small perturbation to the equilibrium point of the system of Equations (1)-(4), where everyone is

susceptible. Linearise the resulting equations and show that (2) can be written in matrix form as

$$\frac{d\boldsymbol{x}}{dt} = \beta \mathrm{M}\boldsymbol{x}, \tag{5}$$

where M is a matrix that should be defined.

2. (3 marks) By considering an appropriate form for the perturbation, $\sim e^{\lambda t}$, determine how the perturbation changes with time. In particular, comment on how the spread of disease will depend on the parameters of the model, and the highest degree in the network.

   You may find it useful to use the result from Linear Algebra that given a diagonalisable, $n \times n$ matrix B with eigenvectors $\boldsymbol{v_i}$ and eigenvalues λ_i, the matrix $\bar{\mathrm{B}} = \mathrm{B} + b\boldsymbol{I}$ has the same eigenvectors $\boldsymbol{v_i}$ with eigenvalues $\lambda_i + b$.

2 Part 2

We now look at the SIR model (1)-(4) applied to a given network and use computational analysis to determine how the disease spreads for different recovery and infection rates.

1. (4 marks) First, we will apply the theory of Part 1 to a network.

   (a) Generate a graph using a Barabasi-Albert model in Networkx with $N = 100$ and $m = 2$. Write a code that finds the required eigenvalue of A and M for the graph, to check the theory and results you deduced in Part 1 Question 2 holds for this graph when $\beta = 0.5$ and $\gamma = 0.2$.

   (b) Then consider $\beta = 0.5$, with any γ. What is the critical value of γ, γ_c, for which $\gamma < \gamma_c$ results in a spread of disease on this graph at short times, and for $\gamma > \gamma_c$ the disease will not spread.

   (c) How do the results of (a) and (b) change when $m = 4$, but all other parameters are kept the same? Why is this?

2. (7 marks) Now, we will solve the system of equations (1)-(4) numerically, subject to an initial condition and determine if our linearisation in Part 1 is accurate. You are provided with a function *SIR_model* in the ipynb file that numerically solves the SIR model (1)-(4) on a given network. The network must be in the form of a networkx graph, and other parameters must be prescribed- see the documentation for the function. Note, some aspects of the function are missing.

   (a) Suppose the initial condition for node $i = i0$ is $x_{i0}(0) = x0$, where for all other $i \neq i0$, $x_i(0) = 0$, as well as $r_i(0) = 0$ for all i. Determine what s_i should be at time $t = 0$. Amend the function to include the required initial condition, where $i0$ and $x0$ are prescribed as an input to the function. Furthermore, the function *RHS*, contained within the function *SIR_model*, is incomplete. Add code to complete this function. You should assume that *RHS* will be called a large number of times within one call to *SIR_model* and construct your code accordingly. Keep in mind it should be efficient for large complex simple graphs, though it is recommended that you develop and test your code with much smaller graphs. Note that variables set "above" *RHS* will be available within *RHS*. Note: You have been provided with a code below the function, which can be used as a check for the numerics.

   (b) We will now use the function *SIR_model* to find the numerical solution to the SIR model on the graph generated in Part 2 Q1(a). Consider $\beta = 0.5$, and node $i0$ to be infected at time $t = 0$ with $x0 = 0.01$. Use the time interval $0 \leq t \leq 10$, with

the time step $Nt = 10000$. Consider two different γ, where one $\gamma < \gamma_c$ and $\gamma > \gamma_c$. Pick an initial node to infect (make sure that this node has a degree greater than 2), and carefully analyse and investigate the spread of disease from this initial node on the network, for both γ. For a better understanding of the trends, you may vary the parameters, but conclusions should be made on the given parameters. You may find it useful to focus and present results on the neighbouring nodes of the initially infected node. Compare the growth rate found using the linear theory from Part 1 and Part 2 Q1, with the numerical solution.

3 Details

- You should submit a pdf, with Parts 1 and 2 compiled together. For example, both Parts 1 and 2 could be completed in a Jupyter notebook and a pdf compiled. Or you could compile a pdf version of a latex file answering Part 1, with a pdf of the notebook of Part 2, into one pdf file. Please also submit a ZIP file of the .ipynb file with your code for Part 2. To submit your assignment, go to the *Assesments and Mark Schemes* folder on the module Blackboard page. In the folder there is another folder called *Coursework 2 Drop Box Spring 24* within this folder there are two dropbox's one for the pdf and one for a ZIP file which should include your ipynb file. To convert a file to ZIP, right click on the file and click "compress" and this will generate a ZIP. (these should be named project2_CID.ipynb and project2_CID.pdf.

- Please make sure your CID is written at the top of the project and both pdf and ipynb files.

- You may use numpy, scipy, and matplotlib as needed. You may use networkx as needed. Please do not use any other packages without explicit permission.

- Marking will be based on the correctness of your work, coherence, structure and clarity of the project, as well as efficiency of your codes, and the degree to which your submission reflects a good understanding of the material covered up to the release of this assignment.

- This assignment requires sensible time-management on your part. Do not spend so much time on this assignment that it interferes substantially with your other modules. If you are concerned that your approach to the assignment may require an excessive amount of time, please get in touch with the instructor.

- Questions about the assignment should be asked in private settings. This can be a "private" question on Ed (which is distinct from "anonymous"), asking for a one-on-one meeting during office hours, or during a problem class.

- Please regularly backup your work. For example, you could keep an updated copy of your notebook on OneDrive.

- In order to assign partial credit, we need to understand what your code is doing, so please add comments to the code to help us. Similarly, please include all working and steps for solutions to Part 1.

- It may be helpful to initally develop your code in a Python module (outside of a function) and run it in a qtconsole (or similar Python terminal) so that you can readily access the values of the variables you are using.

- Feel free to use/modify codes that I have provided during the term.