Amsterdam University College

Modelling Real World Problems

Networks: Final Project Lab

- The python-mesa project "SIR modelling" contains a basic cellular automata model for spread of an infectious disease. The following questions are aimed at understanding the model, and providing ideas on how to extend the model and/or use it for your research project.

  1. Run the model and describe what kind of spatial patterns the epidemic forms. Describe how this deviates from the mean field situation, and think about consequences of the pattern formation for the epidemic dynamics.
  2. In the model, the distance at which individuals can infect can be varied. Find the code where this is specified and explore what happens for larger or smaller infection distances. Think on how this affect the R_0 of the disease.
  3. What is the mean field equation for the CA model? How does this relate to the commonly used models for epidemic disease modelling?
  4. Think of how you could add an incubation period to the model (there are several possibilities for doing this). How would this affect the mean field equation?

- In this part of the LAB we will simulate and visualize **SIRS** models over a network. For the second part of the lab you will need to have the plotly module installed. If you did not do this already you can install it using:

  pip install plotly

  as usual.

  - Recall that in SIRS models each individual can be in one of the following states:
    - **S**usceptible: the individual is healthy and can potentially be infected by the virus.

    - **I**nfected: the individual is infected and can infect other individuals.
    - **R**ecovered: the individual is cured and has developed (some) immunity from the virus.

  The probability of going to one state from the other are the following:

    - β: probability that a susceptible individual who comes in contact with an infected individual becomes infected.
    - γ: probability that an infected individual recovers.
    - ξ: probability that a recovered individual becomes susceptible again (loses his immunity).

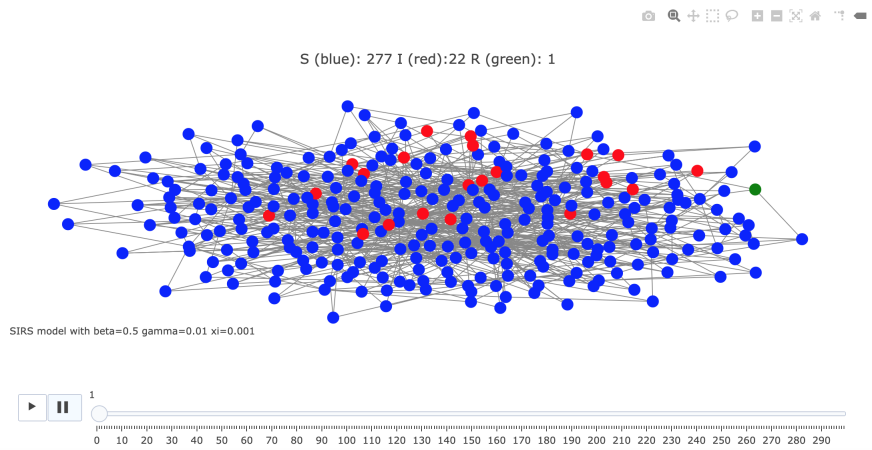
- Write a function that simulates an SIRS model over a graph G. The function should take a graph G, the three parameters of the SIRS model β, γ, and ξ, an integer f representing the length of the simulation, and two disjoint lists of nodes of G, one representing the initial carriers and one representing the initial immune individuals. The function modifies the graph G by adding to each node an attribute called *state* (you can do so by using `nx.set_node_attributes(G,attDict)`). The value of *state* for a node n in G should be a list L of length f. For each $t < f$, $L[t]$ represent the state of the node n at time t in the simulation of the SIRS model. In particular, $L[t] = 0$ if the node n is in state **S** at time t, $L[t] = 1$ if n is in state **I** at time t, and $L[t] = 2$ if n is in state **R** at time t. Finally, the function should return a list D of length f. Each element $D[t]$ of D is a list of three numbers $[S, I, R]$ where S, I, and R, are the number of susceptible, infected, and recovered individual at time t, respectively.
- Visualize the result. In the Canvas module for this lab you can find the code of a mini-library that allows you to visualize the SIRS model and the resulting data. Generate a BA graph with 100 nodes using Networkx and simulate an SIRS model over it using your previous function (decide how to instantiate all the needed parameters). Then, import the Visualizer module in your file and use the function showSIRS to visualize your simulation.

  The function showSIRS works as follows:

    - Input:
        - G: a graph obtained after an SIRS simulation as described before.
        - filename: the name of the html file to be generated which will contain the simulation.
        - beta, gamma, and xi: the parameters of the SIRS model.
        - nb_frames: the length of the simulation (the integer f in the previous exercise).
        - data: the data obtained in the simulation in the format specified before.

* Output: the function outputs a file which allows to visualize the simulation of the SIRS model given as input.

The result of your simulation should look as follows:

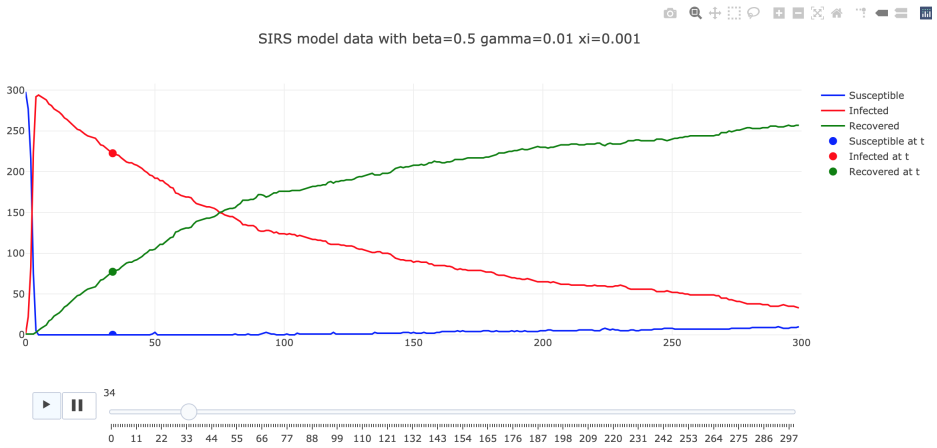


Finally, use the function showData to generate a visualization of the data obtained from the SIRS simulation (the list D that your function returns).

The function showData works as follows:

* Input:
  - data: SIRS data whose format is that described in the previous exercise.
  - filename: the name of the html file to be generated.
  - beta, gamma, and xi: the parameters of the SIRS model.
* Output: the function outputs a file which allows to visualize the data given as input.

The result of your simulation should look as follows:


SIRS model data with beta=0.5 gamma=0.01 xi=0.001
Susceptible
Infected
Recovered
Susceptible at t
Infected at t
Recovered at t
300
250
200
150
100
50
0
0
50
100
150
200
250
300
34
0 11 22 33 44 55 66 77 88 99 110 121 132 143 154 165 176 187 198 209 220 231 242 253 264 275 286 297