Computational Medicine: 02-518/02-718

Carnegie Mellon University

Homework 4

Version: 1.0; updated 11/1/2020

Due: November 15 by 11:59pm

Hand-in: A single PDF to Gradescope that contains the following items:

1. A cover page that lists your name and Andrew id
   - If you worked on a team, indicate who/whom your teammate(s) is/are by their name(s) and Andrew id(s).
     - Each team should have no more than 3 people.
   - Note: Each team should only hand in one pdf. It does not matter who does the actual upload. We will make sure that the grades are entered appropriately.
   - Whoever uploads the pdf to GradeScope should make sure to tell it who your teammates are (if applicable).
2. A PDF export of the Jupyter notebook for question 1

You can combine all the PDFs into one using Adobe Acrobat, or similar tool.

Overview

In this assignment, you will implement and test a simple Agent Based Model (ABM) in question 1 to study the spread of a virus throughout a population. There is no Machine Learning involved. You will just need to implement some routines, run some simulations, and plot some graphs.

This assignment will give you the opportunity to apply methods related to the material covered during lectures 17-20 (weeks 9 & 10).

The goals of the assignment are as follows:

1. To give you the opportunity to implement a simple Agent Based Model
2. To give you the opportunity to simulate different scenarios, and see the impact of various preventative actions on controlling the spread of an infectious agent.

Question 1 (100 points)

This question involves implementing and testing a simple Agent Based Model (ABM).

To complete this question, use the provided jupyter notebook.

Part 1.1 (70 points)

- (10 points) Implement a program that creates a virtual *n*-by-*n* world, where *n* is a user specified parameter. The virtual world will be a 2-dimensional torus, meaning that if you are in cell (1,1) and take one step left, you are in cell (*n*,1) ; if you take one step up, you are in cell (1,*n*); if you move to the left *and* up, simultaneously, you are in cell (*n*,*n*).
    - Note: you may want to implement a visualization for the *n*-by-*n* world, to facilitate debugging your code
- (10 points) Implement an agent object. Each agent has the state variables that keep track of the following:
    - Location in the virtual world (i.e., a coordinate (*x,y*))
    - Whether the agent is susceptible to infection (*S*), currently infected (*I*), in quarantine (*Q*) or recovered from infection (*R*).
    - If they are currently infected (*I*), a counter indicating how long they have been infected (measured in days, where one time step in the ABM simulation equals one day).
    - Whether the agent is wearing a mask
    - Whether the agent is practicing physical distancing
    - Whether the agent is in asymptomatic (assuming they are infected)
- (50 points) Implement a simulator that, given a virtual world and a list of agents, does the following - for each agent in the list:
    - Assigns a random starting location to each agent (ensuring that there is at most one agent per cell)
    - Sets a user-specified number of agents to state *I.*
    - Sets a user-specific number of agents to state *R,* to simulated vaccinated individuals.
    - Simulate the movement of each agent (iterating through the agents in some order).
        - Each agent will identify the set of allowable moves from the set {*left; right; up; down; left & up; left & down; right & up; right & down*}, taking into consideration the presence of other agents in its vicinity. An agent may not move into a cell that is already occupied by some other agent. Also, if the agent is practicing physical distancing, then it will always make sure there is at least one empty cell between it and the nearest other agent.

        - The agent will randomly select one of the allowable moves (if any), and then move to the new cell.
        - If there are no allowable moves (due to crowding), then that agent will stand still for one simulated day.
    - After each agent has taken a step, update its state
        - If the agent is type *S*, and it is in a cell that is adjacent to a cell occupied by an agent of type *I*, the *S* becomes an *I* with probability 25%, unless *S* or *I* is wearing a mask, in which case the probability of infection is 1%. If both *S* and *I* are wearing masks, then the probability of infection is 0.01%.
        - When an agent becomes infected*,* the probability it is asymptomatic is 20%.
        - If the agent is type I and has been infected for at least 3 days, and is symptomatic, it goes into quarantine by becoming type *Q*.
        - If the agent is an *I* or *Q,* increment the counter keeping track of the number of days the agent has been infected
            - If the counter is > 14, set the agent's type to *R.*
            - There is a 0.2% chance the agent dies. If it dies, it is removed from the simulation.

Part 1.2 (10 points)

(10 points) Run 10 simulations, each of which tracks 250 agents over 365 days in a 2525 world with the following initial conditions:

- 5 infected agents (*I*), the rest are susceptible (*S*)
- 0% of agents wear masks
- 0% of agents practice physical distancing

Keep track of the number of *S, I,Q,R*, and *D* agents over time in each simulation. Plot the averages of these trajectories with standard error bars.

What is the final value of *R*?

What is the peak number of active cases (*I*+*Q*)?

How many days does it take for the virus to go extinct (i.e. *I+Q=0*)?

Part 1.3 (10 points)

(10 points) Run 10 simulations, each of which tracks 250 agents over 365 days in a 2525 world with the following initial conditions:

- 5 infected agents (*I*), the rest are susceptible (*S*)
- *p*% of agents wear masks
- *p*% of agents practice physical distancing

Keep track of the number of *S, I,Q,R*, and *D* agents over time in each simulation. Plot the averages of these trajectories with standard error bars.

What value of *p* reduces the final value of *R* to roughly half of the final value of *R* you obtained for part 1.2?

What is the final value of *R*?

What is the peak number of active cases (*I*+*Q*)?

How many days does it take for the virus to go extinct (i.e. *I*+*Q*=*0*)?

Part 1.4 (10 points)

(10 points) Run 10 simulations, each of which tracks 250 agents over 365 days in a 2525 world with the following initial conditions:

- 5 infected agents (*I*), p% of agents have been vaccinated (i.e., start in state *R*), the rest are susceptible (*S*)
- 0% of agents wear masks
- 0% of agents practice physical distancing

Keep track of the number of *S, I,Q,R*, and *D* agents over time in each simulation. Plot the averages of these trajectories with standard errors.

What value of *p* reduces the peak number of active cases (*I*+*Q*) to roughly half of the final value of (*I*+Q) you obtained for part 1.2?

What is the final value of *R*?

What is the peak number of active cases (*I*+*Q*)?

How many days does it take for the virus to go extinct (i.e. *I*+*Q*=*0*)?

Resources & Tips

- Although it's not necessary, it may be helpful and easier to organize your code if you use an object-oriented approach for the world and agents: https://realpython.com/python3-object-oriented-programming/
- Plots w/ error bars: https://matplotlib.org/api/_as_gen/matplotlib.pyplot.errorbar.html

- Adding legends to Plots: https://matplotlib.org/3.1.1/api/_as_gen/matplotlib.pyplot.legend.html
- Randomly sampling values from a probability distribution: https://docs.scipy.org/doc//numpy-1.10.4/reference/generated/numpy.random.choice.html