COMS30014 (Artificial Intelligence)

Week 7 – Multiagent Systems – Lab 2

Simulating Schelling's Segregation Model

Segregation is a separation of people in different groups. Although segregation is outlawed, segregation is often observed. Thomas Schelling developed a striking model of racial segregation wherein Schelling showed how mild preferences of people to be with similar neighbors could results in widespread segregation [1].

In this lab, we will first develop an agent-based simulation to model Schelling's Segregation Model. We will use Mesa for this lab. Luckily Mesa examples include Schelling's model, so we do not need to start from scratch.

Next, we will explore ideas on how introducing agents of different types could reduce segregation and promote mixed neighborhoods.

Overview of Schelling's Segregation Model

The model is setup as n x n grid. Agents occupy space on this n x n grid such that there are certain number or percentage of free cells on the grid. One agent can occupy only one cell in the grid. Agents are of two types. In our code base, we represent the agent types by colors red and blue. Figure 1 shows 14 agents in a 4x4 grid with 2 spaces vacant. The two agent types are represented via X and 0.

X	X	0	0
X			0
0	X	0	X
0	0	X	X

Figure 1. Agents of two types in a grid.

Agents have a preference to be with a certain percentage of similar neighbors; this preference is represented by homophily in the code base. The agents who have similar neighbors above the homophily threshold are happy, whereas the ones with homophily below the threshold are unhappy. Consider homophily is 40%. Figure 2 shows unhappy agents whose homophily is less than 40% in yellow.

X	X	0	0
X			0
0	X	0	X
0	0	X	X

Figure 2. Unhappy agents in yellow.

At each step in the simulation, happy agents continue to occupy the same cell whereas, whereas all unhappy agents move to an empty cell. Note that, all movements take one step. These movements could be either random or based on a certain strategy.

X	X	0	0
X	X	0	0
0			X
0	0	X	X

Figure 3. Unhappy agents moved and now are happy.

Note that, when unhappy agents move to empty cell, they may make previously happy agents unhappy.

The agents continue to move until all agents are happy.

[1] Thomas C Schelling. Models of Segregation. *American Economic Review*, 59(2):488–493, 1969.

Lab Tasks

You are given a code base which contains three files --- model.py, run.py, and server.py.

- model.py contains the code for Agent model and Schelling simulation model
- run.py launches the server (you will not need to change anything in this file)
- server.py contains the code to setup the UI elements

Task 1.

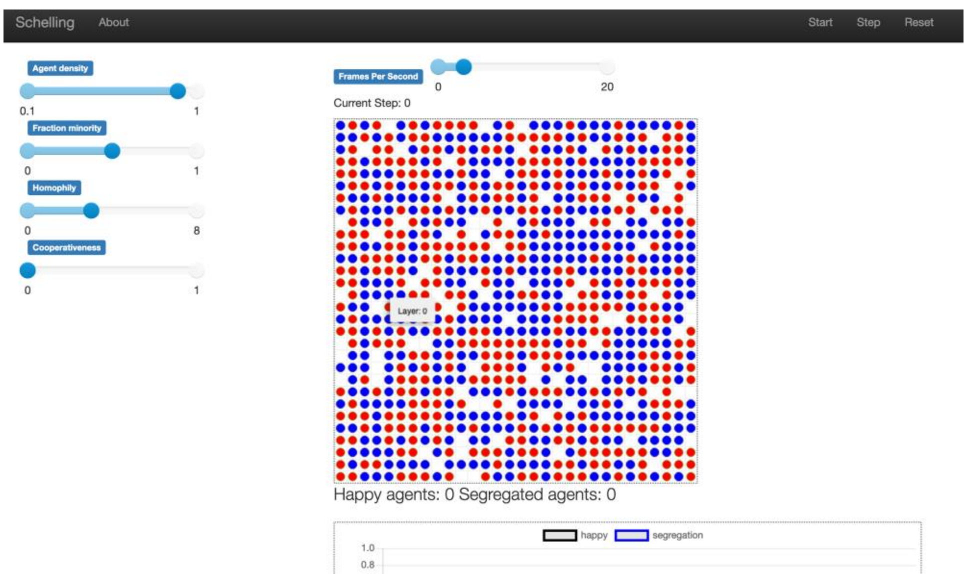
Your first task is to get familiar with the code base. In the first 15-20 minutes, you should try executing the simulation and understand how to the simulation runs. Play with the simulation controls.

To start the simulation:

On the command prompt, type: *mesa runserver*

If Mesa is correctly setup on your computer, a browser window shall open pointing to the URL: http://127.0.0.1:8521/

You shall see a window like the figure below:



Task 2.

Segregated agents are those who are surrounded by agents of the same type. In Figure 3, there are four segregated agents in the four corners. You would have noticed that the simulation GUI does not show the segregated agents count.

Your second task is to first compute the count of segregated agents, and next show the count in the GUI. Hints:

- Number of happy agents are computed in the step function (def step) of the SchellingAgent class which resides in model.py
- DataCollector in Schelling class passes the happy agent value to server.py
- happy agent count is reset in step function of Schelling class
- def render in server.py renders the count of happy agents on the GUI

Now that you have the count of segregated agents on the screen, run the simulation for a few times and observe how homophily relates to segregated agent count.

Task 3.

Society is not always red and blue. There are some people who are liked by everyone. Will having more of such people reduce the segregation? We shall find that out in this task.

In this task, you shall introduce this new property in agents. Agents with this property are cooperative and thus are liked by everyone. We add this cooperative property by including a Boolean attribute *is_cooperative* to the Agent class.

If an agent's neighbor is likeable, i.e., *is_cooperative* is True for a neighbor, the agent considers that neighbor agent as a similar agent.

You could render cooperative blue and red agents in a different shade of blue and red than normal agents.

You shall also introduce a new attribute to the society called as cooperativeness, which indicates what percentage of agents are cooperative.

Now restart the simulation and observe how adding cooperative agents influence the count of happy agents and segregated agents.

Additional Exploratory Tasks.

- Cooperative agents developed in Task 3 are liked by others, but they themselves are not flexible --- they have preference to live with agents of similar color. How will the count of happy agents and segregated agents change if cooperative agents are flexible --- they are happy with anyone as their neighbors?
- In Schelling's model and in our simulation, happy agents do not move. Consider happy agents are not selfish. They move (albeit less often than unhappy) if they notice that unhappy agents have moved more than a threshold number of steps but haven't found a cell which is favorable to them.