Final Project / Schelling's Model of Segregation

Mahyar Albalan

403201223

For doing this simulation, as mentioned in the text I calculate neighbors of each cell and count the same neighbors and different neighbors. Based on predefined thresholds, I change each cell's value(position) randomly (you can check the code for more details). Here is the final result of simulation for network of size 60 and $\rho = 0.1$ and $B_m = 0.75$ is as follows: (you can see animation of network's evolution in the Jupyter file in my github).

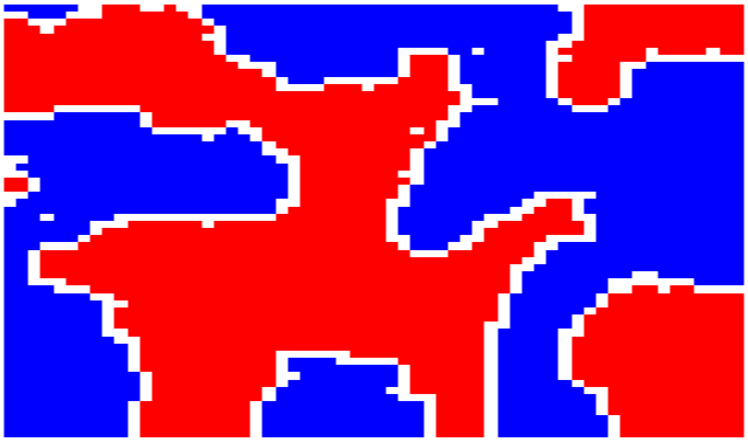

Figure 1: final state of the network

as you can see as expected two classes are completely seperated.

I calculated the total number of displacements for different thresholds by simply assigning a counter for each step to count the number of displacements in that step, and then summing all of them to get the total number of displacements. I simulated this number using several different thresholds (there are only 8 distinct thresholds since each cell has at most 8 neighbors). I simulated the process for $\rho = 0.1$ and different thresholds; the result is as

follows:

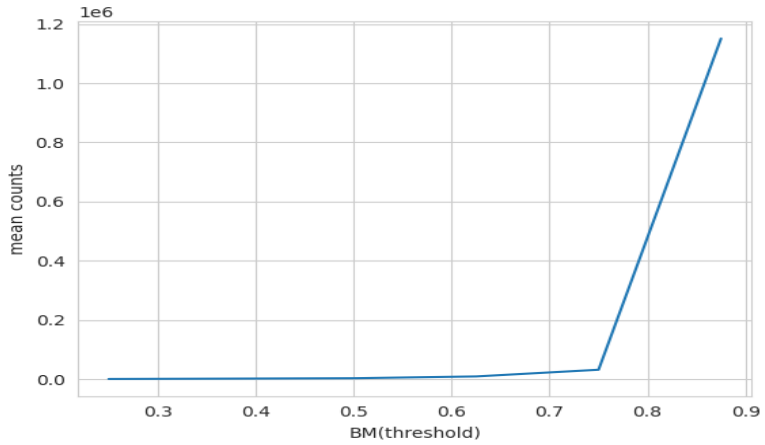

Figure 2: total number of displacements vs thresholds

as expected by increasing threshold total number of displacements grows aswell.

For simulating the third part, I used $B_M = 0.7$ and different densities for empty cells. As expected, by increasing the density, the total number of calculations decreases. The result is as follows:

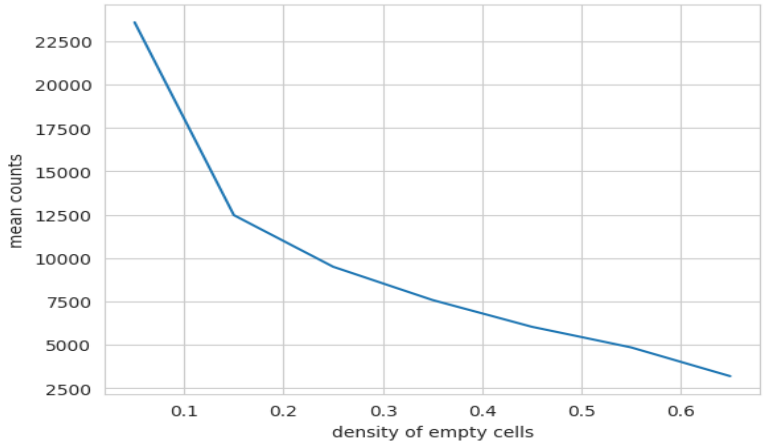

Figure 3: total number of displacements vs density of empty cells

For calculating segregation coefficient, I used the relation in the text. First I wrote a function to calculate the size of each cluster, and using that calculate segregation coefficient. Here is the results for 3 different densities:

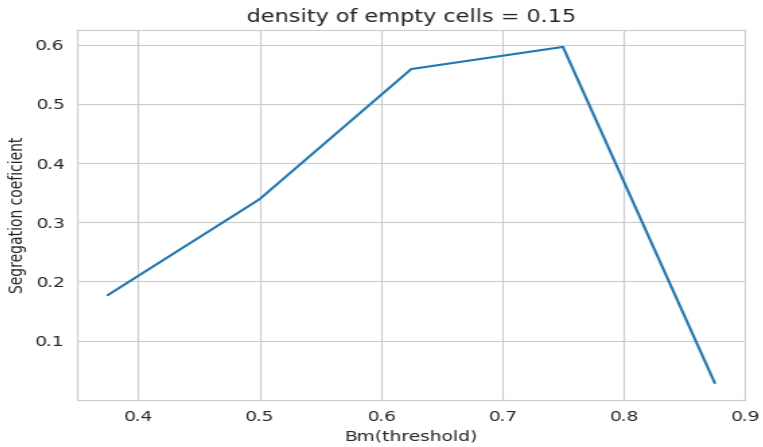

Figure 4: segregation coeficient vs thresholds

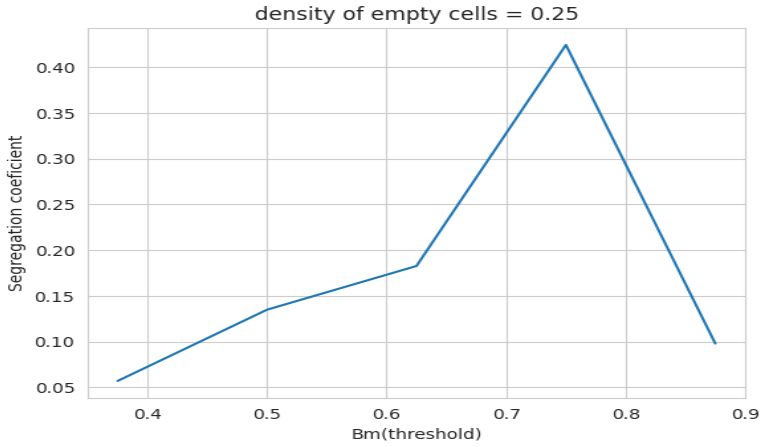

Figure 5: segregation coeficient vs thresholds

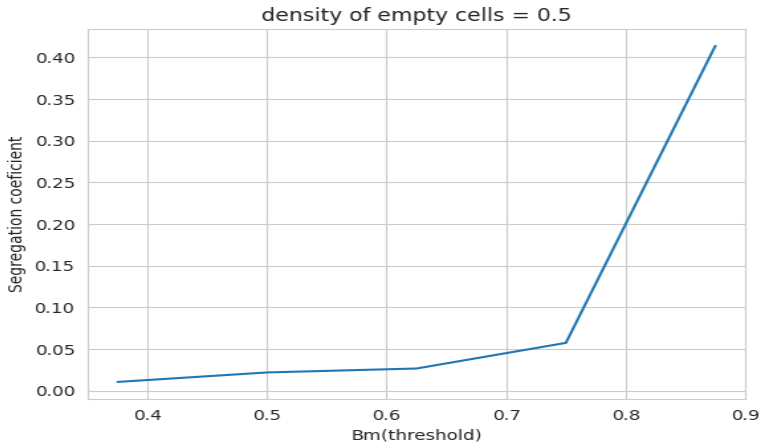

Figure 6: segregation coeficient vs thresholds