Project 2 Report: Creation of Spatial Structure by Activator / Inhibitor Cellular Automaton

Patrick Elam

CS420

February 20, 2015

Introduction:

For this project, we examined the creation of spatial structure by activator / inhibitor cellular automaton in terms of spatial correlation and mutual information. To do this, we programmatically created samples of 30 x 30 cellular automaton for simulation. These cellular automatons were then advanced until they stabilized according to the following formula:

$$s_i(t+1) = \text{sign}\left[h + J_1 \sum_{r_{ij} < R_1} s_j(t) + J_2 \sum_{R_1 \le r_{ij} < R_2} s_j(t) \right]$$

In this formula, R1 represented the radius in which the given cell wanted to replicate its state. The radius between R1 and R2 was where the given cell wanted to have an opposite state. The value of h was there to inject a bias into the formula. I ran these simulations with many variations of R1, R2 and h, keeping $J_1 = 1$ and $J_2 = -0.1$, with a randomly generated starting value for each cell.

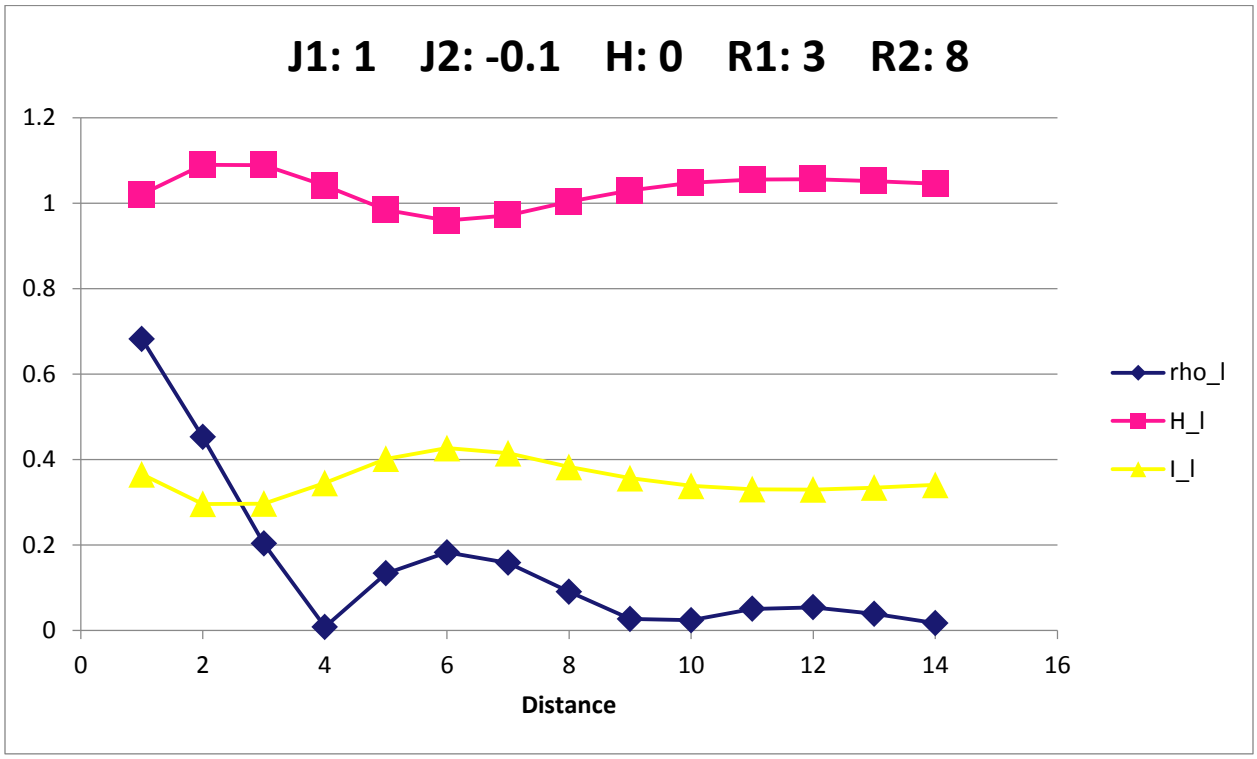
Simulator:

I, along with Alex Chaloux, wrote a simulator that generated a 30 x 30 cellular automaton for the purpose of simulating automata in c++. The simulator would generate a cellular automaton and fill each cell with a random value. Then, we defined a set of R1, R2 and h values to be used for each simulation. The values for R1 were 1, 3, 5, 7, 9, 11. The values for R2 were 2, 4, 6, 8, 10, 12. The values for h were -5, -3, 0, 3, 5. For each combination of these values, a new cellular automaton was generated and ran. During each run, we asynchronously updated the cells of the automaton according the formula above. The code continued to do this until the cellular automaton had settled into a stabilized state. The simulator then ran several calculations on the stabilized cellular automaton. For each set of unique parameters, we ran 4 iterations, each with a random starting point, and then took the average results from each calculation and set them aside. For each run, a pgm image file was generated that represented each cellular automaton, as well as a .csv file that contained the results of each calculation.

Calculations:

The calculations done on each run of the simulator were as follows. The first calculations were that of correlation for each possible distance (0-14). The second set of calculations was that of entropy of the entire cellular automaton. The third set of calculations was that of the joint entropy for each possible distance (0 -14). The final set of calculations was that of mutual information, again for each possible distance (0-14).

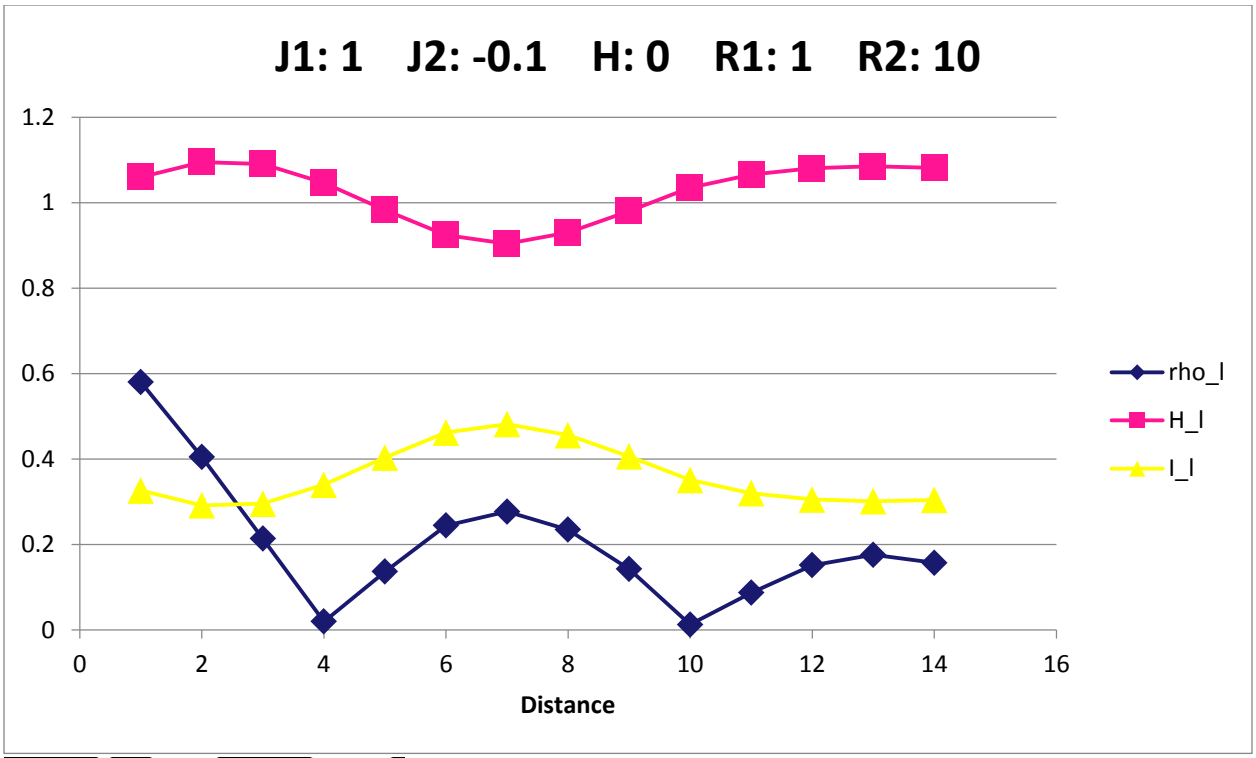
Once we obtained the average values for each calculation for each set of parameters, we graphed the I_l (mutual information), rho_l (correlation), and H_l (joint entropy) against the l values. Below is an example where R1 = 3, R2 = 8 and H = 0.

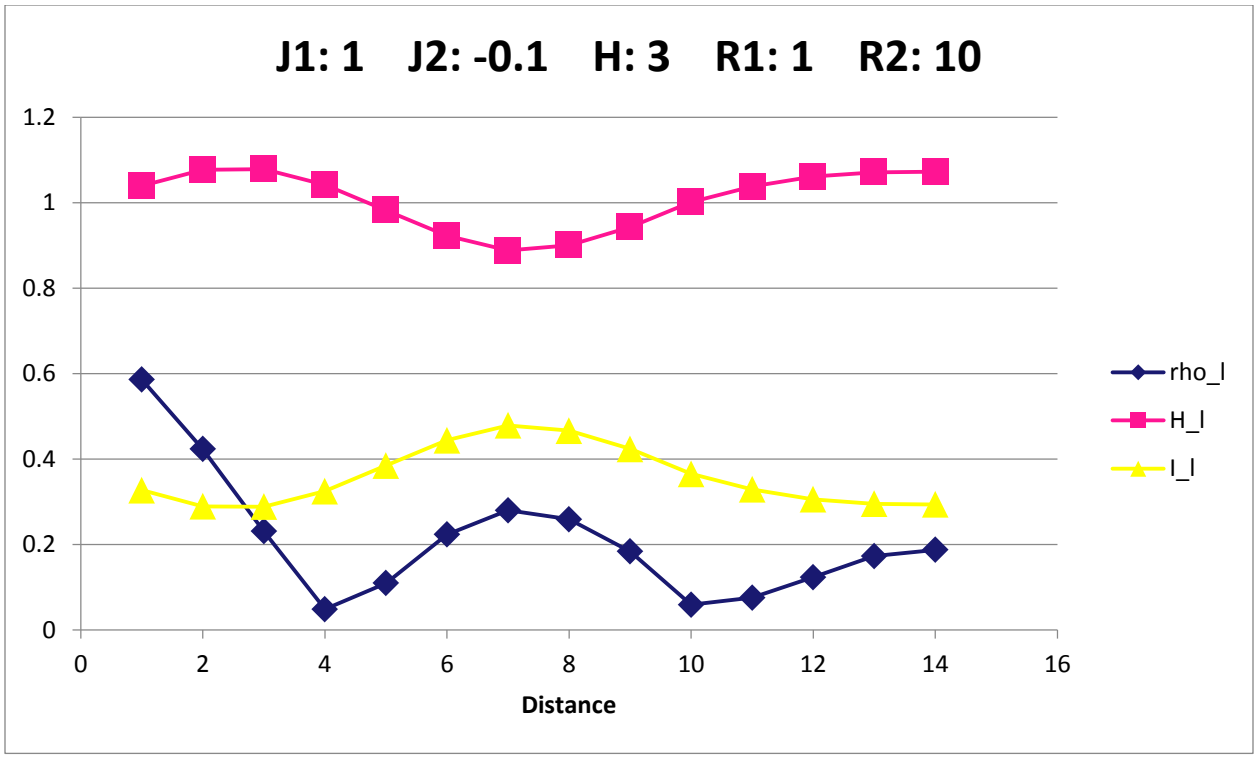


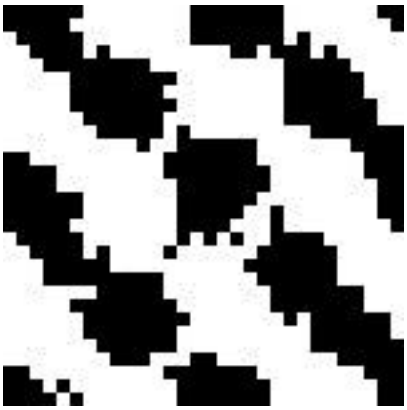
One example of the images from this set of parameters is below:

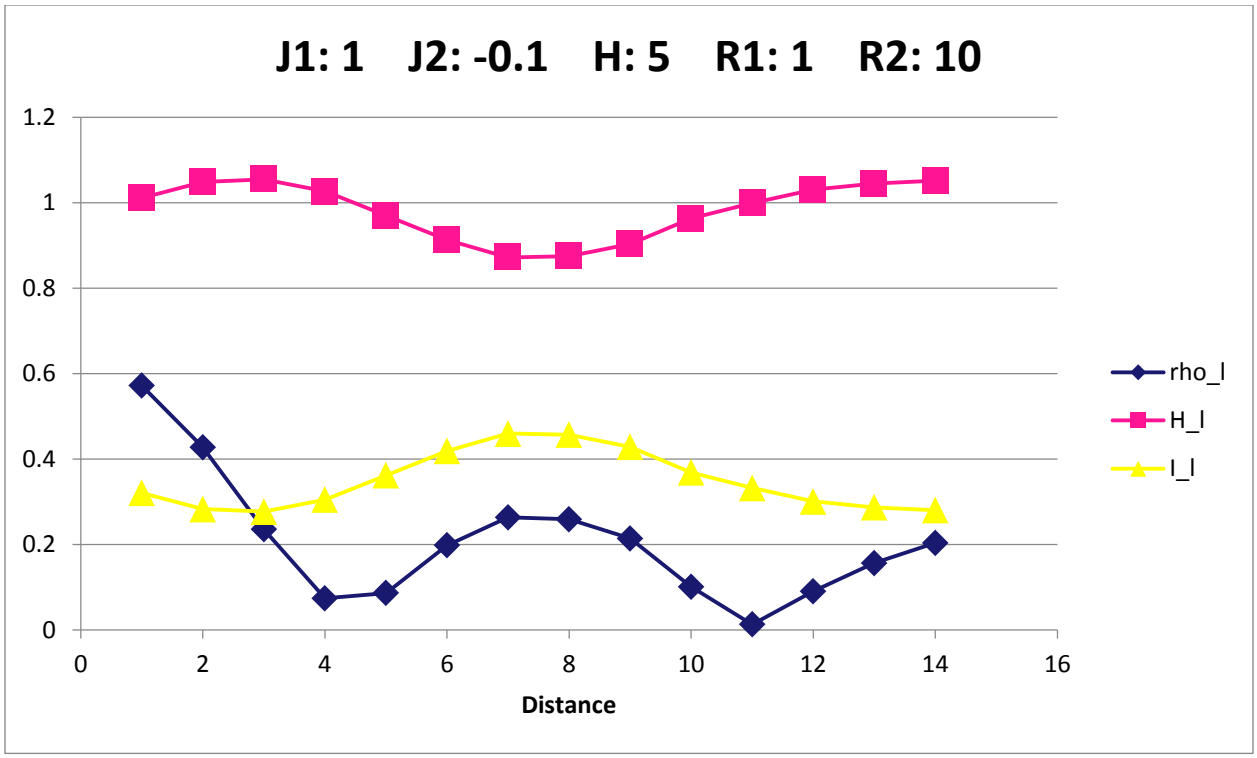
Patterns Found:

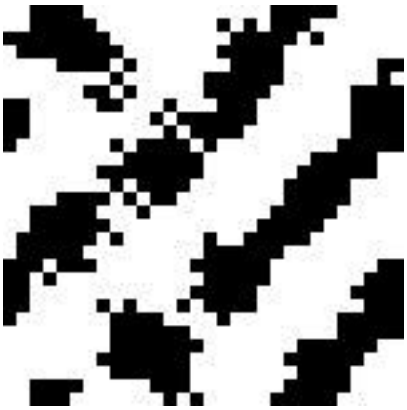
Upon investigation of the data, several patterns began to emerge. When the R1 value was low, the more of a difference in the cellular automaton the value of h made. In these cases, the further h got from 0, the higher the entropy values. Consider the following example of graphs and their associated images. As h grows, the entropy values decrease, and the image becomes more stable.


J1: 1 J2: -0.1 H: 0 R1: 1 R2: 10
1.2
1
0.8
0.6
0.4
0.2
0
0
2
4
6
8
10
12
14
16
Distance
rho_l
H_l
I_l

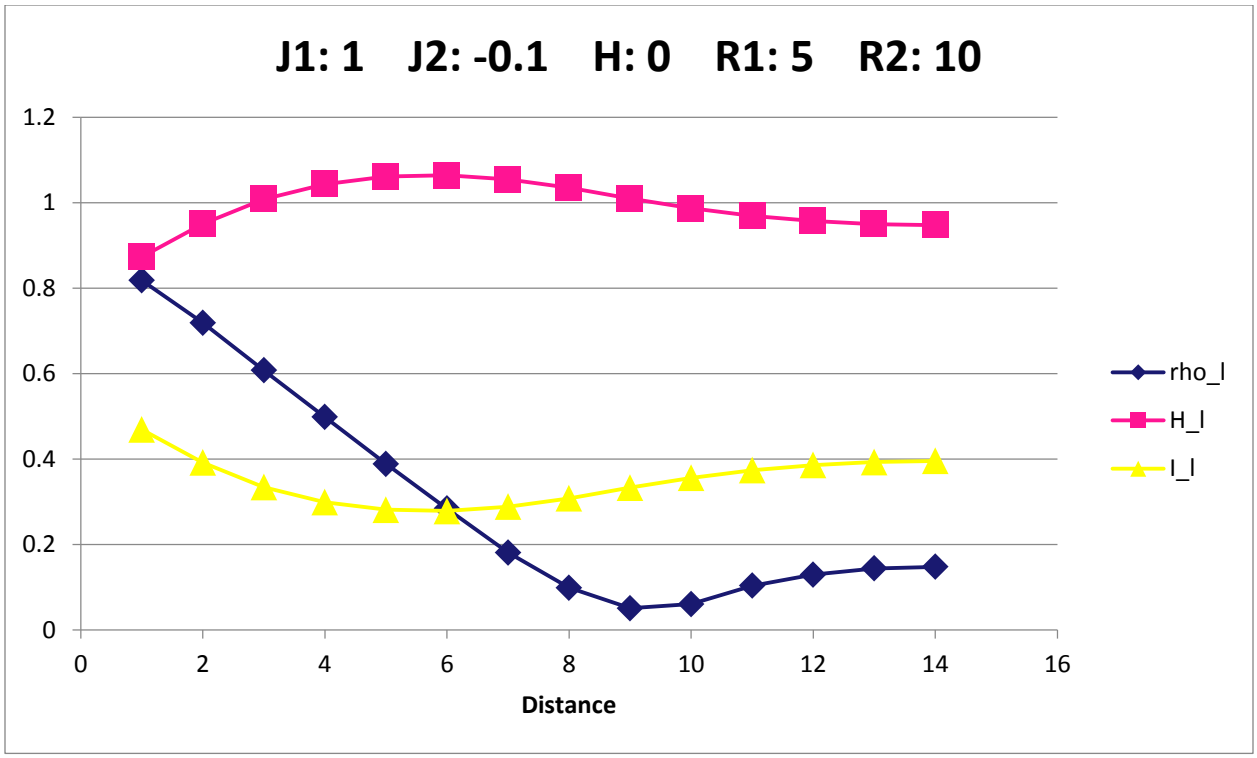

J1: 1 J2: -0.1 H: 3 R1: 1 R2: 10
1.2
1
0.8
0.6
0.4
0.2
0
0
2
4
6
8
10
12
14
16
Distance
rho_l
H_l
I_l

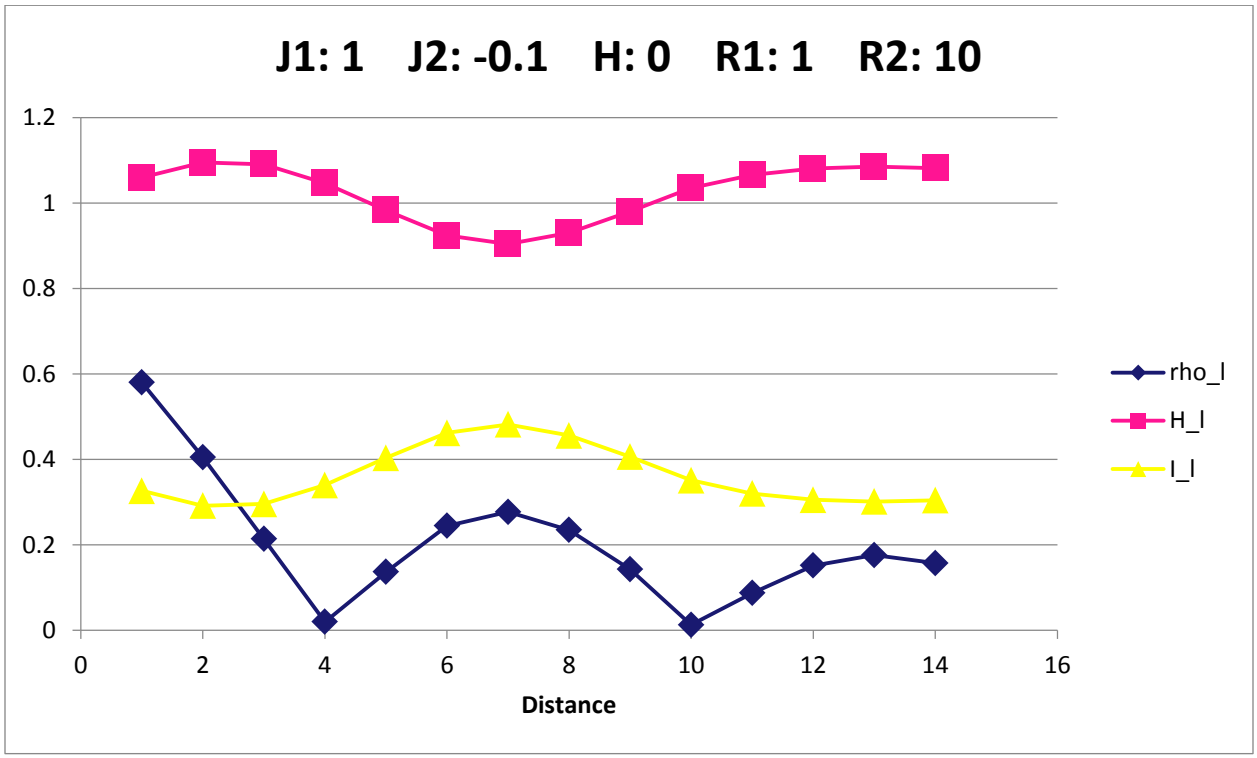


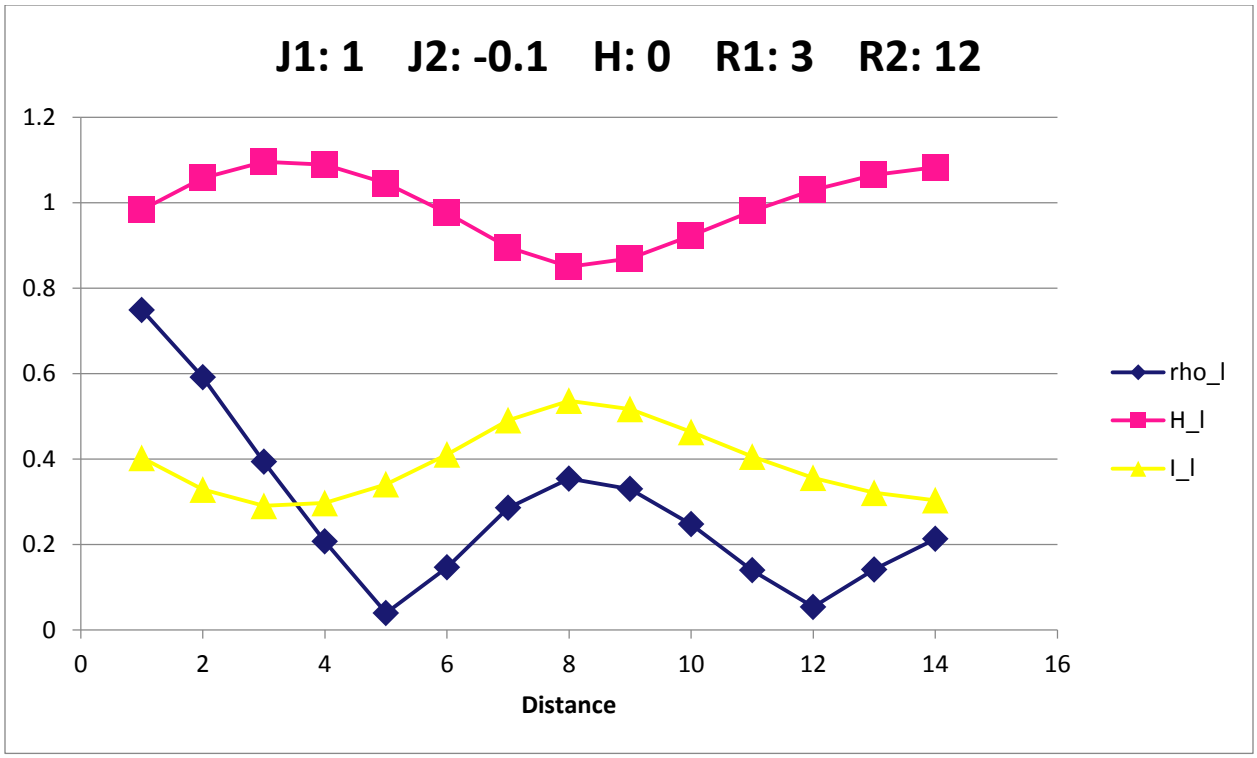




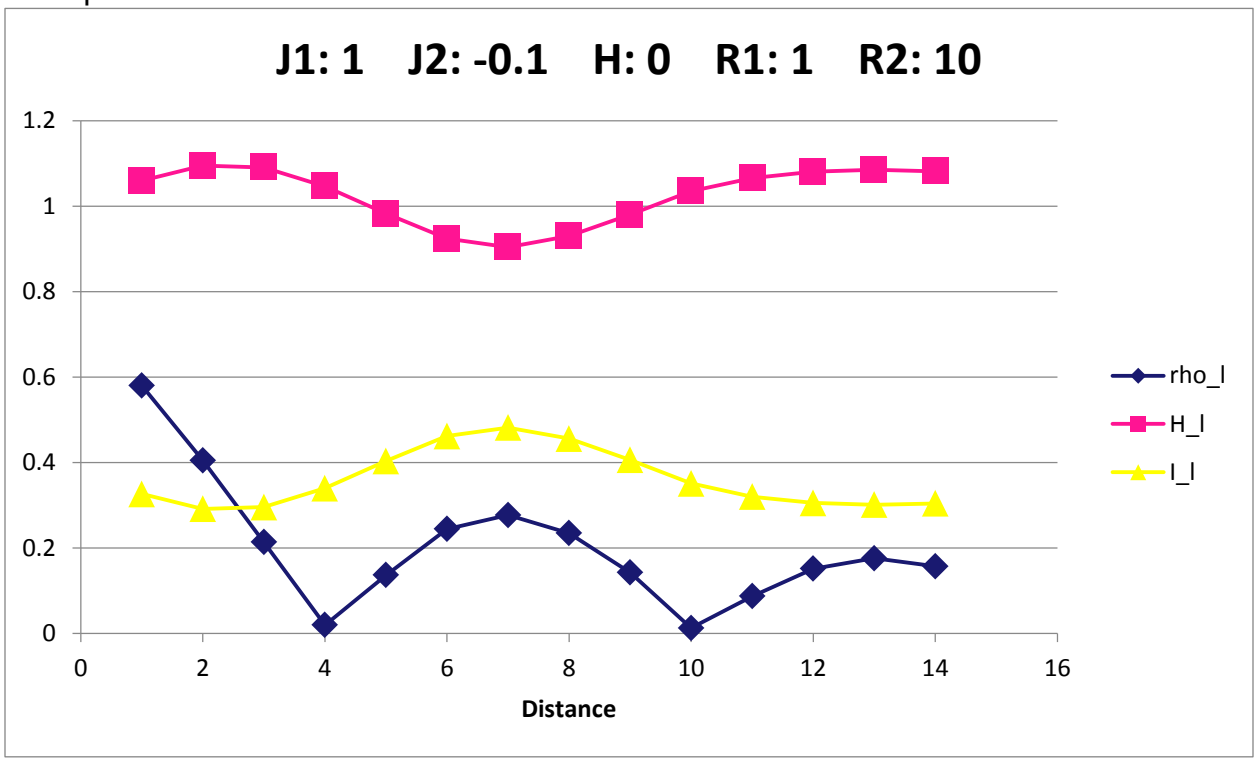
Another pattern that became evident was that of the correlation between the correlation values for the cellular automaton and the values of R1 and R2. It became evident that the correlation values dipped heavily when they reached the boundary at R1, rose between R1 and R2, and dipped heavily again at the boundary of R2. Below are a few examples of this behavior. It seems that the cause of this behavior is that within the R1 radius, the cells are most often similar, and then as the cells get further away from the target cell and closer to the R1 radius, they become less effected. The cells between the radii R1 and R2 have a higher correlation with the target cell as well. This behavior seems to exist because near the radii, there are cells that want to go either way, and thus there is less correlation.

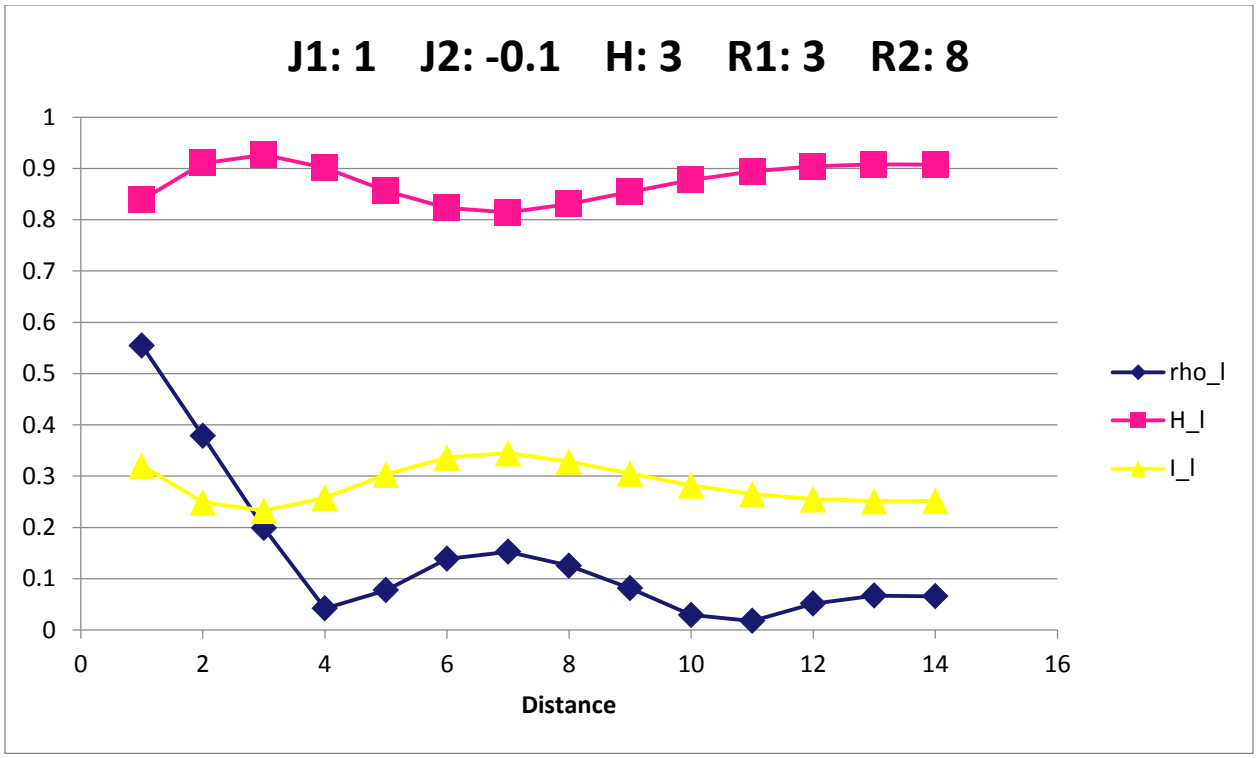

J1: 1 J2: -0.1 H: 0 R1: 5 R2: 10
1.2
1
0.8
0.6
0.4
0.2
0
0
2
4
6
8
10
12
14
16
Distance
rho_l
H_l
l_l

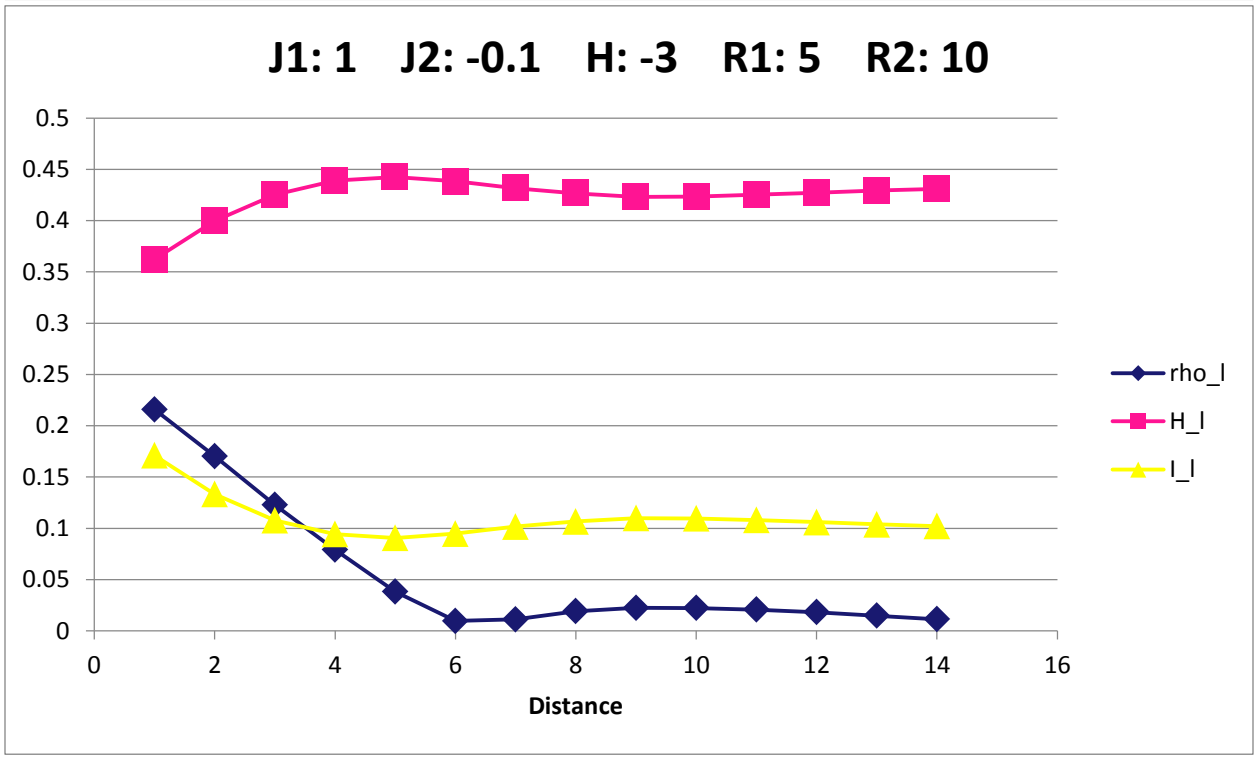

J1: 1 J2: -0.1 H: 0 R1: 1 R2: 10
1.2
1
0.8
0.6
0.4
0.2
0
0
2
4
6
8
10
12
14
16
Distance
rho_l
H_l
l_l



Another interesting pattern we found in the data is that close to the distances of R1 and R2, entropy tends to spike, while mutual information declines. Below are some examples of this behavior.




J1: 1 J2: -0.1 H: 3 R1: 3 R2: 8
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
0
2
4
6
8
10
12
14
16
Distance
rho_l
H_l
I_l


J1: 1 J2: -0.1 H: -3 R1: 5 R2: 10
0.5
0.45
0.4
0.35
0.3
0.25
0.2
0.15
0.1
0.05
0
0
2
4
6
8
10
12
14
16
Distance
rho_l
H_l
I_l

Conclusion:

It seems evident from the data that as cells are closer to the R1 and R2 distances, they tend to have less correlation to the target cell. Within R1, the cells are more like the target cell, as the formula suggests. Between R1 and R2, the cells also correlate to the target cell, but in the opposite state, again, as the formula suggests. At these values of R1 and R2, there is more chaos in the cells, and thus more entropy. With the increase in entropy, we get a decrease in mutual information. This can be seen in the data, as well as the formulas, and is also logically sound. In conclusion, the distances of R1 and R2 from the target cell highly correlate with expected changes in the given calculations.