Exploring the Resilience of Emergent Properties using Langton's Ants

Logan Wrinkle

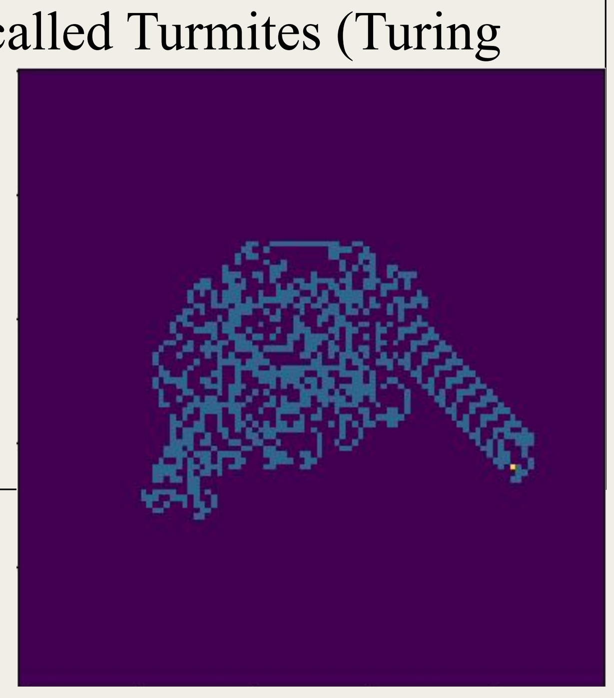
Introduction

Langton's Ants are two-dimensional Turing machines and Cellular Automata with very simple rules that generate emergent behavior. The basic rules for an ant are that if it is on a white square, it should turn right, flip the color to black, and move forward. If it is on a black square, it should turn left, flip the color to white, and move forward.

This basic Ant starts by creating simple patterns that are sometimes symmetrical before moving on to more chaotic shapes. Around step 11000 of the simulation, the ant begins building a "highway" indefinitely. This is shown below.

These Ants have been further explored in various ways, such as by adding additional ants and adding more states and transition rules. When varying the states and rules, they are called Turmites (Turing machine termites).

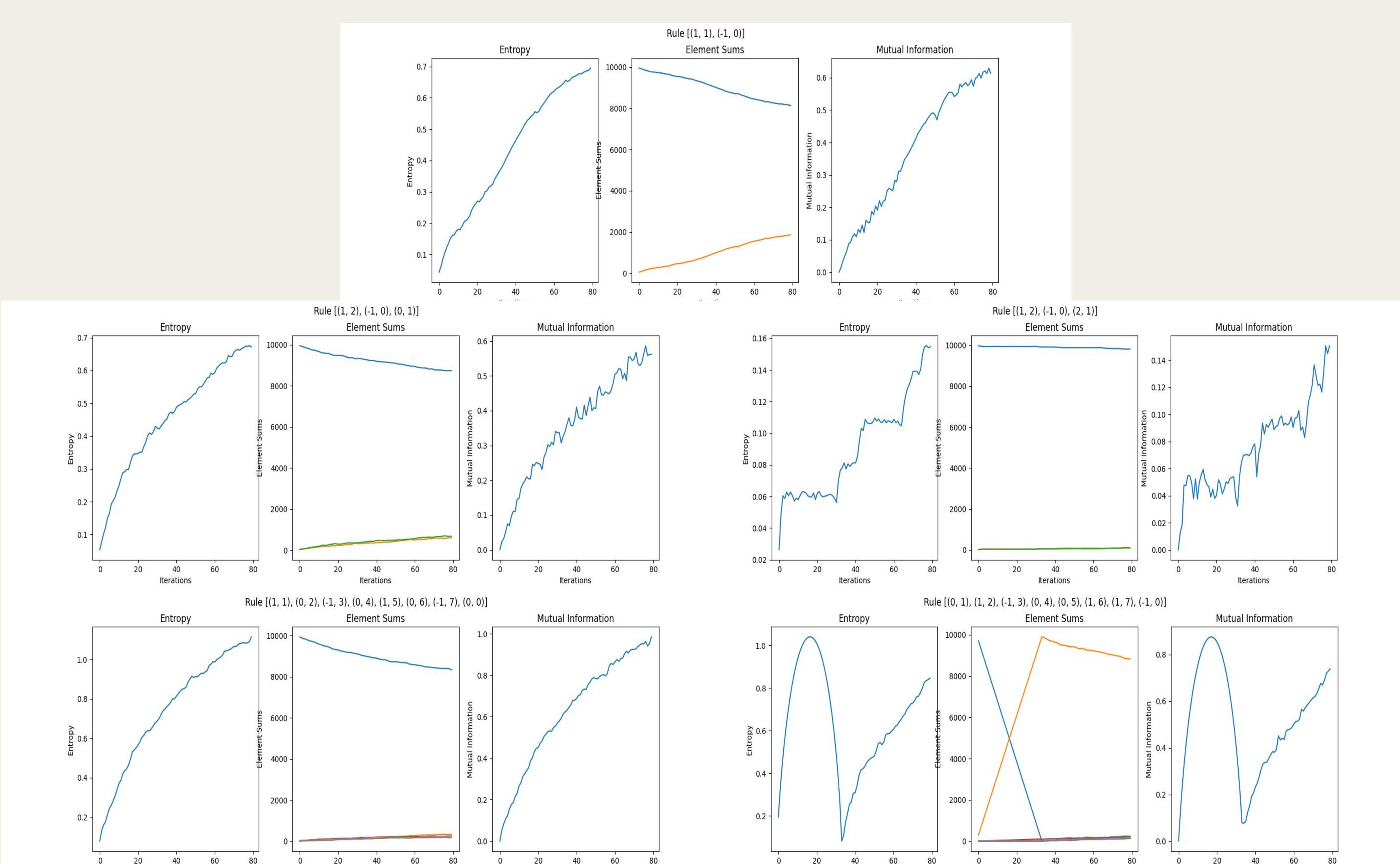
The inspiration for the project was to explore these Cellular Automata and test the resilience of emergent phenomena by introducing chaotic/random elements.



Results

The following 5 graphs represent each of our rulesets with N = 1, random tiles = 0, and Chaos Ants=0

The rulesets on top are interpreted as Rule[current_state] = (direction to turn, state to set), so for the default ruleset, if the ant was on a white tile (state value 0), then Rule[0] = (1,1), so set the white tile to black (state value 1), turn right, and move forward.



Methods

Langton's Ants can be treated as Cellular Automata where you have a number of states equal to the number of colors * 2. This encodes each state's color and a boolean for whether an ant is on the tile. However, I chose an easier option, which is to have only a number of states equal to the number of colors with the ant acting on top of the board. The two methods are equivalent, however.

In order to measure the complexity of the behavior I used three metrics

- Entropy - described as $-\sum_{x\in\mathcal{X}} p(x)\log_b p(x)$ where p(x) is the occurrence of state x / total number of cells.
- Total number of each state
- Mutual information - described as $\sum_{x\in\mathcal{X}}\sum_{y\in\mathcal{Y}} P(x,y)\log\frac{P(x,y)}{P(x)P(y)}$ where x and y are states.

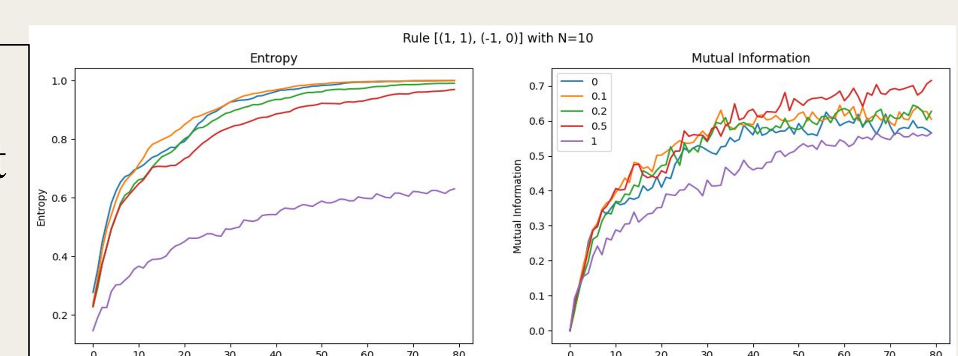
The graphs show these metrics as the simulation progresses.The x-axis is in snapshots with 300 steps between each x value. (1=300, 2=600, etc)

The tests we will run are every permutation of the following variables,

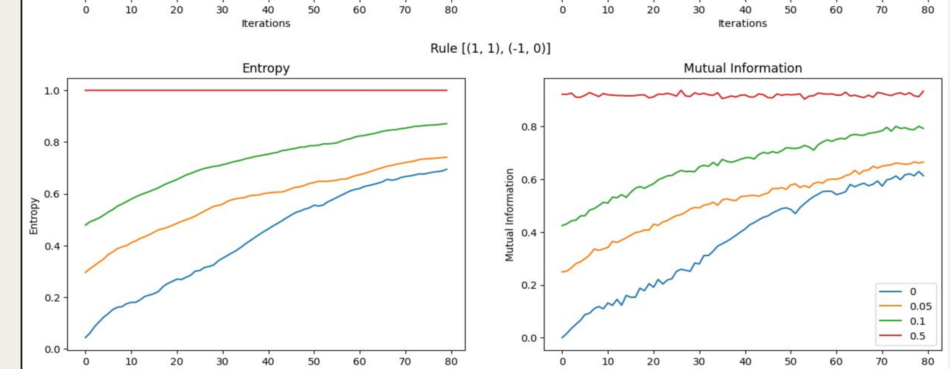
- rules(5 rulesets)
- Number of ants(1,3,10,20)
- Odds of starting tiles not being 0 (0,.05,.1, .5)
- Odds of an ant spawning with a randomized ruleset (0,0.1,0.2,0.5,1)

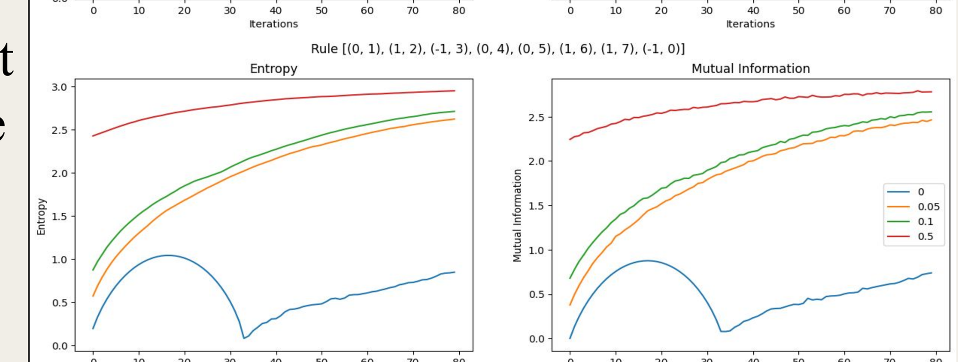
Each permutation was run 3 times. The total number of experiments was 1200, with each simulation gathering data every 300 steps until step 24000.

The graph to the right shows the default ruleset with varying odds of Chaotic Ants (N=10)

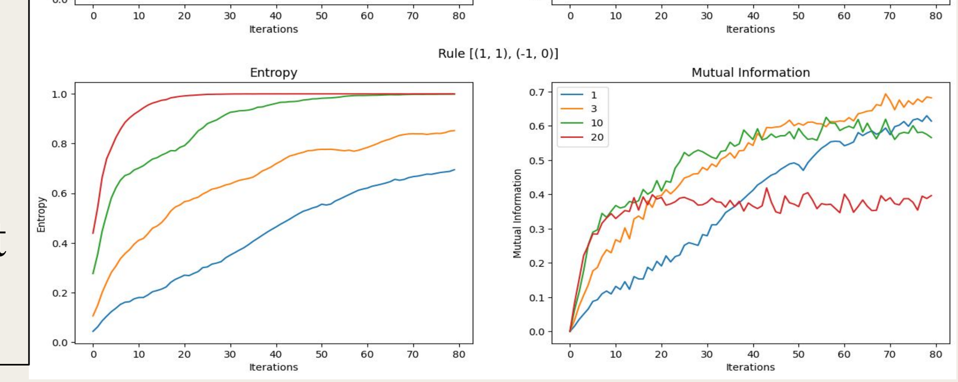


The two graphs to the right shows the default ruleset and 8-state ruleset with varying random tile odds





The graph to the right shows the default ruleset with varying N values



Conclusion

I have issues with the metrics I chose to evaluate the complex behaviors. As the first lab has shown, entropy isn't a good metric of determining complexity in Cellular Automata.

Despite that, we can see clear disruptions of the patterns at any level of additional chaos. As we increase the number of ants, we get more jagged graphs which hit a convergence point much quicker. As we introduce more chaotic ants we see a drop in entropy as it increases with mutual information being surprisingly constant. Finally the randomized starting tiles, even at low odds, entirely breaks the unique pattern of the final 8-state ruleset (that unique parabolic shape).

With even the smallest amount of randomness/chaos introduced the patterns and behaviors almost entirely breakdown.

THE UNIVERSITY OF TENNESSEE KNOXVILLE