STAT 37830 – Final Project Report

Group 7: Matt Frazier, Robert Winter, & Simone Zhang

1 Introduction

While the internet has made humans more connected than ever before, it has also enabled the creation of virtual echo chambers and polarized opinions within online networks. In his 2017 paper entitled *Opinion Cascades and Echo-Chambers in Online Networks: A Proof of Concept Agent-Based Model*, Toby D. Pilditch proposed a simple agent-based model of such opinion propagation.[1] In particular, he modeled agents in a virtual community who arrive at opinions about an event by (1) attending to the opinions declared by their neighbors, and (2) evaluating (stochastic) evidence about the event, independent of the opinions of their neighbors. This project aims to employ agent-based modeling techniques to gain a deeper understanding of the mechanisms underlying opinion polarization and consensus formation in digital environments. In particular, we reproduce Pilditch's model in Python and attempt to replicate some of his key findings. We also attempt to extend his model by introducing (1) "influencers" whose opinions affect those of a wider range of agents, (2) population-wide bias regarding an event, and (3) "open-minded" agents who are capable of changing their minds about an event.

2 Methods

2.1 Agent-Based Modeling

Agent-based models are a class of computational simulations that use a set of agents to model individual parts of a system (usually people in a population). Each agent is assigned attributes of interest, and real-world interactions are simulated using stochastic interactions between agents. Due to the specific attribute structure and interaction scheme, this method lends itself most to object-oriented programming using a specific agent class and methods for interaction.

2.2 Opinion Spreading Model

In this project, agent-based modeling is applied to a binary opinion about a neutral event. For modeling opinion spread, we used agents who have a "gradient opinion" taking on a value in $[-1, 1]$, and a "final opinion," which takes one of the values 'L'(left), 'R'(right), or 'neutral'. At the beginning of a simulation, all agents have neutral opinion and their gradient opinion is set to zero. An important simplification (which we relax in Section 5.3) used in

[1] Toby D. Pilditch, *Opinion Cascades and Echo-Chambers in Online Networks: A Proof of Concept Agent-Based Model*, 2017.

this context was precluding any re-learning of an opinion: once an agent's final opinion is 'L' or 'R', it does not change for the rest of the simulation. In addition, the agents' positions are randomly assigned throughout the $[-1,1]\times[-1,1]$ square space. To determine which agents interact, we set an "interconnectivity" parameter. The interconnectivity of a population is defined as the proportion of the population each agent interacts with — specifically, the proportion of the population that each agent listens to when forming their own opinion. The agent's 'neighbors' (agents to whom it attends during opinion formation) are then its k nearest neighbors, where k = interconnectivity $\times$ population.

The dynamics of opinion spread are defined by a simple learning model with three steps: attention, learning, and declaring. At each time step, a neutral agent will listen to its neighbors (attention) to see if they have declared an opinion. Supposing any of its neighbors declare an opinion, the attentive neighbor's gradient opinion changes by a set amount (the "opinion strength" parameter) in a direction determined by the neighbor's binary opinion. Once an agent hears an opinion (or opinions, if multiple of their k nearest neighbors are declaring), the learning step is performed, which represents an agent independently assessing the opinion based on evidence. We consider a simple model for reinforcement learning represented by the following equation (Rescorla, 1972):

$$Q_i(t+1) = Q_i(t) \; + \underbrace{\sum_{j=1}^{n}(\lambda\times\gamma_j\times\mathbb{1}_{\text{neighbor } j \text{ declares opinion}})}_{\text{Contribution from Attending}} \; + \underbrace{(\beta\times\delta_i(t))}_{\substack{\text{Contribution}\\ \text{from Learning}}} \tag{1}$$

where

$i = 1,\ldots,N$ where N is the number of agents in the population

$j = 1,\ldots,n$ where n is the number of neighbors of agent i

$n =$ interconnectivity $\times$ population size ($+$ number of influencers[2] within radius r of agent i)

Here Q_i is agent i's gradient opinion, λ is the opinion strength of each agent (assumed to be constant throughout the population), γ_j equals 1 if agent j's final opinion is 'R' and equals -1 if agent j's final opinion is 'L,' δ_i is a standard normal random variable representing the prediction error (essentially capturing the agent's leaning based on the learning step), and β is the learning parameter (standard deviation of learning, or how much the learning step affects the agent's gradient opinion). An important feature of using a normal learning parameter is its symmetry, which determines a neutral event (the event itself is not biased toward either opinion). Once an agent completes learning, it almost surely has a non-zero gradient opinion, and is assigned a final opinion accordingly. Finally, agents who have a non-neutral opinion declare that opinion at each time step according to a Boolean random variable computed based on the probability of declaration. We model agents as deciding whether to declare independently for each of their i.i.d for each non-neutral agent.

[2]We make precise the notion of "influencers" in Section 5.2 below.

To initiate an opinion-spreading simulation, a neutral ‘event’ occurs at $(0,0)$, the center of the grid space. This is represented by initiating the learning method on a set number of agents in the center of the space who ‘witness’ the event, so that the initial distribution of opinions is normal, and the initial set of agents who have non-neutral opinions are concentrated at the center of the grid. The learning process described above is executed for all agents in discrete time steps until no agent changes opinion.

2.3 Programming Methods

The Python packages used for this project were limited to `numpy` for standard numerical tools and random number generators, `scipy` (specifically the `spatial` module) for nearest neighbor queries,[3] and `matplotlib` for visualization.

In addition, `multiprocessing` was used to run simulations more efficiently. Multiprocessing takes advantage of multiple CPU’s to run independent processes simultaneously on separate CPU’s. As long as the data in one process doesn’t depend on the data in the others, this decreases the execution time by the number of CPU’s available. The `Pool` method was used to pass multiple arguements to the same function which are executed on separate CPU’s simultaneously.

3 Structure of `src` Package

Our source code is partitioned into three modules: `agents.py`, `population.py`, and `tools.py`.

- In `agents.py`, we define a class `agent` of agents and equip them with the attributes and methods needed to participate in an opinion propagation network. In particular, we provide agents with the cognitive architecture to identify their neighbors (`find_neighbors()`), attend to their neighbors’ opinions (`attend()`), evaluate their own opinions (`learn()`), and declare their opinions to other agents (`declare()`).
- In `population.py`, we defined a class `population`, a collection of `agent` objects constituting a network. A `population` is able to implement the dynamics of opinion propagation, including the generation of an event (`event()`), opinion formation among witnesses of the event, opinion propagation by witnesses, and subsequent opinion formation and propagation by downstream agents (each using `time_step()`). We also equip `population` objects with the ability to print a visual summary of its agents’ locations and opinions (`display()`).
- `tools.py` contains functions useful for modeling and extracting data from models. Specifically, the `model()` function runs a “cover-to-cover” opinion propagation simulation, from initiating an event to allowing agents to form and spread their opinions.

[3] We note that using the `spatial` module’s nearest neighbor queries is much more computationally efficient than a “brute-force” approach to nearest neighbor computations. See Appendix A.

The `opinion_count()` function determines the percentages of a population holding each opinion, 'L,', 'R,' and neutral. Lastly, the `cluster()` function measures the degree of clustering, or the percentage of each agent's neighbors having the same opinion, in a population.

4 Unit Testing

In the `test.py` file, we built test classes using the `unittest` package for all the methods in the `agent` class that are written in the `agent.py` file. In particular, we create a population of 5 agents and verify the methods are implemented correctly. We also built test classes for all the methods except the display method in the `population` class.

- `TestNeighbor`:
  Initialize a population of agents using the `population` class. Manually modify the position of all agents and recalculate each agent's neighbors using a KD-tree. Choose a test agent and verify that the `find_neighbors` method correctly records as the test agent's neighbors those agents with the closest Euclidean distances and any influencers within a set radius, not including the agent itself.

- `TestAttend`:
  Verify results of the `attend` method. This class contains three methods, which test whether an agent's gradient opinion is correctly updated given his neighbors' declared opinion and the specified opinion strength. The three directions are divided by whether his neighbor has declared Right, Left, or Neutral.

- `TestLearn`:
  Verify results of the `learn` method corresponding with the learning formula equation (1). In particular, this class verifies both the numerical value `gradient_opinion` and the string `final_opinion`. Contain three methods which are divided by $\delta < 0, > 0, = 0$.

- `TestDeclare`:
  Verify results of the `declare` method. Run 100 simulations of drawing random numbers of uniform distribution (0.5) where 0.5 is the pre-specified declaration probability and verify that the `declare` method should always return `TRUE`.

- `TestPopulationInitialization`:
  Initialize a population of agents, generate agent objects, and test whether the attributes in the `population` class are correctly set up and initialized.

- `TestPopulationEvent`:
  Initialize a population of agents. Specify the number of witnesses in the population

and call the `event` method in the `population` class. Check there are in total 2 agents (witnesses) having a final opinion that is not "neutral"

- `TestPopulationTimeStep`:
  Initialize a population of agents and call the `time_step` method in the `population` class. Verify that each time step should output a value (time) greater than 0.

All the tests have passed successfully.

5 Results

5.1 Basic Model

A few interesting example results are shown in Figures 1 and 2. A neutral 'event' is initiated at the center of the space, and is allowed to spread by opinion sharing through neighboring agents until no opinions have changed. The factor that most influences the structure of the outcome is interconnectivity, which controls the number of neighbors an agent interacts with. In general, the lower the interconnectivity, the smaller the clusters of like-minded agents are. When interconnectivity drops below 0.5%, there are often groups of agents that are totally disconnected from the rest of the population, which occasionally causes opinion spread to stop before any opinion has spread through the entire population. For example, in Figure 1(a), we model a population of 1,000 agents with 0.5% interconnectivity; notice that the northwest corner of the grid contains a cluster of agents with neutral opinions and that smaller groups of neutral agents can also be found near $(-0.25, 0.5)$, $(0.5, 0.5)$, and $(1.0, 0.75)$. In Figure 1(b), we model another population of 1,000 agents with 0.5% interconnectivity; this time, opinion spread has halted near the center of the grid due to the low interconnectivity. In contrast, Figures 2(a) and (b) feature 1,000-agent populations with a $10\times$ increase in interconnectivity, to 5%. Observe that neither of these simulations includes neutral agents, unlike in Figure 1. Each simulation was run using a population of 1000 agents. To test our agent-based modeling framework, a similar series of simulations to Pilditch (2017) was run to test the reproducibility of results.

In particular, two indicators of information propagation were investigated: (1) the overall portion of the population with a particular opinion, and (2) the degree of opinion clusters, which is defined as the average proportion of an agent's neighbors with the same opinion over the entire population.

Figure 3 shows the average percentage of the population holding opinion 'L' as a function of population interconnectivity and opinion strength, as well as standard deviation, over 10 simulations at each sample point. As in Pilditch (2017), we find that the average proportion of each opinion is about 50 percent, as expected for an event with neutral opinion bias. However, this result was only valid over an average of a large number of simulations. In particular, the variance in the opinion percentage across the population increases with interconnectivity. Figure 5 shows the distribution of opinion percentage for 100 simulations for a high (20

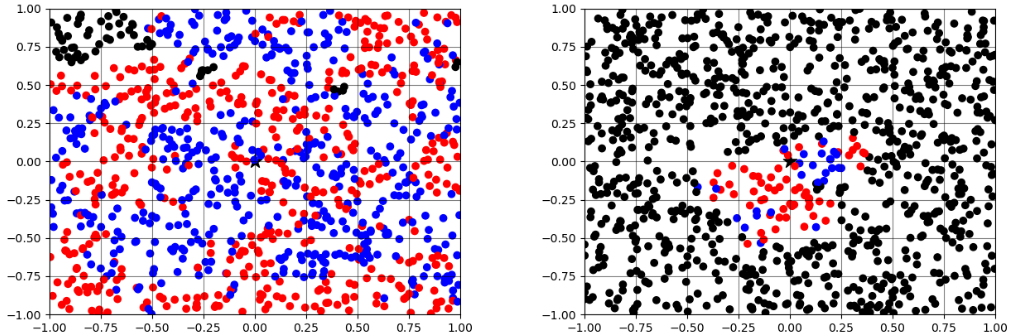

Figure 1: Two examples using interconnectivity of 0.5% (each agent having 5 neighbors). Each point represents an agent, with red for opinion 'R', blue for opinion 'L', and black for neutral. The first example is typical, showing sparsely connected areas of each opinion, and a few clusters of disconnected agents. The second example illustrates when the small number of neighbors causes a halt in opinion spread near the center.

percent), moderate (1 percent), and low (0.5 percent) level of interconnectivity. The variance in opinion spread increases with interconnectivity and is in all cases between 10–20%. The large amount of variability seen in Figure 2 reflects this high amount of variance over just 10 samples. Pilditch (2017) shows that averaging over 1000 simulations consistently produces an opinion split of 50/50, but our results show that there is in fact a substantial amount of variance associated with this average. This is a significant oversight in the conclusions of the original work. The average 50/50 split shown in Pilditch (2017) is purported to suggest that, although clusters or echo chambers opinion exist, the overall opinion split given a neutral event will always be close to 50/50. However, our results show that this is only true for low values of interconnectivity, and opinion splits with interconnectivity above 1% will often create lopsided opinion splits in a population even with a neutral event.

Figure 4 shows the degree of clustering across a population. Although we were only able to complete simulations up to 20% interconnectivity, the degree of clustering decreases with interconnectivity consistent with the results from Pilditch (2017). For real-world online communication networks, 20% is in fact an unreasonably high value of interconnectivity. As expected, the clustering also depends on opinion strength, though weakly. The effect of declaration probability was not thoroughly investigated but is expected to be small.

5.2 Model Extension: "Influencers" and Nonuniform Interconnectivity

Throughout Section 5.1 above, we assumed that all agents were equally interconnected; that is, all agents attended to the same number of neighbors (not all of whom were necessarily declaring their opinions) when forming their opinions. For example, in a population of 1,000 agents with an interconnectivity parameter of 0.5%, we assumed that each agent attended to her $1,000 \times 0.5\% = 5$ nearest neighbors when forming her own opinion. In the real world,

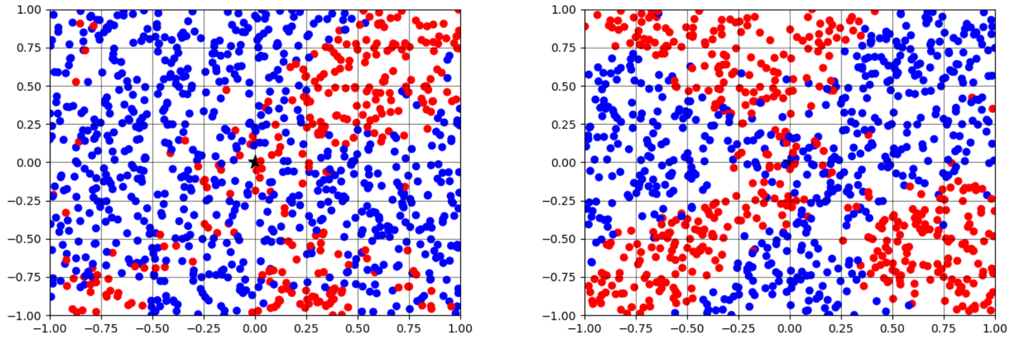

Figure 2: Two examples using interconnectivity of 5% (50 neighbors). Note that although the areas of each opinion are geometrically more compact, clustering in fact decreases due to the large number of neighbors.

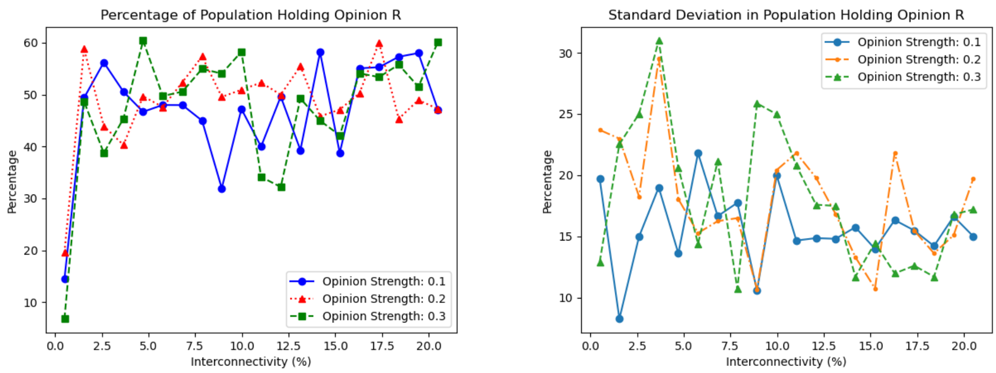

Figure 3: Average percentage of population holding opinion 'R' at the end of a simulation along with the standard deviation over 10 samples for each data point. Note that although the average opinion split is between 40–60% regardless of interconnectivity, the standard deviation is almost always above 10%, (opinion split 40/60), and up to 30% (opinion split 80/20). The degree of noise suggests a much higher sample size would be beneficial.

however, some people's declared opinions have wider reach than just their nearest neighbors. For example, when forming her opinion of a current event, a person may consult her family, friends, and other people in her network (her nearest neighbors) *as well as* the opinions of political commentators, or even "Average Joes" with large social media followings.

We extend our model to account for this fact by introducing a new type of agent, called an "influencer," who is endowed with a "radius of influence" r. Now, an agent's neighbors—the agents she attends to when forming her own opinion—consist of (a) her k nearest neighbors, where $k = \{\text{the number of people in the population}\} \times \{\text{the interconnectivity parameter}\}$, and (b) any influencers who are within distance r of her. For simplicity, we assume that each influencer has the same radius of influence r, and that influencers and non-influencers have

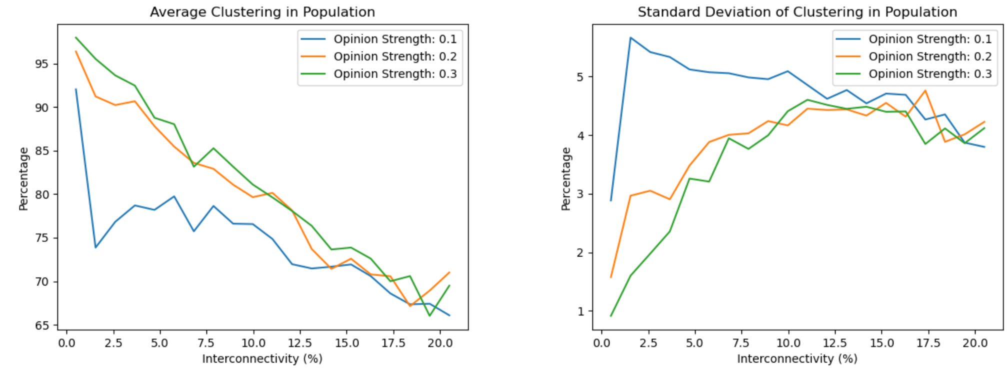

Figure 4: Degree of clustering in the population after opinion spread (average percentage of neighbors with like opinions across the population). Unlike the average opinion split, clustering shows a clear downward trend towards 50% (even mixing) with increasing interconnectivity.

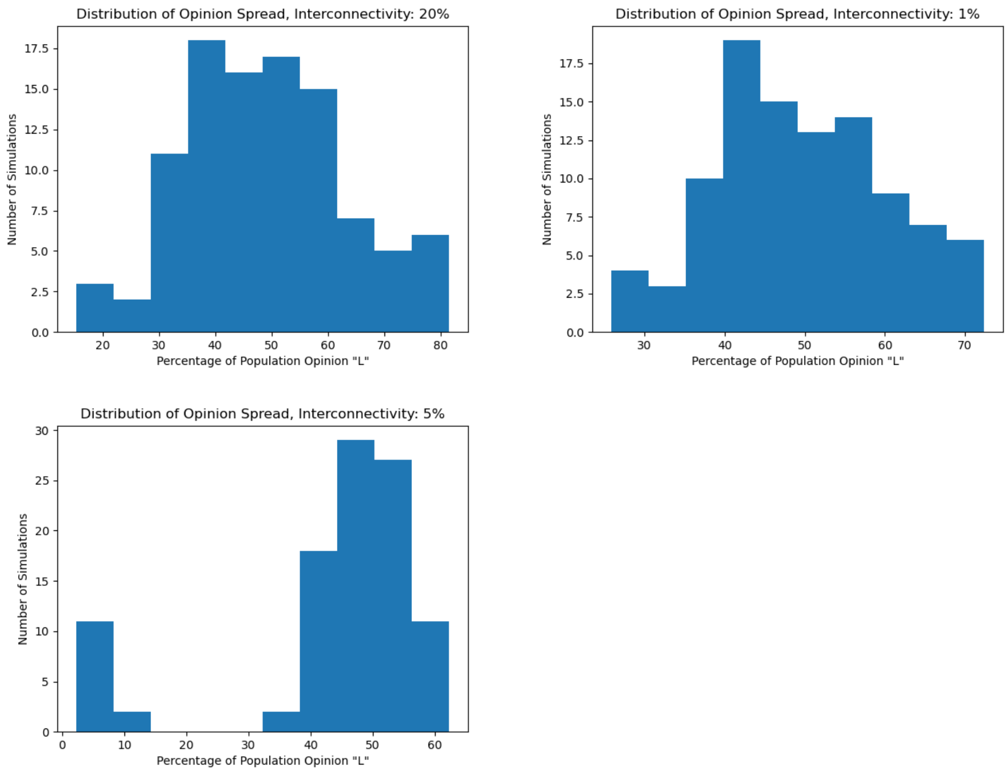

Figure 5: Distribution of opinion splits for three different interconnectivity levels (N = 100 simulations). The mean and standard deviation were 49.6% $\pm$ 14.3%, 49.7% $\pm$ 10.5%, and 43.3% $\pm$ 15.7% respectively. The mean of the 0.5% interconnectivity model was distorted by simulations where propagation stopped early (large numbers of neutral agents), but overall higher interconnectivity causes a greater probability of an uneven opinion split.

the same opinion strength (so that an agent's opinion is swayed just as much by one of her influencer neighbors as by one of her non-influencer neighbors).

As an illustration, in Figure 6 below, we show the outcome of a population with 1,000 agents, approximately 10% of whom are influencers with a radius of influence 0.5, where

- the interconnectivity parameter is set to 0.5% (so that each agent has $1,000 \times 0.5\% = 5$ nearest neighbors that she attends to in addition to her influencers);
- each agent has a declaration probability of 0.5;
- each agent has an opinion strength λ of 0.1; and
- each agent has a learning parameter β of 0.2.

Influencers are marked with squares ■, and non-influencers are marked with circles ●.

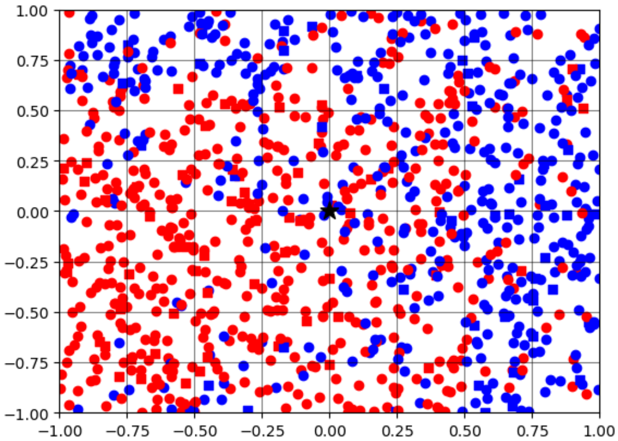

Figure 6: The final opinions of a population comprised of 1,000 agents, roughly 10% of whom are influencers.

Beyond simply modeling populations with influencers, we were interested in studying the extent to which influencers can affect the opinions of a population—or, in short, "How much influence do influencers have?" To do so, we simulated 100 populations, each with 1,000 agents, approximately 10% of whom are influencers with a radius of influence 0.5, where the interconnectivity parameter, declaration probability, opinion strength, and learning parameter values were 0.5%, 0.5, 0.1, and 0.2, respectively (as in the example in Figure 6 above). For each simulation, we recorded the share of influencers on the Left (*i.e.*, with final opinion 'L') and the share of non-influencers on the Left, which we plot in Figure 7(a) below. Observe that there is a clear positive linear relationship between the two shares: when a small share of influencers are on the Left, a small share of non-influencers are on the Left, and when a large share of influencers are on the Left, a large share of non-influencers are on the Left.

We note that the above finding is not normal: in a population without influencers, we would not expect to see this kind of linear relationship between randomly selected subpopulations. To illustrate this, we simulated an additional 100 influencer-free populations, each with 1,000 agents and the same propagation parameters described above. For each

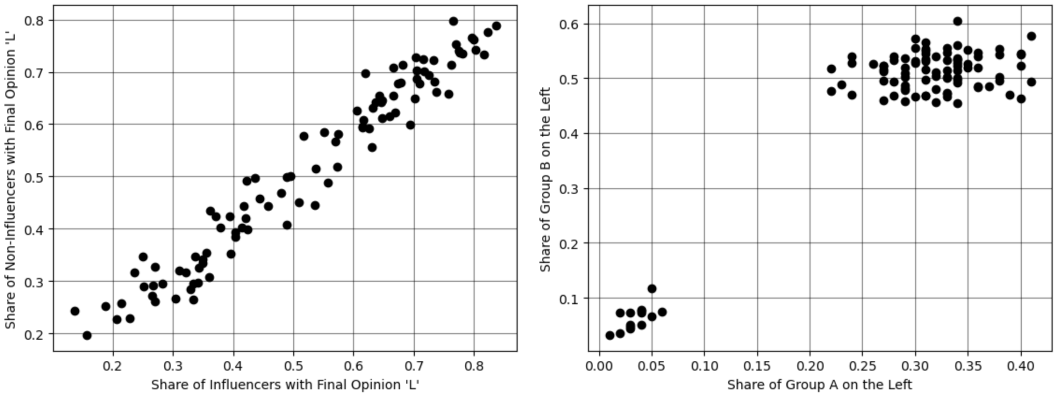

Figure 7: (a) The proportions of one hundred 1,000-agent populations whose final opinion is 'L', broken out by influencers ($\sim$10% of the population) and non-influencers ($\sim$90% of the population). Observe the positive linear relationship between the two sets of proportions. (b) The proportions of one hundred 1,000-agent populations whose final opinion is 'L', broken out into groups of 100 and 900 randomly chosen agents. Observe the lack of any relationship between the two sets of proportions, particularly in the upper-right corner of the plot.

of these simulations, we randomly partitioned the population into a group of 100 agents ("Group A," taking the place of approximately 10% of agents who were influencers in the previous example) and a group of 900 agents ("Group B"). We recorded the shares of Groups A and B on the Left and plotted these in Figure 7(b). This time, observe that there is no correlation:[4] In nearly every simulation, around 50% of agents in Group B were on the Left, even as the share of agents in Group A on the left varied from 20% to 40%. This result is natural—since there is nothing "special" about the randomly chosen agents in Group A, we would not expect Group B (the rest of the population) to follow their opinions.

Taken together, Figure 7(a) and 7(b) showcase an important causal relationship. The positive linear relationship in Figure 7(a) did not simply emerge because the opinion shares of a small subset of a population are bound to resemble the opinion shares of the rest of the population—if this was the case, we would also have observed a positive linear relationship in Figure 7(b). Rather, the opinions of the non-influencers in Figure 7(a) were *shaped by* the opinions of the influencers. Put differently, the shares of opinions held by influencers (roughly) determine the shares of opinions held by the population at large—even, as in our example, when just 10% of the population are influencers—provided that their radii of influence are large enough. That such a small share of a population can have such an outsized influence on the population's overall opinion spread is an important (albeit concerning) result.

[4]For this discussion, we ignore the cluster of simulations in the lower-left of Figure 7b, where very small shares of Groups A and B are on the Left. When a very small fraction of the population is on the Left to begin with, we would expect the shares in Groups A and B to both be small, hence correlated.

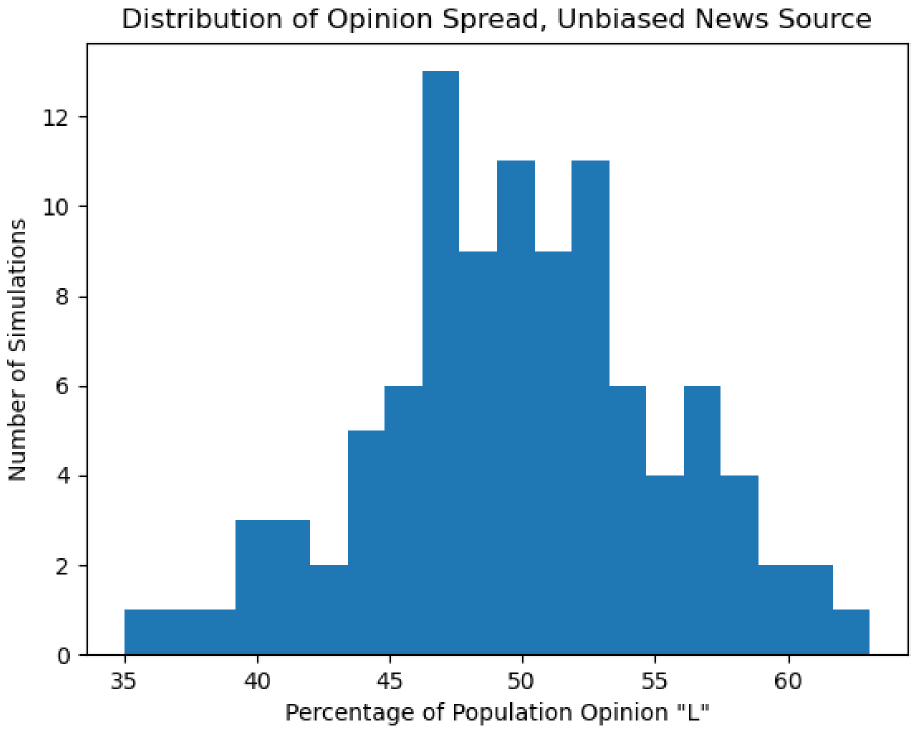

Figure 8: Distribution of opinion spread across a population for $N = 100$ simulations. Note the much more concentrated of this distribution around 50/50 than in Figure 5. The mean of the distribution was $50\% \pm 5.4\%$ The parameters evaluated were 1000 agents, interconnectivity = .5%, $\beta = 0.1$, opinion strength = 0.1, $P_{decl} = 0.5$.

5.3 Model Extension: News Sources and Population-Wide Learning

In the modern world (and perhaps more so throughout history) opinions and facts are not communicated only person-to-person, but are communicated more or less uniformly throughout the population by a (sometimes) trusted source, e.g. the news. Two ways of incorporating this way of communication through a population were explored.

First, we modeled an unbiased news source. Given an unbiased event, an unbiased news source should allow an agent to learn about an event, but not bias that agent towards one opinion or the other. This naturally fits into our learning model by allowing each agent to learn in each step. Where previously, the learning step was reserved for an agent to evaluate an opinion she heard from a neighbor, now each agent completes the learning routine in each time step regardless of whether they heard an opinion during that time step. To represent an unbiased source, the same standard normal distribution was used to model this learning.

However, an increasingly debated topic currently is the bias of news sources. In our model, a biased news source was modeled as an agent with no neighbors (i.e., it doesn't attend to any other agents or change its opinion) that has a probability of declaration of

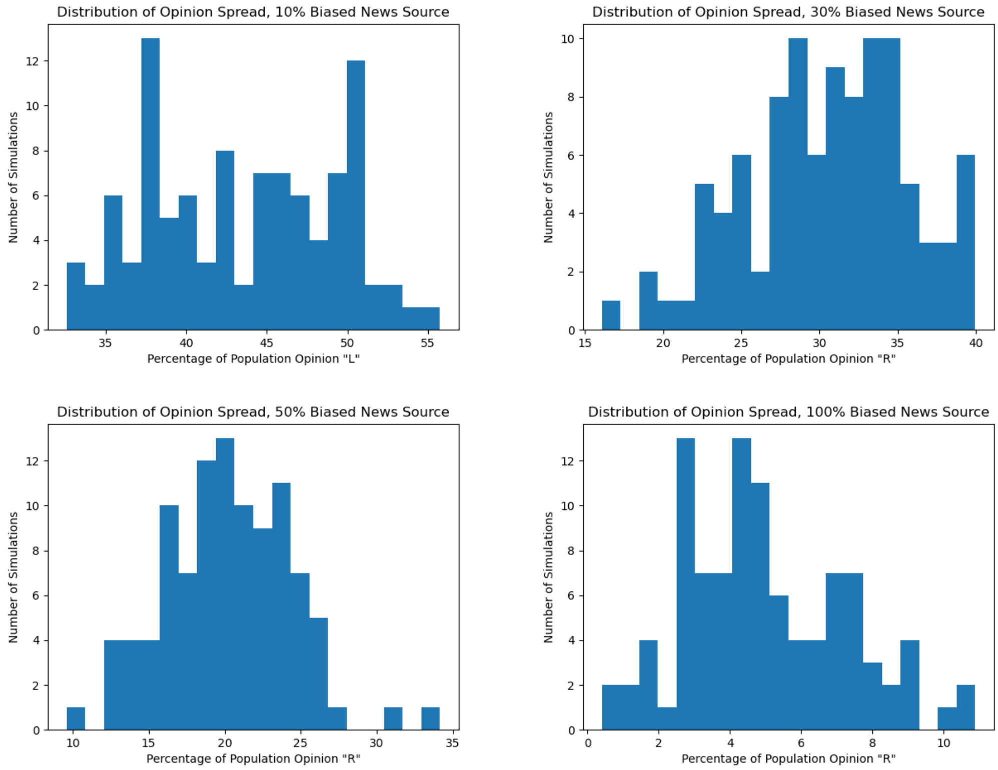

Figure 9: Biased news source with an opinion strength of 0.01, 0.03, 0.05, and 0.1 respectively. The means and standard deviations were: $43\% \pm 5.8\%$, $30\% \pm 5.2\%$, $20\% \pm 4.2\%$, and $5\% \pm 2.3\%$. The parameters evaluated were 1000 agents, inter-connectivity .5%, β 0.1, opinion strength 0.1, P_{decl} 0.5.

one, but which communicates with all other agents. The communication in this case is asymmetric and constant. The asymmetric nature of communication is realistic: in reality, the news is received by viewers not influenced by any individual viewer's opinion (but could be affected by the aggregate opinion). However, the fact that all agents see the same news sources is a mild simplification. Modeling a biased news source as an agent, which applies a deterministic instead of stochastic influence, is a slight simplification as well. But each agent still undergoes stochastic learning after being influenced by a biased source, so the distinction is most likely negligible.

Two important notes are necessary. First, the original model states no re-learning occurs once a final opinion is reached, and that opinion is reached whenever the gradient opinion is non-zero. If some stochastic learning occurs every step from an unbiased news source at each time step, then the probability that any agent's gradient opinion is 0 after any time step is 0. Therefore the simulation would stop after 1 time step. To prevent this, a final opinion is now declared only if the gradient opinion reaches a certain level, taken to be 0.25 in all examples. Another simplistic assumption that was relaxed is that each agent can't re-learn an opinion after having a final opinion. Relaxing this assumption in fact did not produce significantly different results.

Results of opinion spread with an unbiased news source across 100 simulations with identical parameters are shown in Figure 8. Most noticeably, the distribution of opinion spread given an unbiased news source is much closer to 50/50. With opinion spread only occurring through neighbors the opinion spread was on average 50/50, but the standard deviation was typically 15-20% regardless of parameter choices, indicating that for any given event, the opinion spread in the population was likely to be as uneven as 70/30. However, the introduction of an unbiased news source through unbiased learning reduced the standard deviation of opinion spread to about 5%, indicating that the most extreme opinion split for an unbiased event is only likely to by 55/45.

However, this regularizing effect is easily defeated by biased news sources, as shown in Figure 9. To test the effect of biased news, the opinion strength of the biased news agent was set at 0.01, 0.03, 0.05, and .1, representing 10%, 30%, 50%, and 100% of the effect that a neighbor or unbiased learning would have on an agent's opinion. Even with the opinion strength only at 10%, the effect on overall opinion split is noticeable. With 10% opinion strength, the average opinion split was 43%/57%. For the values of 30%, 50%, and 100% of learning strength, the average opinion split grew more uneven to 30/70, 20/80, and 5/95 respectively. One factor which may be amplifying this effect is the fact that all agents listen to the biased news source at all time steps uniformly, but the noticeable effect on opinion spread even for very small relative values of opinion strength is notable.

A Appendix: Nearest Neighbor Computational Complexity

As an aside, it is worth appreciating the computational efficiency of `scipy.spatial`'s `KDTree` class. In our initial construction of the `agent` class, we computed each agent's neighbors using a "brute force" approach, wherein for each agent $i \in \{1, \ldots, N\}$, we computed the distances between agent i and every other agent $j \in \{1, \ldots, N\}$, and then identified the k smallest of those (using `numpy`'s `argpartition()` function). Since we are finding N distances for each of N agents, this approach is $\mathscr{O}(N^2)$. Indeed, observe in Figure 10 that nearest neighbor-identification runtimes for the "brute force" approach roughly coincide with the parabola $g(N) = 0.000003N^2$.

As we discussed in class, KDTrees are much more efficient because they partition the $[-1,1] \times [-1,1]$ square into sub-squares, allowing us to "get away with" only computing the distances between each agent i and those agents in i's sub-square and the eight surrounding sub-squares. Indeed, as shown in Figure 10, using KDTrees brings our nearest neighbor-identification complexity closer to $\mathscr{O}(N)$, with runtimes roughly coinciding with the line $f(N) = 0.0002N$. For sufficiently large numbers of agents (certainly $N \geq 100$), the KDTree-based approach is noticeably more efficient.

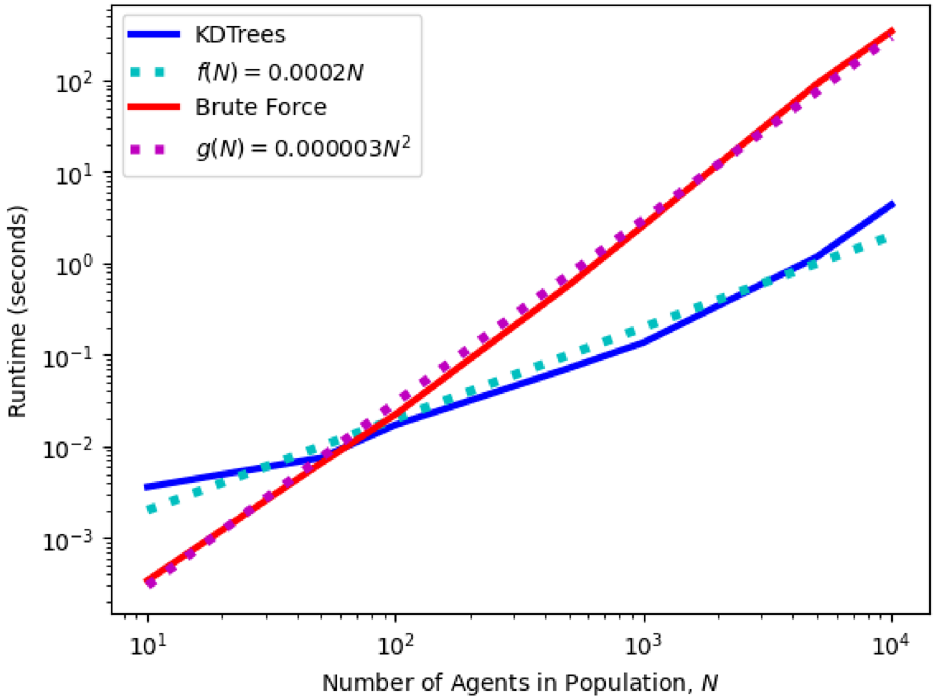

Figure 10: The computational complexity of nearest neighbor identification using KDTrees vs. a “brute force” approach. Observe that the “brute force” runtimes coincide with a quadratic function of the number of agents, N, while the KDTree-based approach more closely resembles a linear function of N.

B Appendix: References

1. Diep, Hung T., Miron Kaufman, and Sanda Kaufman. "An Agent-Based Statistical Physics Model for Political Polarization: A Monte Carlo Study." Entropy 25, no. 7 (2023): 981.

2. Lu, Peng, Zhuo Zhang, and Mengdi Li. "Big data-drive agent-based modeling of online polarized opinions." Complex & Intelligent Systems 7 (2021): 3259-3276.

3. Pilditch, Toby D. "Opinion cascades and echo-chambers in online networks: A proof of concept agent-based model." Cognitive Science Society, 2017.

4. Rescorla, Robert A. "A theory of Pavlovian conditioning: Variations in the effectiveness of reinforcement and non-reinforcement." Classical conditioning, Current research and theory 2 (1972): 64-69.

5. Song, Hyunjin, and Hajo G. Boomgaarden. "Dynamic spirals put to test: An agent-based model of reinforcing spirals between selective exposure, interpersonal networks, and attitude polarization." Journal of Communication 67, no. 2 (2017): 256-281.