An Analysis of the Voter model with an application

Demetrios Papakostas

APM-541 Final Project

Dr. Nicholas Lanchier

December 3, 2020

1 Introduction

In this project, we will examine the voter model and its relation to other models in the course, as well as seeing an application to a fake hypothetical election problem. We compare various parameterizations of our initial conditions in the model, as well as how investigating how the voter model behaves over time, and discuss the implications of the behaviors we study.

Broadly speaking, the models we studied in class can be broken up into three main categories: those that have no spatial interaction, those with some sort of "implicit" spatial interaction, and those with explicit structure. The first is the easiest case, and include the branching process and birth/death processes. A level up in sophistication is the implicit spatial dependence. The spatial dependence here simply means that "space matters", i.e. the locations of units (individuals) are *considered* but there are no constraints as to where in the state space units in the model are allowed to travel. These models were studied in chapters 11 and 12, and include the logistic growth process, the Wright-Fisher model, and the Moran model, described in detail in Lanchier [2017] and first introduced in Verhulst [1845],Wright [1942], and [Moran, 1958] respectively.

While the implicit spatial structure is a step up from the no-space models, and is easier to study analytically, the model is not too enticing when applied to a "real-world problem". Take, for example, the spread of a virus such as the covid pandemic. Were one to model the number of infected with the logistic growth process, for example, they may run into issues with the assumption that all pairs of locations are equally likely to interact. To be explicit, to consider modeling the entire United States with a spatially implicit model such as that seems unwise[1].

Naturally, we might want to consider a model with an *explicit spatial structure*. One such model is the voter model. first introduced independently by Clifford and Sudbury [1973] and Holley and Liggett [1975]. The voter model, in simplest terms, is modeled by a $d-$dimensional integer lattice occupied by an individual of type 0 or type 1. Independently of one another, the individuals update their opinion at a generation of the model by mimicking one of their nearest neighbors (within $2d$), where the neighbor *within* this distance of the individual of interest is chosen uniformly at random.

An interacting particle system is a continuous time Markov chain with state at time t defined by the function:

$$\xi_t : \mathbb{Z}^d \to \{0, 1, \dots, \kappa - 1\} \tag{1}$$

That is elements in the $d-$lattice, **sites**[2] in our model, are mapped to different **types**. The **interaction neighborhood** is given by N_x and the transition rates are given by $c_{0\to1}(x,\xi)$ and $c_{1\to0}(x,\xi)$, described in equations 2 and 3 below:

$$N_x = \left\{ y \in \mathbb{Z}^d : \|x - y\| = 1 \right\} \forall x \in \mathbb{Z}^d \tag{2}$$

$$\begin{aligned} c_{0\to1}(x,\xi) &= \left(\frac{1}{2d}\right) \sum_{y \in N_x} \xi(y) \\ c_{1\to0}(x,\xi) &= \left(\frac{1}{2d}\right) \sum_{y \in N_x} \big(1 - \xi(y)\big) \end{aligned} \tag{3}$$

The following algorithm is used to simulate the voter model. The algorithm is relatively easy to implement in a number of languages; we choose python for ease of visualization. Chapter 16 of Lanchier [2017] provides a graph theoretic approach to study properties of

Algorithm 1 How to simulate the Voter model

procedure VOTER(N, T) ▷ The size of our grid and # of generations
  Create an $N \times N$ lattice(grid)
  $T = \min\{T_1(x,y) : (x,y) \in E\} \sim \text{Exp}(N^2)$ ▷ Time of first update
  First potential update is at vertex uniformly chosen $X \sim \text{uniform}\{1, 2, \dots, N\}^2$
  initialize grid with different types ▷ Set probability of each type per site
  while $t < T$ **do**
    Choose one of the 4 neighbors of vertex X uniformly at random from the grid
  return the type at site

[1] This example is meant to be illustrative; we do not plan to model the covid pandemic in this write-up.
[2] **vertex** is the term we use when approaching the problem from a graph-theoretic approach.

the voter model, exploiting the duality relationship between the voter model and random walks. The voter model has been applied extensively to a wide suite of problems, with several interesting recent applications. Gastner and Ishida [2019] used the voter model to study how opinions spread between social networks, while Gastner et al. [2018] studies a variant of the voter model to study *concealed* voter opinions. Bhat and Redner [2020] used the voter model as a basis to study the influence on contrasting news sources on political polarization

2 The problem at hand

Although relatively simple, the voter model has been used to tackle many interesting problems. Her is the problem we think would be interesting to solve. Say we have an upcoming election to declare the next head librarian in the fictional country of Tredfylton. We are deployed to work for the campaign of candidate A, who we call "A", with an opponent named "B". Upon entry, we are dealt with the issue of two differing opinions within the campaign. Some in the model seem to think a national TV blitz is the game. If a voter in Arizona sees a voter in New Hampshire[3] voting for A, then they may make up their mind to vote for A. The other camp in the campaign thinks this approach is nonsensical. Voters in certain areas will only vote for A if those around them are, or if the campaign directly sends surrogates to *directly* interact with these voters. In a sense, they are hedging their bets on a polarized electorate and want to take advantages of clusters and know how rigid they are. We tend to buy the argument of the latter group more and will explore their claims in some detail in the next section.

2.1 Wright Fisher model

2.2 The voter model

First things first, we present the results of simulating the voter model for 4 different realizations at 4 different time iterations. We assume a vote for A is a win, and a vote for the opponent, a loss. These are coded as going to vote for A as a 1 (blue in our map), otherwise 0 (white in the maps).

Note, we are reporting *realizations* of the voter model. Our point is to show the general idea of more clustering as T increases, but the difference in clusters and final results vary dramatically from realization to realization. For a more thorough analysis, a robust monte carlo simulation should be performed to give a better idea of how the system evolves over

[3] This hypothetical country has the same state structure and nomenclature as the United States.

a course of different simulations. While we do report Monte Carlo results for "low" T situations, the computational burden of doing so for more interesting situations[4] is too great for this exercise.

Figure 1 shows different realizations of our model for 3 different initial types of configurations. All of them at higher T show significant clustering of similar overall ratios, despite the very different places they all started out as. For the polarized scenarios, it shows that while polarization remains steady at "low" T (even though $T = 1$ still includes about 10,000 iterations), the polarized regions will eventually shift and the clustering will be more randomly distributed across the grid.

Figure 2 shows the difference in outcome for win/loss (i.e. preferring our candidate our the opponent for the head librarian position) for the low polarization and high polarization model over time. This time, we have the outcome for every realization in the run, but each time point only gives 2 numbers (technically 1 since win+loss=N^2) to summarize the thought patterns. While we lose some information in these plots, they still remain informative in showing how the system evolves over time. Specifically, we see how there is little change in the high polarization for quite a while, which when combined with our knowledge of the initial conditions, is unsurprising as we expect little change since so many people are surrounded by like-minded people in this configuration.

We also introduce a method for quantifying clustering/polarization. We calculate the % of neighbors a unit has the same opinion as, and do so for all units in the grid. Albeit crude, it is somewhat informative. Figure 3 shows histograms of the % of neighbors (the 8 touching units on the grid) that share the same opinion as a unit. The histogram represents the frequency of each % across the $N \times N = 10000$ units on the grid at the specific time realization. Notice, in the low polarization state, we start with the low polarization but end at $T = 500$ with nearly half of units having *every single* neighbor have the same unit. In the heavy polarization initial state, we start heavily polarized, but end at about the same place as the low-polarization state, and if we were running the models longer we conjecture they'd both converge to the same percentage of all neighbors thinking the same.

[4] For example we could look at regional variations over different runs for example

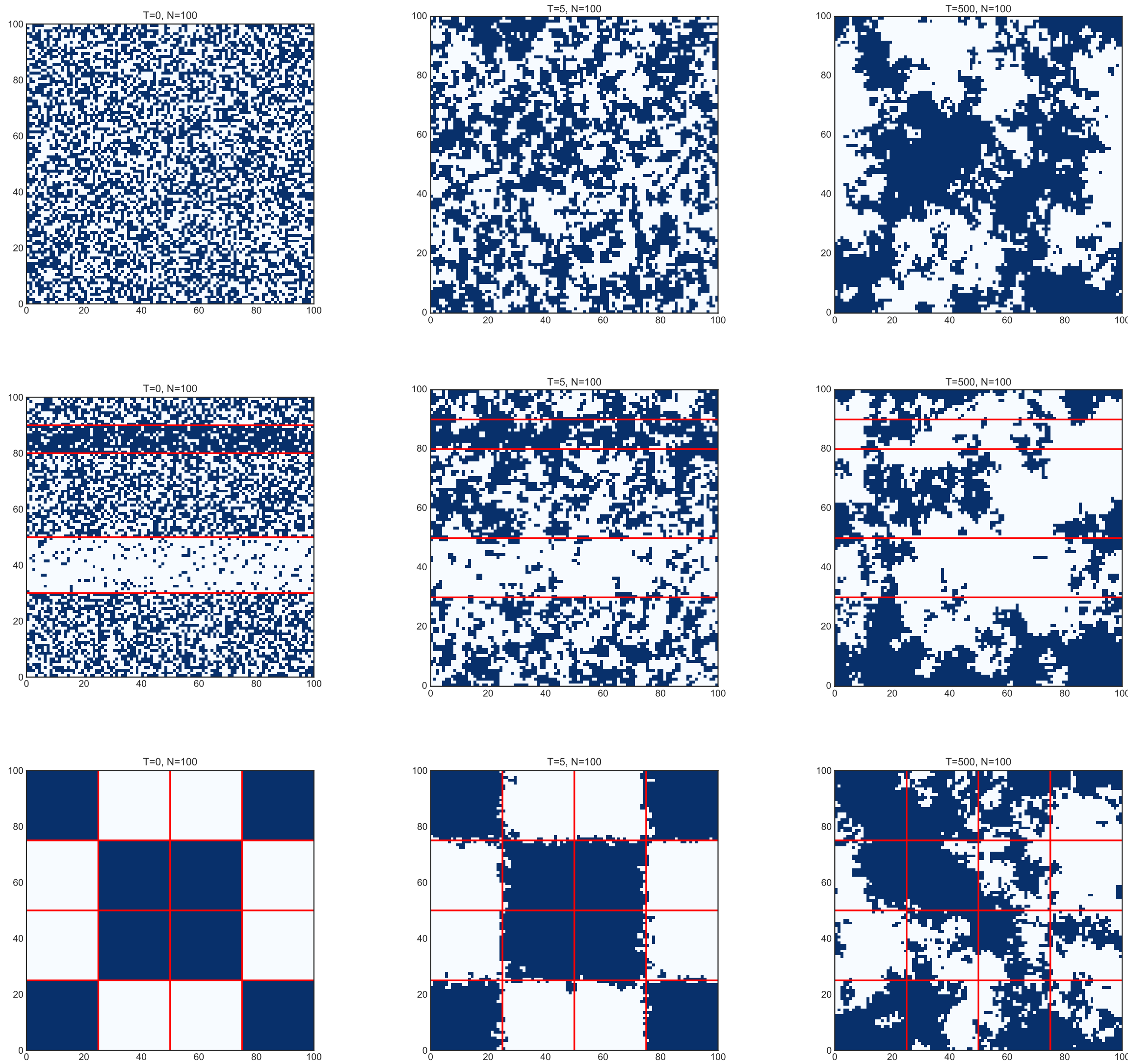

Figure 1: Realizations at different snapshots in time. $T = n$ does not indicate n runs, because the time is exponentially distributed. Rather $T = 1$, for example, translates to about 10000 iterations. Top is low polarization, mid level is medium polarization, where certain parts of the initial map that that are pre-disposed to vote a certain way. We assume in region 30-50, the opponent has a 90% chance of being favored, whereas from 80-90, there is an 80% chance. The regions are outlined by red. The final row is the high polarization, where voters in certain regions are predisposed to vote one way or another. At $T = 2000$ the 3 models clustering patterns all appear randomly distributed, regardless of initial configuration, see figure 4 in appendix.

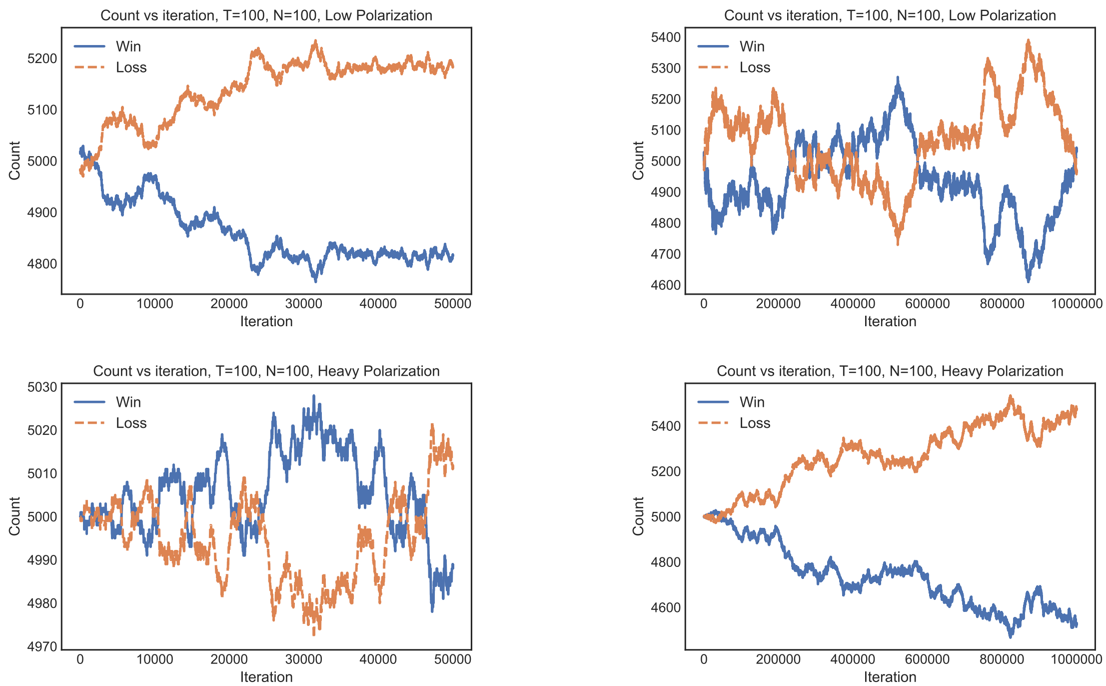

Figure 2: We compare the totals across iterations. The top plots are zoomed in. Notice the scale of the y-axis on the heavy polarization state in the zoomed in plot. There is very little movement during the first 20000 iterations.

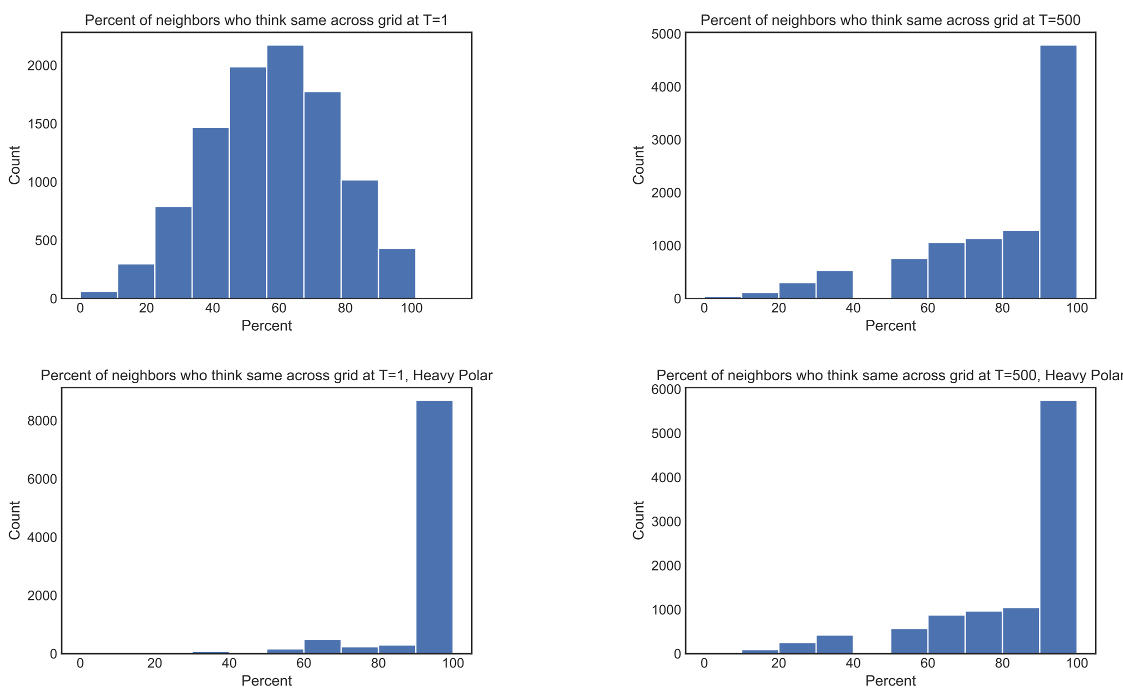

Figure 3: The top row is a histogram of the probability a unit has the same opinion as their neighbor for the low polarization initial set up, at two different realizations of T. The bottom is the same plot but for the high-polarization initial state.

2.3 Monte Carlo Simulations

Here we look at repeating the process at a fixed grid-size and the number of generations pre-determined as well.

Initial polarization	T	Mean P(win)	Var P(Win)	mean % all neighbors think same	Var % all neighbors
Low	10	0.502	0.0004	21.2	0.016
Low	100	0.44	0.001	30.0	0.8
Medium	10	0.484	0.0015	30.4	0.56
Medium	100	0.478	0.0018	33.7	0.78
Heavy	10	0.483	0.0016	79.6	0.40
Heavy	100	0.487	0.0013	63.3	0.64

Table 1: Monte Carlo Simulation: 100 repetitions without any common seed. Repeat the $N = 100, T = 10$ model.

3 Conclusions

For the campaign that employed us, we would likely report back that our model seems to converge towards a polarized electorate, though the rate of that convergence is highly dependent on initial configurations, and predicting where polarization occurs is also a random process. The takeaway is then probably to have a very firm understanding of the initial conditions of the problem.

As for the investigation of polarization, we simply explored one configuration of heavy polarization. In this setting, the spatial configuration of the initial conditions is likely very important. For example, if all the A voters were on the top half of the grid and all the others were on the bottom half of the grid, the results would likely vary *dramatically* for different realizations of the model.

A better metric to quantify clustering [5] would also be another step to look at. Due to computational concerns, calculating an index at *every iteration* with our current methodology would be cumbersome. We'd have to perform the method at every point in the $N \times N$ grid for every of the likely millions of iterations. Our current implementation is in python using the just in time compiler from NUMBA, which helps speed up the process. However, the process is still a burden computationally. Further, due to memory concerns, keeping track of clustering of every site at every iteration is not feasible, and thus choosing a subset of sites, or a single index to summarize the entire grid is necessary.

[5] in our example, clustering corresponds to polarization.

As a mathematician/statistician, one could argue that it is not our place to debate which campaign assumption is correct. Rather, perhaps it is appropriate to model *both* these scenarios under various underlying parameter conditions and report back to camp with our findings. To do so, it could be interesting to also deploy the Moran model and study some properties of it. However, the model feels unrealistic for the problem, which is why it's omitted from this project.

References

D. Bhat and S. Redner. Polarization and consensus by opposing external sources. *Journal of Statistical Mechanics: Theory and Experiment*, page 013402, 2020. URL https://arxiv.org/pdf/1910.01200.pdf. 3

P. Clifford and A. Sudbury. A model for spatial conflict. *Biometrika*, 60:581–588, 1973. 2

M.T. Gastner and K. Ishida. Voter model on networks partitioned into twocliques of arbitrary sizes. *Journal of Physics A: Mathematical and Theoretical*, 52:505701, 2019. URL https://iopscience.iop.org/article/10.1088/1751-8121/ab542f/pdf. 3

M.T. Gastner, B. Oborny, and M. Gulyas. Consensus time in a voter model with concealed and publicly expressed opinions. *Journal of Statistical Mechanics: Theory and Experiment*, page 505701, 2018. URL https://iopscience.iop.org/article/10.1088/1742-5468/aac14a. 3

R.A. Holley and T.M. Liggett. Ergodic theorems for weakly interacting infinite systems and the voter model. *Ann. Prob.*, 3(4):643–663, 1975. 2

N. Lanchier. *Stochastic Modeling*. Springer, Cham, Switzerland, 2017. 1, 2

P.A.P. Moran. Random processes in genetics. *Mathematical Proceedings of the Cambridge Philosophical Society*, 54(1):60–71, 1958. URL https://www.cambridge.org/core/journals/mathematical-proceedings-of-the-cambridge-philosophical-society/article/random-processes-in-genetics/9EEED52D6AE22A026036F32D9B1CA07C. 1

P.F. Verhulst. Recherches mathematiques sur la loi d'accroissement de la population. *Nouveaux M´ emoires de l'Acad´ emie Royale des Sciences et Belles-Lettres de Bruxelles*, 18: 1–42, 1845. 1

S. Wright. Statistical genetics and evolution. *Bulletin of the American Mathematical Society*, 48(4):223–246, 1942. URL https://projecteuclid.org/euclid.bams/1183504254. 1

A Appendix

Figure 4 looks at realizations at $t = 2000$ for three different initial configurations. The take-away is the clustering seems to no longer have the initial patterns in the medium and heavy polarization set-ups.

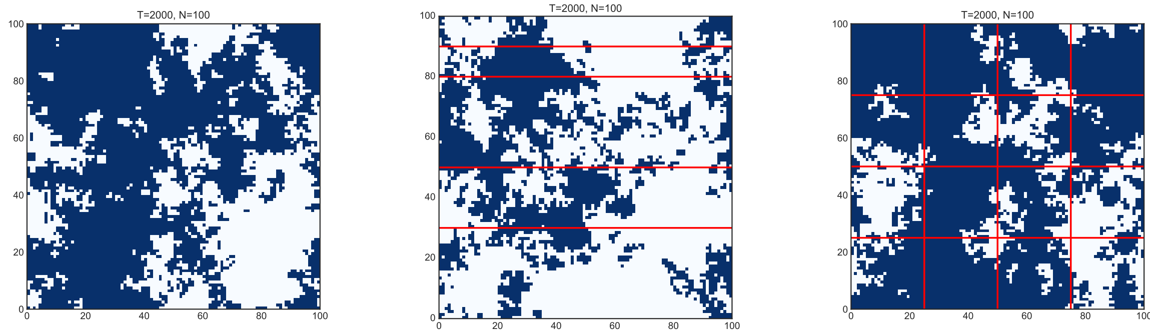

Figure 4: At $T = 2000$, with the model going through many iterations, the clustering appears random and it is hard to tell which initial grid we started from.

We also introduce a "prophet" right at the center of our grid at the onset of the model. The prophet cannot change their mind, and everyone who's come into contact with the prophet not only changes their opinion to align with voting for the prophet, but they no longer are allowed to change their mind. In the sense, the prophet is an "invasive" species, and this wrinkle in the model mirrors the contact process. Figure 5 shows a plot of the voting alignment of the units in the grid over time. Eventually the prophet will take all the votes.

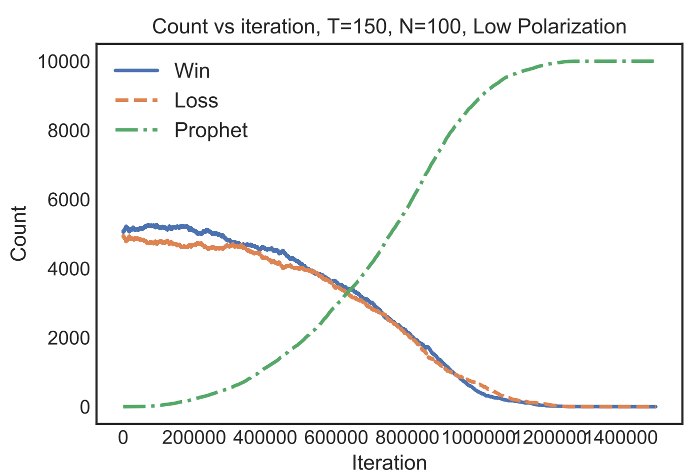

Figure 5: Involving the "Prophet", who's views are unchangeable and whose followers are also unchangeable after their interaction.