Homogeneity of population and it's relationship to partisan gerrymandering

Mahdi Saeedi

Abstract

This article presents an overview of testing An impossibility theorem for gerrymandering written by Boris Alexeev and Dustin G. Mixon. A Districting system and Three Conditions are defined for districts to satisfy. As you read on, you will encounter the definition of the Districting system and its associated conditions. At the theory's heart lies in a relationship between how homogeneous the population is and if the minority party can win. This work will cover the process to test the claims authors made about this districting system, its implications, and possible future work.

1 Introduction

To understand the United States population changes, the census bureau conducts a survey and publishes the results of population changes in counties in each state every ten years. Based on this published data, the states redraw the political districts that elect representatives to the house and congress. We still do not have a clear understanding of the implications of the redistricting process. This means incumbent politicians have the ability to redraw districts in a manner that leads to them staying in power. In this paper, I will explore a form of redistricting that focuses on splitting a state based on the party preference of residents. This is called partisan Gerrymandering. This paper focuses on the homogeneity of the population with respect to party preference and the outcome it has on election results. In [1] the authors propose a theory that states if over 79% of the population are in favor of one party and the population is well mixed, drawing a district that leads to the minority party winning any districts and satisfy three conditions, which we'll introduce shortly becomes impossible. The authors justify this claim throughout the paper by proposing a districting system that must follow three rules inspired by court cases regarding redistricting. Furthermore, variables are presented in an inequality that aims to serve as a mechanism for finding the level of homogeneity that allows for impossibility to occur. We will design a simulation testbed using python to study this theorem. Furthermore, we will use deep learning to investigate the relationships between variables that can be extracted from a district. To comprehend the material in this paper, a solid understanding of operations possible on mathematical inequalities and manipulation methods is required. Furthermore, we will design the system using concepts studied in graph theory (see A.) and associated attributes. To simulate our study, we will be using python programming language

[2], and Google's OR tools [3], a combinatorial optimization library that allows us to iterate through possible districting assignments.

1.1 Graph Theory Definitions

In our context a graph is made up of vertices aka nodes and edges connecting the nodes. In this study we will be using an undirected graph. A graph is an ordered pair G = (V,E). V is a set of vertices, which represent our voters. E is a set of edges representing a connection between two nodes.

$$E \subseteq \{\{x, y\} \mid x, y \in V and\, x \neq y\}$$

We will use nodes to represent voters and establish physical connections between edges.

1.2 Definition of Gerrymandering

Gerrymandering is the process of establishing an unfair political advantage for your party by redrawing the voting districts in a way that maximizes your wins. Flaws in the voting system create politically charged challenges, which further leads to a lack of representation. The United States justice system has been struggling to define a specific mechanism to cope with partisan Gerrymandering. "Excessive partisanship in districting leads to results that reasonably seem unjust," wrote Chief Justice John Roberts. [[4] "But the fact that such gerrymandering is 'incompatible with democratic principles' does not mean that the solution lies with the federal judiciary. We conclude that partisan Gerrymandering claims present political questions beyond the reach of the federal courts. In order to obtain a more comprehensive view of the challenges classifying a district as gerrymandered presents, we should look at North Carolina's 12th congressional district and Illinois's 4th congressional district. N.C. 12th district was the reason behind Rucho V. Common Cause. The argument for Illinois 4th district bizarre shape is to connect to Hispanic communities. I believe this points to the complexity of our redistricting process. Should we allow districts to display weird shapes that would not be categorized as convex shapes? Or is it necessary to draw bizarre shapes to bring representation? The question gets more complex when we try to define representation.

In order to better understand representation, we need to ask citizens what they consider to be part of their communities. What kinds of industries do people work are? What are the religions of people? How does the income distribution look? The central aspect of representation I will focus on in this work will revolve around partisan Gerrymandering. However, it was worth mentioning the diverse set of questions we need to study in order to understand the challenges in redistricting.

1.3 Lack of mechanism to cope with partisan Gerrymandering

In recent years multiple cases [6] in Wisconsin, Maryland, and North Carolina have made their way to the United States' highest courtrooms. The variable for flagging a district as gerrymandered is still not clearly defined. Three ideas that have been quantified using tools offered by mathematics are of

Figure 1: Illinois 4th congressional district [5]

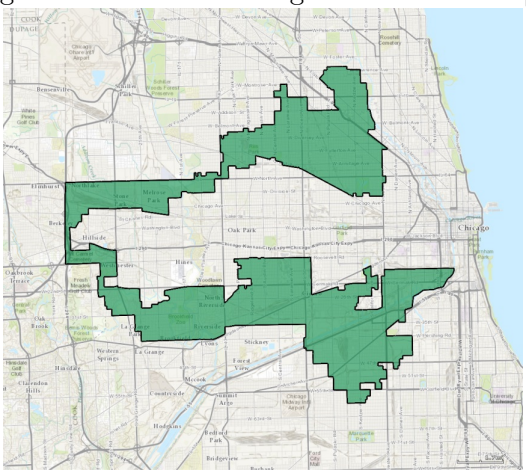
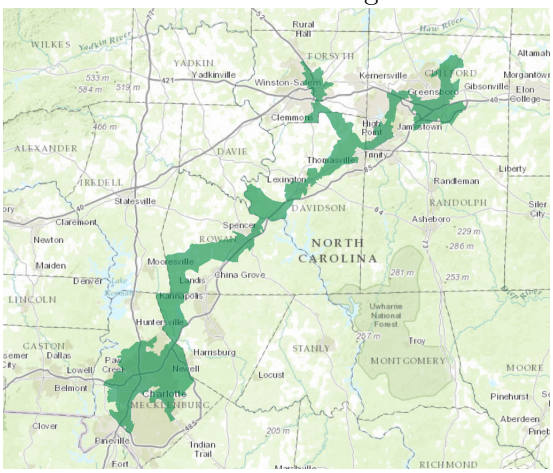


Figure 2: North Carolina 12th congressional district [5]



particular interest in our discussion. Polsby-Popper Compactness [7], Efficiency Gap [8], One Person One Vote [9], Polsby-Popper Compactness: In short, referred to as PP-Compactness, is inspired by the idea of Isoperimetric inequality [10]. Which states circle is the most efficient use of perimeter to enclose an area. We can use the notion of geographic compactness to classify district shapes as bizarre. Meaning we can quantify what it means to be an odd shape. These notions can impose constraints on the distracting process. Efficiency Gap: EG, for short, is concerned with the results of the elections. It considers votes for each party from each district and quantifies how many votes are wasted by each party. A vote is deemed to be wasted for anything over 50% if the party wins. All votes are considered wasted if the party loses a district. EG could raise a flag for situations where a minority party wins a majority of seats. One Person One Vote: Districts must be near equal in size. Most tolerance is set to a difference of 1. As authors of [1] showcase in the above image taken from the paper; we can observe that in a case where we have a lean blue state, it is possible to design

"nice-looking" districts as shown in the middle image and still have no representation for the red voters. The process of making districts with weird shapes can also be used to extract representation.

1.4 Mechanisms to cope with Gerrymandering

I hope by now I have convinced you of the fact that just using shapes to classify districts will not be sufficient. Factors such as forcing districts to be near exact size have been put in place by courts. This leads us to the bread and butter of this paper. We can design computer simulations that can help us quantify the relationship between all the different moving pieces in the redistricting problem. In this work, I will focus on simulating a state population inscribed in a square grid. The nodes are then assigned party preference based on methods described in section 3.1, which enable us to study the effects of population homogeneity and its impact on partisan Gerrymandering. What follows is a definition of a districting system that must satisfy three rules based on our earlier discussions on PP-Compactness, EG, and One Person One Vote.

2 Mathematics of our Districting System

From [1] let the districting system be a function that takes in disjoint finite sets $A, B \subseteq [0,1]^2$ (Here A and B are our voter locations) and a positive number indicating a number of districts we intent on splitting the population into. The system outputs a partition $D_1 \sqcup ... \sqcup D_k = [0,1]^2$ This districting system aims to satisfy the following three rules. However, the Impossibility theorem states that it is impossible to satisfy all of them at once.

2.1 Mathematical definitions of our rules

- (i) One person One Vote: The districts we draw have to contain roughly the same number of voters. There exists $\delta \in [0,1)$ such that the districts always satisfy
  $(1-\delta)\lfloor \frac{|A \cup B|}{K} \rfloor \leq |(A \cup B) \cap D_i| \leq (1+\delta)\lceil \frac{|A \cup B|}{K} \rceil \; \forall i \in [k]$
- (ii) Polsby Popper Compactness: Involves taking ratio of the area of the district to the area of a circle whose circumference is equal to the perimeter of the district (P_D). $PP(D_i) := \frac{4\pi |A_D|}{P_D^2}$ Lowest PP(D) will be $\gamma > 0 \; \forall i \in [k]$
- (iii) Efficiency Gap: This process calculates the wasted votes from each party, first for each district and summing the results. $EG(D_1, ..., D_k; A, B) := \frac{1}{|A \cup B|} \sum (w_{A,i} - w_{B,i})$ For some values of α and β there exists $\alpha, \beta > 0$

2.2 Theory of impossibility

Theorem 1. *There is no districting system that simultaneously satisfies all three desiderata above. In particular, for every δ, γ , α, β, and k, there exist A and B such that every choice of districts*

$\{D_i\}_{(i=1)}^{k}$ *violates one of (1), (2), and (3) defined in section 2.*

2.3 How to set up the simulation

One of the efficient methods of testing this theorem is to use computer simulation. A simulation architecture proposed by Dr. Zerr takes advantage of an open-source combinatorial optimization library called Google OR tools to draw districts, additional functionalities to calculate metrics for produced districts, and visualize results. * Add reference to code on GitHub.

- Creates a grid
- Assigns attributes such as voter preferences to each node.
- Calculates compactness
- Visualizes results

To create homogeneity within our grid of voters, the authors suggest a process by which you group your nodes into smaller squares and create random (location-wise) voter diversity within those squares following a certain proportion of voter distributions. We'll call the number of squares n. This variable will come in handy later on.

2.4 Testing the theory of impossibility

Here n represents the number we split our grid into. A and B correspond to voter locations. b and a represent the number of voters in each square from parties A and B. Step 5 is the inequality we will calculate in the simulation.

$$\epsilon = \frac{1}{n}$$

$$|A \cap D_i| \geq a(\frac{|D_i|}{\varepsilon^2} - E) \geq b(\frac{|D_i|}{\varepsilon^2} + E) \geq |B \cap D_i|$$

$$\frac{b}{a} \leq \frac{|D_i| - \varepsilon^2 E}{|D_i| + \varepsilon^2 E}$$

$$\frac{b}{a} \leq \frac{|D_i| - 4\pi\gamma^{-}1\varepsilon^2 E}{|D_i| + 4\pi\gamma^{-}1\varepsilon^2 E}$$

$$|(A \cup B) \cap D_i| \geq (1-\delta)\lfloor\frac{|A \cup B|}{K}\rfloor \geq (1-\delta)\frac{|A \cup B|}{2k} = (1-\delta)\frac{n^2 l^2}{2k}$$

$$\frac{b}{a} \leq \frac{F^2 - 4\pi\gamma^{-}1F\sqrt{2}\varepsilon - 8\pi^2\gamma^{-}1\varepsilon^2}{F^2 + 4\pi\gamma^{-}1F\sqrt{2}\varepsilon + 8\pi^2\gamma^{-}1\varepsilon^2} \quad (5)$$

Where

$$|\phi D_i| \geq \sqrt{(1-\delta)/(2k)} = F$$

3 Components of the simulation

This section contains pseudo-code steps that should give you a general knowledge of the steps involved in the simulation.

3.1 Grid Setup

The following code snippet will create our nodes and edges based on the population we define. We first build the list holding the node labels corresponding to the sources for the graph's edges. The initial source is labeled 0 and is at the tail of edges that terminate at each census block's nod. Next, we build the list holding the node labels corresponding to the sinks for each graph's edges. Next, create a list of each edge's capacity. For edges connected to census block nodes, this will be the population of the census block. The edges connecting each district to the final sink node will be the sum of the districts' populations divided by the number of districts – i.e., the ideal district size. Next, use the populations passed to the function to create edge capacities. We will also include the tolerance for deviation from an acceptable population size into the capacities array. A final setup step involves attaching costs to each edge in the graph. The costs will be the distances between each census block's centroid and the centroids of the possible districts to which it could be assigned. For this example, I assume census blocks have centroids located at lattice points spaced 1 unit apart. The next procedure involves district centers and an array to hold supplies (i.e., the population of each node).

In the code, these tasks are split into two functions called: 1) setup 2) grid_setup Within grid_setup, we'll make a function call to add_party_preference. This function takes in a grid, a number to split the grid into, and outputs party preference assignments to the nodes.

3.2 District Assignments

We'll call the function dist_assign. First, we'll call grid_setup, which allows us to create the grid and district centers. We'll then pass this information to our optimization process that uses Google OR tools to iterate through district assignments. We'll then take our district assignments, calculate pp-compactness, efficiency gap, and variables from part 2, section D.

3.3 Optimization

This function takes in our nodes, capacities, and costs. It then calls the min_cost_flow function from pywrapgraph [11]. Optimize function returns block_assignments as a pandas data frame.

3.4 Data Collection

Throughout the program, we'll keep track of the variables we are interested in using a mixture of key/value pairs, arrays, and dictionaries with keys pointing to arrays. All the variables are joined together in a master pandas data frame. This will aid in preparing our machine learning data set.

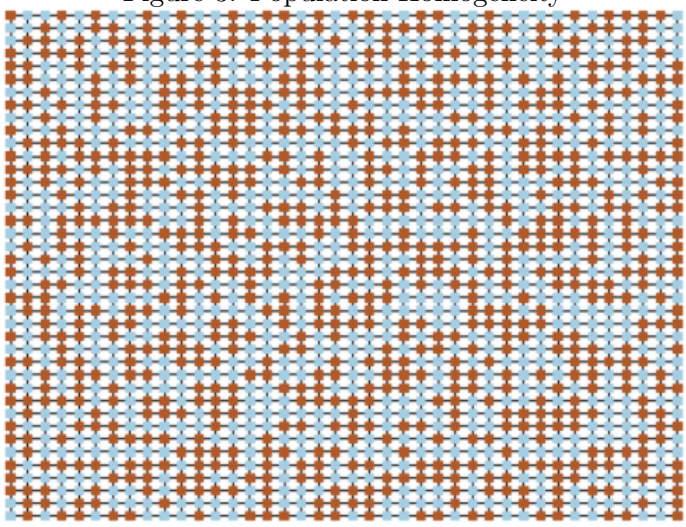
3.5 Putting it all together

All of our functions come together after we define initial values for K (how many districts), n (how many small squares to split the grid into), and L (how many nodes (voters) in each square).

3.6 Data generated

In this section I have included a sample district with 200 voters. The level of zoom aka the N value has been set to 10, with a 0.5 ratio voter preference spread. The simulated sample of voter population homogeneity is displayed in figure 3. If you would like to test out scenarios of districting, you can experiment with the simulation tool using the following link: https://pg-calculator.anvil.app/

Figure 3: Population Homogeneity



We then use the map from figure 3 to create our district assignments, a possible district assignment along with a table of calculated quantifications we discussed earlier are included in figures 4 and 5.

Figure 4: District Assignments

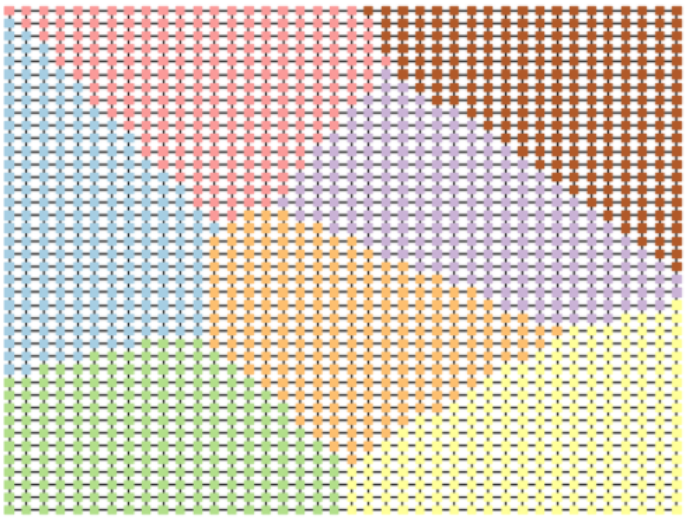


Figure 5: District Metadata

0 votes	1 votes	Efficiency Gap	State Votes / Total Votes (0_X)	State Votes / Total Votes (1_Y)	0 wins / total states (x)	1 wins / total states (y)	x/X	y/Y	district area	district perimeter	pp compactness
114	112	-0.073	0.5	0.5	0.571	0.429	0.875	1.167	56.5	58	0.209
114	115	-0.073	0.5	0.5	0.571	0.429	0.875	1.167	57.25	56	0.228
117	112	-0.073	0.5	0.5	0.571	0.429	0.875	1.167	57.25	60	0.199
116	113	-0.073	0.5	0.5	0.571	0.429	0.875	1.167	57.25	53	0.255
113	116	-0.073	0.5	0.5	0.571	0.429	0.875	1.167	57.25	60	0.199
113	116	-0.073	0.5	0.5	0.571	0.429	0.875	1.167	57.25	54	0.246
113	116	-0.073	0.5	0.5	0.571	0.429	0.875	1.167	57.25	57	0.22

4 Can Machine Learning help?

From our discussion so far, I hope you have gotten a sense of just how many different variables we can extract from the redistricting process. The multidimensionality of our problem set raises the question of is the possible relationships or tensions between the various variables we can track.

4.1 Getting our data ready for ML operations

There are two ways we can go about exploring variables. Figure 7 shows a full list of the variables we can calculate. In this table, we can include an imp_flag column to mark the instances where the impossibility theorem happens. Unfortunately, given that this problem suffers from a combinatorial explosion, I have not been able to simulate enough situations to train a machine learning model yet.

Figure 6: District metadata

votesA	votesB	area	perimeter	pp_compactness	eg	imp_flag	win_count_a	win_count_b	total_votes	...	p_a	p_b	pOverP_a	pOverP_b	n	L	K
100	81	45.25	53	0.201210	-0.391111	0	5	0	900	...	1.0	0.0	0.555556	0	10	9	5
101	80	45.25	54	0.193827	-0.391111	0	5	0	900	...	1.0	0.0	0.555556	0	10	9	5
102	79	45.25	47	0.255862	-0.391111	0	5	0	900	...	1.0	0.0	0.555556	0	10	9	5
99	82	45.25	48	0.245312	-0.391111	0	5	0	900	...	1.0	0.0	0.555556	0	10	9	5
98	78	44.00	54	0.189520	-0.391111	0	5	0	900	...	1.0	0.0	0.555556	0	10	9	5

We can also treat each voter location as a pixel similar to a image recognition problem (see figure 8). This means each column represents a voter, and the value is set to 0 or 1 based on party preference. We will then use the ratio of state votes to total votes over the ratio of wins over total states to create levels that allow us to describe wasted votes. In this case, we can have levels ranging from 0.0 to 1.0 incremented by 0.1.

4.1.1 Statistics about our data

In this section I will go over statistics about a sample space of 2850 instances generated used high-performance computer clusters available at UND. Figure 9 shows the distribution of K (number of districts) we have split the grids into compared to total votes based on N.

Figure 7: Voter location data

voter887	voter888	voter889	voter890	voter891	voter892	voter893	voter894	voter895	voter896	voter897	voter898	voter899	label_a_0	label_b_1
0	0	1	0	0	1	1	0	1	1	0	1	1	0.555556	0
0	0	1	0	0	1	1	0	1	1	0	1	1	0.555556	0
0	0	1	0	0	1	1	0	1	1	0	1	1	0.555556	0
0	0	1	0	0	1	1	0	1	1	0	1	1	0.555556	0
0	0	1	0	0	1	1	0	1	1	0	1	1	0.555556	0
0	1	1	0	0	1	0	1	1	0	0	0	1	0.553333	0
0	1	1	0	0	1	0	1	1	0	0	0	1	0.553333	0
0	1	1	0	0	1	0	1	1	0	0	0	1	0.553333	0
0	1	1	0	0	1	0	1	1	0	0	0	1	0.553333	0
0	1	1	0	0	1	0	1	1	0	0	0	1	0.553333	0

Figure 8: K distribution based on N

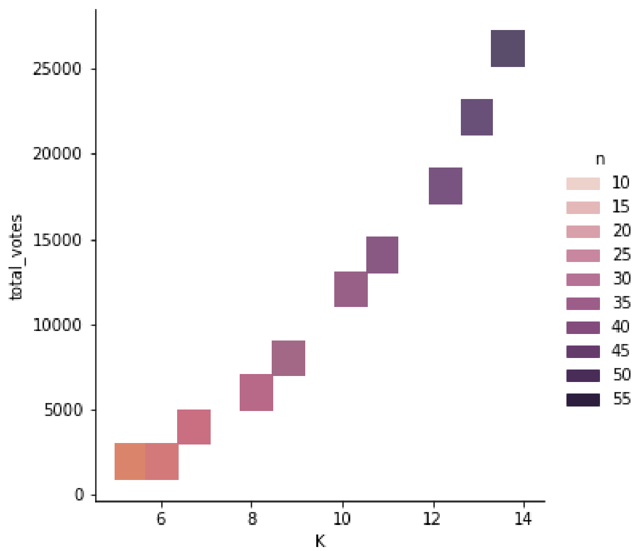


Figure 10 shows how efficiency gap and Polsby-Popper compactness are related as N changes. Figure 11 shows the relationship between Polsby-Popper compactness and K colored coded by N.

4.2 Background on Machine Learning

At the core, machine learning is a method of data analysis that automates analytical model building by examining the data we provide. Machine learning can be separated into two categories: 1) supervised and 2) unsupervised learning. We will focus on unsupervised learning. The main idea behind unsupervised learning is that we should let our algorithm cluster our data, AKA giving our algorithm unlabeled data.

4.2.1 What is Deep Learning

The motivation for applying unsupervised learning is to allow an algorithm to find any connections among the variables. Computers can excel at comparing vast amounts of data compared to naked human eyes.

Figure 9: Efficiency Gap vs PP Compactness based on N

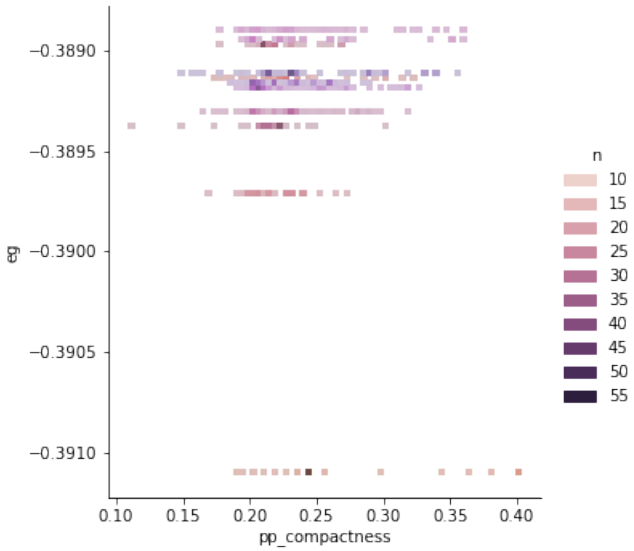
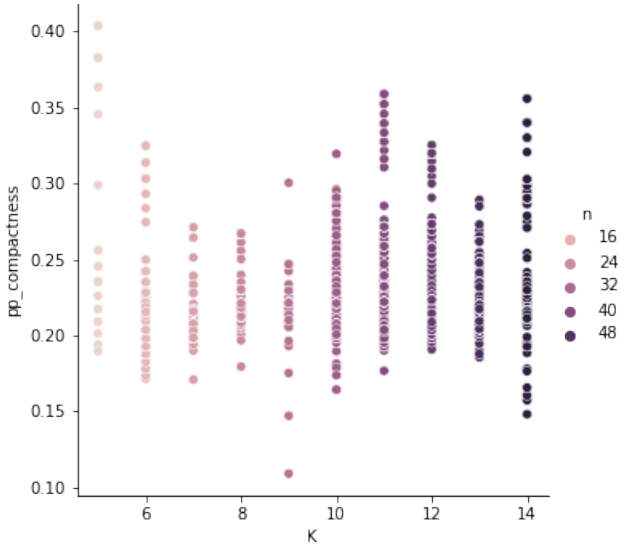


Figure 10: PP compactness vs K based on N



4.2.2 Computation Tools required

We will be using Python to create our statistical models. Since the project's primary focus is studying redistricting, we will use an open-source neural network algorithm from TensorFlow [12], a machine learning library designed by Google. An interactive version of the code will be provided via Kaggle.com.

4.3 Challenges in Designing a Model

The challenges in producing substantial samples with various levels of homogeneity that can represent an actual state require more work in code optimization. Due to this fact this paper only covers the steps involved in simulating the samples and producing the data.

5 Conclusion

This study has covered some of the critical questions courts ask when assessing a legitimacy of a district. I have also pointed out the struggle of the supreme court in deciding on the issue of partisan Gerrymandering. The main contribution of this paper is PG Calculator that allows us to adjust the resolution of homogeneity with respect to party preference to study the quantifications used in Courts. I hope the material I presented so far will get you to think more deeply about what makes your community yours and what you wish to get out of voting for your legislators, congress members, and presidents.

6 Future Directions

6.1 Expansion of Simulation to study more complex communities of interest

The simulation in this work only looked at voter distributions with respect to party preference. Redistricting can happen to protect a diverse set of communities of interest. They can range from party affliction, religion, work industry, etc. Expanding the attribute list of nodes in our graph can allow us to keep track of different characteristics. One idea that seems promising is to take census block data, party preference distributions, race, income distributions for each precinct (the smallest unit electoral district is divided into). Treat each precinct as a square grid. This allows us to develop multiple grids with preferences assigned based on real-life data. Doing this could help us mitigate the challenge of combinatorial explosion, which study redistricting suffers from.

6.2 Make simulation accessible to redistricting communities

Another important future step for this project is delivering an analytical tool that enables redistricting committees across states to quantify the outcome of proposed plans.

7 Acknowledgements

I would like to express my gratitude to my supervisor, Dr. Ryan Zerr, who guided me throughout this project.

References

[1] Alexeev. Boris and Mixon. Dustin G. An impossibility theorem for gerrymandering. 2017.

[2] https://www.python.org/.

[3] https://developers.google.com/optimization.

[4] https://www.supremecourt.gov/opinions/18pdf/18-422_9ol1.pdf.

[5] https://nationalatlas.gov.

[6] https://www.brennancenter.org/our-work/court-cases/current-partisan-gerrymandering-cases.

[7] Polsby. Daniel D. and Popper. Robert D. The third criterion: Compactness as a procedural safeguard against partisan gerrymandering. *9 Yale L. Pol'y Rev.*, 1991.

[8] Stephanopoulos. Nicholas O. and McGhee. Eric M. Partisan gerrymandering and the efficiency gap. *The University of Chicago Law Review*, 82(2):831–900, Spring 2015. Copyright - Copyright University of Chicago, acting on behalf of the University of Chicago Law Review Spring 2015; CODEN - UCLRA2.

[9] https://www.law.cornell.edu/wex/one-person_one-vote_rule.

[10] Osserman. Robert. The isoperimetric inequality. *Bulletin of the American Mathematical Society*, 84(6):1182–1238, 1978.

[11] https://google.github.io/or-tools/python/ortools/graph/pywrapgraph.html.

[12] https://www.tensorflow.org.