Estimating the Probability Distribution of Party Representation as a Result of Political Redistricting Using a Random Walk Monte Carlo Technique

J. Brian Adams
Penn State Harrisburg

Nathaniel Netznik
Penn State Harrisburg

With each decennial census states create the boundaries that are to be used for their legislative districts for the next ten years. In this paper we present a Random Walk Monte Carlo technique that can be used to determine the probability that a set of districts has been drawn without partisan bias – gerrymandered. This is done through the creation of random spanning trees to form the representative districts. Historical election results will then be used to estimate the party representation of that random redistricting map. Through bootstrapping a probability distribution can estimated. This distribution will be used to test the hypothesis that a particular redistricting plan does not disenfranchise voters of that state.

Keywords: gerrymandering, redistricting, Monte Carlo simulation, bootstrapping, probability distribution

INTRODUCTION

The constitution of the United States requires that the federal government conduct a decennial census. The primary purpose of this census is apportionment: using the current population data to create state and federal legislative districts. The population data from the data determines the number of districts in each state. The state then has the responsibility of creating the districts. The only constitutional constraint is that all districts have approximately equal populations. (United States Constitution, Article I, Section 2, 1787)

Each state determines their own process for redistricting. Questions will often arise when the process is controlled by a single party. In the Commonwealth of Pennsylvania, for example, a redistricting commission is created. This five member commission consists of four people selected two each from the Republican and Democratic caucuses. A single fifth member is then selected by both parties together. If they cannot decide on this fifth person, and history has shown that they will not, the Supreme Court of Pennsylvania appoints a person (Constitution of the Commonwealth of Pennsylvania, Article II, Section 17, 1968). The Supreme Court has normally selected a representative of the party that has the majority on the court. As a final step, any maps created must be approved by the current governor of the Commonwealth. While the process is always partisan, since the majority of the Supreme Court will always be from one of the parties, the process has famously been one sided.

It is this partisan nature of redistricting that has resulted in gerrymandering: a process going back over 200 years in which the districts are drawn to favor the party who controls the process. This approach to redistricting became known as gerrymandering despite the fact that Elbridge Gerry, Governor of

Massachusetts in 1812 from whom it is named was opposed to the district map that was created and approved (Spann, 2020).

Gerrymandering is not new; in fact it has been common since the nineteenth century. Until recently it was limited by the time consuming nature of the process. But modern computers have made it possible to create districts in which one party has such complete control that it has eliminated any interparty competition (Kang, 2020). Many of the issues with today's political climate have been attributed to this gerrymandering. The theory is that if the district has no possibility of competition from the outside – another party – then the competition becomes internal. In this case any challenge to the incumbent will be in the primary election by a member of their own party. It is well documented that since voters in primaries tend to the extreme of their party's ideology, the threat of being primaried will often push candidates to these more extremes.

While there has been a push over the past ten or so years to eliminate the partisan redistricting in favor of a non-partisan committee to draw the maps, in most states the change has not been implemented. Instead, most states continue to have elected representatives from the political parties to create the districts.

This is not to say it has not had legal challenges. On the state level, the Supreme Court of Pennsylvania ruled that Congressional maps drawn in 2012 disenfranchised voters by not having their votes count equally. They ordered the legislature to redraw the maps for the 2018 election (League of Women Voters of Pennsylvania, et al. v. Commonwealth of Pennsylvania et. al., 2018).

At the federal level the Supreme Court of the United States did not take the same course. Instead, while ruling that the individual states have the right to create biased districts, they also stated that there is not a standard or mandated method to test if the districts were indeed unfair. In this case, the Robert's court stated

> "determining when political gerrymandering has gone too far" cannot be grounded in a "limited and precise rationale" because the issue "lacks judicially discoverable and manageable standards for resolving." (Rucho v. Common Cause, et al , 2019)

The legal aspects are best left to attorneys and legislators. But the issue of manageable standards may be more addressable.

The issue of the probability of a particular outcome has recently become a topic of research. In a recent paper an attempt is made to evaluate the degree of partisanship that results from the hyperpartisan redistricting (Burden & Smidt, 2020). In this paper the authors have used the simulation techniques to estimate the probability distribution of a known set of district maps.

It is hypothesized that if a probability distribution of the expected party representation in the districts were known, then an interval estimate can be made as to what a fair district map might be. In this case we would define fair as one that is likely – within some probability p to occur if the districts had been assigned randomly. This has been attempted using Markov Chains (Barkstrom, Dalvi, & Wolfram, 2018). In this model the authors created random districts then calculated an estimate election outcome. Their hypothesis is that if the probability of attaining a mix that is the same as the actual legislative model is small then the process must be a result of partisan gerrymandering. In their model they create the districts using Markov Chains.

We aim to take a different approach. Since the precincts in a state can be modeled as nodes of a graph, and the districts formed from those nodes are trees, a set of random districts can be formed by creating k connected trees from the graph. Once the formed the precinct level election results can be used to form an expected outcome of the districts with respect to an actual state or national election. By repeating this random district creating thousands of times a probability distribution for each possible election result can be estimate. This bootstrapping technique creates the interval estimation of this party representation. With the expected distribution of party representation a probability that the actual representation could have occurred without partisan interference can be predicted.

METHODOLOGY

There are two parts to the development of the interval estimate of party representation: the network graph and a set of N trees formed from the graph, and the estimation of the probability distribution. The probability distribution will be estimated using a Monte Carlo technique known as bootstrapping. The development of the N trees will be the application to which the bootstrapping will be applied.

Bootstrapping is a simulation technique in which you generate random values that represent the outcome of a process (Murphy, 2012). With each iteration the outcome is recorded. By repeating the random number generation thousands or even millions of times, the frequency distribution of each outcome can be estimated. This relative frequency will be an estimate of the probability of that outcome.

For the bootstrapping, the random variable, **X** will be a tuple representing the number of representatives of each party. The sum of the two values will always be equal to the number of districts. Thus, if there are *N* districts then the sum of each value in X_k will equal *N*.

$$X_k = (x_{k1}, x_{k2}) \tag{1}$$

$$N = x_{k1} + x_{k2} \tag{2}$$

This creates a probability distribution with the random variables

$$\mathrm{X} = \{(0, \mathrm{N}), (1, \mathrm{N}-1), \cdots, (\mathrm{N}, 0)\} \tag{3}$$

Each outcome will have a probability estimated by the bootstrapping. This involves generating random numbers to determine the outcome – in this case the number of representatives from each party. For each outcome a frequency is generated by the simple matter of counting the number of times each outcome occurs. The probability is then estimated as the relative frequency,

$$\mathrm{P}\big(\mathrm{X_k} = (\mathrm{x_{k1}}, \mathrm{x_{k2}})\big) = \frac{\mathrm{f_k}}{\mathrm{n}} \tag{4}$$

where n is the number of times that the simulation is run. The individual outcomes will be generated by creating random trees from a network graph that describes the state.

In this model, each voting district, heretofore a precinct, will be represented by a node on the graph. By creating an adjacency list of the nodes, random trees will be created. Each tree will represent a possible legislative district. The number of trees will thus be equal to the number of legislative districts, in this case N. As this is a simulation to determine the probability of a particular outcome tree, there will be no assumption that the trees are anything other than that they are connected. They will not be, nor are they intended to be, minimum cost trees (Wayne & Sedgewick, 2011).

Beginning with the unconnected graph where each node is a precinct, we will begin connecting them into a tree to form a district. Since the decision to stop adding nodes to the graph is determined by a preset population value, an accumulator for the population is set to zero. Then a currently unconnected node will be chosen at random and its population is added to the accumulator. Using the nodes in its current adjacency list, a second node is randomly selected, and a third and fourth until the tree is complete. Each time the population of the additional precinct is added to the accumulator. The process is repeated until a maximum population threshold is reached.

Once the first district – the first tree – is created, the process will be repeated starting with another, currently unconnected, node. This tree creation will be repeated until N districts have been formed.

The N trees create the legislative districts, but the goal is the probability distribution of the political party representation. After each district is created, we will use a vote total for each candidate from all of the precincts to determine the district's winning party by popular vote. Once all districts are constructed,

we will record the number of Republican districts and number of democratic districts in the region. This will be a single outcome for the probability distribution.

Each time that the party representation is calculated the accumulator for that outcome is incremented. By repeating this process n times the discrete probability distribution of the party representations will be estimated.

This probability distribution can now be used to predict the probability that the current arrangement could happen by random assignment. If this probability is small, then a statistical argument can be made that the hypothesis of no bias is invalid and should be rejected. This could be done for both the federal districts, state districts, state senate districts, or any state. To test the algorithm, a data file will be created of random precincts. Each precinct will have an ID, a population, an adjacency list of its surrounding precincts, the number of Republican votes and the number of Democratic votes.

The inputs to the model are the maximum regional population size *m*, the desired number of districts *N*, and the desired number of runs in the simulation, *n*. Throughout each run, the program maintains an adjacency list for all of the unconnected precincts. Note that by definition, two precincts are adjacent if they have at least one vertex in common. As precincts are added to districts in a run, they are removed from the adjacency list. The adjacency list is reset at the beginning of each run.

A tree object represents each tree (district). The object contains the following attributes, which are each dynamically modified throughout the district construction process:

- nodes: a list of precincts included in the district
- choices: a list of precincts outside of the district that are adjacent to at least one precinct within the district. This yields possible choices for selecting a new precinct to add to the district.
- population: the total population of the district
- r_k: the total number of Republican votes in district *k*
- d_k: the total number of Democrat votes in district *k*

As the outcomes $X_k = (x_{k1}, x_{k2})$ are updated throughout the simulation, the program maintains a list *d* of length *n+1*. The value at index i of the list, $d_k[i]$, is the number of times that a region has *i* Democratic district wins in the run. Once a region is created and the winning party of each district decided, the $d_k[i]$ list is updated accordingly – if the region has *i* predominantly Democratic districts, then $d_k[i]$ is increased by one. At this point, the adjacency list is reset to contain all precincts again; new regions are created and processed according to the procedures outlined above until the script reaches the specified number of runs. Once all of the regions have been constructed, the program generates a probability distribution from the information stored in the d list. The first two columns correspond respectively to the number of Democratic districts and number of Republican districts that could occur in a run. The third column reports the frequency - that is, the number of times that the arrangement defined in the first two columns occurs (notice that the frequency for row *i* simply corresponds to $d_{wins}[i]$). The fourth column converts this frequency to a probability, dividing by the number of runs.

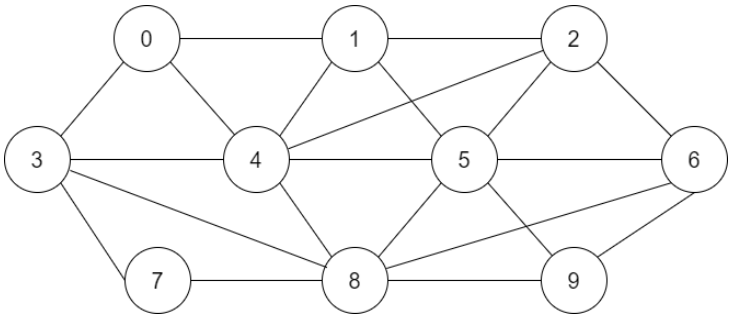
TABLE 1
STRUCTURE OF TABLE GENERATED BY SCRIPT THAT DISPLAYS PROBABILITY DISTRIBUTION

D	R	F	P
0	N	$dk[0]$	$dk[0]\ /n$
1	$N-1$	$dk[1]$	$dk[1]\ /n$
⋮	⋮	⋮	⋮
n	0	$dk[N]$	$dk[N]\ /n$

RESULTS

We ran the simulation on a set of sample precinct data for the geographic district modeled by the network given below.

FIGURE 1
A NETWORK REPRESENTATION FOR A DISTRICT



In this example, the population for a precinct is uniformly distributed between 500 and 1500. The number of Republican votes is uniformly distributed between 25% to 75% of the corresponding precinct's population, and the remainder of the precinct's population is accounted for by the other party.

TABLE 2
DATA FOR THE EVALUATED REGION

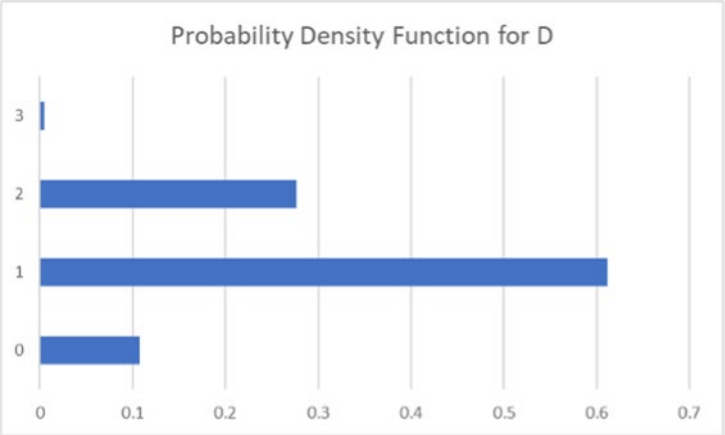
ID	POP	ADJ List	Rep	Dem
0	1411	1 3 4	656	755
1	971	0 2 4 5	541	430
2	1163	1 4 5 6	319	844
3	744	0 4 7 8	397	347
4	760	0 1 2 3 5 8	276	484
5	625	1 2 4 6 8 9	216	409
6	1329	2 5 8 9	975	354
7	1006	3 8	592	414
8	1379	3 4 5 6 7 9	897	482
9	1073	5 6 8	537	536

We ran the simulation on this data set with a maximum population size of 10,000, 3 districts per region, and 10,000 total runs. The probability distribution generated by the simulation for the data set was generated and is shown in table 3.

TABLE 3
PROBABILITY DISTRIBUTION GENERATED FOR THE GIVEN GEOGRAPHIC DISTRICT

D	R	F	P
0	3	1074	0.1074
1	2	6117	0.6117
2	1	2762	0.2762
3	0	47	0.0047

FIGURE 2
PROBABILITY DENSITY FUNCTION FOR D



Of all ten precincts in the example given above, four of them have a democratic majority. The expected value of D is 1.1782, while the expected value of R is 1.8218. The most likely outcome is 1 D, 2 R; the least likely outcome is 3 Democrats, 0 Republicans.

The original question posed by the Supreme Court opinion was whether it can be shown that the redistricting was biased. From a statistical approach bias would be determined by whether or not it would be reasonable to accept that the districting results would be expected to happen. Using a critical value of 5% we can expect the three Democratic representatives to happen less than 5% of the time. As such we would reject a hypothesis that a three Democrat, zero Republican redistricting was not gerrymandered to provide partisan favoritism.

CONCLUSIONS

In the case of an election in the provided region, it is most likely that a Republican candidate would win the region. In particular, it is very unlikely that all three regions would be dominated by the Democratic party.

We are currently in the process of setting up and conducting the simulation on historical data for Pennsylvania. We began the process by writing a Python script that reads in shapefiles containing data for each of Pennsylvania's precincts. The resulting data set contained data for 9253 precincts, each with an id, name, population, and adjacency list. After obtaining voting data for each precinct, we will perform the same simulation but now with eighteen districts per run.

The distribution generated by the simulation will then be used to determine the probability of an associated historical outcome occurring. If historical records reveal an outcome that is unlikely by random assignment according to the distribution, we reject the hypothesis of no partisan gerrymandering and conclude that there is statistically significant evidence that the historical election's district map is unfair.

REFERENCES

Barkstrom, J., Dalvi, R., & Wolfram, C. (2018). *Detecting Gerrymandering with Probability: A Markov Chain Monte Carlo Model*. (Unpublished paper).

Burden, B., & Smidt, C. (2020). Evaluating legislative districts using measures of partisan bias and simulation. *Sage Open*, *10*(4), 1–12.

Constitution of the Commonwealth of Pennsylvania, Article II, Section 17. (1968).

Kang, M. (2020, October). Hyperpartisan Gerrymandering. *Boston College Law Review*, *61*(4), 1379–1445.

League of Women Voters of Pennsylvania, et al. v. Commonwealth of Pennsylvania et. al. (2018). J-I-2018.

Murphy, K.P. (2012). *Machine Learning: A Probabilistic Perspective Adaptive computation and machine learning series* (First ed.). MIT Press.

Rucho v. Common Cause, et al. (2019). 318 F. Supp. 3d 777 and 348 F. Supp. 3d 493.

Spann, G.A. (2020). Gerrymandering Justiciability. *Georgetown Law Review*, *108*(4), 981–1025.

United States Constitution, Article I, Section 2. (1787).

Wayne, K., & Sedgewick, R. (2011). *Algorithms* (First ed.). Addison Wesley.