Empirical Validation of Automated Redistricting Algorithms on the Virginia House of Delegates District Map

Madeleine Goertz

International Community School

AP Research

Randall S. Huberman

May 20th, 2021

Method

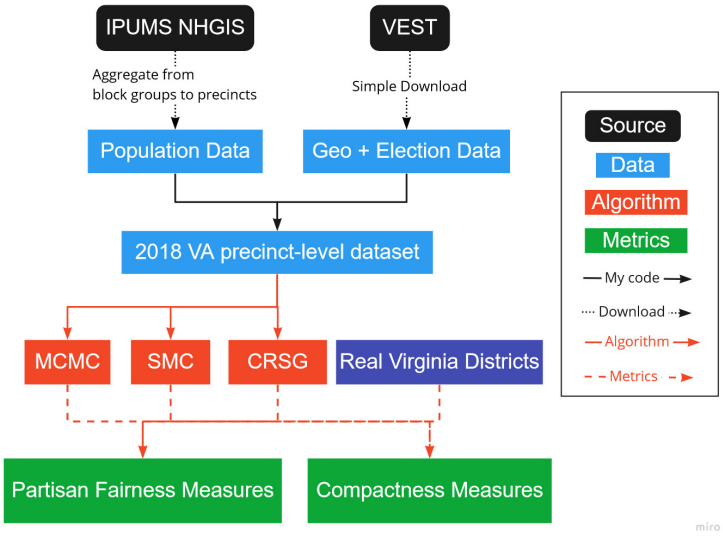

Figure 7

Graphical Overview of project

My research method simulates redistricting the legislative districts for the Virginia House of Delegates using three different algorithms in the years 2015, 2017, and 2019. Figure 7 provides and overview of this process. I begin by explaining my overall research method, explaining how I'm aligned to both the components and principles of experimental designs, and then explaining the data collection and cleaning process.

Choice of Research Method

For this study, I chose to use the experimental design method because it will allow me to isolate the hypothetical impact of the redistricting algorithm from other possible confounding variables. This method also includes the use of a control group, which allows the researcher to establish causation.

Components of Experimental Design

Experimental Units. The experimental units for this study are the complete datasets for each election year in Viriginia.[10] I have one dataset for each of these years: 2015, 2017, 2019. Every row in each dataset corresponds to a precinct, the smallest geographical unit by which votes are tabulated in Virginia. For each precinct, I have the following attributes: total population, population by race, total voting-age population (VAP)(population over the age of 18), VAP by race, total votes for the democratic House of Delegates (HOD) candidate, total votes for the Republican HOD candidate, and the total votes for any other HOD candidate. Additionally, each precinct has a polygon associated with it that represents its geographical shape.

Treatments. The treatments for this study are the three different redistricting algorithm that I'm comparing: Markov chain Monte Carlo (Fifield, Higgins, et al., 2020), Sequential Monte Carlo (McCartan & Imai, 2020), and Random Seed Growth (Chen & Rodden, 2013).[11] I'm using the implementations in the R programming language "redist" package (Fifield, Kenny, et al., 2020). See the Literature Review section for a deeper dive into these algorithms. Broadly, I chose them because they are deterministic. Much of the literature focuses on creating many possible redistricting plans for a commission to choose from, but these three aim to create an "ideal" map.

Response Variables. Broadly, the goal will be to evaluate how "fair" each redistricting plan generated by each algorithm for each year is.[12] Within the literature, partisan symmetry is the primary principle used to evaluate redistricting plans.

Chamber Power Balance. Since the redistricting that's occuring is hypothetical and I have precinct-level election results for each of these years, I can simulate what the power distribution in the VA House of Delegates would be if the proposed

---

[10] This corresponds to the blue rectangles in Figure 7.

[11] This corresponds to the red rectangles in Figure 7

[12] This corresponds to the green rectangles in Figure 7.

redistricting plan had been used.

Control Group. The official VA House of Delegates map used in the years 2015-2019 will serve as the control group for this experiment. I will compute the same metrics for this map as I will for my hypothetical redistricting plans.[13]

Principles of Experimental Design

The primary principles of experimental research design are randomization, replication, and local control. This is how I plan to address them.

Randomization. Every experimental unit will receive each treatment, and every experimental unit can be replicated many times without issue, so thereâĂŹs no error from a lack of randomization. Think of each treatment operating within a separate parallel universe.

Replication. There is no need for me to run my trials several times (run the same algorithm on the same data set several times) because these are deterministic algorithms, and the datasets will be immutable.

Local Control. All of the redistricting will be happening in controlled environments, so there will be no way for lurking variables to creep in and confound my results.

Data Cleaning

To create my datasets, I cleaned and compiled three different types of data: demographic data, Geographic Information Systems (GIS) data, and election data.[14]

Demographic Data

One required piece of data in order to redistrict is demographic data at the precinct level. This means both the total population and the Voting-Age Population (VAP) broken

[13] This corresponds to the purple rectangle in Figure 7.

[14] This is an explanation of the black and blue rectangles in Figure 7 and the transitions between them.

down into the Non-Hispanic White, Non-Hispanic Black, and Hispanic categories. In order to run the most accurate redistricting simulations, these data needed to be recent for the year being redistricted, meaning I needed different data sets for 2015, 2017, and 2019. Comprehensive population counts are only conducted by the US Census Bureau every 10 years, so I instead used the 5-year American Community Survey results at the block-group level. This is a sample survey, not a population count, but that is offset by the aggregation of sample data over a 5 year period. I downloaded this data from the IPUMS National Historic GIS project (Manson et al., 2020). Using the "maup" Python Library (Hully, n.d.), I disaggregated the data from the block-group level to the block level, prorating the demographic data based on population. This data was then aggregated up to the precinct level.

However, IPUMS did not have VAP by race data, so I downloaded that separately from the US Census Bureau (Bureau, n.d.) and cleaned it in a similar fashion, merging it into my precinct tables.

GIS Data

In order to redistrict, the algorithms need to know the shape and relative location of each precinct. In practice, this means every precinct has a "polygon" associated with it and a Coordinate Reference System that describes where these polygons fall in space. These data tables with the geometry column are known as "shapefiles." I accessed these shapefiles from the Voting and Election Science Team on their Harvard Dataverse (Voting and Election Science Team, 2019a, 2019b, 2019c). I then merged in my precinct-level demographic data tables, so I now have shapefiles with the necessary demographic data. [15]

---

[15] Since election administrators are free to change the precincts between elections, precinct shapefiles are unique to both a place and a time. This was the reason that I only ran simulations in the years 2015, 2017, and 2019, since these were the only years for which I was able to find reputable precinct shapefiles.

Election Data

The last necessary component needed to evaluate redistricting plans is the number of votes one by each party in each precinct in each election.[16] The Virginia Department of Elections publishes historical records of every election on their website (Virginia Department of Elections, n.d.), which I cleaned and aggregated to arrive at a party vote count for each precinct for each year. These data were then merged into my precinct shapefiles.

---

[16] The algorithms I'm comparing assume a 2 party system, so I only tracked Democratic and Republicn votes won in each election.

References

Bureau, U. C. (n.d.). Citizen Voting Age Population by Race and Ethnicity. Retrieved February 2, 2021, from https://www.census.gov/programs-surveys/decennial-census/about/voting-rights/cvap.html

Chen, J., & Rodden, J. (2013). Unintentional Gerrymandering: Political Geography and Electoral Bias in Legislatures. *Quarterly Journal of Political Science*, *8*(3), 239–269. https://doi.org/10.1561/100.00012033

Fifield, B., Kenny, C. T., McCartan, C., Tarr, A., Higgins, M., Kawahara, J., & Imai, K. (2020). Redist: Simulation Methods for Legislative Redistricting. Retrieved January 29, 2021, from https://CRAN.R-project.org/package=redist

Fifield, B., Higgins, M., Imai, K., & Tarr, A. (2020). Automated Redistricting Simulation Using Markov Chain Monte Carlo. *Journal of Computational and Graphical Statistics*, *0*(0), 1–14. https://doi.org/10.1080/10618600.2020.1739532

Hully, M. (n.d.). Maup: The Geospatial Toolkit for Redistricting Data. Retrieved February 18, 2021, from https://github.com/mggg/maup

Manson, S., Schroeder, J., Van Riper, D., Kugler, T., & Ruggles, S. (2020). National Historical Geographic Information System: Version 15.0. https://doi.org/10.18128/D050.V15.0

McCartan, C., & Imai, K. (2020). Sequential Monte Carlo for Sampling Balanced and Compact Redistricting Plans. *arXiv:2008.06131 [cs, math, stat]*. Retrieved January 18, 2021, from http://arxiv.org/abs/2008.06131

Virginia Department of Elections. (n.d.). Results/Reports. Retrieved January 13, 2021, from https://www.elections.virginia.gov/resultsreports/

Voting and Election Science Team. (2019a). 2015 Precinct-Level Election Results. https://doi.org/10.7910/DVN/KTXHEW

Voting and Election Science Team. (2019b). 2017 Precinct-Level Election Results. https://doi.org/10.7910/DVN/VNJAB1

Voting and Election Science Team. (2019c). 2019 Precinct Shapefiles.
https://doi.org/10.7910/DVN/A0VJ3B