

CS 760 FINAL PROJECT

APPLICATIONS OF MACHINE LEARNING FOR ALGORITHMIC REDISTRICTING

Daniel Szabo
9074762189

1 Introduction

In this paper I'll analyze some methods for quantifying gerrymandering using machine learning. Theoretically I will assume we are given a truly uniform random representative sample of all possible districting plans of some geographic entity, and then discuss what can be done to extract information relevant to the study of gerrymandering from this sample. This includes using unsupervised learning to see the structure and significance of certain gerrymandering metrics from the political science literature, as well as using supervised learning to identify gerrymandering with CNNs.

2 Background and Data

Gerrymandering is the partisan process of drawing election maps in particular ways to gain a political advantage, and results in the disenfranchisement voters. It is a central issue in political science that has no clear solution, and remains a contentious issue in courts all over the United States. One of the key difficulties is quantifying gerrymandering in a way that can hold up in court. The most popular solution to this is currently to draw a *representative* sample of all possible elections maps (districting plans) and use those to classify a given plan as gerrymandered or not. Usually this is done using one of the many metrics available in the political science literature, and conducting a statistical test for the significance of whether the given plan comes from the sampled distribution.

Algorithmic redistricting refers to the methods for sampling districting plans. This has mostly been done with Markov Chain Monte Carlo algorithms [2], but also using topological data science [3], as well as methods from statistical mechanics [1]. One of the issues with all of these methods is that they have no theoretical justification to being a representative sample of the space of all possible legislative maps, which is huge. My work with Jin-Yi Cai, Ashwin Maran, Kenneth Mayer and Song Gao among others has resulted in a new algorithm that can truly uniformly sample from this space by counting exactly the total number of possible partitions, and using that to sample in subexponential time (in the number of precincts).

More formally, we are given a planar adjacency graph G where vertices are precincts (the building blocks of districts) and can sample a non-self-intersecting path p on the edges of G . Because each edge corresponds to a boundary between two districts, p is also a path on the edges of H , the dual of G , where precincts are faces—this corresponds to an actual map we are used to seeing.

A significant political caveat to this entire field is that partisan gerrymandering is not currently illegal. In 2019 the supreme court ruled in *Rucho v. Common Cause* that only the supreme court can regulate claims for gerrymandering, and other cases showed that they can only really decide based on racial disenfranchisement. In other words, even if we were to show that some map is drawn in a completely partisan way, if it does not intentionally disenfranchise a target population, it is not illegal. Nonetheless the mathematics of this area remains interesting, as the distinction remains a very gray area.

2.1 Data

Unfortunately the practical case for sampling is quite different from the theory. The data comes from legislative districts 26 and 27 near Sheboygan, WI from the 2012 election, and only looks at approximately 200 precincts. The algorithm to sample districting plans currently is only implemented for partitioning into 2 districts. Thus I am trying to measure gerrymandering using a relatively small republican majority area that can only be split into two

districts, a case for which many of the most popular metrics may not even have any significance. The algorithm to sample districting plans produced 3446 sample plans out of the enormous space of 6080174112134899589734 possible plans.

Another issue is with the nature of the underlying geographic data. Precincts are drawn by humans and modified slightly over time for a plethora of reasons, and often have inconsistencies. Because of this the requirement for the population of each district to be similar had to be waived, as the real districting plan did not adhere to those requirements either. This means we are sampling from a much larger space that is not really representative of any districting plans, but the methods here can still be applied to better data.

For supervised learning, I used 33 images from Dr. Mayer's 2018 Expert Report [4] on specific Assembly districts in Wisconsin to train the CNN. There he describes how these plans are gerrymandered, and gives alternate districting plans as well. The images came from a word document, and had some errors or inconsistencies that I did not know how to heal.

2.2 Metrics

The metrics I used were mostly those available from the Metric Geometry and Gerrymandering Group's (MGGG) GerryChain algorithm [2]. They include the efficiency gap, population, vote percentage, partisan Gini score, and Polsby-Popper compactness.

Efficiency Gap: The efficiency gap is a measure of each party's wasted votes. It is calculated as the difference in the two parties' total wasted votes, divided by the total number of votes cast.

Vote Percentage: The vote percentage is only applicable for two districts, and simply measures the difference between the largest percentage of votes won in a district and the average percentage of the two districts.

Partisan Gini Score: The partisan Gini score is defined as the area between the seats-votes curve and its reflection about $f(x) = x$. The seats-votes curve plots the percentage of seats a party controls as a function of its share of the overall vote summed across all districts.

Population: Due to the difficulties mentioned in Section 2.1, it is also relevant to collect the population of each district.

Polsby-Popper compactness: The only compactness measure I collect is the Polsby-Popper score PP , which is the following ratio of the area A of a district and its perimeter P :

$$PP = \frac{4\pi A}{P^2}.$$

3 Unsupervised Learning

The space of metrics evaluated from the sampled redistricting plans is inherently unlabeled. Using this data, we want to say something about the single real districting plan. Previous methods simply look at where the true plan lies in this distribution, however after visualizing a 2-dimensional embedding generated by PCA, it seemed the data was naturally clustered, so I tried using k -means to get a sample space that better represents the behavior of these metrics within a cluster.

3.1 Hypothesis

The statistical behavior within a k -means cluster is a good predictor of whether a districting plan is gerrymandered.

3.2 Methods

First, to choose the hyperparameter k , I used principal component analysis to visualize the metrics. The reasoning is that the different metrics measure similar aspects of the data, so it likely can be well represented by some lower dimensional space. To implement this, I used sklearn's PCA algorithm [5].

After choosing k , we need to implement k -means, which I also did using sklearn. The computational benefits of the lower dimensional embeddings are not necessary as we only collected 8 metrics, so we can use the original data after normalization.

Once we have a clustering, we can look at the behavior of our sample within its cluster. Although this can be done using multivariate analysis, a simpler approach is to see where it lies in the distribution of signed distances from

the cluster center. The sign of the euclidean distance to the center can be determined by which side it is of any hyperplane crossing the center, as it's only needed for the distribution of distances to be approximately normal (symmetric).

3.3 Results

The singular values suggested that the districting plans could be well represented in two dimensions. These dimensions may correspond to partisan metrics versus compactness metrics, as those are significantly different while partisan metrics tend to be correlated. The singular values were approximately

$$[2.0107 \times 10^6, 9.0811 \times 10^5, 6.5658, 8.6638 \times 10^{-1}, 6.8599 \times 10^{-1}, 3.5867 \times 10^{-1}, 1.2918 \times 10^{-1}].$$

The 2-dimensional visualization in the largest two eigenbases is shown in Figure 1. This suggests a choice of $k = 5$ would be appropriate.

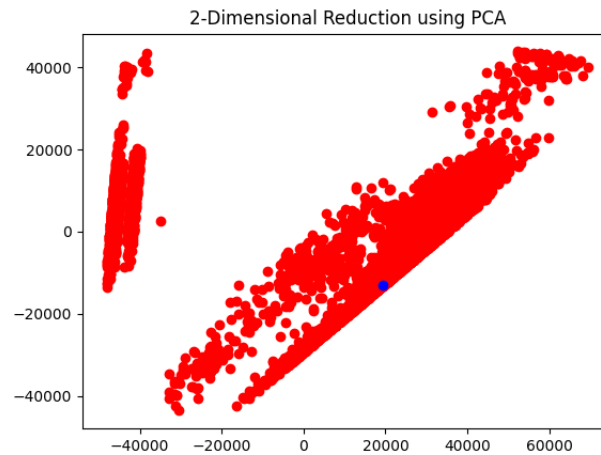


Figure 1: The lower dimensional representation of the randomly sampled metric data. The red dots represent sampled plans, and the blue dot the existing districting plan.

The distribution of the cluster along with the current districting plan can be seen in Figure 2.

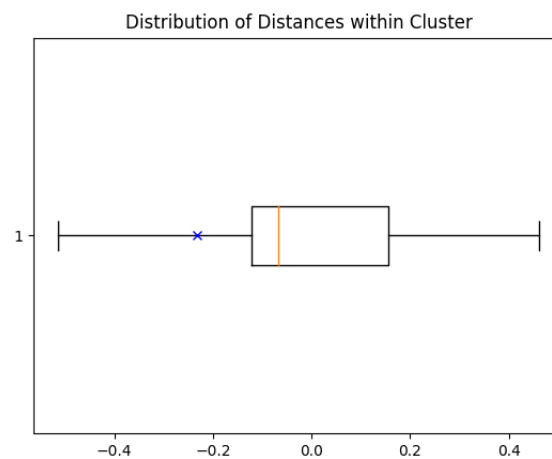


Figure 2: The location of the existing districting plan (labeled by a blue x) in the distribution of signed Euclidean distances.

3.4 Analysis

We do not have a realistic enough example to accept or refute our hypothesis.

The reason the data is clustered as seen in Figure 1 is not inherent to the problem, but to the particular samples. They were generated by going over all possible “opposite” starting and ending points for the district border, and looking at the statistics of the ordered samples it is clear that the clusters are due to the location of these starting and ending points. Nonetheless it is known that larger samples the space of districting plans is in some way clustered, as it is one of the shortcomings of algorithms that can only optimize locally (exploit rather than explore).

The actual example failed to correctly classify the real districting plan that was gerrymandered, however the methodology still has potential in larger instances.

4 Supervised Learning

For my supervised learning approach, I trained a convolutional neural network on images of districting plans.

4.1 Hypothesis

A CNN trained on labeled images of districting plans is a good predictor of gerrymandering.

4.2 Methods

Before training, we need to transform the images to a regular size. I chose a smaller size of $3 \times 100 \times 100$, as the most important feature we had to preserve in these images were the boundaries. I would have then liked to somehow increase the contrast of these boundaries so the network treats them with more weight, however I could not find a way to do that with the data already having some faults. The transformed data can be seen in Figure 3.

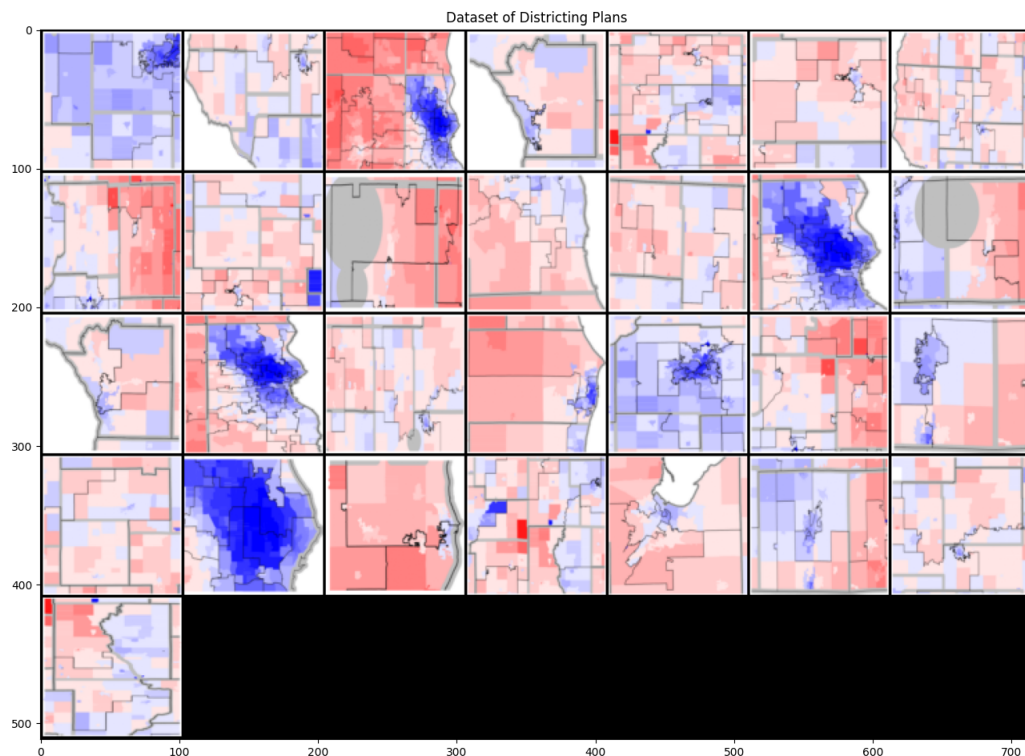


Figure 3: Images of current gerrymandered districts and alternative fair districting plans in an arbitrary ordering.

Then after sampling a training and validation set, the CNN has the following structure:

We first have two similar convolution blocks that increase the dimension to $64 \times 100 \times 100$ using small (3×3) kernels in an effort to highlight the sharp local features, the boundaries. ReLU activation is used between layers. Then I tried using a convolution layer that preserves the dimension however has a large kernel to account for larger scale relationships. Between convolution blocks I used max pooling layers. Finally, I reduce the dimension to the output dimension of 2 with 3 fully connected linear layers.

Training was done using SGD and cross-entropy loss over 50 epochs with a learning rate of 0.01. The batch size had to be low due to the small number of samples (33), only being 4. Cross-validation showed that changing these hyperparameters did not significantly improve results.

4.3 Results

Due to the small number of samples, the validation set could not be too large and still be independent from the training set, so validation accuracy was not the best metric. Instead I used the loss function of the CNN, which is plotted in Figure 4. Despite the relatively small learning rate and high number of epochs, the loss showed no sign of converging to anything with good accuracy.

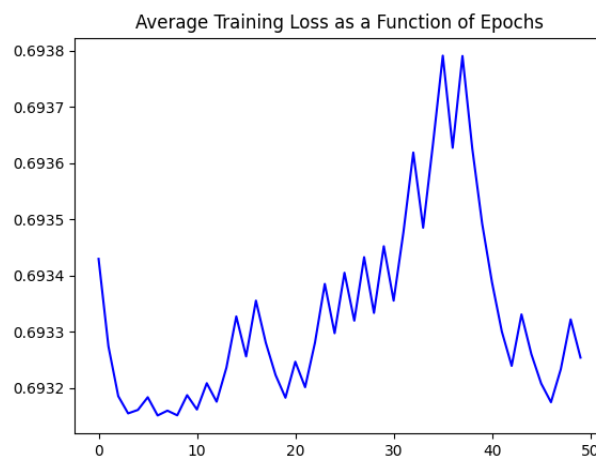


Figure 4: The average training loss of the CNN per epoch.

4.4 Analysis

Although there are other confounding factors, the CNN could not effectively detect gerrymandering on this dataset. Not only was the dataset too small, but the CNN had no way of identifying which feature of these maps was important. This could have been solved by pairing the data points in some way (as in, say, a paired t -test in statistics because the original data was paired), or using a more sophisticated model that knows to focus on the boundary lines rather than the underlying voter distribution.

There is also a deeper issue of what is defined as gerrymandering. It is not always the case that straighter boundaries that seem uncorrelated to the underlying voter distribution are not gerrymandered, despite this being the usual adhoc definition. Geographic data can inherently have strange nonlinear boundaries, and because of this gerrymandering is better measured by the outcomes of elections being far from the actual voters' percentages. Nonetheless neural networks can learn surprising things, and this experiment alone is not enough to refute the potential of artificial intelligence in detecting gerrymandering. With larger quantities of more thorough and carefully processed data, some deeper networks could be able to learn the features necessary to identify gerrymandering.

References

- [1] Chung-I Chou and S.P. Li. Taming the gerrymander—statistical physics approach to political districting problem. *Physica A: Statistical Mechanics and its Applications*, 369(2):799–808, 2006.
- [2] Daryl DeFord, Moon Duchin, and Justin Solomon. Recombination: A family of markov chains for redistricting. *Harvard Data Science Review*, 3 2021. <https://hdsr.mitpress.mit.edu/pub/1ds8ptxu>.

- [3] Moon Duchin, Tom Needham, and Thomas Weighill. The (homological) persistence of gerrymandering. *Foundations of Data Science*, 0:–, 2021.
- [4] Kenneth R. Mayer. William Whitford v. Beverly Gill 3:15-CV-00421-jdp (10/26/2018) Expert Report.
- [5] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.