The Racial Voting Power Gap: Analyzing Racial Gerrymandering Through Solving Ecological Inference with The Discrete Voter Model

a thesis presented
by
Hakeem Olakunle Isa Angulu
to
The Department of Computer Science and Statistics

in partial fulfillment of the requirements
for the degree of
Bachelor of Arts
in the subject of
Computer Science and Statistics

Harvard University
Cambridge, Massachusetts
May 2020

Thesis advisor: Professor Moon Duchin — Hakeem Olakunle Isa Angulu

The Racial Voting Power Gap: Analyzing Racial Gerrymandering Through Solving Ecological Inference with The Discrete Voter Model

Abstract

Section 2 of the Voting Rights Act of 1965 (VRA)[vra] prohibits voting practices or procedures that discriminate based on race, color, or membership in a language minority group, and is often cited by plaintiffs seeking to challenge racially-gerrymandered districts in court.

In 1986, with Thornburg v. Gingles[tho], the Supreme Court held that in order for a plaintiff to prevail on a section 2 claim, they must show that:

1. the racial or language minority group is sufficiently numerous and compact to form a majority in a single-member district
2. that group is politically cohesive
3. and the majority votes sufficiently as a bloc to enable it to defeat the minority's preferred candidate

All three conditions are notoriously hard to show, given the lack of data on how people vote by race.

In the 1990s and early 2000s, Professor Gary King's ecological inference method tackled the second condition: racially polarized voting, or racial political cohesion. His technique became the standard technique for analyzing racial polarization in elections by inferring individual behavior from group-level data. However, for more than 2 racial groups or candidates, that method hits computational bottlenecks.

A new method of solving the ecological inference problem, using a mixture of contemporary statistical computing techniques, is demonstrated here. It can be used for multiple racial groups and candidates, and is shown to work well on randomly-generated mock election data.

Contents

0 Introduction 1

1 Background and Literature Review 6
1.1 Racial Gerrymandering and the Courts 6
1.2 Ecological Inference 11
1.3 Discretization 15

2 Methods 22
2.1 phc 23

3 Results 30

4 Discussion 36

5 Conclusion 39

Appendix A Appendix 43

References 51

This is the dedication.

Acknowledgments

I would like to express my deep gratitude to Professor Latanya Sweeney, Jinyan Zang, and the students of the Fall 2019 iteration of GOV93B at Harvard University. They provided me with patient guidance, enthusiastic encouragement and useful critiques of the scope and design of this research work. In addition, I would like to thank Professor Moon Duchin of Tufts University for her guidance of this research and my overall thesis work.

0

Introduction

The Voting Rights Act of 1965 (VRA)[vra] was the result of decades of activism and advocacy around unencumbered suffrage in the United States. Specifically, this act was meant to "enforce the fifteenth amendment to the constitution," which was ratified 95 years prior, and came after African Americans in the South protested the tremendous obstacles to voting they encountered, including poll taxes, literacy tests, harassment, intimidation, physical violence, and other systemic and infrastructural facets of the practical application of the right to vote. The United States Department of Justice has cited it as "the single most effective piece of civil rights legislation ever passed by Congress."[7]

Section 2 of that act specifically prohibits any jurisdiction in the nation from implementing any "voting qualification or prerequisite to voting or standard, practice, or procedure...which results in a denial or abridgment of the right of any citizen of the United States to vote on account of race or color." Today an often challenged voting procedure is the drawing of political districts themselves, on the grounds that a political tool, called gerrymandering, dilutes the effectiveness of the vote.

Gerrymandering is the process of manipulating the boundaries of electoral districts to establish an unfair political advantage for some group of people. This process occurs at all levels of government in the United States of America, and regularly occludes the representative characteristic of the democracy.

Gerrymandering is the abusive version of the more general *redistricting*, which is simply the process by which elected and appointed officials redraw electoral districts. In order to guarantee fair representation, electoral districts ought to be drawn based on data about the population and its distribution in space. There is a consensus that "fairness" requires population balance, hence redistricting is done at least every ten years in the United States, immediately after a census. When redistricting is corrupted, it can become a powerful political tool for diluting or magnifying a demographic group's voting power in elections. When that demographic group is a race, gerrymandering becomes addressable by the VRA.

Quantifying the effects of voting standards, like the electoral map, on racial groups is notoriously difficult. Both Congress and the Supreme Court have attempted to set aside rules to prove that voting standards that sound neutral have racially disparate effects.

In *Thornburg v. Gingles* (1986)[tho], the Court established that plaintiffs needed to show that:

1. the racial or language minority group is sufficiently numerous and compact to form a majority in a single-member district
2. that group is politically cohesive
3. and the majority votes sufficiently as a bloc to enable it to defeat the minority's preferred candidate

This amounts to two kinds of evidence, now known colloquially as the Gingles Test, needed to successfully prove a violation of the Voting Rights Act:

1. the geographic conditions to draw an effective district
2. racial polarization that blocks the will of minority voters

The racially polarized voting test requires showing that a marginalized racial group is *politically cohesive*, and that the "majority votes sufficiently as a bloc to enable it to usually defeat the minority's preferred candidate." Both of these are uniquely hard to show given the secret ballot; that is, all voter choices in an election are anonymous. Thus, it is impossible to access the ground truth about how racial groups in the majority and minority vote. This information is necessary to challenging an electoral map that one believes is gerrymandered.

To fill this gap, statistical methods, most notably King's Ecological Inference (EI)[17], exist to infer that information from publicly available data.

EI was pioneered in the 1990s and 2000s by Harvard professor Gary King and Columbia professor Andrew Gelman, and it became the standard technique for attempting to analyze racial polarization in elections in service of mounting VRA litigation. As King himself billed EI, it is "the process of learning about discrete individual-level behavior by analyzing data on groups"[17].

From only the aggregate vote counts and aggregate racial distribution, EI produces inferred candidate preferences for each racial group. This is important because the courts that litigate VRA cases typically do not accept data from non-public sources. Often, this means that the only data that plaintiffs have to bring forward are Census data and election results.

One method of solving the Ecological Inference problem is King's own Ecological Inference method. The simplest versions of this method use Normal distributions, and the most complex hinge on Binomial-Beta hierarchical models for Bayesian inference. As a point of departure, this paper explores a new way of providing data to the hierarchical model pipeline: by discretizing the whole parameter space of possible vote outcomes. With this discretized space, one can replace the limited Beta class of distributions with a much more flexible description of voter behavior. This new method for solving the Ecological Inference problem, the *discrete voter model* (DVM), was not computationally tractable at the full scale of a U.S. state in the 1990s and early 2000s, but is now, due to advances in algorithm design and computing power.

While gerrymandering is difficult to prove, it is clear that it is one of the most pressing contemporary voting rights problems. The nation's electorate, and thus the integrity of the democracy, has to constantly contend with confusing and dilutive maps that are drawn by people, parties, and agencies with specific agendas. It is important that the methods for analyzing electoral maps are constantly updated and improved with new knowledge and computing power. Additionally, it is critical that organizers, potential plaintiffs, and members of various affected communities all have the ability to quickly and intuitively synthesize the work of mathematicians, statisticians, geographers, political scientists, and other key stakeholders towards resolving this longstanding and unavoidably complex problem.

tbd

tbd

1

Background and Literature Review

1.1 Racial Gerrymandering and the Courts

Gerrymandering is the process of manipulating the boundaries of electoral districts to establish an unfair political advantage for some group of people. Racial gerrymandering has been affirmed by the court as a direct violation of constitutional rights, with the Voting Rights Act (VRA) used to further bolster those rights. A racial gerrymander constitutes a "systemic and infrastructural facet of the practical application of the right to vote,"[14] and plaintiffs often invoke the VRA to force officials to redistrict.

Racially polarized voting, defined as different racial groups voting for different candidates in an election, has been central to redistricting law since the 1982 amendments to section 2 of the Voting Rights Act.[14] Two kinds of evidence, now known colloquially as the Gingles Test, after the 1986 *Thornburg v. Gingles*[tho] case, further codified that in order to prove a violation of the Voting Rights Act one needed to show that:

1. the racial or language minority group is sufficiently numerous and compact to form a majority in a single-member district
2. that group is politically cohesive
3. and the majority votes sufficiently as a bloc to enable it to defeat the minority's preferred candidate

The final two requirements require showing racially polarized voting. "Political cohesion" is often interpreted as racial groups voting for one candidate, and that racial voting information is necessary to determine if racial minority group's candidates are often defeated by the majority racial group's candidates.

This means, practically, that in order to win a lawsuit plaintiffs must prove that voters in districts vote along racial lines. While not every instance of racially polarized voting occurs because of a gerrymandered electoral map, proof of racially polarized voting is often the first step in successfully litigating a racial gerrymandering case as a violation of the VRA.

1.1.1 Estimating Racially Polarized Voting

Although racially polarized voting is critically important to racial gerrymandering litigation, it is quite difficult to prove. The *secret ballot*, an essential part of the United States' democracy, ensures that one's vote is anonymous. Hence, any data collection around voting patterns that separates along any axis of identity, is impossible, and all information about those voting patterns must be inferred.

The *Gingles* decision noted that two techniques were "standard in the literature for the analysis of racially polarized voting"[tho]: bivariate ecological regression and homogeneous precincts.

Dozens of court cases, from the 1980s to now, have relied on either of the above methods for litigating racial gerrymandering cases[14]. However, some appellate courts and judges have been hostile to them. In *Lewis v. Alamance County (1996)*[Lew], the court said that the critical assumption of regression "runs counter to common sense." In *Holder v. Hall (1994)*[Hol], said that "bivariate regression analysis...does not directly control for...factors [other than race]."

These criticisms were often ignored by peer and other courts because they did not posit new ways of deciding cases without using regression. Chief Justice Roberts chimed into the debate as well, saying "At trials, assumptions and assertions give way to facts. In voting rights cases, that is typically done through regression analyses of past voting records."[LUL]

However, the problem has gotten increasingly more complex. Jim Greiner, Professor of Law at Harvard Law School, stated in his paper *Ecological Inference in Voting Rights Act Disputes: Where Are We Now, and Where Do We Want To Be?*[14]

> According to some scholars, the patterns themselves have shifted in that a certain percentage of white citizens in some jurisdictions [like Chicago] have become willing to vote for minority-preferred candidates. This shift suggests that the days in which "documenting racially polarized voting is generally...just beating the obvious" may be gone. Racial bloc voting, often a hotly contested issue in an "ordinary" redistricting dispute, may now become even more so, and as a result, using the best methodology is more critical.[14]

Greiner's statement is a good introduction to the future of redistricting. He notes not only that the electorate's voting patterns are becoming more complex, but also that to ensure that the courts are equipped with the correct tools to tackle the complexity, the methodologies supplied by and to expert witnesses must be continually improved.

The ecological regression and homogeneous precincts methods had several disadvantages. Ecological regression regularly produces impossible estimates of support, like that -400% of Hispanic voters in a district supported a candidate. The analysis of homogeneous precincts depends on the assumption that white people living in an all-white precinct vote similarly to white people living in a more racially-diverse precinct, which has been shown empirically, and can be thought intuitively, to be false.[14] In addition, neither method generalizes well past 2 racial groups and 2 candidates in an election.

The dominant racial bifurcation in both legal and lay discourse is that between Black and white citizens. Thus, much of the litigation of racial gerrymandering and VRA cases, along with the methods used, have focused on these two groups. However, many contemporary cases explore the more complex racial dynamics that exist in our more, and increasingly, diverse electorate.

Take, for example, a recent case that used ecological regression: *League of United Latin American Citizens v. Perry, 126 S. Ct. 2594 (2006)* (LULAC). In this case, Latinx plaintiffs, a mixture of private citizens and representatives of the Mexican American Legal Defense and Educational Fund, argued that Texas' congressional districts were racially gerrymandered to negatively affect Latinx citizens in Texas. With the help of a dubiously-adjusted ecological regression model (to be used for a larger combination of racial groups), the court decided to throw out District 23 within a new plan proposed in 2003, ordering a remedial redistricting.[LUL] The court also stated that it need not rule that District 25 was a racial gerrymander, given that the remedial changes to District 23 would necessarily affect District 25.

Although not verbally verified by the court, District 25 is widely believed to have been drawn to pack Latinx voters into a district and dilute their voting power, so the redrawing of it and District 23 was a win for that section of the electorate. The change had incredibly practical, and almost immediate, effects. In the year of the remedial redistricting, the affected districts held new primary elections. Henry Bonilla, the Republican representative for the 23rd congressional district, was defeated by Democrat Ciro Rodriguez, a favorite of the Latinx electorate.

Figure 1.1 shows the creation of District 25 after the 2003 redistricting in question. Its shape is reminiscent of popular gerrymanders.

In that decision, the judges referred to a previous one from 1993, where the court stated:

> In countless areas of the law weighty legal conclusions frequently rest on methodologies that would make scientists blush. The use of such blunt instruments in examining complex phenomena and corresponding reliance on inference owes not so much to a lack of technical sophistication among judges, although this is often true, but to an awareness that greater certitude frequently may be purchased only at the expense of other values.[cle]

Here, the court explicitly acknowledges the disadvantages of the methods used to infer racially po-

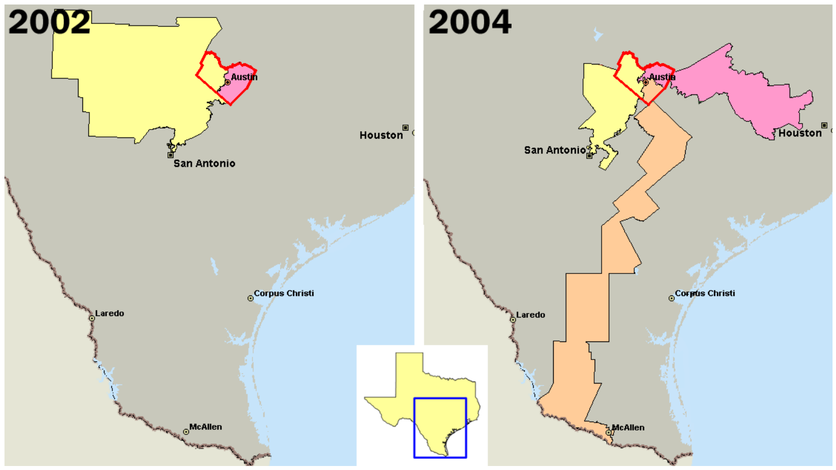


Figure 1.1: The Creation of Texas' 25th Congressional District (orange) in 2003 [10]

larized voting. These disadvantages, the criticisms lain on ecological regression and homogeneous precincts by some courts, and improvements in theory inspired a burst of energy and innovation in the academy in the late 1990s and early 2000s. At the center of this burst is King's EI.

1.2 Ecological Inference

Ecological inference (EI) is the process of using aggregate (or, ecological) data to infer discrete individual-level relationships of interests when individual-level data are not available.[17] This is distinct from the *ecological regression* discussed above, which is the statistical method of running regressions on aggregates and interpreting those regressions as predictive relations on the level of individual units.[13]

In a world with two categories, groups $\{1, 2, \ldots\}$ and groups $\{A, B, \ldots\}$, both methods seek to answer the question "what percentage of members in group 1 are also in group A?" This is precisely the answer that plaintiffs need in racial gerrymandering cases, where groups $\{1, 2, \ldots\}$ are races and

Table 1.1: A 3×3 Table of Voting By Race, using Counts of Voting Age Population

	Cand. 1	Cand. 2	$\sum$
Race 1	-	-	Race 1 Pop.
Race 2	-	-	Race 2 Pop.
$\sum$	Cand. 1 Votes	Cand. 2 Votes	Precinct Pop.

groups $\{A, B, \ldots\}$ are candidates.

Greiner provides an alternate description of EI that fits well within mathematical models.[14] EI can also be understood as the attempt to predict internal cell values of a set of contingency tables, like Table 1.1, when only the column and row totals (the margins of the table) are observed.

Each precinct has its own table, with an election's vote counts as the column totals and the voting age population, or demographic split of the district, often from the Census, as the row totals. The internal cells, how many people of each racial group that voted for each candidate, are protected by the secret ballot. Those internal cells are exactly the evidence needed in racial gerrymandering litigation, and their values have been inferred by ecological regression and homogeneous precinct analysis in the past.

Theoretically, these tables can be expanded to be as large as necessary: $R \times C$, where R corresponds to the number of rows (demographic groups or races) and C corresponds to the number of columns (candidates). This would be ideal given that the diversification of the electorate has created more politically powerful demographic groups, and that there are frequently more than 2 candidates in elections, especially in local cases (where many infringements of the VRA are suspected to occur).

Professor Gary King developed and proposed his own method to solve this problem in the late

1990s. It has been used in a few court cases, mainly for the advantage that unlike ecological regression, this method will never produce impossible estimates. It is also understood to be more intuitive than regression, addressing the court's seldom complaint.

1.2.1 King's Ecological Inference

Professor King introduced his method to solve the ecological inference problem in his 1997 book *A Solution to the Ecological Inference Problem: Reconstructing Individual Behavior from Aggregate Data*[15]. His method, which he expands upon with several papers from 1997 to the late 2000s, uses Bayesian inference and bounds methods to infer the values of the empty cells in Table 1.1.

Using Table 1.1 as a reference, Table 1.2 describes the quantities that King used to describe this problem[17].

Table 1.2: Quantities in King's EI

Quantity	Description
Observed	
X_i	fraction of population i in race 1
T_i	fraction of population i who voted for candidate 1.
N_i	size of population i
T_i'	number of people in population i in race 1
Unobserved	
$b1_i$	fraction of members of race 1 who voted for candidate A
$b2_i$	fraction of members of race 2 who voted for candidate A

King's method then uses the observed quantities, and a few parameters, to find the unobserved quantities and answer the ecological inference question. The parameters used are described in Table

1.3.

Table 1.3: Parameters in King's EI

Parameter	Description
c_1, d_1	hyperparameters for the Beta distribution for b_1
c_2, d_2	hyperparameters for the Beta distribution for b_2.
ϑ	the expected fraction of the population of i registered to vote i
λ	the fixed hyperparameter for the distributions of c_1, c_2, d_1, d_2

The model is a hierarchical Bayesian inference model that proceeds as follows:

1. $c_1, c_2, d_1, d_2 \sim \text{Exponential}(\lambda)$
2. $b1_i | c_1, d_1 \sim \text{Beta}(c_1, d_1)$
3. $b2_i | c_2, d_2 \sim \text{Beta}(c_2, d_2)$
4. $T_i' | b1_i, b2_i, X_i \sim \text{Binomial}(N_i, \vartheta_i)$

The Python implementation of King's EI is given in Listing A.1. Using the observed quantities, King's method produces a model for the unobserved quantities of interest, which can then be used in litigation as described above.

However, in King's original paper, the bulk of his examples are for 2×2 tables, and proposed extensions of that method beyond that bound have been dubious.

In King's second iteration of his method and accompanying software, *EI2*, King implements a "Multiple Imputation Approach" to try to handle 2×3 tables. In essence, this approach collapses the 2×3 table into multiple 2×2 subtables, then applies the original method to each of those subtables.[16] Statisticians and scholars, like Jim Greiner, have noted that this method as applied is statistically invalid, as it does not, and cannot, fully adhere to combinatorial rules that the pioneer of multiple imputation, Professor Donald Rubin of Harvard University, notes in his writings.[21]

The problems increase if the size of the table increases to 3×3 or higher. Professor Karen Farree of the University of California, San Diego investigated King's EI applied to tables of that size, and found that the method produced biased estimates and relied on inappropriate modeling assumptions.[12]

Hence, although King's method has proved quite useful for some cases of ecological inference, with an increasingly diverse electorate, and complex multi-candidate elections, a method that generalizes well to an $R \times C$ table, using the same data available to previous methods, is necessary.

1.3 Discretization

King's method assumes the quantities of interest, the demographic voting patterns of a district, can be modeled completely with Beta distributions. The hierarchical model allows some freedom to those distributions, generating their parameters from Exponential distributions.

Beta distributions are known to be approximately bimodal if both shape parameters are below 1. However, Beta distributions can never be *fully* bimodal, which is the expectation for racially polarized voting: that two different groups vote quite differently. Figure 1.2 is an example of a very, but not fully, bimodal Beta distribution.

Perhaps more importantly, these distributions can never be multimodal (with more than 2 modes), which is the case that is necessary for multiple racial groups and candidates. This is where discretization becomes quite useful. Figure 1.3 is an example of a true multimodal distribution over two variables.

A distribution like this could be created easily with Neural Networks, but that architecture requires

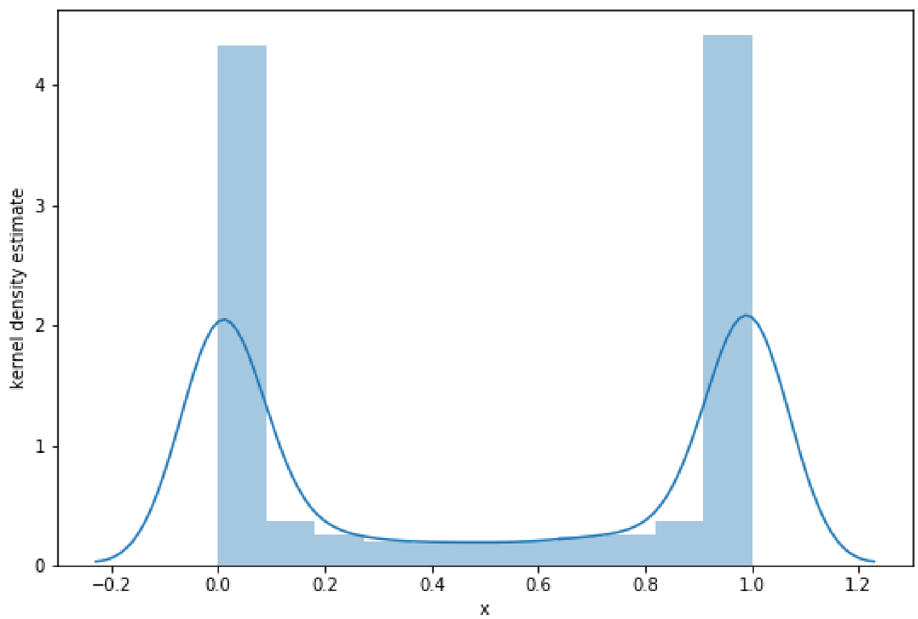


Figure 1.2: Kernel Density Plot of the Beta(0.1, 0.1) Distribution

tens of thousands or millions of data points to be reliable, where a "data point" would probably be an electoral result. The problem of ecological inference, especially in local elections, is simply too small for those methods.

However, by specifying very little about the space (even less than something like a Beta distribution), then allowing the space to take the shape of the available data, more complex, interesting, and possibly accurate distributions can be found. This is called the *Discrete Voter Model (DVM).*

For a given candidate, each state of this space is a hypercube (the n-dimensional abstraction of a 2-D square/grid or a 3-D cube) of values, where each dimension corresponds to a demographic group of interest. Each cell of the hypercube has a probability, where the sum of the probabilities of all cells

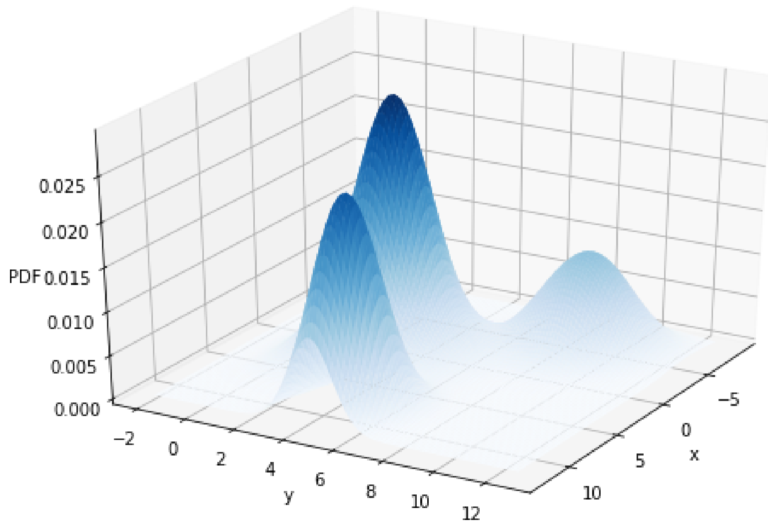


Figure 1.3: A Bivariate Multimodal Distribution

is 1. Each cell also uses its position (the values on each axis) in the full hypercube to represent the demographic voting patterns of a given precinct.

For example, in the 2-dimensional case, for a hypercube with granularity 10 (each axis goes from 0 to 10), let the cell at (2, 5) have an internal probability of 0.4. This corresponds to there being a 40% chance that this candidate's demographic voting pattern in this precinct is:

- 20% of group 1 votes for them
- 50% of group 2 votes for them

Hence, a single hypercube represents the distribution of possible demographic voting patterns in a precinct. Therefore, the distribution of hypercubes represents possibilities for all precincts in a map, and for an election.

The procedure by which the state space of hypercubes takes the shape of the data is Markov chain Monte Carlo.

1.3.1 Markov Chain Monte Carlo

Markov chain Monte Carlo (MCMC) is a class of algorithms that sample from probability distributions that are hard to sample from, typically called *target distributions*.[9] With many distributions, like the Normal, Beta, and Exponential mentioned above, obtaining a sample or specifying those distributions is simple and fast. There exist many packages for Python, R, and other programming languages that generate samples instantly with closed form methods.

The target distribution for the Discrete Voter Model, that over the space of hypercubes that could have possibly resulted in the outcome of an election, is not well-specified, and differs not only by election, but also by precinct. Attempting to apply a well-known distribution requires many approximations and assumptions. In lieu of a well-known distribution, the Discrete Voter Model uses discretization and MCMC to find the estimates of the quantities of interest.

All MCMC algorithms are iterative, and the samples converge to the target distribution over time, so the understanding of that target distribution grows with each iteration. All MCMC algorithms construct Markov chains, which means that each sample is generated from, and only depends on, the prior sample in the process, establishing the Markov relationship.[18] At each step, a candidate for the next sample generated according to some transition rules, determined by a *transition kernel*. That candidate is either accepted or rejected based off of some acceptance probability function. More precisely:

Let the chain be represented by $X_0, X_1, \dots, X_t$, where X_i is a step in the chain at iteration or time i.

Let the target distribution be $\pi(\cdot)$, which has an unnormalized density π_u.

Let there be some proposal distribution, $Q(\cdot)$, which generates new candidates at every step of the chain. Therefore, as a function $q(x, \cdot)$, it generates Y_{i+1} as a candidate for X_{i+1}, given X_i.

Let there be an acceptance function:

$$\alpha(x, y) = \min\left[1, \frac{\pi_u(y) \cdot q(y, x)}{\pi_u(x) \cdot q(x, y)}\right]$$

The candidate state Y_{i+1} is accepted with the probability $\alpha(X_i, Y_{i+1})$. If the state is accepted, $X_{i+1} \leftarrow Y_{i+1}$. If not, $X_{i+1} \leftarrow X_i$.

The differences in transition kernels, $Q(x, \cdot)$, and acceptance probability functions, $\alpha(x, y)$, are they key delineators of different types of MCMC algorithms. The Discrete Voter Model uses two types of transition kernels: *Random Walk Metropolis* and *Hamiltonian Monte Carlo*.

1.3.2 Random Walk Metropolis

One Markov chain Monte Carlo method which uses the Random Walk Metropolis transition kernel is the Metropolis-Hastings algorithm. Metropolis-Hastings is one of the simplest MCMC methods, but is still quite reliable and widely used for many applications.

This transition kernel requires that $q(x, y) = q(y - x)$. This is one way to ensure that the chain is reversible and detail balanced, necessary conditions for the chain to converge to the target distribution.[20]. Examples of such distributions are $Q(x, \cdot) = N(x, \sigma^2)$ and $Q(x, \cdot) = Uniform(x - 1, x + 1)$.

The Discrete Voter Mode's proposal randomly selects two cells in the probability hypercube, and adds a small value, ε, to one and subtracts it from the other. This is a reversible chain – one could get theoretically get back to the probability hypercube that was changed later in the chain – which is essential for convergence.[9]

1.3.3 Hamiltonian Monte Carlo

Hamiltonian Monte Carlo (HMC) differs from Random Walk Metropolis in that its proposal distribution adapts to the current state and its position in the parameter space.[8]

HMC uses the gradient of the target distribution's density, and adapts the proposal distribution toward that direction. This helps combat some of the inefficiencies of Random Walk Metropolis, like its tendency to explore parts of the state space where there is very little probability density. Hence, fewer samples have to be created to approximate the target distribution.

The "Hamiltonian" in "Hamiltonian Monte Carlo" refers to the inspiration of this kernel: Hamiltonian mechanics, a reformation of Newtonian mechanics that reconsiders the basis of time evolution of a system.[11] This kernel generates some *potential function/energy* from the target density, then uses the curvature of that function, some random seed, and the current state to determine the candidate for the next state.[19]

The primary benefit of HMC to the Discrete Voter Model is that it can theoretically produce a better sample of the target distribution with fewer steps in the algorithm. In addition, HMC has been shown to work well on state spaces with many parameters, like the potentially large probability hypercubes in this model.[19]

1.3.4 Robustness

The Markov chain Central Limit Theorem guarantees convergence and accurate sampling with the correct specifications.[9] Furthermore, for both the kernels above, chains are robust to their assumptions – they often converge to reliable samples in enough time, even when assumptions (like a perfectly ergodic chain) are broken.

The main disadvantages of using Markov chains for such complex state spaces are that it is difficult to assess convergence, and the methods themselves are computationally expensive. However, given the increasing availability of computing power, including the prevalence of GPU-based computing and the maturity of programming packages like `Stan`, `PyMC3`, and `TensorFlow Probability`, these methods can be more widely applied.

This is some random quote to start off the chapter.

Firstname lastname

2

Methods

The Discrete Voter Model is implemented in Python 3, and the accompanying code is found in the repository noted in Appendix A. A set of 6 modules supports the Discrete Voter Model, with each implementing an aspect of the statistical framework. Those modules are:

1. `phc`: creates probabilistic hypercubes
2. `expec_votes`: finds the expectation of votes for a candidate
3. `prob_votes`: finds the probability of an electoral outcome

4. `elect`: aggregates election and electorate data
5. `tools`: provides auxiliary functions for the model
6. `dvm`: executes the Markov chain on a state space of hypercubes

The modules work together to run the Discrete Voter Model on an `Election` object to generate a distribution of probabilistic hypercubes for the election. Each probabilistic hypercube encodes a probability distribution within a potential precinct of the district, and is used to extract the likely demographic voting probabilities.

These demographic voting probabilities are the goal of the ecological inference, and provide information to answer whether voting in an election is polarized by race.

2.1 PHC

The Discrete Voter Model introduces the probabilistic hypercube (PHC) to represent the distribution of possible demographic voting probabilities in a precinct.

A probabilistic hypercube is defined as a rank n tensor (matrix, multidimensional array, or holor) in $\mathbb{R}^n$ space with the following characteristics:

1. the component values of the cells sum to 1
2. each dimension corresponds to a demographic group in an electorate
3. each cell's position, or index, represents the demographic voting probabilities of a precinct
4. each cell's component value represents the inferred probability of that cell representing the true demographic voting probability of a precinct

One derives the demographic voting probabilities (DVP) by

The `make_grid` subroutine starts the discretization by initializing an R^n space as a probabilistic hypercube (or grid in the 2-dimensional case), where n is the number of demographic or racial groups in the electorate.

This hypercube represents the probability distribution of the voting patterns of different demographic groups in a precinct. Each cell in the hypercube corresponds to some collection of b_i, the unobserved quantity of interest from King's specification and the necessary quantity for determining racially polarized voting. The cell itself contains the probability of it being the collection that produced the observed outcomes.

The following four subroutines support and implement the MCMC section of the Discrete Voter Model.

`shift_weight`

The `shift_weight` subroutine implements the proposal step of the MCMC method. It reversibly shifts the weight of the hypercube to create another hypercube. *Reversibility* refers to an assumption of MCMC that guarantees convergence. A Markov chain is said to be reversible if there is a probability distribution, π, over the states such that:

$$\pi_i \Pr(X_{n+1} = j | X_n = i) = \pi_j \Pr(X_{n+1} = i | X_n = j)$$

for all iterations n and all states i and j. This is also called the *detailed balance* condition.

`shift_weight` allows for 5 types of shifting:

1. `uniform`: add a uniform random hypercube to the hypercube
2. `single_uniform`: add a single uniform random variable to each cell in the hypercube
3. `shuffle`: shuffle the hypercube randomly
4. `right`: shift weight in the hypercube to the right
5. `left`: shift weight in the hypercube to the left

Each of the above types of shifting is reversible, with the `uniform` as the default method. At each step of the chain, the algorithm uses this subroutine to propose a new hypercube.

`expec_votes`

This is the first of two subroutines that score a candidate hypercube in every step of the Markov chain.

`expec_votes` calculates the expectation of votes for a given candidate from some probabilistic hypercube.

The subroutine iterates over every cell in the hypercube and finds the expectation of votes that it dictates.

The notation is as follows:

- $i = 0, 1, \ldots, n$ represents cells in the hypercube
- p_i represents the probability of being in cell i
- $j = 0, 1, \ldots, r$ represents demographic groups
- q_j represents the probability of demographic group j to vote for the given candidate
- m_j represents the population of demographic group j in the district

The expectation of a hypercube is thus given by Equation 2.1.

$$\sum_{i=0}^{n} p_i \left(\sum_{j=0}^{r} q_j \cdot m_j \right) \tag{2.1}$$

This expectation is then compared to the observed outcome of votes, with the $L1$-norm (absolute value). The lower the norm, the better the hypercube, and the more likely it will be accepted in the Markov chain.

`prob_votes`

The is the second of two subroutines that score a candidate hypercube in every step of the Markov chain.

`prob_votes` calculates the probability that a given hypercube produced the observed election outcome.

With the same notation as `expec_votes`, and additionally:

- a_j represents the number of people in demographic group j that voted for a candidate
- d represents the observed number of votes cast for the given candidate

$$\sum_{\forall a_j \text{ s.t. } \sum_{\forall j \in [0,r]} a_j = d} \left(\prod_{j=0}^{r} q_j^{a_j} \cdot (1 - q_j)^{m_j - a_j} \right) \cdot \frac{d!}{\prod_{a_j} a_j!} \tag{2.2}$$

In essence, Equation 2.2 uses the Binomial distribution to calculate the probability of members of different demographic groups voting together in a way to produce the desired outcome. This is extensible to any number of demographic groups and can be repeated for any number of candidates.

A crucial step in this process is generating all possible partitions of the observed electoral outcome into the demographic groups. For example, if 10 people voted for a candidate, if there are three demographic groups, possible partitions include:

- 4 people from group 1, 3 people from group 2, and 3 people from group 3 voted for the candidate
- 2 people from group 1, 0 people from group 2, and 8 people from group 3 voted for the candidate
- 4 people from group 1, 4 people from group 2, and 2 people from group 3 voted for the candidate

The code for this partitioning process can be found in the appendix, as Listing A.2.

The probability of some candidate hypercube producing the electoral outcome given by that expression is then compared to the probability that the current hypercube produced the outcome, and the candidate is accepted if the probability is higher, or with some acceptance probability if lower.

`mcmc`

The final subroutine, `mcmc`, runs the Markov chain Monte Carlo method with the Metropolis-Hastings algorithm, employing either `expec_votes` or `prob_votes` to score candidates.

The algorithm is as follows:

1. initialize some hypercube with `make_grid`
2. iterate some number of times. at each iteration:
    (a) generate a candidate hypercube with `shift_weight`
    (b) score that candidate with `expec_votes` or `prob_votes`

(c) accept (assign it as the current hypercube and record the score) or reject that candidate based on the scores in the previous step

(d) if the score is better than all scores seen up until this point, save the score and the hypercube

3. output all hypercubes explored, and identify the best scoring one

The subroutine returns a collection of the best scoring hypercube, the highest score it received, and a list of all the hypercubes explored and their scores.

Evaluation

All of the methods for inferring racially polarized voting noted above, including ecological regression, homoegenous precincts, and King's EI, are necessarily imperfect – as is the Discrete Voter Model. The extent of that imperfection, however, can be evaluated and compared.

King's EI and the Discrete Voter Model are evaluated in this paper in two ways: accuracy and runtime. Both models are run on generated election data with different demographic distributions. The evaluation is only possible because with this generated election data, the ground truth, that is: the true demographic voting patterns, is known. That ground truth is fixed as an input in the `generate_random_election` subroutine, whose Python implementation can be found in Listing A.3.

The accuracy of the model is determined by how close the model's result is to the ground truth. The runtime of the model is how long it took the model to reach that result. Both are critically important to the model's use in practice.

DVM's runtime and accuracy are scored in comparison to King's EI for 2×2 examples, and demonstrations of its use on $R \times C$ examples are given in the Results section.

Nulla facilisi. In vel sem. Morbi id urna in diam dignissim feugiat. Proin molestie tortor eu velit. Aliquam erat volutpat. Nullam ultrices, diam tempus vulputate egestas, eros pede varius leo.

Quoteauthor Lastname

3

Results

Experiment Specification

2 × **2** Case

For the 2 demographic group case, 5 random elections were generated for each of the demographic distributions of the precicnt in Table 3.1 for precincts of size 100. Each of these distributions is labeled

$d_i \forall i \in [1, 3]$ (without loss of generality) for reference in Table 3.5.

The true demographic voting patterns for candidates a and b in Table 3.2. Each of these voting patterns is labeled $v_i \forall i \in [1, 2]$ (without loss of generality) for reference in Table 3.5.

Table 3.1: Demographic Distributions Tested in the 2×2 Case

Label	Group 1 (%)	Group 2 (%)
d_1	50	50
d_2	25	75
d_3	10	90

Table 3.2: Demographic Voting Patterns Tested in the 2×2 Case

Label	Group	Cand. a (%)	Cand. b (%)
v_1	1	50	50
	2	30	70
v_2	1	25	75
	2	20	80

3×2 Case

For the 3 demographic group cases, 5 random elections were generated for each of the demographic distributions of the precinct in Table 3.3 precincts of size 100. Each of these distributions is labeled $d_i \forall i \in [4, 6]$ (without loss of generality) for reference in Table 3.5.

The true demographic voting patterns for candidates a and b in Table 3.4. Each of these voting patterns is labeled $v_i \forall i \in [3, 4]$ (without loss of generality) for reference in Table 3.5.

Accuracy and Runtime

Accuracy is reported as the mean squared error (MSE) between the models' outputted demographic voting distribution and the true demographic voting distribution, averaged over all trials. The lower

Table 3.3: Demographic Distributions Tested in the 3×2 Case

Label	Group 1 (%)	Group 2 (%)	Group 3 (%)
d_4	50	50	0
d_5	25	25	50
d_6	33	33	34

Table 3.4: Demographic Voting Patterns Tested in the 3×2 Case

Label	Group	Cand. a (%)	Cand. b (%)
v_3	1	20	80
	2	80	20
	3	40	60
v_4	1	30	70
	2	40	60
	3	50	50

the MSE, the more accurate the model. The runtime is in seconds. The lower the runtime, the faster the model.

Table 3.5 summarizes the results of running DVM and King's EI on all of the experiments, for a total of 12 experiments replicated 5 times each, in Python 3.7.4 on a 2018 MacBook Pro with a 2.3 GHz Quad-Core Intel Core i5 processor, 16 GB of 2133 MHz LPDDR3 RAM, and a Intel Iris Plus Graphics 655 1536 MB graphics card. All numbers are given with 3 significant figures of precision. The DVM's MCMC was configured with 200 steps, and the hypercubes had a granularity of 10.

In addition to the presentation of the numeric evaluation, a presentation of select visualizations of the models produced by the Discrete Voter Model show its expressive power.

Figure 3.1 shows the distribution of probability over the 2D hypercube for a model in the 2×2 case. The figure is 3-dimensional, with the third dimension being the probability of being in one of the cells of the hypercube. This is multinomial and much more expressive of possible outcomes.

Table 3.5: Accuracy and Runtime of the Discrete Voter Model (DVM) and King's EI (KEI)

	DVM		KEI	
Label	MSE	Runtime (s)	MSE	Runtime (s)
$d_1 \times v_1$	0.0825	3.08	0.0253	13.9
$d_1 \times v_2$	0.0893	3.04	0.0589	14.0
$d_2 \times v_1$	0.153	4.05	0.0665	16.9
$d_2 \times v_2$	0.194	2.89	0.0604	11.5
$d_3 \times v_1$	0.107	2.74	0.0134	9.77
$d_3 \times v_2$	0.103	2.72	0.0620	9.55
$d_4 \times v_3$	0.0692	1500	-	-
$d_4 \times v_4$	0.0825	1470	-	-
$d_5 \times v_3$	0.236	1570	-	-
$d_5 \times v_4$	0.156	1530	-	-
$d_6 \times v_3$	0.149	1530	-	-
$d_6 \times v_4$	0.125	1500	-	-

Figure 3.2 shows the distribution of probability over the 2D hypercube for a model in the 3×2 case. Since there are 3 dimensions for demographic groups, Figure 3.2 presents the fourth dimension, probability, as directed arrows across a cross section of the dimensions. The colors of these arrows give the corresponding probability.

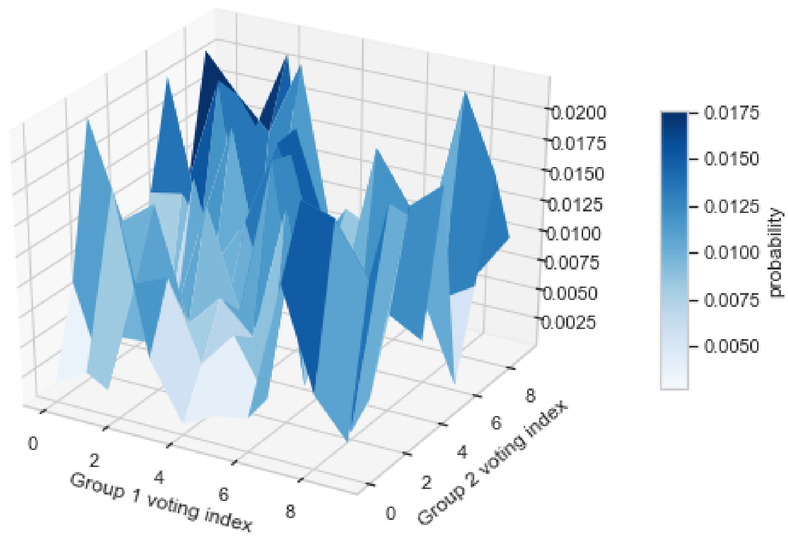


Figure 3.1: Distribution of Probability over the 2D Hypercube for a Model in the 2×2 Case

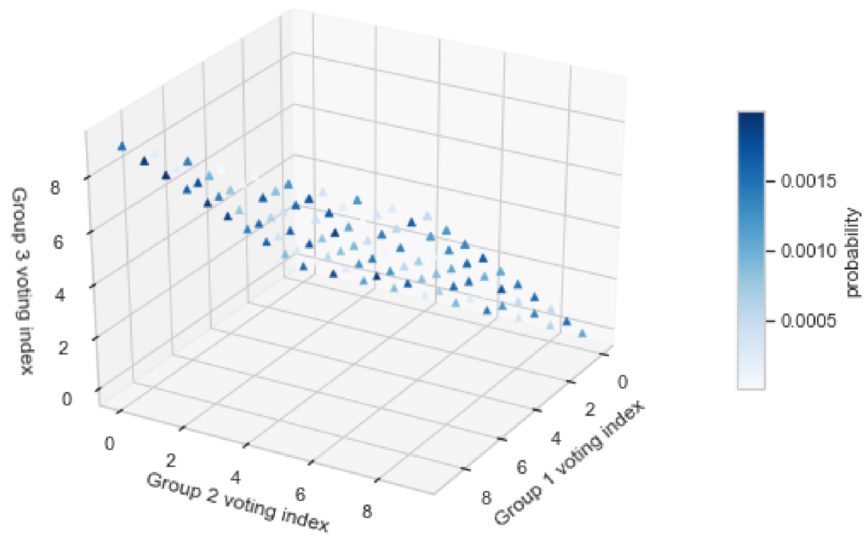


Figure 3.2: Distribution of Probability over the 3D Hypercube for a Model in the 3×2 Case

Nulla facilisi. In vel sem. Morbi id urna in diam dignissim feugiat. Proin molestie tortor eu velit. Aliquam erat volutpat. Nullam ultrices, diam tempus vulputate egestas, eros pede varius leo.

Quoteauthor Lastname

4

Discussion

The results presented in Table 3.5 show three primary trends:

1. the Discrete Voter Model performs nearly as well as, but a little worse than, King's Ecological Inference in the 2×2 case (two demographic groups and two candidates)
2. the Discrete Voter Model is remarkably faster than King's Ecological Inference in the 2×2 case (two demographic groups and two candidates)
3. unlike King's EI, the Discrete Voter Model can produce reliable results for the 3×2 case (three demographic groups and two candidates)

The Discrete Voter Model was limited to 200 steps, with a hypercube granularity of 10 (that is, each dimension of the cube went from 0 to 9). Further testing corroborates the theory that, generally, with more iterations, Markov chain Monte Carlo converges to the true distribution. Hence, allowing DVM to run longer may increase the accuracy of the model, at the expense of runtime.

With that said, DVM was able to use those 200 steps to perform comparably to King's EI in the 2×2 cases. This also corresponded to a lower runtime than King's EI in all cases, which is a testament to improvements in computing power and algorithms.

DVM was also able to do what King's EI cannot in any amount of time: run on the larger 3×2 tables. Higher dimensional tables are also possible with DVM, with no changes to the statistical theory or implementation.

If King's EI is taken to be a standard method, as it has been by some courts, DVM's performance on these generated elections show that it is a good contender. Not only does it perform comparably, but:

- it is inherently extensible and can be tuned. Generally, at the expense of time:
  - one can increase the number of iterations to get closer to the true distribution
  - one can increase the size of the hypercube to get more granular estimates for the racial voting patterns
- it can produce visualizations of more complex voting pattern distributions

The Discrete Voter Model leverages the power of discretization and Markov chain Monte Carlo to provide more expressive models within a more malleable and sound structure. Its flexibility has been shown on generated election data, and improvements continue to be made to its speed.

As Jim Greiner said: ”Racial bloc voting, often a hotly contested issue in an ”ordinary” redistricting dispute, may now become even more so, and as a result, using the best methodology is more critical.”[14] The Discrete Voter Model, as presented here and in future iterations, is a direct response to the need for more flexible and robust methods in the redistricting and voting rights space.

5

Conclusion

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Morbi commodo, ipsum sed pharetra gravida, orci magna rhoncus neque, id pulvinar odio lorem non turpis. Nullam sit amet enim. Suspendisse id velit vitae ligula volutpat condimentum. Aliquam erat volutpat. Sed quis velit. Nulla facilisi. Nulla libero. Vivamus pharetra posuere sapien. Nam consectetuer. Sed aliquam, nunc eget euismod ullamcorper, lectus nunc ullamcorper orci, fermentum bibendum enim nibh eget ipsum. Donec porttitor ligula eu dolor. Maecenas vitae nulla consequat libero cursus venenatis. Nam magna enim, accumsan eu, blandit sed, blandit a, eros.

Quisque facilisis erat a dui. Nam malesuada ornare dolor. Cras gravida, diam sit amet rhoncus ornare, erat elit consectetuer erat, id egestas pede nibh eget odio. Proin tincidunt, velit vel porta elementum, magna diam molestie sapien, non aliquet massa pede eu diam. Aliquam iaculis. Fusce et ipsum et nulla tristique facilisis. Donec eget sem sit amet ligula viverra gravida. Etiam vehicula urna vel turpis. Suspendisse sagittis ante a urna. Morbi a est quis orci consequat rutrum. Nullam egestas feugiat felis. Integer adipiscing semper ligula. Nunc molestie, nisl sit amet cursus convallis, sapien lectus pretium metus, vitae pretium enim wisi id lectus. Donec vestibulum. Etiam vel nibh. Nulla facilisi. Mauris pharetra. Donec augue. Fusce ultrices, neque id dignissim ultrices, tellus mauris dictum elit, vel lacinia enim metus eu nunc.

Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Vestibulum tortor quam, feugiat vitae, ultricies eget, tempor sit amet, ante. Donec eu libero sit amet quam egestas semper. Aenean ultricies mi vitae est. Mauris placerat eleifend leo. Quisque sit amet est et sapien ullamcorper pharetra. Vestibulum erat wisi, condimentum sed, commodo vitae, ornare sit amet, wisi. Aenean fermentum, elit eget tincidunt condimentum, eros ipsum rutrum orci, sagittis tempus lacus enim ac dui. Donec non enim in turpis pulvinar facilisis. Ut felis.

Cras sed ante. Phasellus in massa. Curabitur dolor eros, gravida et, hendrerit ac, cursus non, massa. Aliquam lorem. In hac habitasse platea dictumst. Cras eu mauris. Quisque lacus. Donec ipsum. Nullam vitae sem at nunc pharetra ultricies. Vivamus elit eros, ullamcorper a, adipiscing sit amet, porttitor ut, nibh. Maecenas adipiscing mollis massa. Nunc ut dui eget nulla venenatis aliquet. Sed luctus posuere justo. Cras vehicula varius turpis. Vivamus eros metus, tristique sit amet, molestie dignissim, malesuada et, urna.

Cras dictum. Maecenas ut turpis. In vitae erat ac orci dignissim eleifend. Nunc quis justo. Sed vel ipsum in purus tincidunt pharetra. Sed pulvinar, felis id consectetuer malesuada, enim nisl mattis elit, a facilisis tortor nibh quis leo. Sed augue lacus, pretium vitae, molestie eget, rhoncus quis, elit. Donec in augue. Fusce orci wisi, ornare id, mollis vel, lacinia vel, massa.

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Morbi commodo, ipsum sed pharetra gravida, orci magna rhoncus neque, id pulvinar odio lorem non turpis. Nullam sit amet enim. Suspendisse id velit vitae ligula volutpat condimentum. Aliquam erat volutpat. Sed quis velit. Nulla facilisi. Nulla libero. Vivamus pharetra posuere sapien. Nam consectetuer. Sed aliquam, nunc eget euismod ullamcorper, lectus nunc ullamcorper orci, fermentum bibendum enim nibh eget ipsum. Donec porttitor ligula eu dolor. Maecenas vitae nulla consequat libero cursus venenatis. Nam magna enim, accumsan eu, blandit sed, blandit a, eros.

Quisque facilisis erat a dui. Nam malesuada ornare dolor. Cras gravida, diam sit amet rhoncus ornare, erat elit consectetuer erat, id egestas pede nibh eget odio. Proin tincidunt, velit vel porta elementum, magna diam molestie sapien, non aliquet massa pede eu diam. Aliquam iaculis. Fusce et ipsum et nulla tristique facilisis. Donec eget sem sit amet ligula viverra gravida. Etiam vehicula urna vel turpis. Suspendisse sagittis ante a urna. Morbi a est quis orci consequat rutrum. Nullam egestas feugiat felis. Integer adipiscing semper ligula. Nunc molestie, nisl sit amet cursus convallis, sapien lectus pretium metus, vitae pretium enim wisi id lectus. Donec vestibulum. Etiam vel nibh. Nulla facilisi. Mauris pharetra. Donec augue. Fusce ultrices, neque id dignissim ultrices, tellus mauris dictum elit, vel lacinia enim metus eu nunc.

A

Appendix

The full set of code for this research and method can be found here. To completely reproduce these experiments, run `methods.ipynb` in full. The following are notable code snippets mentioned in the body of the article.

Listing A.1: Python Implementation of King's Ecological Inference

```
import numpy as np
import pymc3 as pm
```

```python
def ei_two_by_two(demo_pcts, candidate_pcts, precint_populations, lmbda
    =0.5):
    """
    Run King's Ecological Inference method on
    a 2x2 example (2 demographic groups).

    group_demo_pcts (NumPy array): the percentage of people in the
    demographic group for each precinct
    group_voting_pcts (NumPy array): the percentage of people in the
    precinct who voted for a candidate
    precint_populations (NumPy array): the populations of the
    precincts
    lmbda (float):

    return: the probabilistic model
    """
    demo_counts = candidate_pcts * precint_populations
    # Number of populations
    p = len(precint_populations)
    with pm.Model() as model:
        c_1 = pm.Exponential('c_1', lmbda)
        d_1 = pm.Exponential('d_1', lmbda)
        c_2 = pm.Exponential('c_2', lmbda)
        d_2 = pm.Exponential('d_2', lmbda)

        b_1 = pm.Beta('b_1', alpha=c_1, beta=d_1, shape=p)
        b_2 = pm.Beta('b_2', alpha=c_2, beta=d_2, shape=p)

        theta = demo_pcts * b_1 + (1 - demo_pcts) * b_2
        Tprime = pm.Binomial('Tprime', n=precint_populations , p=theta,
            observed=demo_counts)
    return model
```

Listing A.2: Integer Partitioning

```python
from itertools import chain, permutations
```

```python
def integer_partition(n, k, min_size=0):
    """
    Partition an integer.

    n (int): the integer to partition
    k (int): the number of elements in a partition
    min_size (int): the minimum size of an element
    in the partition

    return: a generator of partitions as tuples
    """
    if k < 1:
        return
    if k == 1:
        if n >= min_size:
            yield (n,)
        return
    for i in range(min_size, n // k + 1):
        for result in integer_partition(n - i, k - 1, i):
            yield (i,) + result

def permute_integer_partition(n, k, min_size=0):
    """
    Partition an integer, with all permutations

    n (int): the integer to partition
    k (int): the number of elements in a partition
    min_size (int): the minimum size of an element
    in the partition

    return: a generator of all permutations of partitions as tuples
    """
    return chain.from_iterable(set(permutations(p)) for p in
        integer_partition(n, k, min_size))
```

Listing A.3: Generate a Random Election

```
import numpy as np

def generate_random_election(candidates, demo, beta):
    """
    Generate a random election.

    candidates (string list): the candidates
    demo (dict): the demographics of the electorate
    beta (dict): the theoretical voting percentages of
    each demographic group, for each candidate

    return: a dictionary of candidates and the vote breakdowns by
    demographic group
    """
    # Set up the result dictionary
    num_groups = len(demo)
    result = {'a': (0, [0] * num_groups),
              'b': (0, [0] * num_groups),
              'c': (0, [0] * num_groups)}

    # Iterate through each demographic group
    for group_index, group in enumerate(demo):
        # Simulate each voter
        for voter in range(demo[group]):
            vote = np.random.choice(candidates, 1, beta[group])[0]
            prev_total, prev_breakdown = result[vote]
            prev_breakdown[group_index] += 1
            result[vote] = prev_total + 1, prev_breakdown
    return result
```

Listing A.4: Evaluate and Compare DVM and King's EI

```
import numpy as np
from operator import mul
import functools
```

```python
import random
import pymc3 as pm
import time

from itertools import chain, permutations

def dvm_king_evaluator(election, demo, beta, label, n_iter=1, met_iter=200)
    :
    """
    Run and compare the results of the Discrete
    Voter Model and King's EI on generated
    election data.

    election (dict): the random election to
    evaluate on
    demo (dict): the demographic dictionary of
    a precinct
    label (string): the label of the experiment
    n_iter (int): the number of times to repeat
    the experiment
    met_iter (int): the number of iterations to use
    for Metropolis-Hastings

    return: a dictionary of the label and times and MSEs for
    the Discrete Voter Model and King's EI
    """
    results = {}
    dvm_total_time = 0
    dvm_total_mse = 0

    king_total_time = 0
    king_total_mse = 0

    temp_pcts_array = [pcts[0] for group, pcts in beta.items()]
    true_pcts = np.fromiter(temp_pcts_array, dtype=float)

    for _ in range(n_iter):
        # Get the observed votes for candidate a
```

```
candidate_a_obs = random_election_1_1['a'][0]

# Run Metropolis-Hastings and time it
dvm_total_time -= time.time()
initial_grid = make_grid(len(demo), 10)
met_result = metropolis_hastings(met_iter, initial_grid,
    candidate_a_obs, demo, scoring_type='prob')
dvm_total_time += time.time()

# Find the best grid and output the result
best_grid = met_result['best_grid']
best_cell = get_most_probable_cell(met_result['best_grid'])
vote_pcts = get_vote_pcts(best_cell, 10, demo)

# Find the MSE of the result
dvm_mse_array = np.fromiter(vote_pcts.values(), dtype=float)
dvm_total_mse += mse(dvm_mse_array, true_pcts)

# Run King's EI and time it, if at the right dimension
if len(demo) > 2:
    continue
king_demo = list(demo.values())[0] / 100
king_cand_vote = candidate_a_obs / 100
king_prec_pop = np.array([100])

king_total_time -= time.time()
king_model = ei_two_by_two(king_demo, king_cand_vote, king_prec_pop
    )
with king_model:
    king_trace = pm.sample()
king_total_time += time.time()

# Find the MSE of the result
king_mse_array = np.fromiter([king_trace.get_values('b_1').mean(),
                              king_trace.get_values('b_2').mean()],
                             dtype=float)

king_total_mse += mse(king_mse_array, true_pcts)
```

```
    return {'label': label,
            'dvm_time': dvm_total_time / n_iter,
            'dvm_mse': dvm_total_mse / n_iter,
            'king_time': king_total_time / n_iter,
            'king_mse': king_total_mse / n_iter}
```

References

[Hol] *Holder v. Hall, 512 U.S. 874, 904 n.13 (1994).*

[cle] *League of United Am. Citizens v. Clements, 999 F.2d 831, 860 (5th Cir. 1993) (en banc).*

[LUL] *League of United Latin American Citizens v. Perry, 126 S. Ct. 2594 (2006).*

[Lew] *Lewis v. Alamance County, 99 F.3d 600, 604 n.3 (4th Cir. 1996.*

[tho] Thornburg v. Gingles. *Oyez.*

[vra] Voting Rights Act (1965). *Our Documents - Voting Rights Act (1965).*

[7] (2015). Introduction to federal voting rights laws. *The United States Department of Justice*.

[8] Betancourt, M., Byrne, S., Livingstone, S., & Girolami, M. (2017). The geometric foundations of hamiltonian monte carlo. *Bernoulli*, 23(4A), 2257–2298.

[9] Christian, R. & Casella, G. (2012). A short history of markov chain monte carlo: Subjective recollections from incomplete data. *arXiv.org*, 26(1).

[10] Commons, W. (2014). File:traviscountydistricts.png — wikimedia commons, the free media repository. [Online; accessed 15-November-2019].

[11] Deruelle, N. & Uzan, J.-P. (2018). Hamiltonian mechanics. In *Relativity in Modern Physics*. Oxford University Press.

[12] Ferree, K. E. (2004). Iterative approaches to r x c ecological inference problems: Where they can go wrong and one quick fix. 12(2).

[13] Gelman, A., Park, D., Ansolabehere, S., Price, P., & Minnite, L. (2001). Models, assumptions and model checking in ecological regressions. *Journal Of The Royal Statistical Society Series A-Statistics In Society*, 164, 101–118.

[14] Greiner, D. J. (2007). Ecological inference in voting rights act disputes: Where are we now, and where do we want to be? *Jurimetrics*, 47(2), 115–167.

[15] King, G. (1997). *A Solution to the Ecological Inference Problem: Reconstructing Individual Behavior from Aggregate Data*. Princeton, New Jersey: Princeton University Press.

[16] King, G. (2013). *A Solution to the Ecological Inference Problem: Reconstructing Individual Behavior from Aggregate Data*. Princeton: Princeton University Press.

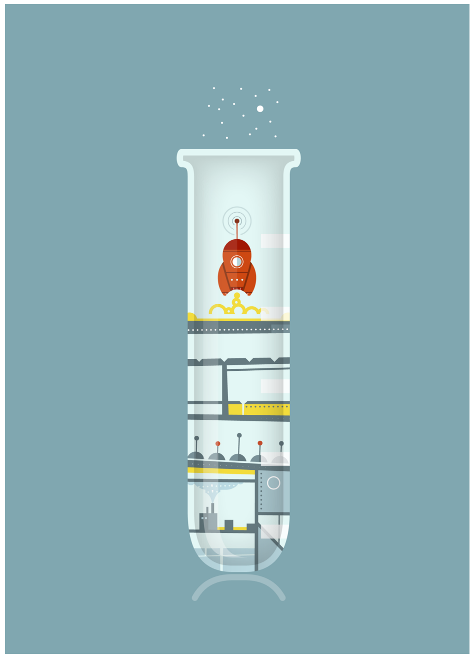
[17] King, G., Rosen, O., & Tanner, M. A. (1999). Binomial-beta hierarchical models for ecological inference. *Sociological Methods & Research*, 28(1), 61–90.

[18] Liang, F. F. (2010). *Advanced Markov chain Monte Carlo methods : learning from past samples*. Wiley series in computational statistics. Chichester, England: Wiley.

[19] Neal, R. (2012). Mcmc using hamiltonian dynamics. *arXiv.org*.

[20] Roberts, G. O. & Rosenthal, J. S. (2004). General state space markov chains and mcmc algorithms. *Probab. Surveys 1*, 1(20-71), 20–71.

[21] Rubin, D. B. (1987). *Multiple imputation for nonresponse in surveys*. Wiley series in probability and mathematical statistics. Applied probability and statistics. New York ;: Wiley.



THIS THESIS WAS TYPESET using LaTeX, originally developed by Leslie Lamport and based on Donald Knuth's TeX. The body text is set in 11 point Egenolff-Berner Garamond, a revival of Claude Garamont's humanist typeface. The above illustration, *Science Experiment 02*, was created by Ben Schlitter and released under CC BY-NC-ND 3.0. A template that can be used to format a PhD dissertation with this look & feel has been released under the permissive AGPL license, and can be found online at github.com/suchow/Dissertate or from its lead author, Jordan Suchow, at suchow@post.harvard.edu.