Physica A 506 (2018) 170–178

Contents lists available at ScienceDirect

Physica A

journal homepage: www.elsevier.com/locate/physa

Minireview

Sampling from complicated and unknown distributions

Monte Carlo and Markov Chain Monte Carlo methods for redistricting

Wendy K. Tam Cho [a,*], Yan Y. Liu [b]

[a] *Departments of Political Science, Statistics, Mathematics, Asian American Studies, and the College of Law and Senior Research Scientist at the National Center for Supercomputing Applications, University of Illinois at Urbana-Champaign, United States*

[b] *Department of Geography and Geographic Information Science and the National Center for Supercomputing Applications, University of Illinois at Urbana-Champaign, United States*

ARTICLE INFO

Article history:
Received 21 February 2018
Received in revised form 28 March 2018
Available online 16 April 2018

Keywords:
Markov Chain Monte Carlo
Monte Carlo simulation
Redistricting

ABSTRACT

Sampling from complicated and unknown distributions has wide-ranging applications. Standard Monte Carlo techniques are designed for known distributions and are difficult to adapt when the distribution is unknown. Markov Chain Monte Carlo (MCMC) techniques are designed for unknown distributions, but when the underlying state space is complex and not continuous, the application of MCMC becomes challenging and no longer straightforward. Both of these techniques have been proposed for the astronomically large redistricting application that is characterized by an extremely complex and idiosyncratic state space. We explore the theoretic applicability of these methods and evaluate their empirical performance.

Contents

1. Introduction

For many interesting problems, the technique of random sampling is powerful because it allows us to draw a representative sample from an underlying population of interest. Random samples allow us to make inferences about a population without the need to examine each population unit. If the population distribution is known, drawing samples can be straightforward. When the distribution is unknown, the problem is more complex.

* Corresponding author.
E-mail address: wendycho@illinois.edu (W.K.T. Cho).

https://doi.org/10.1016/j.physa.2018.03.096
0378-4371/

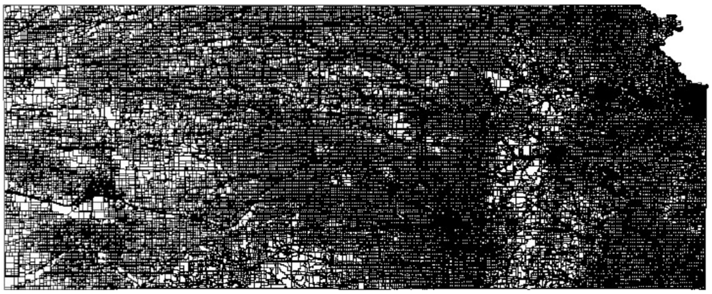

Fig. 1. The census blocks in the state of Kansas.

Recently, there has been some interest in statistical methods for uniformly sampling the space of possible electoral maps. This is helpful if one wants to assess the properties of a particular map because it allows us to place the map of interest in context [1] and provides the potential to assess, for instance, whether the current map is an outlier in the space of feasible alternative maps. This idea animates a line of legal reasoning that the U.S. Supreme Court has struggled to establish to adjudicate partisan gerrymandering cases in the United States. The U.S. Supreme Court has stated that while partisan information may be used in devising electoral maps (*Gaffney v. Cummings*, 412 U.S. 735 (1973)), "*excessive* injection of politics is *un*lawful" (*Vieth v. Jubelirer*, 541 U.S. 267 (2004)). One might equate "excessiveness" with a map that is an outlier in its partisan effect. A way to assess whether a map is an outlier in this sense is to examine where that map lies within the possible values or in the distribution of feasible maps. Since the entire set of feasible maps is too large to enumerate, uniformly sampling from the set of feasible maps is a possible recourse for redistricting litigation. If one could characterize the distribution of feasible alternative maps, then one could make an argument for whether the disputed map is an outlier in that distribution. The courts seem amenable to this line of reasoning, but the goal of uniformly sampling feasible electoral maps is neither simple nor straightforward.

While several different sampling schemes have been proposed, these proposals have not been accompanied with much rigorous analysis of their empirical performance or discussion of their theoretical properties in the redistricting realm. We explore the redistricting problem and conduct an empirical analysis of the ability of these estimators to uniformly sample the idiosyncratic set of legal and feasible electoral maps.

2. The redistricting state space

Redistricting in the United States is equivalent to partitioning an n-element set into k aggregated units. We can visualize the problem in Fig. 1 that shows the census blocks for the state of Kansas, which has 238,600 census blocks. Census blocks are the smallest unit of census geography that may be used as a building block for electoral districts. In Kansas, these geographic units must be aggregated into 4 total congressional districts. It is visually apparent that there is an astronomical number of ways in which small census geographies can be aggregated into electoral districts. Moreover, Kansas is one of the more simple cases. In California, there are more than 700,000 census blocks that need to be partitioned into 53 congressional districts.

The total number of possible unconstrained partitions of n elements into k aggregated units is a Stirling number of the second kind, $S(n, k)$ [2], which can be computed recursively as $S(n, k) = kS(n-1, k) + S(n-1, k-1)$, for $n \geq 1$ and $1 \leq k \leq n$. Even with only a modest number of units, the scale of the unconstrained map-making problem is awesome. Partitioning only $n = 55$ units into $k = 6$ districts already presents a formidable 8.7×10^{39} number of possibilities. The state space becomes smaller when the legal constraints that govern the redistricting process are imposed, though there is no known way to place an upper bound on the number of possibilities with constraints, and the number of possibilities remains prohibitively large.

The state space that includes only contiguous partitions can be defined in graph theoretic terms as a continuous space. Here, each geographic unit, i, is a node or vertex in the graph, $v_i \in V$, and the edges, E, represent node adjacency. The problem is then a graph partitioning problem where the set of nodes is partitioned into d districts such that every node is in only 1 district, $V_{d_i} \cap V_{d_j} = \emptyset$, for $i \neq j$, where V_{d_i} is the set of nodes in district i, $i = 1, \ldots, d$; all nodes are in one of the d districts, $\bigcup_{i=1}^{d} V_{d_i} = V$; and the subgraphs of each district are contiguous. In this setup, if one begins with a valid partition, one can always ensure a move to another valid partition by considering only units along district boundaries as possibilities for movement to another partition. Hence, this state space is continuous. However, even in this smaller continuous state space, an actual redistricting application remains astronomically large and non-trivial such that an exhaustive search of this space within the available computing environment is not possible.

In addition, while traversing a continuous state space is more straightforward, when even a single constraint in addition to contiguity is imposed, it is no longer obvious how to construct a continuous state space. When the state space is not continuous, uniformly sampling the non-continuous feasible space is challenging because while the state space may be

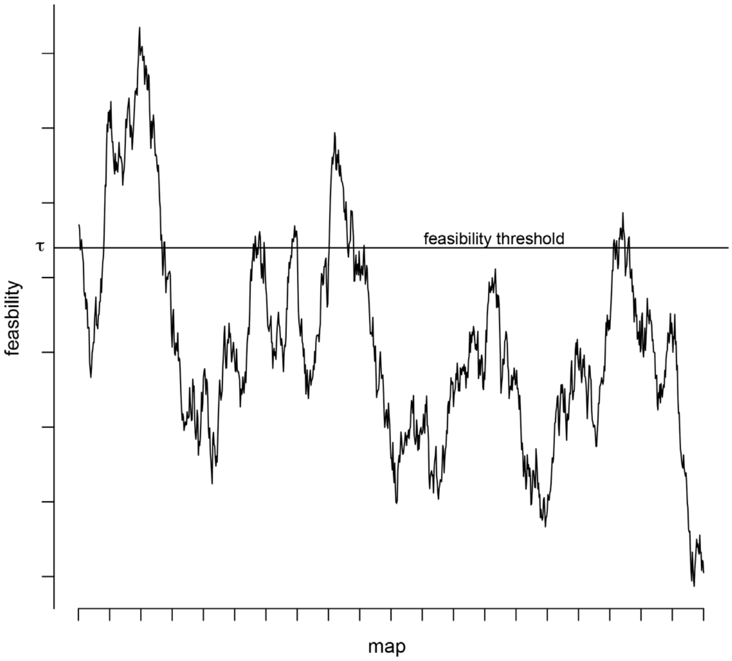

Fig. 2. Rugged State Space with Feasibility Threshold. Only maps with feasibility scores above some threshold are feasible maps.

rugged with many peaks and valleys, most of the state space is infeasible. We can visualize this in Fig. 2, which is a 2D rendition of the state space. Each map is associated with a feasibility score. Only those maps above some feasibility threshold satisfy the legal constraints and are thus viable maps. Importantly, if a map does not satisfy the legal constraints, its properties are irrelevant in a comparison with a legal and enacted map. As the number of legal constraints increases, the number of maps excluded by the feasibility threshold quickly proliferates, presenting a vexingly difficult traversal space. In short, because there is an astronomically large set of feasible electoral maps located in a state space that is not easily traversable, uniformly sampling from this set is difficult.

3. Empirical performance of samplers

Various methods have been proposed for sampling legal electoral maps. We examine the empirical performance of these samplers with a small data set consisting of 25 precincts from the state of Florida (FL25).[1] This data set is well-suited for our purposes because it is both large enough to be non-trivial (the number of ways to partition 25 precincts into 3 districts without constraints is $S(25, 3) = 141{,}197{,}991{,}025$), and small enough that we can enumerate the entire set of feasible maps. Since we have the underlying population distribution, we know the correct answer for these data. If we impose a contiguity constraint, the number of valid partitions reduces by several orders of magnitude to 117,688. If we further impose a population constraint that requires the population deviation from the ideal population to be less than 10%, the number of valid partitions drops to 927.

Small data sets with known answers are essential for designing algorithms for large problems such as redistricting. To be sure, if an algorithm performs poorly for small problems, it would be ill advised to apply the same algorithm to an astronomically larger and far more complex problem. To gain a sense of the difference in magnitude between an actual redistricting application and our small FL25 application, note that redistricting Pennsylvania entails partitioning more than 9,000 voter tabulation districts into 18 districts while satisfying a significant set of complex legal constraints (including population equality, contiguity, compactness, preservation of political boundaries, compliance with the Voting Rights Act, and incumbency protection). For FL25, we partition only 25 precincts into 3 districts with one (only contiguity) or two (contiguity and population) constraints. Moreover, it is customary when designing algorithms for large problems to work with smaller problems because this strategy allows us to understand the large problem and its complexities better, which, in turn, helps us design efficient and effective heuristics more intelligently. We use FL25 to examine the performance of two proposed methods for characterizing the space of feasible electoral maps, ordinary Monte Carlo type algorithms and Markov Chain Monte Carlo implementations.

[1] This data set was created by and examined in [3].

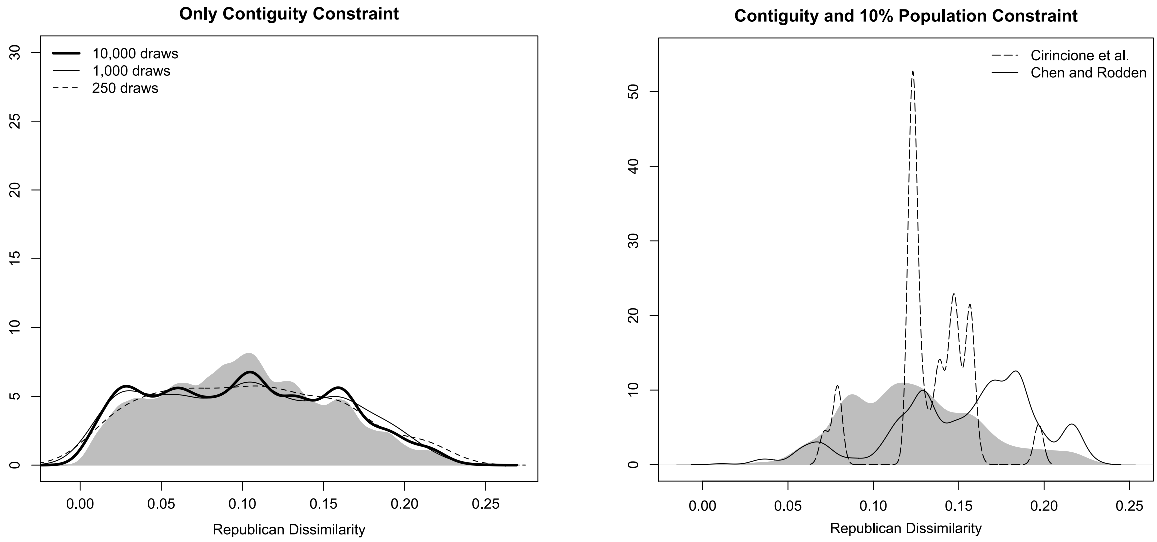

Fig. 3. Florida 25 precincts, 3 districts. Monte Carlo Sampler Performance.

3.1. Monte Carlo algorithms

An ordinary Monte Carlo algorithm is a computational method that relies on repeated random sampling to recover an unknown quantity [4]. Suppose we are interested in the expected value of some map metric, $f(X), f : X \rightarrow \mathbb{R}$, where X is a map, but we are unable to calculate this metric for every map. If we can simulate maps, $X_1, X_2, \ldots, X_n$, such that the simulated set of maps is independent and identically distributed with the same distribution as the set of all maps, then we can approximate the unknown mean of all maps, μ, as

$$\hat{\mu}_n = E[f(X)] = \frac{1}{n}\sum_{i=1}^{n} f(X_i). \tag{1}$$

Chen and Rodden [5] and Cirincione et al. [6] seek to fulfill this theory with their Monte Carlo type map simulation algorithms. Both algorithms begin with a randomly selected unit. Both then build on to the randomly selected unit by merging another randomly selected unit on the border of that unit. Cirincione et al. [6] repeat this procedure until the merged set of units is within the ideal population, at which point it begins to build another district. Chen and Rodden [5] first create contiguous districts and then exchange units so that the districts are close in population.

Neither embarks on a study of the theoretical properties of their algorithms. That is, though these algorithms generate or "simulate" electoral maps, whether the set of maps that they identify is a uniform sample of the set of all feasible maps has never been shown by its authors. Moreover, skepticism has been raised by several scholars that this process does not uniformly sample from the full set of feasible maps even though it has stochastic components. Fifield et al. [3], commenting on these types of algorithms, state that "no theoretical justification is given for these existing simulation algorithms, and some of them are best described as ad-hoc…the algorithms come with no theoretical result and are not even designed to uniformly sample redistricting plans even though researchers have a tendency to assume that they are". Chikina et al. [7, p. 2862] raise similar concerns, commenting that "…this work has relied on heuristic sampling procedures, which do not have the property of selecting districtings with equal probability (and more generally, distributions that are not well-characterized), undermining rigorous statistical claims about the properties of typical districts".

When we examine the performance of these samplers on FL25, our empirical evidence supports these suspicious. Though these methods have some stochastic components and are able to identify maps satisfying particular criteria, they do not uniformly sample the underlying population and thus do not produce a representative sample of the underlying population of interest.

Fig. 3 shows the results from our Monte Carlo simulation. On the left, we use the Chen and Rodden algorithm to draw samples of maps from the simple case where maps are only constrained to have three contiguous districts. Using the Chen and Rodden algorithm, we drew samples of electoral maps of size 250, 1000, and 10,000.[2] Each of the electoral maps, X, is associated with a metric, $f(X) \in \mathbb{R}$. Here, for each map, $X, f(X)$ is "Republican Dissimilarity", which is a measure of the

[2] In the Cirincione et al. [6] algorithm, contiguous maps are built until a population criterion is met. For the problem we set up in the plot on the left, since there is no population constraint, we cannot evaluate that algorithm for this simple case.

relative Republican concentration using the Massey and Denton [8] segregation index measure. We note that the measure itself is not of particular importance in our assessment of sampling behavior as long as it provides enough granularity to enable us to distinguish maps from one another so that we can assess whether the underlying maps are uniformly sampled. We can see from the plots that when we are considering only contiguity, the sample size seems to matter little. The samples drawn from any of the sample sizes do not comprise a uniform sample from the underlying population of contiguous maps, though they are not wildly off.

The plot on the right in Fig. 3 shows the result when we simulate maps that have both a contiguity constraint as well as a population constraint. These are not substantively interesting maps since there are a number of other legal constraints that must be satisfied for legally viable maps. However, adding another constraint allows us to examine how these Monte Carlo samplers behave as the types of maps that need to be sampled become more constrained and the state space is no longer continuous. The gray area shows the true underlying distribution of maps satisfying the contiguity and population constraint. The distribution for the sample drawn with the Cirincione et al. algorithm is outlined with a dashed line while the distribution for the sample drawn by the Chen and Rodden algorithm is outlined by a solid line. Neither algorithm comes close to producing a representative sample of the underlying population. Both sample non-uniformly, oversampling in some areas and undersampling in other areas. These algorithms plainly fail to produce a uniform sample of the set of feasible maps. The problem is not sample size since the entire population is of size 927 and 10,000 draws are produced. Instead, the problem is fundamental and does not improve with more draws.

Note that while both the Chen and Rodden and the Circincione et al. algorithms have random components that randomly choose, for instance, from among a defined set of possible units, these stochastic components are not sufficient for ensuring that a uniform sample of the underlying true distribution is drawn. A uniform sample of the underlying distribution is what we seek, not maps that arise from an algorithm that has some stochastic components. Indeed, while the algorithm must embody stochastic components, as we have demonstrated, simply incorporating stochastic components does not guarantee a uniform sample.

3.2. Markov Chain Monte Carlo

Markov Chain Monte Carlo (MCMC) algorithms are computational methods intended to recover the properties of an unknown probability distribution by constructing a Markov chain that has the desired distribution as its stationary distribution [9,10]. The Fundamental Theorem of Markov Chains is the following.

Theorem. *Let $\mathcal{X}$ be a finite set and $K(x, y)$ a Markov chain indexed by $\mathcal{X}$. If there is n_0 so that $K^n(x, y) \geq 0$ for all $n > n_0$, then K has a unique stationary distribution π, and as $n \to \infty$,*

$$K^n(x, y) \to \pi(y) \qquad \text{for each } x, y \in \mathcal{X}.$$

In other words, from any starting state, x, the nth step of a Markov chain has a chance close to $\pi(y)$ of being at y if n is large. Essentially, if we define a Markov chain with the desired properties, and this chain is run for a sufficiently long time, "mixing", or the point at which the stationary distribution is reached from an arbitrary start state, occurs.

The theoretic properties of MCMC are promising and have encouraged a number of different scholars to devise random walk MCMC methods (particularly, Metropolis–Hastings algorithms) [11,12] for redistricting [3,13–15]. However, many hurdles must be overcome before the theory can be realized. In particular, while the MCMC theorem ensures mixing at some point, the theorem does not offer reassurance or any indication of how long a Markov chain must run before its iterations are distributed approximately according to the stationary distribution. It also offers no estimate about the size of the error for any MCMC estimate. Moreover, formulating a random walk given the many legal constraints on electoral maps is not straightforward since the state space is not continuous, and discarding infeasible solutions that are encountered on the random walk dramatically increases the mixing time. In short, while it is theoretically possible to construct a Markov chain that will produce a representative sample of electoral maps, the empirical problem remains quite difficult and defies simple implementations.

Here, we explore the Fifield et al. [3] MCMC implementation.[3] We can make this choice without loss of generality since the general framework, the issues that arise, and the challenges that must be overcome with any MCMC implementation are identical. Indeed, there is no magic bullet that will overcome the general limitations of the MCMC method. This point is made more strongly by Chikina et al. [7], who express pessimism that MCMC can be successfully applied to the redistricting problem at all.

> In an attempt to establish a rigorous framework for this kind of approach, several groups [3,13,16] have used Markov chains to sample random valid districting for the purpose of such comparisons. Like many other applications of real-world Markov chains, however, these methods suffer from the completely unknown mixing time of the chains in question. Indeed, no work has even established that the Markov chains are irreducible (in the case of districting,

[3] We make this choice because the other papers provided less in the way of algorithmic detail. There is enough detail, however, to feel confidant that all of the current implementations are very similar. They all describe a state space defined in a graph theoretic framework where the entire state space is composed of the space of contiguous districts and the random walk consists of shifting units from the border of one district to a neighboring district.

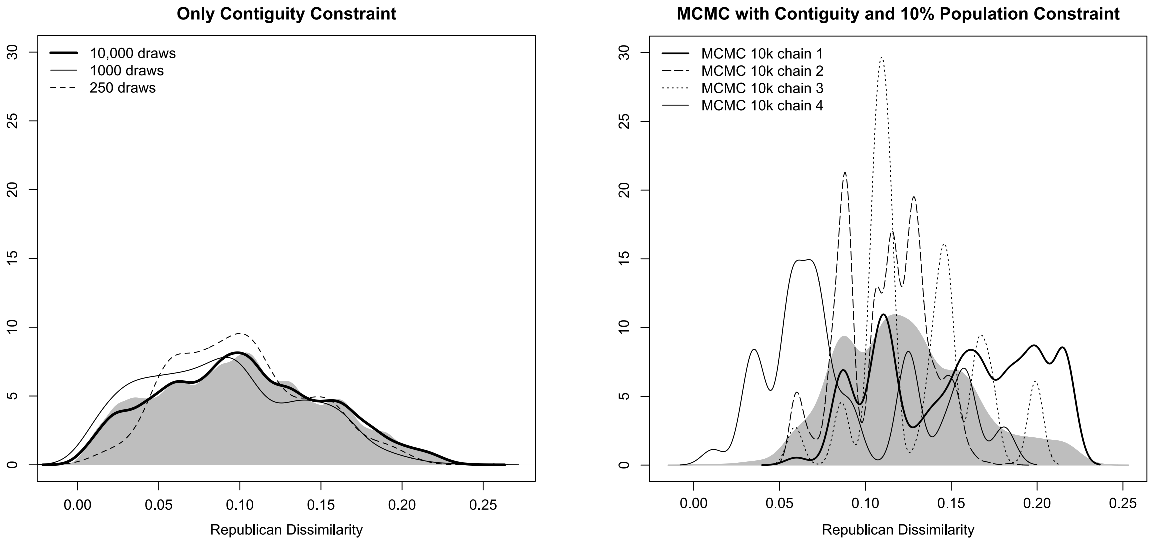

Fig. 4. MCMC performance for sampling uniformly from the FL25 data set.

irreducibility means that any valid districting can be reached from any other by a legal sequence of steps), even if valid districting was only required to consist of contiguous districts of roughly equal populations. Additionally, indeed, for very restrictive notions of what constitutes valid districting, irreducibility certainly fails.

The plot on the left in Fig. 4 shows the MCMC results when we impose only a contiguity constraint. We can see that when contiguity is the only constraint, as the length of the MCMC chain increases from 250 to 1000 to 10,000, the MCMC estimate steadily improves. While the MCMC chains of length 250 and 1000 are not long enough to mix, the MCMC chain of length 10,000 mixes and provides a uniform random sample of the underlying state space. In this example, where there are 117,688 feasible solutions, the MCMC chain did not mix until its length was about 10% of the size of the entire space. Recall that, here, the state space is continuous—every move results in another feasible solution, so the traversal of an MCMC chain in this space is more straightforward.

The results are less encouraging in the plot on the right in Fig. 4 that shows the performance of the same MCMC implementation when attempting to characterize the space that includes a 10% population constraint in addition to a contiguity constraint. The additional population constraint produces a non-continuous feasible space so that each move is not guaranteed to produce another feasible map since the population constraint may not be satisfied. This condition induces a very difficult state space to traverse, but is not fatal for a Metropolis–Hastings MCMC since the acceptance ratio makes it possible to move through infeasible regions or maps and emerge at a different disconnected feasible region. For this problem, we ran multiple Markov chains of length 10,000. The various lines show the results from four MCMC runs that begin at different random starts. As we can see, there is quite a bit of variation between the four chains. Chain 2 is not particularly close to the true distribution, but we might regard it as the closest to the true distribution of these four MCMC runs. Chain 4 significantly oversamples on the left side of the distribution and very poorly samples the distribution in general. Chain 1 does the opposite, significantly oversampling on the right side while poorly sampling elsewhere. Chain 3 generally samples where the distribution is most dense, but fails to sample uniformly. Though a chain of length 10,000 was sufficient for mixing in the simpler problem, it is insufficient for mixing when just one more constraint is required.

We also ran multiple chains of length 100,000 to examine whether chains that are 10 times longer would mix. However, the longer chains of length 100,000 performed no better than the chains of length 10,000. They exhibited similar behavior to the shorter chains—sampling in only particular regions of the underlying space. Since the underlying state space is no longer continuous, this particular MCMC implementation exhibits great difficulty in traversing the state space. It is strongly constrained to one region and does not appear to easily move to other disconnected regions of the state space.

Fig. 5 provides another view of the data. Here, the sample size and population size are normalized so that we can examine whether this MCMC implementation samples uniformly. The plots show the true distribution (in dark gray), the MCMC sample (in light gray), and the overlap (in medium gray). On the left, the results for the problem with only the contiguity constraint is shown. As we can see, the overlap is substantial, illustrating that mixing occurred—for sampling from the unknown distribution where only contiguity is required, this MCMC algorithm is able to uniformly sample from the underlying distribution.

The plots on the right show the results of the more difficult problem with both the contiguity constraint as well as a 10% population constraint. For at least three of the plots, we see that some values are substantially oversampled while other values are not sampled at all. Chain 2 shows the best overlap, but completely misses a particular set of feasible maps on the

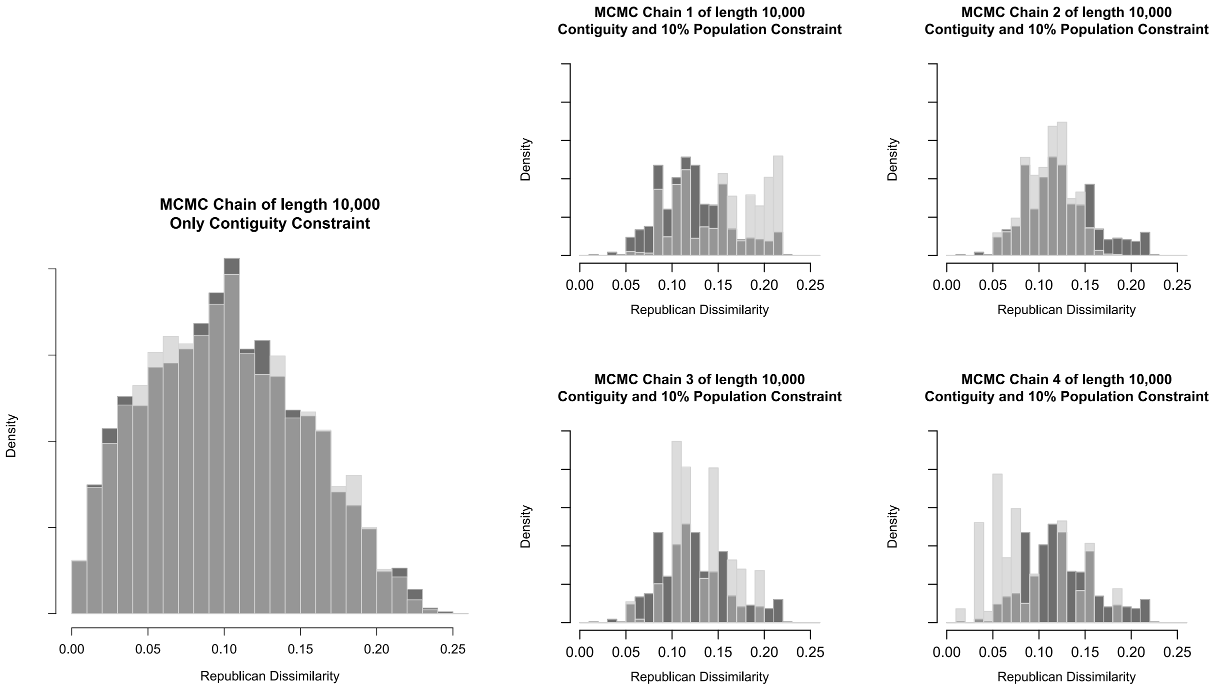

Fig. 5. Florida 25 precincts, 3 districts: MCMC Overlap and Uniform Sampling. The true underlying distribution is shown in dark gray. The MCMC sample is shown in light gray.

right side of the distribution. Chain 1 substantially oversamples on that same side of the distribution while poorly sampling overall. Strikingly, all four of these chains sample very differently from one another. Perhaps even more curious is that, despite there being only 927 feasible maps and a chain of length 10,000, the MCMC algorithm fails to find many of the feasible maps. The percentage of feasible maps visited varies significantly by chain. Chain 4 visited only 3.7% of the feasible maps while Chain 2 visited 52.4% of the full set of feasible maps. So, this MCMC algorithm both fails to identify many of the possible maps, and of the ones that it does find, it is not sampling uniformly from the underlying distribution.

It is clear from the plots that these different chains visit the space very differently, and that where they visit is related to the location of the random start state. The substantial oversampling in areas of low density imply that the chain is "stuck" in these areas. Certainly one reason why a chain might get stuck in a certain portion of the space is that when constraints are added to the problem, the feasible space is no longer continuous or path connected, making the space challenging to traverse. If the chain begins in one part of the space that is not well-connected to other parts of the feasible space, then it may never reach these other parts of the space or, minimally, find it difficult to identify a path to these other areas since it must first travel through possibly very large portions of infeasible space. Moving through infeasible space is possible with a Metropolis–Hastings MCMC, but this obviously becomes less likely as the portions of infeasible space grow, which is precisely the effect in the redistricting problem as additional constraints are imposed. When the ability to identify and traverse the feasible space is compromised, the estimate from a single MCMC chain in this implementation results in a biased and incorrect estimate of the parameter of interest.

MCMC can produce random samples of unknown distributions, in theory. However, in practice, this outcome is not always easily obtained and certainly not guaranteed. Accordingly, MCMC chains must be devised with careful thought and understanding of the underlying state space. For electoral maps, the underlying state space is plainly idiosyncratic and multimodal. If one has to satisfy only a contiguity constraint, it is simple to specify a path connected and continuous state space. However, as additional constraints are added, how one might specify a continuous state space or, alternatively, devise movement to jump from one region to a different disconnected region is not straightforward and must be carefully considered in devising the Markov chain. As more constraints are added, the state space becomes increasingly idiosyncratic, multi-modal, and difficult to traverse.

Further, it is difficult to know when we are in a situation where our MCMC algorithm failed. Fig. 6 shows the trace plots for our four chains. None of the trace plots for our four chains induces immediate or strong concerns. It is arguable that all four of these diagnostics indicate that each of the chains has mixed since there is scant evidence of a trend or a change in spread. In that case, the convergence is only pseudo-convergence. That is, the diagnostics for the chains indicate that they have mixed when they have not. Indeed, it is not unusual for MCMC diagnostics to indicate that a chain has converged when it has not. Brooks et al. [17] are clear that

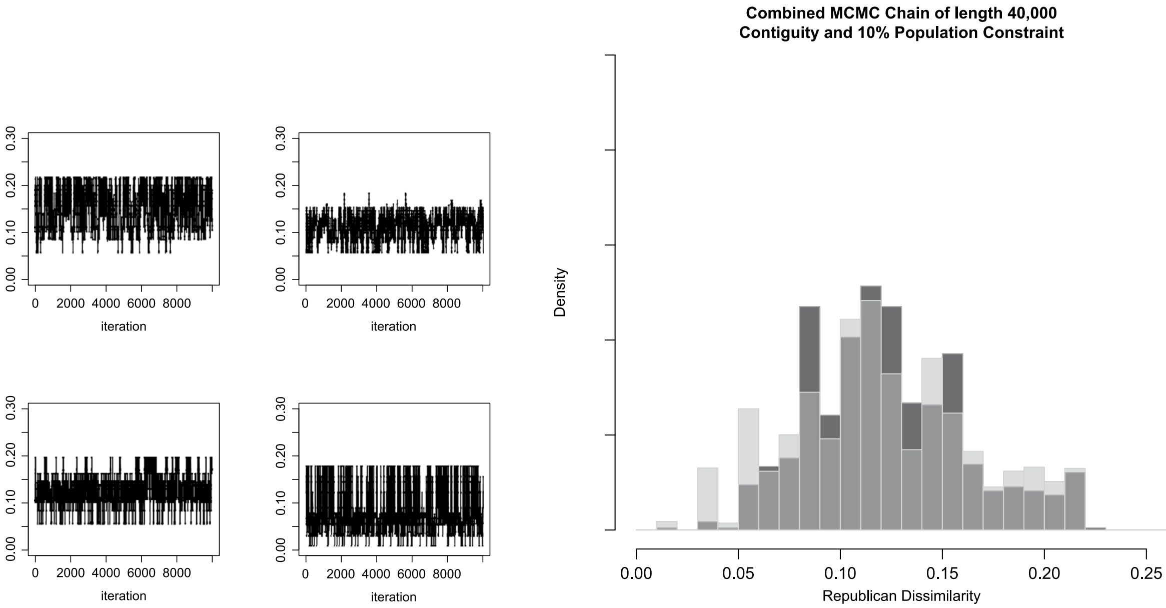

Fig. 6. Florida 25 precincts, 3 districts: Trace Plots and Combined Chain. The true underlying distribution is shown in dark gray. The combined MCMC sample is shown in light gray.

> Diagnostics can only be used to determine a lack of convergence and not detect convergence *per se*. For example, it is relatively easy for a sampler to become stuck in a local mode and naively applied diagnostics would not detect that the chain had not explored the majority of the model/parameter space. Therefore, it is important to use a range of techniques, preferably assessing different aspects of the chains and each based upon independent chains started a range of different starting points. If only a single diagnostic is used and it detects no lack of convergence, then this provides only mild reassurance that the sampler has performed well.

Pseudo-convergence can happen when there are multiple modes in the underlying distribution or non-continuous feasible portions of the state space. Both of these conditions are common in a redistricting application. In FL25, what is happening is that the various chains, depending on where the random initial starting point was, visit different parts of the space, but are unable to transition across infeasible space to other feasible portions of the space.

Since the different chains explored different parts of the space and seemed to have explored those spaces reasonably, if all parts of the space are explored (which has not quite occurred with 4 chains, but certainly is better achieved with four chains than one chain), we can gain traction by combining these various chains. The plot on the right in Fig. 6 shows the overlap of the underlying distribution with a chain that combines our four chains. As we can see, the combined chain captures the underlying distribution better than any of the single chains, which were all unwittingly biased by their starting point. The combined chain visits more parts of the space while the single chains were stuck in particular region of the state space.

In an actual redistricting application, the number of chains needed is quite large and must be initiated in different portions of the space. Even for the FL25 data set that has only 25 precincts, four separate chains with long lengths relative to the number of feasible solutions are needed to approach mixing. There are various strategies for improving MCMC, including simulated tempering, parallel tempering, simulated sintering, evolutionary MCMC, Metropolis-coupled MCMC, and Hamiltonian Monte Carlo, among many others that have been proposed and designed for peculiar state spaces that defy a more simplistic MCMC implementation [18–25]. None of these strategies is foolproof, but rather are a part of model design, which must be informed by a deep and nuanced understanding of the state space for the application at hand. For instance, if we know that the state space is disconnected, our MCMC implementation must embody a mechanism that allows it to visit different and disconnected portions of the feasible space.

For a large and complex application like redistricting, the challenges are daunting. We cannot expect a simplistic MCMC implementation to produce a random sample of the entire space. Feasible maps are governed by many legal constraints (not simply contiguity and population equality, as in our FL25 problem). The underlying space in an actual redistricting problem is far more complex to traverse than the already difficult space presented by the FL25 data set. The complex MCMC implementations that are behooved by the very large redistricting problem are further computationally expensive.

4. Discussion

The Monte Carlo algorithms that have been proposed [5,6] do not uniformly sample the state space and do not yield a representative sample. Their bias is sure and the direction of the bias is unknown, rendering their results unreliable.

Theoretically, MCMC techniques are promising but their success is also far from assured. The chains must be devised in such a way that is cognizant of the underlying disconnected, multi-modal, and enormous state space. Current MCMC implementations focus on a graph partitioning scheme that shifts one unit at a time, making it difficult to move out of the immediate search space or to move through infeasible space to other feasible solutions, limiting the search space encountered and biasing the outcome.

For the redistricting application, it is clear that, in addition to the importance of the mathematical theory underpinning a sampling scheme, one must also address the enormous computational complexity of the problem. For MCMC, the mathematical theory states that success is possible, but the results are only asymptotically assured. The amount of computation required for an MCMC chain to mix in a redistricting application is massive. One promising avenue is to expend effort to extend MCMC algorithms to the high-performance-computing realm by, perhaps, using MPI (Message Passing Interface, a library and communication protocol for parallel programming) to implement an asynchronous inter-process communication framework where parallel MCMC instances can be efficiently deployed. For a problem as complex as redistricting, both harnessing massive computational power as well as carefully devising the traversal mechanism behind an MCMC chain for a problem are essential efforts [26].

Acknowledgments

This research has been funded in part by the National Science Foundation, United States (Grant No. SES-1725418/1728902) and the Guggenheim Foundation, United States, as well as multiple computing allocation grants on the Blue Waters sustained-petascale computing resources, which is supported by the NSF, United States (Grants OCI-0725070 and ACI-1238993) and the state of Illinois. Blue Waters is a joint effort of the University of Illinois at Urbana-Champaign and the National Center for Supercomputing Applications.

References

[1] Wendy K. Tam Cho, Yan Y. Liu, Toward a talismanic redistricting tool: A fully balanced computational method for identifying extreme redistricting plans, Election Law J. 15 (4) (2016) 351–366.
[2] M. Keane, The size of the region-building problem, Environ. Plan. A 7 (1975) 575–577.
[3] Benjamin Fifield, Michael Higgens, Kosuke Imai, Alexander Tarr, A new automated redistricting simulator using Markov chain Monte Carlo, Working Paper, 2017.
[4] George S. Fishman, Monte Carlo: Concepts, Algorithms, and Applications, Springer-Verlag, New York, 1996.
[5] Jowei Chen, Jonathan Rodden, Unintentional gerrymandering: Political geography and electoral bias in legislatures, Q. J. Political Sci. 8 (2013) 239–269.
[6] Carmen Cirincione, Thomas A. Darling, Timothy G. O'Rourke, Assessing South Carolina's 1990s congressional districting, Political Geography 19 (2) (2000) 189–211.
[7] Maria Chikina, Alan Frieze, Wesley Pegden, Assessing significance in a Markov Chain without mixing, Proc. Natl. Acad. Sci. 114 (11) (2017) 2860–2864.
[8] Douglas S. Massey, Nancy A. Denton, The dimensions of residential segregation, Soc. Forces 67 (2) (1988) 281–315.
[9] Steve Brooks, Andrew Gelman, Galin L. Jones, Xiao-Li Meng, Handbook of Markov Chain Monte Carlo, Chapman & Hall/CRC, Boca Raton, FL, 2011.
[10] W.R. Gilks, S. Richardson, D.J. Spiegelhalter, Markov Chain Monte Carlo in Practice, Chapman & Hall/CRC Interdisciplinary Statistics, Boca Raton, FL, 1995.
[11] Nicholas Metropolis, Arianna W. Rosenbluth, Marshall N. Rosenbluth, Augusta H. Teller, Edward Teller, Equation of state calculations by fast computing machines, J. Chem. Phys. 21 (6) (1953) 1087–1092.
[12] W.K. Hastings, Monte Carlo sampling methods using Markov chains and their applications, Biometrika 57 (1) (1970) 97–109.
[13] Sachet Bangia, Christy Vaughn Graves, Gregory Herschlag, Han Sung Kang, Justin Luo, Jonathan C. Mattingly, Robert Ravier, Redistricting: Drawing the line, 2017. arXiv:1704.03360 [stat.AP].
[14] Jonathan C. Mattingly, Christy Vaughn, Redistricting and the will of the people, 2014. arXiv:1410.8796.
[15] Gergory Herschlag, Robert Ravier, Jonathan C. Mattingly, Evaluating Partisan Gerrymandering in Wisconsin, 2017. arXiv:1709.01596.
[16] Lucy Chenyun Wu, Jason Xiaotian Dou, Danny Sleator, Alan Frieze, David Miller, Impartial redistricting: A Markov chain approach, 2015. arXiv:1510.03247 [cs.CY].
[17] S.P. Brooks, P. Giudici, A. Phillipe, Nonparametric convergence assessment for MCMC model selection, J. Comput. Graph. Statist. 12 (2003) 1–22.
[18] Robert H. Swendsen, Jian-Sheng Wang, Replica Monte Carlo simulation of spin-glasses, Phys. Rev. Lett. 57 (1986) 2607–2609.
[19] Simon Duane, A.D. Kennedy, Brian J. Pendleton, Duncan Roweth, Hybrid Monte Carlo, Phys. Lett. B 195 (2) (1987) 216–222.
[20] Charles J. Geyer, Markov Chain Monte Carlo maximum likelihood, in: E.M. Keramidas (Ed.), Computing Science and Statistics: Proceedings of the 23rd Symposium on the Interface, Interface Foundations, Fairfax Station,1991, pp. 156–163.
[21] E. Marinari, G. Parisi, Simulated tempering: A new Monte Carlo scheme, Europhys. Lett. 19 (6) (1992) 451–455.
[22] Charles J. Geyer, Elizabeth A. Thompson, Annealing Markov chain Monte Carlo with applications to ancestral inference, J. Amer. Statist. Assoc. 90 (431) (1995) 909–920.
[23] James Gary Propp, David Bruce Wilson, Exact sampling with coupled Markov chains and applications to statistical mechanics, Random Structures Algorithms 9 (1–2) (1996) 232–252.
[24] Faming Liang, Wing Hung Wong, Real-parameter evolutionary Monte Carlo with applications to Bayesian mixture models, J. Am. Stat. Soc. 96 (454) (2001) 653–666.
[25] Jeff J. Shi, Daniel L. Rabosky, Speciation dynamics during the global radiation of extant bats, Evolution: Int. J. Org. Evolution 69 (6) (2015) 1528–1545.
[26] Wendy K. Tam Cho, Yan Y. Liu, Massively parallel evolutionary computation for empowering electoral reform: Quantifying Gerrymandering via multi-objective optimization and statistical analysis, in: SC17: The International Conference for High Performance Computing, Networking, Storage and Analysis, 2017.