A New Automated Redistricting Simulator Using Markov Chain Monte Carlo

Kosuke Imai

Princeton University

Keynote Talk at
The Fourth Annual Joint Conference of the Upstate Chapters
of the American Statistical Association
April 11, 2015

Joint work with Benjamin Fifield, Michael Higgins, and Alexander Tarr

Motivation

- Redistricting as a central element of representative democracy
- Redistricting may affect:
  - Representation (Gelman and King 1994, McCarty *et. al* 2009)
  - Turnout (Gay 2001, Baretto 2004)
  - Incumbency advantage (Abramowitz *et. al* 2006)
- Substantive researchers simulate redistricting plans to:
  - detect gerrymandering
  - assess impact of constraints (e.g., population, compactness, race)
- Many optimization methods but surprisingly few simulation methods
- Standard algorithm has no theoretical justification
- Need a simulation method that:
  1. samples uniformly from the true underlying distribution
  2. incorporates common constraints
  3. scales to larger redistricting problems

Overview of the Talk

1. Explain the difficulties of simulating redistricting plans
2. Propose new Markov chain Monte Carlo algorithms
3. Validate the algorithms on a small-scale data example
4. Present empirical analyses for New Hampshire and Mississippi

Characterizing the Distribution of Valid Redistricting Plans

- Scholars want to characterize the *distribution* of redistricting plans under various constraints
- Valid redistricting plans must have:
  - geographically contiguous districts
  - districts with equal population
- Other constraints of interest: compactness, community boundary, etc.
- Naive Approach 1: Enumeration
  - Can't enumerate all plans (too many)
  - Enumerating only valid plans is not trivial
- Naive Approach 2: Random assignment
  - Too few plans will have equal population
  - Too few plans will be contiguous

The Standard Simulation Algorithm

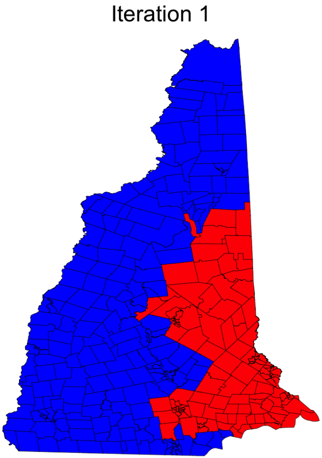
- Random seed-and-grow algorithm (Cirincione *et. al* 2000, Altman & McDonald 2011, Chen & Rodden 2013):
  1. Randomly choose a precinct as a "seed" for each district
  2. Identify precincts adjacent to each seed
  3. Randomly select adjacent precinct to merge with the seed
  4. Repeat steps 2 & 3 until all precincts are assigned
  5. Swap precincts around borders to achieve population parity
- Modify Step 3 to incorporate compactness
- No theoretical properties known
- The resulting sample may not be representative of the population
- Leads to biased inference

The Proposed Automated Redistricting Simulator

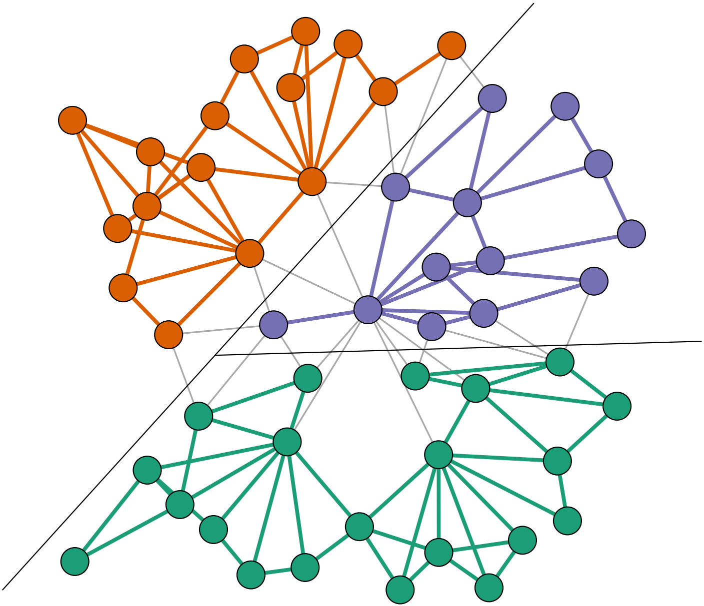
- Independent sampling is difficult
- Markov chain Monte Carlo algorithm
- Can sample uniformly from the target distribution
- Start with a valid plan and then swap precincts in a certain way

The Proposed Automated Redistricting Simulator

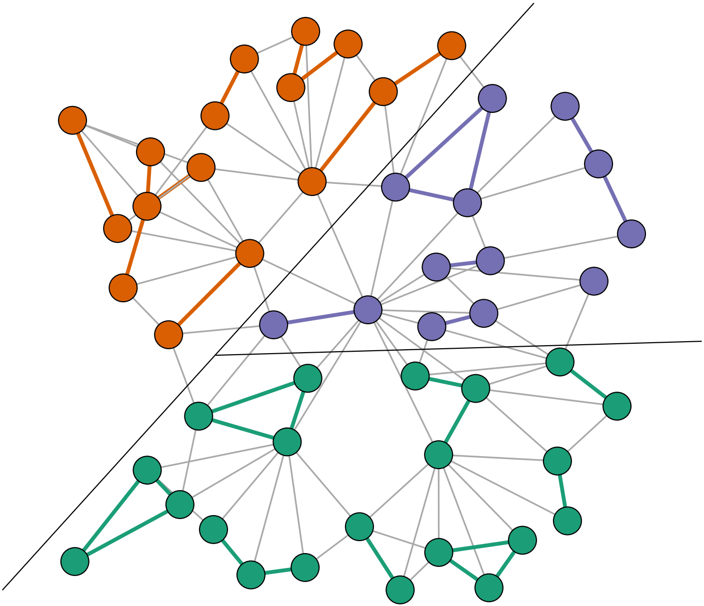
- Independent sampling is difficult
- Markov chain Monte Carlo algorithm
- Can sample uniformly from the target distribution
- Start with a valid plan and then swap precincts in a certain way



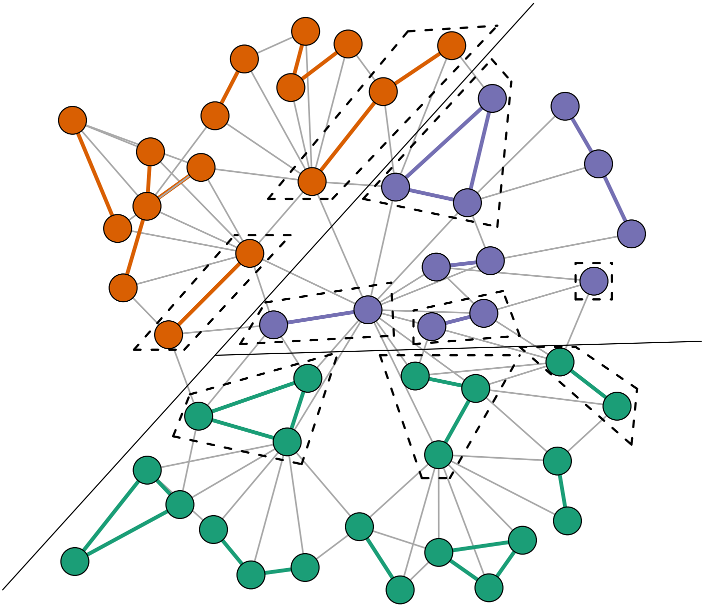
Redistricting as a **Graph-Cut** Problem



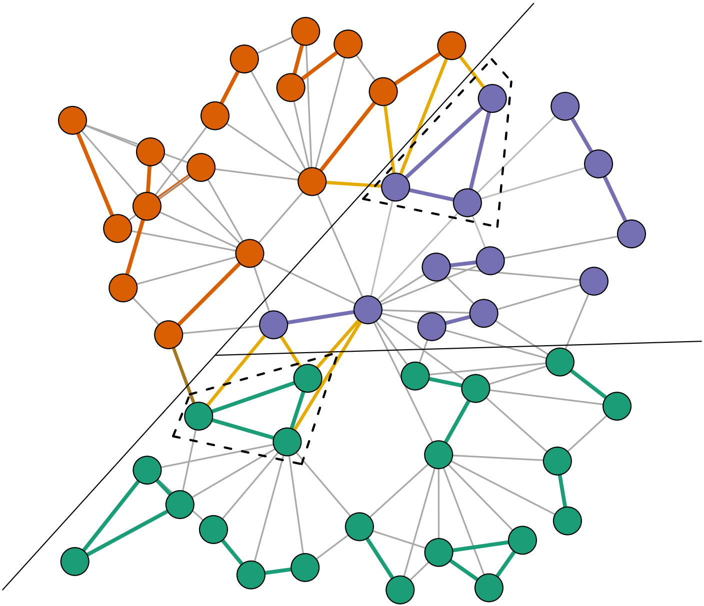
Step 1: Independently "Turn On" Each Edge with Prob. q



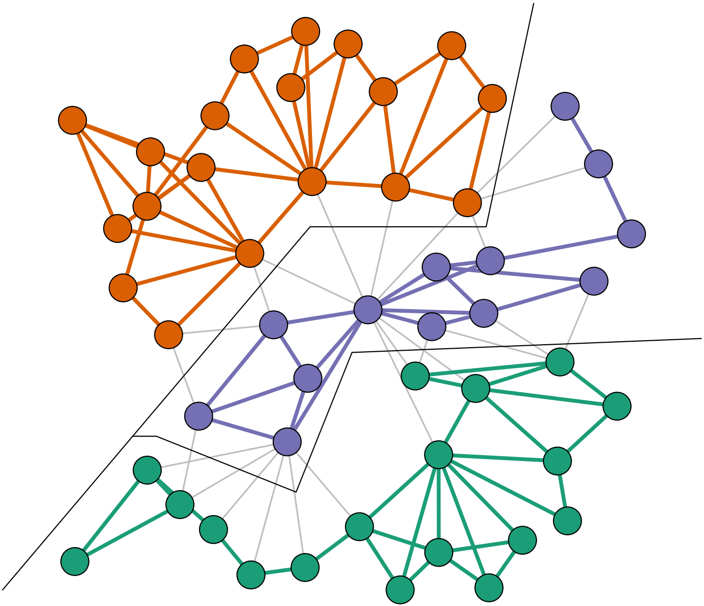
Step 2: Gather Connected Components on Boundaries



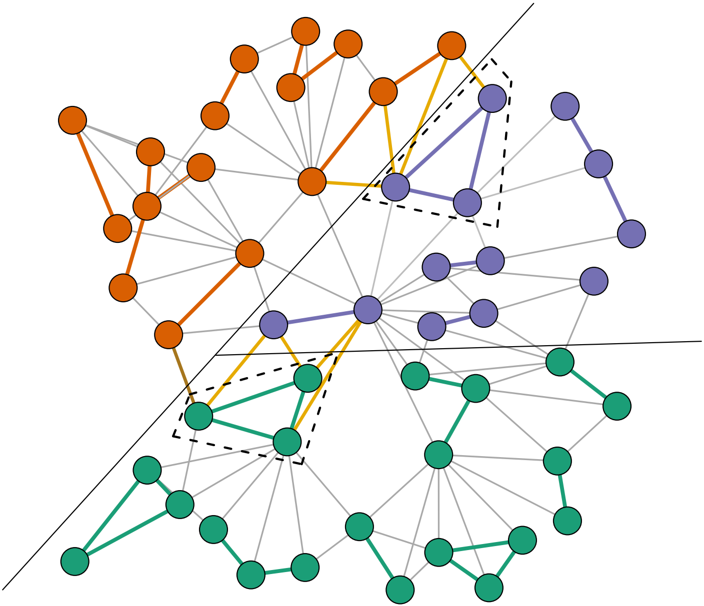
Step 3: Select Subsets of Components and Propose Swaps



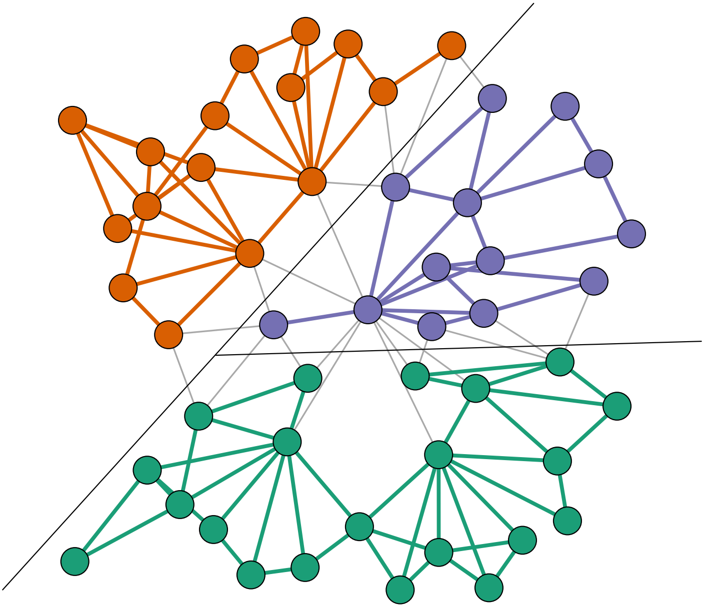
Step 4: Accept or Reject the Proposal



Step 4: Accept or Reject the Proposal



Step 4: Accept or Reject the Proposal



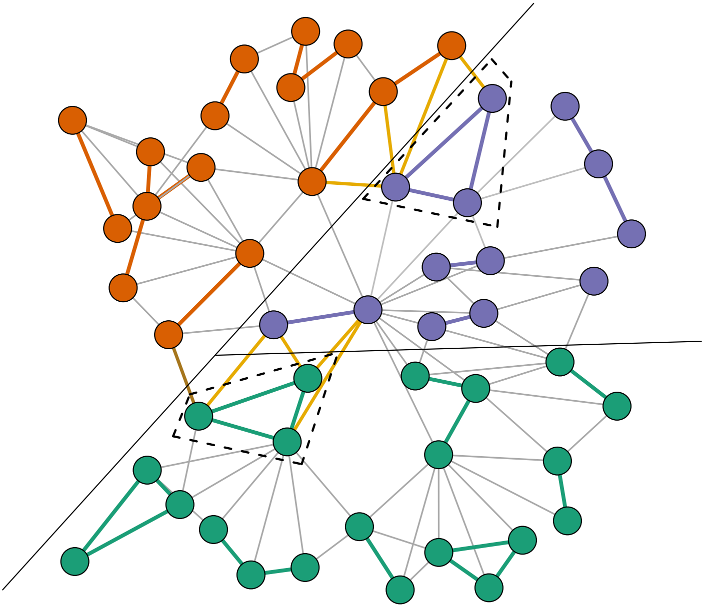
The Theoretical Property of the Algorithm

- We prove that the algorithm samples *uniformly* from the population of all valid redistricting plans
- An extension of the Swendsen-Wang algorithm (Barbu & Zhu, 2005)
- Metropolis-Hastings move from plan $\mathbf{v} \to \mathbf{v}^*$ with acceptance prob.

$$\alpha(\mathbf{v} \to \mathbf{v}^*) \ = \ \min\left(1, \ (1-q)^{|B(C^*,\mathbf{v})|-|B(C^*,\mathbf{v}^*)|}\right)$$

- $|B(C^*, \mathbf{v})|$: # of edges between connected component $C' \in C^*$ and its assigned district in redistricting plan $\mathbf{v}$ $\rightsquigarrow$ Easy to calculate

The Theoretical Property of the Algorithm



Incorporating a Population Constraint

- Want to sample plans where

$$\left|\frac{p_k}{\bar{p}} - 1\right| \leq \epsilon$$

where p_k is population of district k, $\bar{p}$ is average district population, ϵ is strength of constraint

- Strategy 1: Only propose "valid" swaps $\rightsquigarrow$ slow mixing
- Strategy 2: Oversample certain plans and then reweight
  1. Sample from target distribution f rather than the uniform distribution:

$$f(\mathbf{v}) \;\propto\; g(\mathbf{v}) \;=\; \exp\left(-\beta \sum_{V_k \in \mathbf{v}} \psi(V_k)\right)$$

     where $\beta \geq 0$ and $\psi(V_k)$ is deviation from parity for district V_k
  2. Acceptance probability is still easy to calculate,

$$\alpha(\mathbf{v} \to \mathbf{v}^*) \;=\; \min\left(1,\; \frac{g(\mathbf{v}^*)}{g(\mathbf{v})} \cdot (1-q)^{|B(C^*,\mathbf{v})| - |B(C^*,\mathbf{v}^*)|}\right)$$

  3. Discard invalid plans and reweight the rest by $1/g(\mathbf{v})$

Additional Constraints

1. Compactness (Fryer and Holden 2011):

$$\psi(V_k) \propto \sum_{i,j \in V_k, i<j} p_i p_j d_{ij}^2$$

where d_{ij} is the distance between precincts i, j

2. Similarity to the adapted plan:

$$\psi(V_k) = \left| \frac{r_k}{r_k^*} - 1 \right|$$

where r_k (r_k^*) is the # of precincts in V_k (V_k of the adapted plan)

- Any criteria where constraint can be evaluated at each district

Improving the Mixing of the Algorithm

- Single iteration of the proposed algorithm runs very quickly
- But, like any MCMC algorithm, convergence may take a long time

1. Swapping multiple connected components
    - more effective than increasing q
    - but still leads to low acceptance ratio
2. Simulated tempering (Geyer and Thompson, 1995)
    - Lower and raise the "temperature" parameter β as part of MCMC
    - Explores low temperature space before visiting high temperature space
3. Parallel tempering (Geyer 1991)
    - Run multiple chains of the algorithm with different temperatures
    - Use the Metropolis criterion to swap temperatures with adjacent chains

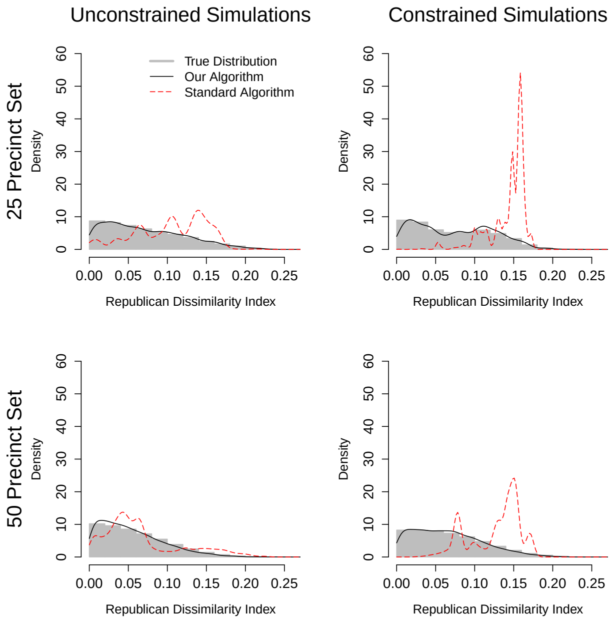
A Small-Scale Validation Study

- Evaluate algorithms when all valid plans can be enumerated

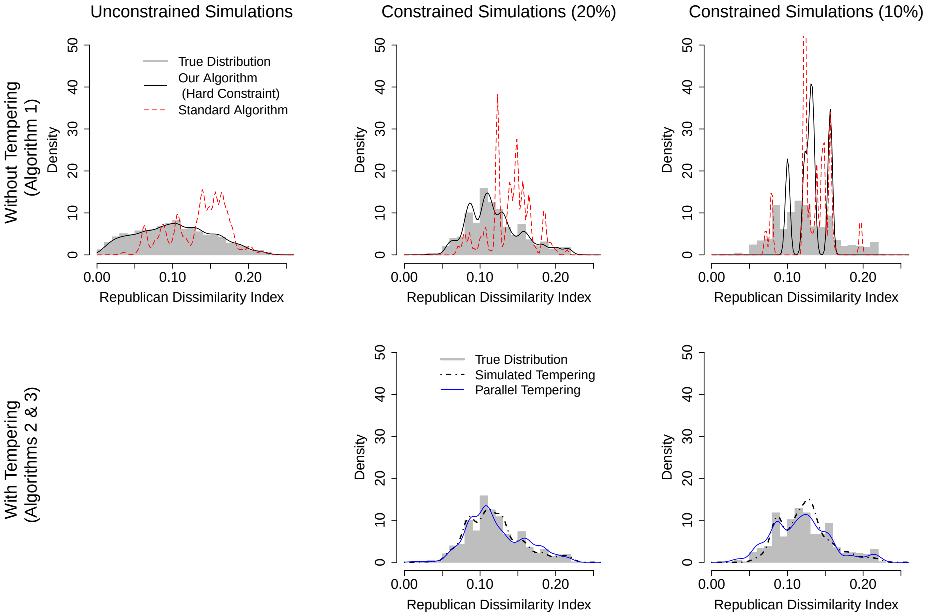
- # of precincts: 25 and 50
- # of districts: 2 and 3 for the 25 set, and 2 for the 50 set
- With and without a "hard" population constraint of 20% within parity
- Also, consider simulated and parallel tempering

- Comparison with the "random seed-and-grow" algorithm via the BARD package (Altman & McDonald 2011)
- 10,000 draws for each algorithm

Our Algorithm vs. Standard Algorithm

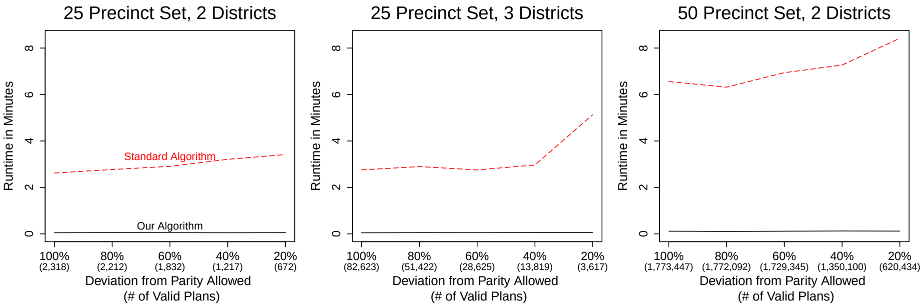


Simulated and Parallel Tempering



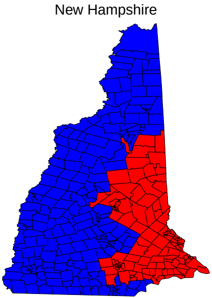
Runtime Comparison

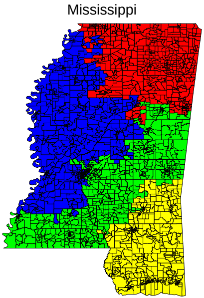
- Run each algorithm for 10,000 simulations under different population constraints



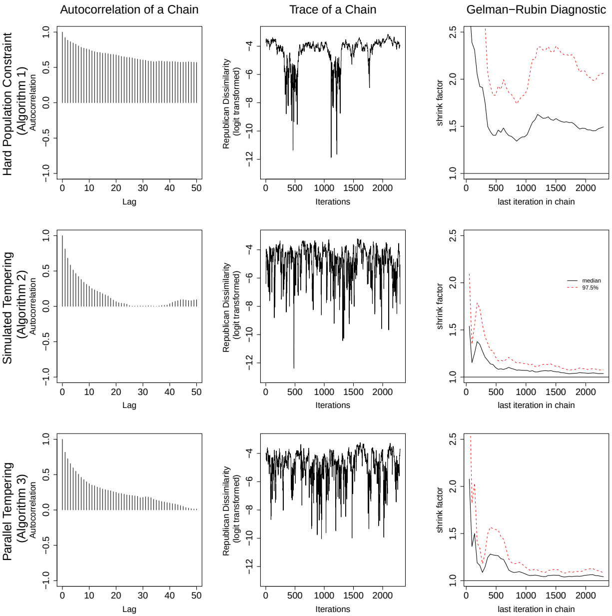
An Empirical Study

- Apply algorithm to state election data:
    1. New Hampshire: 2 congressional districts, 327 precincts
    2. Mississippi: 4 congressional districts, 1,969 precincts
- Convergence diagnostics:
    1. Autocorrelation
    2. Trace plot
    3. Gelman-Rubin multiple chain diagnostic

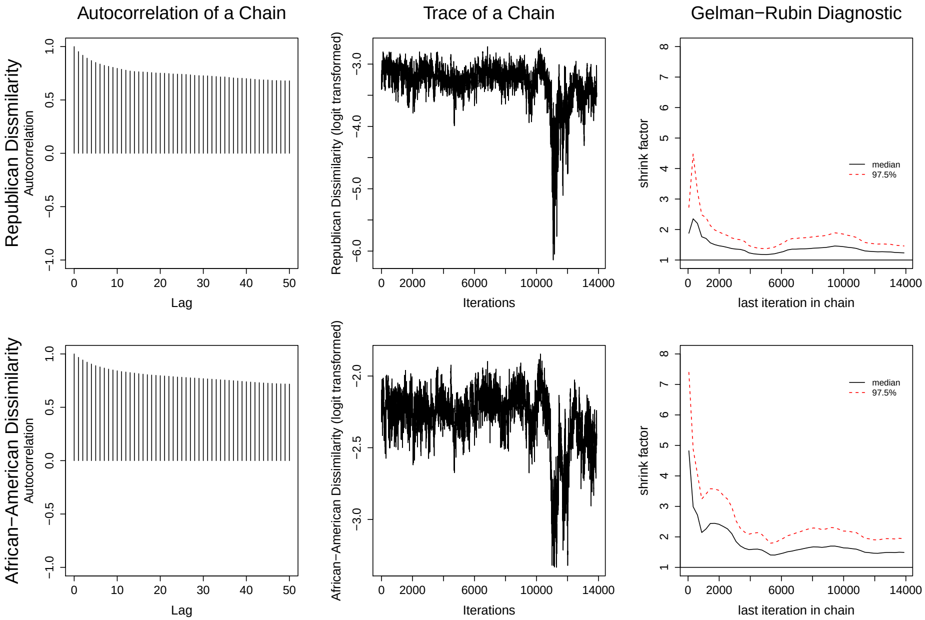




New Hampshire: Simulated and Parallel Tempering Works



Missisippi: Parallel Tempering, More Challenging Case



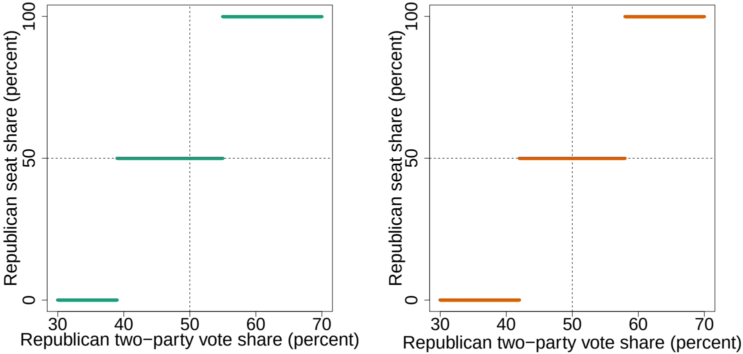
Redistricting Plans that are Similar to the Adapted Plan

- Question: How does the partisan bias of the adapted plan compare with that of similar plans?

- Two measures:
  1. Number of Republican winners under each plan
  2. Partisan bias (Gelman & King, 1994): Deviation from partisan symmetry under each plan

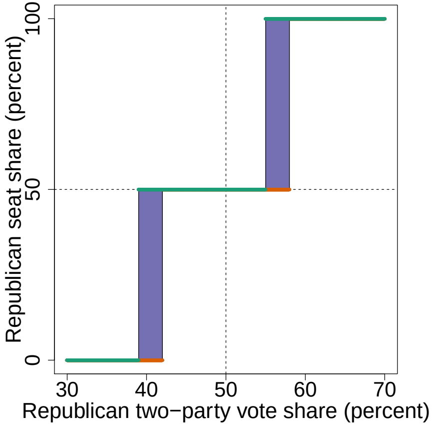
Evaluating Partisan Bias

- Empirical and Symmetric Seats-Votes Curves

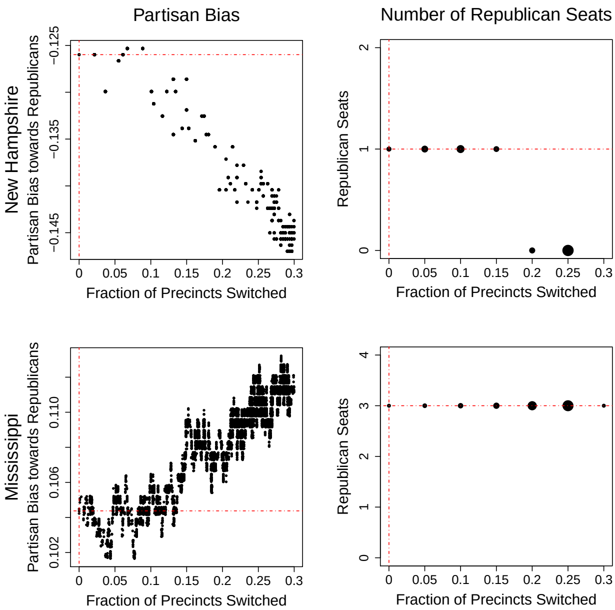


Evaluating Partisan Bias

- Absolute Deviation from Partisan Symmetry



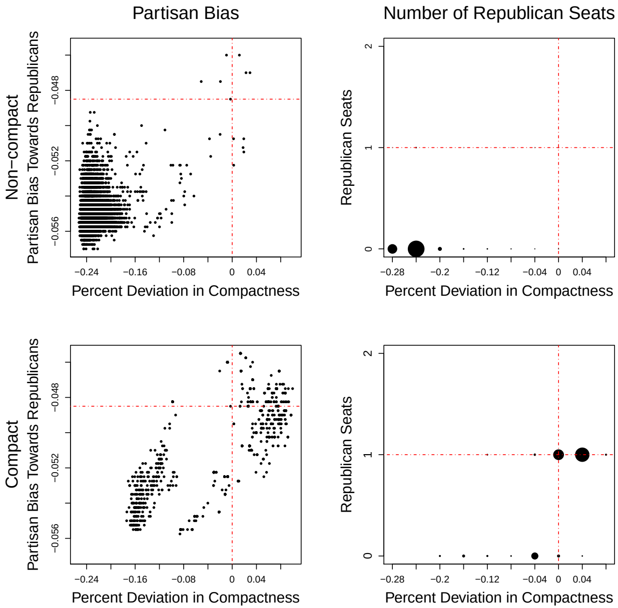
Partisan Implications of "Local Exploration"



Assessing the Partisan Effects of Compactness

- Question: How does a compactness standard limit partisan manipulation of redistricting?
- Two measures:
  1. Number of Republican winners under each plan
  2. Deviation from partisan symmetry under each plan
- Two simulations (10 chains, 50,000 iterations each):
  - Compare without compactness constraint to with compactness constraint with simulated tempering
  - When simulated tempering, inverse reweighting for uniform sampling

Compactness and Partisanship: New Hampshire



Concluding Remarks

- Scholars use simulations to characterize the distribution of redistricting plans
- Many optimization algorithms but very few simulation methods
- No theoretical guarantee for most common algorithms
- We propose a new MCMC algorithm that has:
  - good theoretical properties
  - superior speed
  - better performance in validation and empirical studies
- Future research:
  - Continue to improve the algorithm for large-scale redistricting problems
  - Derive methods for inference to uncover factors driving redistricting

References

1. Paper: available at http://imai.princeton.edu/research/redist.html
2. R package: available at https://github.com/redistricting/redist
3. Comments and suggestions: send them to kimai@princeton.edu