# Metropolis-Hastings

Eric A. Autry

# Winery Storage Problem

# Winery Storage Problem

# Winery Storage Problem: Formal Definition

We are given the following:

- The predicted volume of each type of wine.
- A set of current storage tanks.
- A set of possible storage tank sizes to purchase.
- The cost of each possible storage tank size.

Our goal is:

- Find a set of storage tanks to purchase that properly fits the wine at a minimal cost.

# Markov Chain Monte Carlo (MCMC) Methods

Idea:

- Randomly pick a starting solution.
- Randomly alter that solution according to some specified sampling algorithm and acquire a new solution.
  - (randomly add or subtract a single storage tank)
- Repeat this process to generate a chain of solutions.
- If we track the best solution we've seen so far, maybe we can find a good solution.

Is it possible to improve our odds of finding good solutions?

# MCMC Sampling

Note that MCMC is sampling solutions from the set of all possible solutions (the ‘state space’).

But according to what distribution?

That depends on the actual sampling algorithm we use to acquire the next solution in the chain.

- ‘Randomly add or subtract a single storage tank’ will uniformly sample the state space.

## Imposing a Measure

We want it to be more likely that we sample good solutions.

Idea:

- Let's change the distribution we are sampling from!
- Instead, we will try to sample from a distribution where good solutions are more probable.
- We say we are 'imposing a measure on our state space.'

We need to answer two questions:

- What measure should we use?
- How do we make MCMC sample from this new measure?

# Minimizing Energy

A commonly used measure comes from statistical mechanics:

$$\mu(x) = \frac{1}{z} e^{-\beta J(x)}$$

- $x$ is the current state, i.e., the current solution.
- $\mu$ is the measure.
- $z$ is a constant of proportionality (will cancel out later).
- $\beta$ is a parameter to adjust the strength of the biasing.
- $J(x)$ is the 'energy' of the state.

Note: this distribution makes low energy states more probable.

- For the winery problem, set $J(x)$ to be the cost of the tank set so we minimize cost!

# Sampling with Acceptance/Rejection

How do we make MCMC sample from this new measure?

Idea: allow ourselves to reject a proposed move.

- Randomly pick a starting solution.
- Randomly alter that solution according to some specified sampling algorithm and **propose** a new solution.
  - (randomly add or subtract a single storage tank)
- Calculate an acceptance probability.
  - If we accept, move to the new solution and repeat.
  - If we reject, stay at the current solution and try again.
- Repeat this process to generate a chain of solutions.

# Stationary Distribution

- $\mu(x)$ is the desired measure we would like to impose.
- $Q(x \to y)$ is the probability that our sampling algorithm is currently at state $x$ and proposes a move to state $y$.

Question: does the MCMC process with rejections converge to a stationary distribution?

- Existence: a sufficient condition for convergence to $\mu(x)$ is:

$$\mu(x)Q(x \to y) = \mu(y)Q(y \to x).$$

  - Flux from $x$ to $y$ equals flux from $y$ to $x$.
  - This is called *detailed balance* or *microscopic reversibility*.

- Uniqueness: guaranteed by ergodicity.
  - No state repeats at regular intervals.
  - Each state is expected to occur in finite time.

# Acceptance Probability

Problem: what if detailed balance doesn't hold for our $Q(x \to y)$?

What if $\mu(x)Q(x \to y) > \mu(y)Q(y \to x)$ for some states $x$ and $y$?

We can use our rejections!

- Say we accept a move from $x$ to $y$ with probability $A(x \to y)$.
- Then the flux from $x$ to $y$ is actually $A(x \to y)\mu(x)Q(x \to y)$.
- To balance fluxes, set $A(x \to y)$ so that detailed balance holds:

$$A(x \to y)\mu(x)Q(x \to y) = \mu(y)Q(y \to x).$$

- The acceptance probability should be:

$$A(x \to y) = \min\left\{\frac{\mu(y)\,Q(y \to x)}{\mu(x)\,Q(x \to y)},\ 1\right\}.$$

## Metropolis-Hastings

- Randomly pick a starting solution $x$.
- Randomly alter that solution according to some specified sampling algorithm and **propose** a new solution $y$.
  - (randomly add or subtract a single storage tank)
- Calculate the acceptance probability:

$$A(x \to y) = \min\left\{ \frac{\mu(y)\, Q(y \to x)}{\mu(x)\, Q(x \to y)},\ 1 \right\}.$$

  - If we accept, move to the new solution $y$ and repeat.
  - If we reject, stay at the current solution $x$ and try again.
- Repeat this process to generate a chain of solutions.

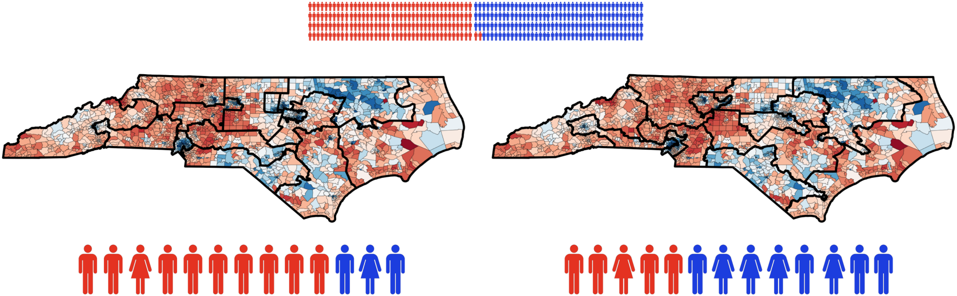
# Winery Results

2011

# Winery Results

2016

# The Gerrymandering Problem



# The Gerrymander

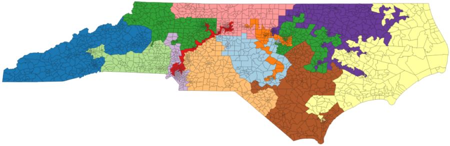
Gerrymandering is a **manipulation** of district boundaries to favor one party or class.

- We think of it as **changing** the outcome of an election:
  "gerrymandered the results."
- Implicit in this discussion:
  a comparison to what should/would have happened without political agendas.

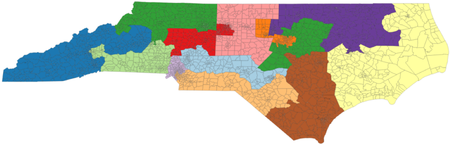

Boston Gazette, 1812

# The Gerrymandering Problem in NC

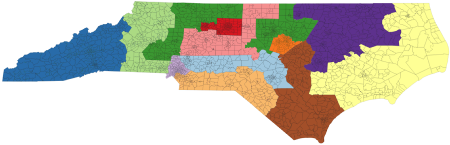
NC 2012



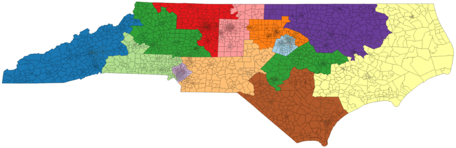
NC 2016



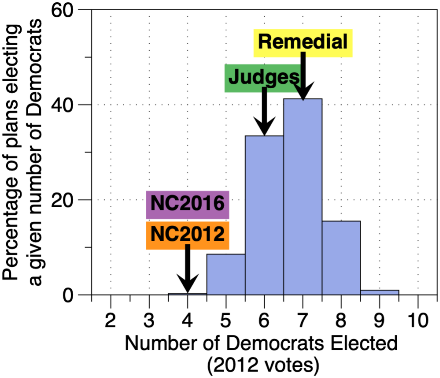
NC 2020 Remedial Plan



Bipartisan Panel of Judges



# The Ensemble Method

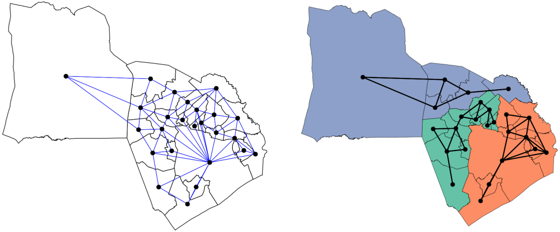



- Create an ensemble of neutral, fair redistricting plans.
- Compare this ensemble with a potential gerrymandered plan.
- Identify if the potential gerrymander is an outlier.

# The Gerrymandering Problem: Formal Definition

- We have $n$ *contiguous* districts to draw.
- We have a set of precincts $p_i \in P$ to be assigned to districts.
- A solution is a mapping $x : P \to \{1, 2, \ldots, n\}$.

An equivalent formulation:

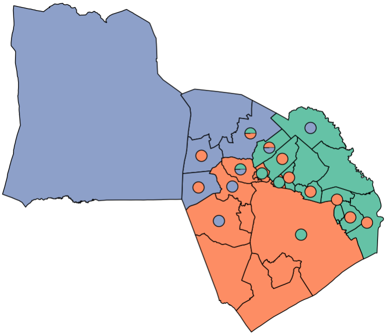


- Let each precinct be a vertex of a graph.
- Edges will represent physical adjacency.
- Contiguous districts are represented by connected subgraphs.
- A solution is a partition of the original graph into $n$ subgraphs.

# Single Node Flip

Idea:

- Randomly flip one boundary precinct to an adjacent district.

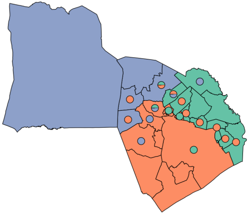


- This will sample plans according to the length of the district borders (as measured by number of adjacent precincts).
- We will want to use Metropolis-Hastings to sample from a politically relevant measure.

# Metropolizing Single Node Flip

1. Assume you are given a function that returns a list of all boundary precincts, how would you calculate the Metropolis-Hastings acceptance probability?

$$A(x \to y) = \min\left\{\frac{\mu(y)\, Q(y \to x)}{\mu(x)\, Q(x \to y)},\ 1\right\}.$$



2. What measure would you sample from?
   - Common preferences include:
     - Near-equal population across districts.
     - Highly compact districts as measured by $(area)/(perimeter^2)$.
     - Preserving municipalities or communities on interest, i.e., splitting these regions as little as possible.

# Metropolizing Single Node Flip: Ratio of $Q$s

1. How would you calculate the Metropolis-Hastings acceptance probability? Count the number of boundary precincts!

When proposing a move, we randomly select a single boundary precinct from state $x$ to flip.

- What if that precinct borders multiple districts?
- We randomly choose one of its neighboring districts to flip to.

We need the probability of making this pair of choices, so define:

- $B(x)$ is the set of boundary precincts of $x$,
- $N(p,x)$ is the number of neighboring districts of precinct $p$ in plan $x$.

This gives us:

- $1/|B(x)|$ is the probability that we chose precinct $p$ to flip,
- $1/N(p,x)$ is the probability that we flipped $p$ to the district that produced plan $y$ instead of some other district.

## Metropolizing Single Node Flip: Ratio of $Q$s

- $1/|B(x)|$ is the probability that we chose precinct $p$ to flip,
- $1/N(p,x)$ is the probability that we flipped $p$ to the district that produced plan $y$ instead of some other district.

This means that the probability that we chose precinct $p$ to flip, and then made the flip that produced plan $y$ is

$$Q(x \to y) = \frac{1}{|B(x)|} \cdot \frac{1}{N(p,x)}.$$

We can make a similar argument for going from plan $y$ to plan $x$, giving

$$Q(x \to y) = \frac{1}{|B(x)| \cdot N(p,x)}, \quad Q(y \to x) = \frac{1}{|B(y)| \cdot N(p,y)}$$

$$\Rightarrow \qquad \frac{Q(y \to x)}{Q(x \to y)} = \frac{|B(x)| \cdot N(p,x)}{|B(y)| \cdot N(p,y)}.$$

# Aside: What if we don't Metropolize the chain?

If we set $\mu = 1$ and sample with acceptance probability

$$A(x \to y) = \min\left\{\frac{Q(y \to x)}{Q(x \to y)}, 1\right\} = \min\left\{\frac{|B(x)| \cdot N(p,x)}{|B(y)| \cdot N(p,y)}, 1\right\},$$

we are uniformly sampling from the set of all districting plans.

Note that this acceptance probability is working to increase the odds of districts with fewer boundary precincts: $|B(x)|/|B(y)|$.

What would happen if we just used MCMC sampling **without** applying the rejections of Metropolis-Hastings?

- Without the acceptance probability, we would see decreased odds for districts with fewer boundary precincts.
- So we would be biased towards precincts with more border precincts, which is not politically relevant!

# Metropolizing Single Node Flip: Measure $\mu$

Recall that the measure $\mu$ defines the distribution we are sampling from: it controls what we prioritize and what we avoid.

The choice of $\mu$ should be policy driven, and is how we mathematically represent what is means for a districting plan to be 'fair.'

In general, there may be many policy decisions we want to consider simultaneously. To do this, we define an energy for each, then define our total energy as a weighted sum of these policy-driven energies.

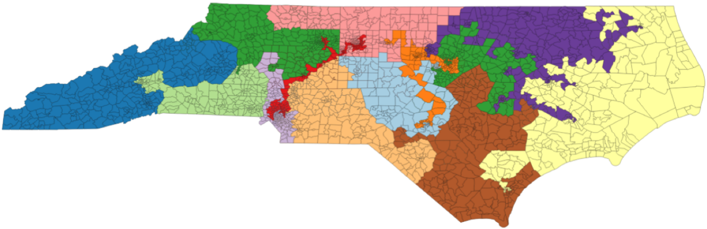
- Near-equal population across districts is required by law. Population energies to prioritize this could look like:
  - $J(x) =$ the max difference in population between any two districts,
  - $J(x) =$ the sum of all pairwise differences in population,
  - $J(x) =$ the sum of the difference between every districts' population and the 'ideal' average population.

# Metropolizing Single Node Flip: Compactness

Another common policy concern is the compactness of the districts:

- Most people agree that compact districts (blobs not snakes) look more fair.
- After all, the original Gerrymander had a very strange shape that arose because it was not fairly drawn.
- And many historical Gerrymandering attempts look similarly ‘snake-y’.

NC 2012



# Metropolizing Single Node Flip: Compactness

- A common compactness energy to use is called the Polsby-Popper score.
- This score is defined as $4\pi A/P^2$, where $A$ is the area and $P$ is the perimeter of the district.
- *Maximizing* the Polsby-Popper score will lead to more compact districts.
- So the compactness energy $J(x)$ is often defined as the reciprocal of the Polsby-Popper score (or just the isoperimetric ratio $P^2/A$).

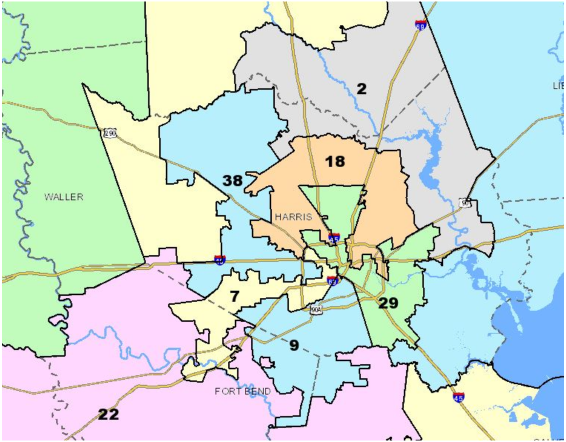
# Metropolizing Single Node Flip: Municipalities

Preservation of Municipalities or Communities of Interest is another popular policy concern:

- People who live together in a community likely share similar concerns and problems.
- So the idea is that they should therefore also share government representation.
- A common gerrymandering technique involving cities is called cracking and packing:
  - Cracking: splitting the city's population among a large number of districts to dilute its influence on any of them.
  - Packing: concentrating a city's population in the fewest number of districts possible to ensure it cannot influence other districts.
  - These are often used together to minimize the influence of a city.

# Metropolizing Single Node Flip: Municipalities

A classic example of packing/cracking is Houston, Texas:

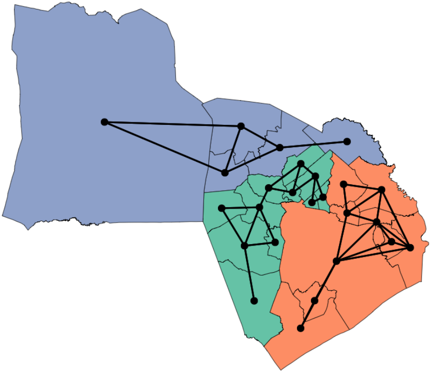



How can we measure community preservation as an energy?

- That is my current research project!

# Problems with Mixing

Problem: how many flips to get large changes in state space?



Single Node Flip tends to have very slow mixing, i.e., it will converge to a representative distribution very slowly.

# Mixing and Convergence of the MCMC Chain

How will we know if we have successfully generated a representative sample of our desired distribution?

- If there exists a unique stationary distribution, then we know that any of our MCMC chains will eventually produce a representative sample.

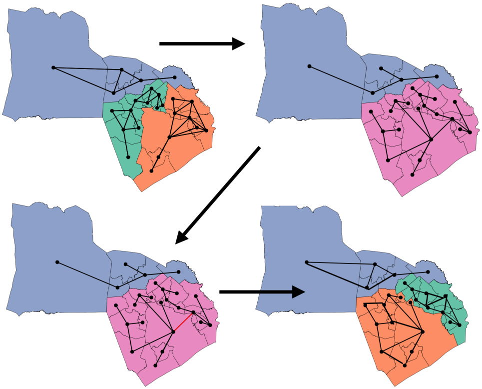
But how long will that take and how will we know if they have?

- A common strategy is to run numerous chains from different starting plans.
- At the end of some fixed number of iterations, you can compare the resulting distributions to see how well they align.
- This can tell you if the chain has converged, and if you do this at regular intervals, you can show the rate of convergence.

Of course, this wont work if there is no unique stationary distribution (non-ergodic problems where some states cannot be reached).

# Forest Recombination

Forest ReCom algorithm (proposed by DeFord, et. al.):



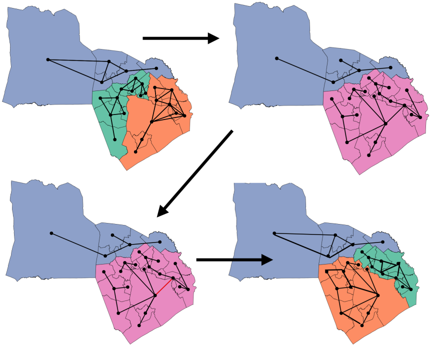
- Pick a pair of adjacent subgraphs to merge.
- Draw a spanning tree on the merged graph (Wilson's algorithm).
- Find permissible cuts (e.g., those that obey a population constraint).
- Cut the merged tree into two connected trees.
- These two new trees define two new subgraphs.

Pro: This can give very fast mixing due to large moves in state space.

# Calculating the Acceptance Probability

Con: We would like to use Metropolis-Hastings, but how do we calculate the acceptance probability?

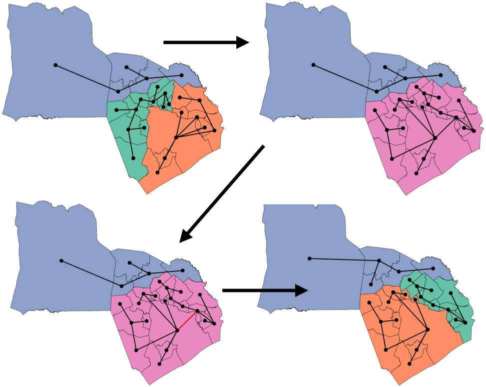
- Consider all sets of spanning tree pairs for the two new districts.
- For each pair, consider all possible edges crossing the cut.
- Compute the probability of choosing those two trees and that edge...
- Multiplied by the probability of cutting that edge.



This is computationally intractable...

# Metropolized Forest Recombination

To use Metropolis-Hastings, a small modification of the algorithm:



- Define the state space as spanning trees instead of graph partitions.
- So now a solution is a set of spanning trees representing the districts.

This gives the computationally tractable Metropolized Forest Recombination algorithm.