Geometry & Data: Algorithmic Approaches to Redistricting

Justin Solomon
MIT EECS
Preliminary.
Welcome!

The MIT Geometric Data Processing Group studies geometric problems in computer graphics, computer vision, machine learning, and other disciplines.

Our team includes students and researchers spanning a variety of disciplines, from theoretical mathematics to applications in engineering and software development. We enthusiastically welcome collaborators and staff at all levels and encourage interested parties to contact us with ideas, challenging problems, and avenues for joint research.

News

Seeking PhD students!
Please apply online by December 15, 2016. Interested students are encouraged to contact Justin.

New website
Please contact Justin with comments or edits.

http://gdp.csail.mit.edu
Geometric Data Processing

- Theory of geometry
  - Differential geometry
  - Optimal transport
  - Geometric PDE
  - Spectral geometry
  - Metric geometry

- Applications
  - Graphics
  - Vision
  - Imaging
  - Learning
  - Optimization
  - Redistricting?
This computer programmer solved gerrymandering in his spare time

By Christopher Ingraham  June 3, 2014

Yesterday, I asked readers how they felt about setting up independent commissions to handle redistricting in each state. Commenter Mitch Beales wrote: "It seems to me that an 'independent panel' is about as likely as politicians redistricting themselves out of office. This is the twenty-first century. How hard can it be to create an algorithm to do the job of an entity that has no power?"
THE COMPUTATIONAL COMPLEXITY OF AUTOMATED REDISTRICTING: IS AUTOMATION THE ANSWER?

MICAH ALTMAN

There is only one way to do reapportionment—feed into the computer all the factors except political registration.

—Ronald Reagan

The rapid advances in computer technology and education during the last two decades make it relatively simple to draw contiguous districts of equal population [and] at the same time to further whatever secondary goals the State has.

—Justice William Brennan

I. REDISTRICTING AND COMPUTERS

Ronald Reagan and Justice Brennan have both suggested that computers can remove the controversy and politics from redistricting. In fact proponents of automation claim that the “optimal” districting plan can be found by using specified values. The Supreme Court’s admissions by Justice Lewis Powell in 1986 are an admission by public bodies of the need to address such mechanization. The history of American redistricting and compactness in two recent redistricting cases (Karcher v. Daggett and United States v. Texas) demonstrates that automation is...
D. **Redistricting is a Computationally Hard Problem**

Redistricting is deeply connected to mathematical partitioning problems. Many researchers in computer science have examined partition problems and have reached some conclusions about their computational complexity. The redistricting problem in general, and even many simpler redistricting sub-problems, are likely to be intractable.
What is the role of computation in political redistricting?
Partners in Redistricting

10^9 computations/second
No legal understanding
No sympathy

?? computations/second
Strong legal understanding
Potentially sympathetic
Clearly easy:
• Visualizing districting plans
• Data collection

Clearly difficult:
• Extracting optimal plans

Huge gray area:
• Improving plans
• Evaluating compactness
• Sampling possible plans
Computational Desiderata

- **Stability**
  Does not change under perturbation or poor measurement.

- **Efficiency**
  Computers can solve the problem in a reasonable amount of time.
Any measure of compactness or fairness that we cannot compute stably and efficiently is not useful.
Pessimistic Approach

435 congressional districts

A lot of shapes!

Image courtesy Wikipedia
Pushing the Limits

11,155,486 census blocks (2010)

\[ e^{11,155,486} \approx 7.984 \times 10^{3,358,135} \]

\[ 11,155,486! \approx 1.435 \times 10^{7,377,397} \]

Comparison: The sun will swallow the earth in \( \sim 5,000,000,000 = 5 \times 10^9 \) years.
Taxonomy of benefits and limitations of computational redistricting tasks.

**Stability:** Micro- and Macro-Scale

**Tractability:** Comparing vs. Sampling vs. Optimizing

**Open challenges**
1. Stability
Stable: Small changes in the input yield small changes in the output
Where will the ball go?
Intuition

Where will the ball go?

Unstable
Point-in-polygon predicate
Micro-Scale Stability: Example

Point-in-polygon predicate
Micro-Scale Stability: Example

Point-in-polygon predicate

Outside!
Micro-Scale Stability: Example

6th Congressional District

Source: maps4news.com/©HERE

Inside!
Micro-Scale Stability: Example
General Principle

Computational geometry is all about border cases.
More Border Cases

Nonconvex shapes and slivers
More Border Cases

Tracts in Texas

Axis alignment
More Border Cases

Disconnected regions

MA tract 980101
More Border Cases

Topological holes

Tracts in Texas
More Border Cases

Census tracts: Unequal population

Discretization artifacts

CA tract 187
“In geometry, we don't compute numbers but structures: convex hulls, triangulations, etc. In building these structures, the underlying algorithms ask questions like ‘is a point to the left, to the right, or on the line through two other points?’ Such questions have no answers that are ‘slightly off.’ …

[I]t's primarily a combinatorial problem, not a numerical one.”

Robust Geometry
Confidence bounds
Robust Geometry

Optimal transport

Image from “Parallel Streaming Wasserstein Barycenters” with M. Staib, S. Claici, S. Jegelka
Principle #1:
Geometric measurements in redistricting software must be provably robust.
Unbiased rounding rules
Macro-Scale Stability: Example

$$\frac{400\pi A}{P^2} \leq 100\%$$


Macro-Scale Stability: Example

Resolution: 1:20,000,000
Perimeter: 10.951
Area: 1.233
$400\pi A/p^2$: 12.92%

Resolution: 1:5,000,000
Perimeter: 14.328
Area: 1.188
$400\pi A/p^2$: 7.27%

Resolution: 1:500,000
Perimeter: 18.266
Area: 1.162
$400\pi A/p^2$: 4.38%

Example courtesy Mira Bernstein and Assaf Bar-Natan

Maryland district 1
Another Example: Alignment

District boundary
Census block boundary

Data from US Census
Another Example: Adjacency of census units
Another Example: Adjacency

District adjacency: Ohio 2011

Connect?
Sources of Large-Scale Instability

- Map projections
- Inaccurate measurement equipment
- Temporal change
- Map resolution
- Security limitations on map data
- Political control of data-gathering
Incredibly Stable Measurement

\[ f(x) \equiv 0 \]
Principle #2: Fairness measures must be stable and informative.
Reasonable Measure of Stability

Small change in districting plan

\[ f(\text{old}) \approx f(\text{new}) \]
District Similarity Measure

\[ d(c_1, c_2) := \inf_{\alpha, \beta} \max_{t \in [0,1]} \left\{ \| c_1(\alpha(t)) - c_2(\beta(t)) \| \right\} \]

Example: Fréchet distance
“Isometry-Aware Preconditioning for Mesh Parameterization”
with S. Claici (MIT), S. Schaefer (TAMU)

Example 2: Killing distance

Killing Energy
\((K, \alpha)\)-Hölder Property

\[ |F(c_1) - F(c_2)| \leq K \cdot d_{\text{Fréchet}}(c_1, c_2)^\alpha \]

Small change in districting plan

Small change in fairness measure
2. Tractability
Tractable [trak-tuh-buhl]: Accomplishable in a reasonable amount of time using computing equipment.
Sad Fact

Some problems are inherently difficult or even impossible to solve on a computer.
Complexity Theory

Number of census blocks vs. CPU time

- My code
- Impossible
Let’s be clear:

Any software extracting the “best possible” districting plan* also resolves the most famous open problem in computer science.

\[ P \neq NP \]

* under any reasonable metric.
Optimal redistricting is **ridiculously** hard.

**Theorem.** There exists a constant $\varepsilon > 0$ such that it is NP-hard to approximate the Euclidean $k$-means to a factor better than $(1 + \varepsilon)$.

[Awasthi et al. 2015]

**Theorem.** For node-weighted multiway cut, ... it is Unique Games hard to achieve a $(2 - 2/k - \varepsilon)$-approximation for a constant number of terminals $k$.

[Ene et al. 2015]
Principle #3:
We must be extremely careful how we talk about computational redistricting.
What Can We Do?

Analysis and comparison
What Can We Do?

Local optimization
Which Objective Function?

- Isoperimetric ratio?
- Graph curvature?
- Dispersion?
- Equal population?
- Minority representation?
- Efficiency gap?

“Pareto optimality”
What Can We Do?

Screenshot from “Quantifying Gerrymandering” (Duke Data+)
https://services.math.duke.edu/projects/gerrymandering/

Sampling/MCMC
Language of Sampling

Uniform
Samples *evenly* from the space of reasonable plans

Mixing time
Amount of computational work until a random sample is truly from a given distribution
Hierarchy of Tractability

Clearly easy:
- Visualizing districting plans
- Data collection

Clearly difficult:
- Extracting optimal plans

Huge gray area:
- Improving plans
- Evaluating compactness
- Sampling possible plans
Principle #4:
Political redistricting standards must be designed with **tractability** in mind.
Principle #1:
Geometric measurements in redistricting software must be provably robust.

Principle #2:
Fairness measures must be stable and informative.

Principle #3:
We must be extremely careful how we talk about computational redistricting.

Principle #4:
Political redistricting standards must be designed with tractability in mind.
Open problems
How should we communicate the advantages and drawbacks of computational redistricting?
What is the role of machine learning in redistricting software?
How complicated is the energy landscape specifically for political redistricting?
How do we ensure transparency for redistricting software?
Geometry & Data: Algorithmic Approaches to Redistricting

Questions?
tinyurl.com/askgerry

This afternoon: Optimal transport!