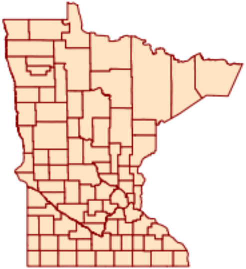
Graph Theory and Gerrymandering: Computationally Assessing Fairness

Garrett J. Kepler
Washington State University
Math 398 Seminar
April 22nd, 2025

Epoch 1: Gary Isaacs

Epoch 1: Gary Isaacs



Epoch 2: Jaime Garcia

Epoch 2: Jaime Garcia

Epoch 3: Andrea Arauza Rivera

Epoch 3: Andrea Arauza Rivera

Epoch 3: Andrea Arauza Rivera (continued)

Epoch 3: Andrea Arauza Rivera (continued)

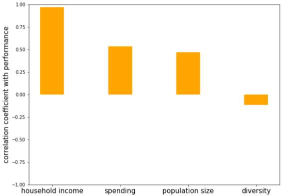


Figure 1: High School Standardized Test Data in RUMBA v1

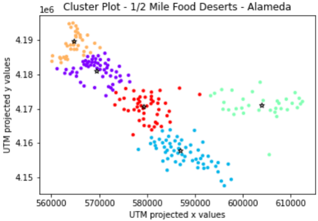
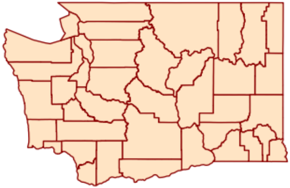


Figure 2: Food Desert Data Clustering in RUMBA v2

Epoch 4: Daryl DeFord

Epoch 4: Daryl DeFord



Epoch 4: Daryl DeFord (continued)

Epoch 4: Daryl DeFord (continued)

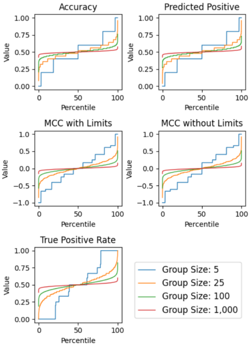


Figure 3: Machine Learning Metrics Bias

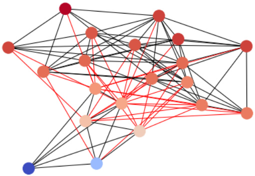


Figure 4: Resulting Spectral Partition from MCMC

Epoch 4: Daryl DeFord (continued)

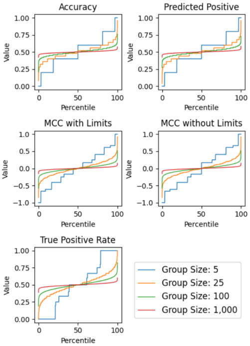


Figure 3: Machine Learning Metrics Bias

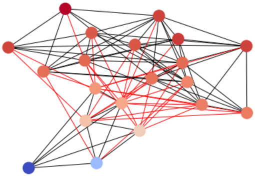


Figure 4: Resulting Spectral Partition from MCMC

The Story of Gerrymandering

The Story of Gerrymandering

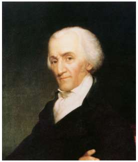


Figure 5: Elbridge Gerry Himself

- Coined in 1812, "gerrymander" arose in the Boston Gazette as a jab at the redrawing of state senate election districts Elbridge Gerry.

The Story of Gerrymandering

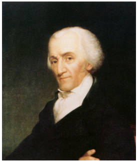


Figure 5: Elbridge Gerry Himself

- Coined in 1812, "gerrymander" arose in the Boston Gazette as a jab at the redrawing of state senate election districts Elbridge Gerry.

Figure 6: His Gerrymandered Salamander

- He blatantly redrew the map so that his part, the Democratic-Republican Party would win the election.

The Story of Gerrymandering

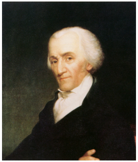


Figure 5: Elbridge Gerry Himself

- Coined in 1812, "gerrymander" arose in the Boston Gazette as a jab at the redrawing of state senate election districts Elbridge Gerry.

Figure 6: His Gerrymandered Salamander

- He blatantly redrew the map so that his part, the Democratic-Republican Party would win the election.

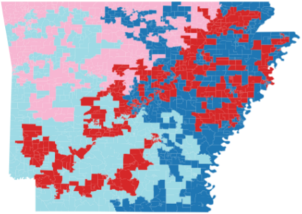
The Story of Gerrymandering

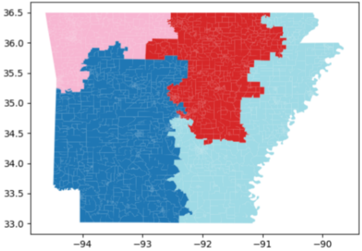
The Story of Gerrymandering

- So you want to be a gerrymanderer. . .

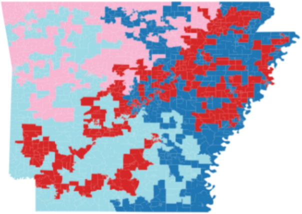
How Do I Know You Gerrymandered?

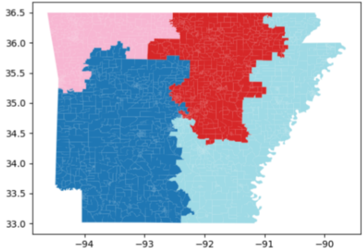
How Do I Know You Gerrymandered?





How Do I Know You Gerrymandered?

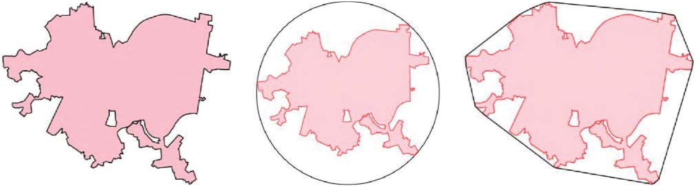




- "I mean. . . look at it!"

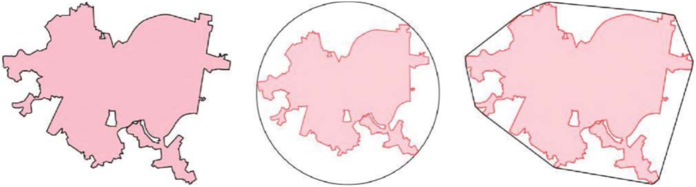
Perhaps Compactness?

Perhaps Compactness?

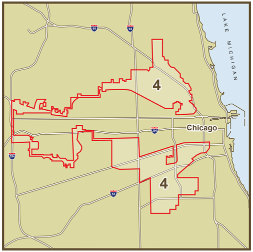


(a) City Boundary (b) Bounding Circle (c) Convex Hull

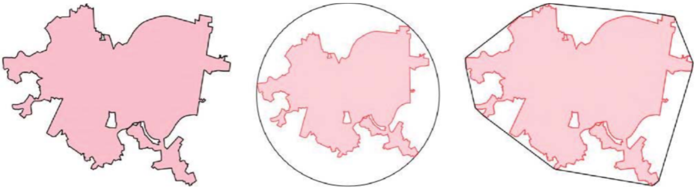
Perhaps Compactness?



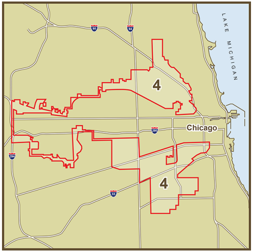
(a) City Boundary (b) Bounding Circle (c) Convex Hull



Perhaps Compactness?



(a) City Boundary (b) Bounding Circle (c) Convex Hull



Metrics

The set of metrics for these maps is infinite! To name some popular ones:

Metrics

The set of metrics for these maps is infinite! To name some popular ones:

- Compactness

Metrics

The set of metrics for these maps is infinite! To name some popular ones:

- Compactness
- Contiguity

Metrics

The set of metrics for these maps is infinite! To name some popular ones:

- Compactness
- Contiguity
- Mean-Median Score

Metrics

The set of metrics for these maps is infinite! To name some popular ones:

- Compactness
- Contiguity
- Mean-Median Score
- Polsby-Popper Score

Metrics

The set of metrics for these maps is infinite! To name some popular ones:

- Compactness
- Contiguity
- Mean-Median Score
- Polsby-Popper Score
- Racial Representation

Metrics

The set of metrics for these maps is infinite! To name some popular ones:

- Compactness
- Contiguity
- Mean-Median Score
- Polsby-Popper Score
- Racial Representation
- Population Balance

Metrics

The set of metrics for these maps is infinite! To name some popular ones:

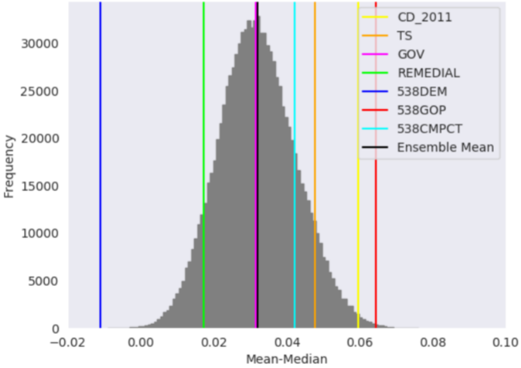
- Compactness
- Contiguity
- Mean-Median Score
- Polsby-Popper Score
- Racial Representation
- Population Balance
- And whatever we decide to make in the future!

Metrics

The set of metrics for these maps is infinite! To name some popular ones:

- Compactness
- Contiguity
- Mean-Median Score
- Polsby-Popper Score
- Racial Representation
- Population Balance
- And whatever we decide to make in the future!

Ensemble Analysis



- "If the map you gave me is soooo fair, it better be fairer than lots of random maps!"

Gerrymandering Resources

https://mggg.org/districtr

https://redistrictingdatahub.org/

RDH

Check-In

1. My Life & I
2. Gerrymandering Goodies
3. Gerrymandering Goodies
4. **Graph Theory Essentials**
5. Spectral Graph Theory
6. Invitation for Exploration

Graph Theory Basics

Graph Theory Basics

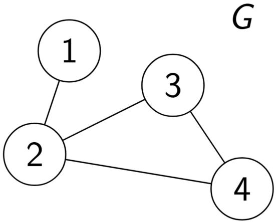
Definition

A graph $G = (V, E)$ consists of a set of vertices (nodes) V with a corresponding set of edges (arcs).

Graph Theory Basics

Definition

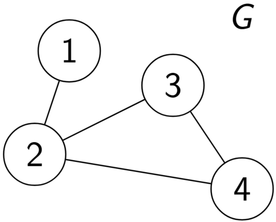
A graph $G = (V, E)$ consists of a set of vertices (nodes) V with a corresponding set of edges (arcs).



Graph Theory Basics

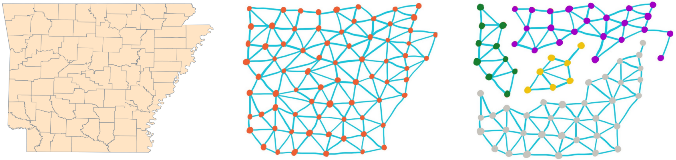
Definition

A graph $G = (V, E)$ consists of a set of vertices (nodes) V with a corresponding set of edges (arcs).

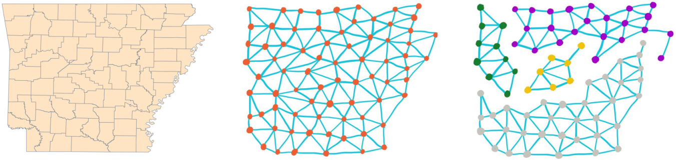


Dual Graph

Dual Graph

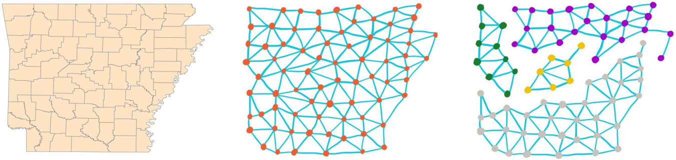


Dual Graph



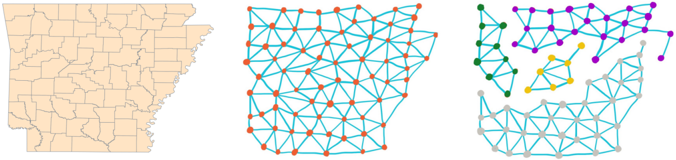
- Assign each district a node

Dual Graph



- Assign each district a node
- Connect nodes if those districts are adjacent in the map!

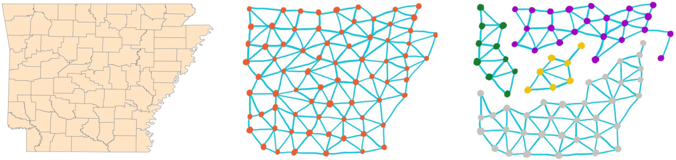
Dual Graph



- Assign each district a node
- Connect nodes if those districts are adjacent in the map!

Main idea: a proposed district "map" will correspond to some partitioning of our dual graph.

Dual Graph



- Assign each district a node
- Connect nodes if those districts are adjacent in the map!

Main idea: a proposed district "map" will correspond to some partitioning of our dual graph.

Motivating Questions

Motivating Questions

In order to detect some level of bias in a proposal map, we want to compare our proposal map to a lot of maps matching similar characteristics.

Motivating Questions

In order to detect some level of bias in a proposal map, we want to compare our proposal map to a lot of maps matching similar characteristics.

Question

- How can we find graphs that are similar with respect to our state's chosen characteristics?

Motivating Questions

In order to detect some level of bias in a proposal map, we want to compare our proposal map to a lot of maps matching similar characteristics.

Question

- How can we find graphs that are similar with respect to our state's chosen characteristics?
- How can we make sure that these graphs are truly random?

Motivating Questions

In order to detect some level of bias in a proposal map, we want to compare our proposal map to a lot of maps matching similar characteristics.

Question

- How can we find graphs that are similar with respect to our state's chosen characteristics?
- How can we make sure that these graphs are truly random?
- Can we do this efficiently?

Motivating Questions

In order to detect some level of bias in a proposal map, we want to compare our proposal map to a lot of maps matching similar characteristics.

Question

- How can we find graphs that are similar with respect to our state's chosen characteristics?
- How can we make sure that these graphs are truly random?
- Can we do this efficiently?

Sampling Maps

Sampling Maps

We now have an object we can more easily manipulate! What we now want is to sample from graphs with properties we desire (or properties we do not want!). We will make a series of steps:

Sampling Maps

We now have an object we can more easily manipulate! What we now want is to sample from graphs with properties we desire (or properties we do not want!). We will make a series of steps:

- Start with some uniformly random graph

Sampling Maps

We now have an object we can more easily manipulate! What we now want is to sample from graphs with properties we desire (or properties we do not want!). We will make a series of steps:

- Start with some uniformly random graph
- Make some perturbation to the node assignments

Sampling Maps

We now have an object we can more easily manipulate! What we now want is to sample from graphs with properties we desire (or properties we do not want!). We will make a series of steps:

- Start with some uniformly random graph
- Make some perturbation to the node assignments
- If our perturbed graph is more desirable to us, we will move forward to that map

Sampling Maps

We now have an object we can more easily manipulate! What we now want is to sample from graphs with properties we desire (or properties we do not want!). We will make a series of steps:

- Start with some uniformly random graph
- Make some perturbation to the node assignments
- If our perturbed graph is more desirable to us, we will move forward to that map
- If not, we will stay where we are and try another perturbation

Sampling Maps

We now have an object we can more easily manipulate! What we now want is to sample from graphs with properties we desire (or properties we do not want!). We will make a series of steps:

- Start with some uniformly random graph
- Make some perturbation to the node assignments
- If our perturbed graph is more desirable to us, we will move forward to that map
- If not, we will stay where we are and try another perturbation
- After repeating this process satisfying some specific conditions, we are guaranteed to converge to some graph with our desired property!

Sampling Maps

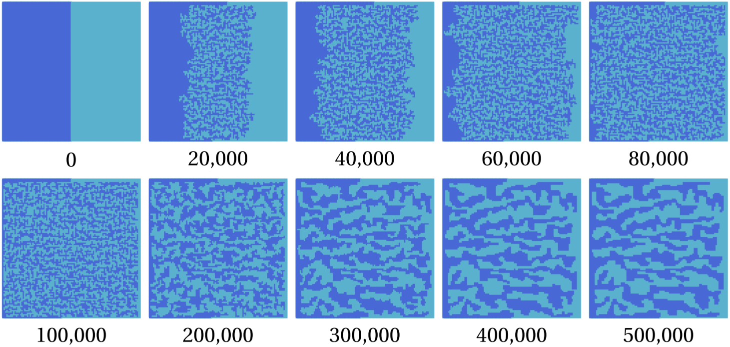
We now have an object we can more easily manipulate! What we now want is to sample from graphs with properties we desire (or properties we do not want!). We will make a series of steps:

- Start with some uniformly random graph
- Make some perturbation to the node assignments
- If our perturbed graph is more desirable to us, we will move forward to that map
- If not, we will stay where we are and try another perturbation
- After repeating this process satisfying some specific conditions, we are guaranteed to converge to some graph with our desired property!

Rabbit hole: this is called Markov Chain Monte Carlo (MCMC)

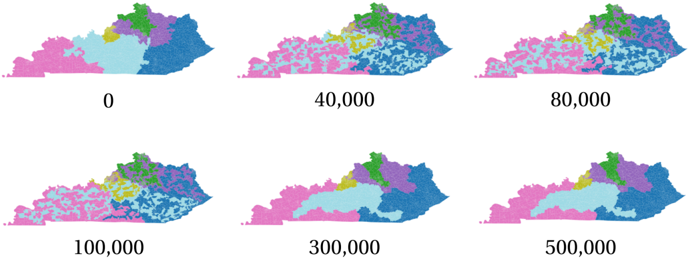
MCMC Example 1

MCMC Example 1



MCMC Example 2

MCMC Example 2



Potential Questions

Potential Questions

Question

- Is it fair to say that I drew a thousand maps that are better than a proposed map so the proposed map is gerrymandered? What about ten thousand? A million? A billion?

Potential Questions

Question

- Is it fair to say that I drew a thousand maps that are better than a proposed map so the proposed map is gerrymandered? What about ten thousand? A million? A billion?
- Is it fair to compare a proposed map to a map that is generated randomly?

Potential Questions

Question

- Is it fair to say that I drew a thousand maps that are better than a proposed map so the proposed map is gerrymandered? What about ten thousand? A million? A billion?
- Is it fair to compare a proposed map to a map that is generated randomly?
- What role should these algorithms play in these social settings?

Potential Questions

Question

- Is it fair to say that I drew a thousand maps that are better than a proposed map so the proposed map is gerrymandered? What about ten thousand? A million? A billion?
- Is it fair to compare a proposed map to a map that is generated randomly?
- What role should these algorithms play in these social settings?

Laplacian

Laplacian

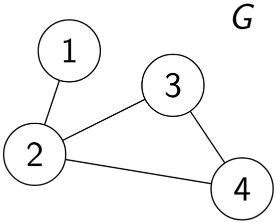
Definition

The *Laplacian* of G is the matrix L defined such that

$$L_{ij} = \begin{cases} \text{degree of node } i & \text{if } i = j \\ -1 & \text{if } i \neq j \text{ and node } i \sim \text{node } j \\ 0 & \text{otherwise} \end{cases}$$

Laplacian Example

Laplacian Example



$$L = \begin{bmatrix} 1 & -1 & 0 & 0 \\ -1 & 3 & -1 & -1 \\ 0 & -1 & 2 & -1 \\ 0 & -1 & -1 & 2 \end{bmatrix}$$

Eigen-Refresher

Eigen-Refresher

Definition

An *eigenvector* of a matrix A is any vector x such that $Ax = \lambda x$ where λ is a scalar. λ is called an *eigenvalue*

Eigen-Refresher

Definition

An *eigenvector* of a matrix A is any vector x such that $Ax = \lambda x$ where λ is a scalar. λ is called an *eigenvalue*

- If my matrix A takes in a vector x and outputs the same vector times some number λ, then I have an eigenvector.

Fiedler Vector

Fiedler Vector

Definition

We call the second smallest eigenvalue of L, λ_2, and its corresponding eigenvector, x_2, the *Fiedler value* and *Fiedler vector* respectively.

Fiedler Vector

Definition

We call the second smallest eigenvalue of L, λ_2, and its corresponding eigenvector, x_2, the *Fiedler value* and *Fiedler vector* respectively.

- Miraculously, we can use the Fiedler vector to partition graphs in two!

Fiedler Vector

Definition

We call the second smallest eigenvalue of L, λ_2, and its corresponding eigenvector, x_2, the *Fiedler value* and *Fiedler vector* respectively.

- Miraculously, we can use the Fiedler vector to partition graphs in two!
- The process:

Fiedler Vector

Definition

We call the second smallest eigenvalue of L, λ_2, and its corresponding eigenvector, x_2, the *Fiedler value* and *Fiedler vector* respectively.

- Miraculously, we can use the Fiedler vector to partition graphs in two!
- The process:
  1. Find the Fiedler vector for your graph G

Fiedler Vector

Definition

We call the second smallest eigenvalue of L, λ_2, and its corresponding eigenvector, x_2, the *Fiedler value* and *Fiedler vector* respectively.

- Miraculously, we can use the Fiedler vector to partition graphs in two!
- The process:
  1. Find the Fiedler vector for your graph G
  2. Look at the nodes assigned negative values in your Fiedler vector and put them in one group

Fiedler Vector

Definition

We call the second smallest eigenvalue of L, λ_2, and its corresponding eigenvector, x_2, the *Fiedler value* and *Fiedler vector* respectively.

- Miraculously, we can use the Fiedler vector to partition graphs in two!
- The process:
  1. Find the Fiedler vector for your graph G
  2. Look at the nodes assigned negative values in your Fiedler vector and put them in one group
  3. Look at the nodes assigned non-negative values to another group

Fiedler Vector

Definition

We call the second smallest eigenvalue of L, λ_2, and its corresponding eigenvector, x_2, the *Fiedler value* and *Fiedler vector* respectively.

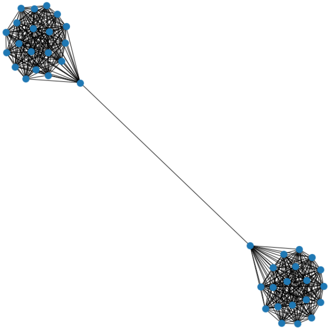
- Miraculously, we can use the Fiedler vector to partition graphs in two!
- The process:
  1. Find the Fiedler vector for your graph G
  2. Look at the nodes assigned negative values in your Fiedler vector and put them in one group
  3. Look at the nodes assigned non-negative values to another group
  4. You now have two connected parts of your graph

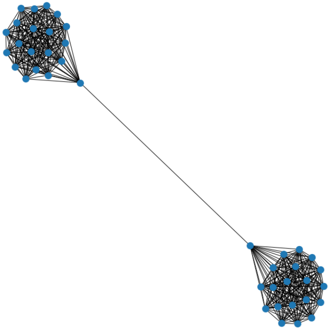
Fiedler Vector

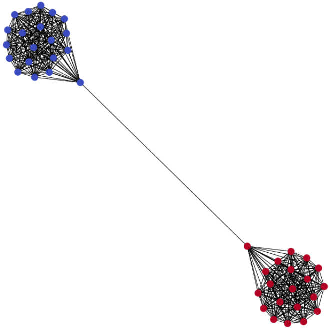
Definition

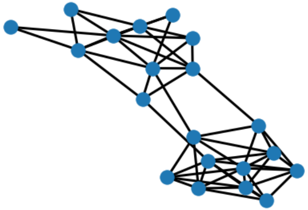
We call the second smallest eigenvalue of L, λ_2, and its corresponding eigenvector, x_2, the *Fiedler value* and *Fiedler vector* respectively.

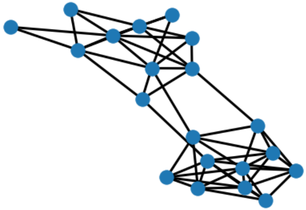
- Miraculously, we can use the Fiedler vector to partition graphs in two!
- The process:
  1. Find the Fiedler vector for your graph G
  2. Look at the nodes assigned negative values in your Fiedler vector and put them in one group
  3. Look at the nodes assigned non-negative values to another group
  4. You now have two connected parts of your graph

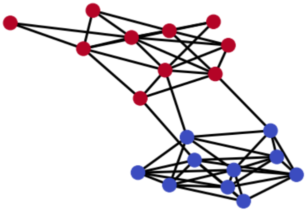


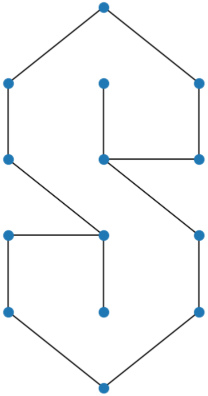


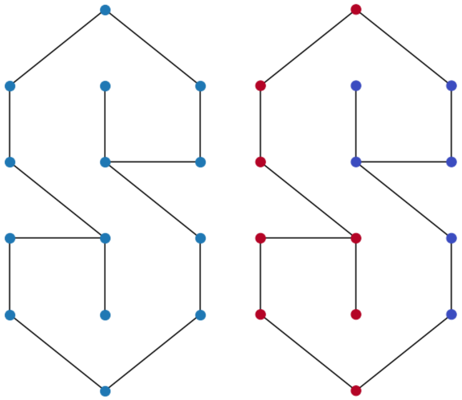


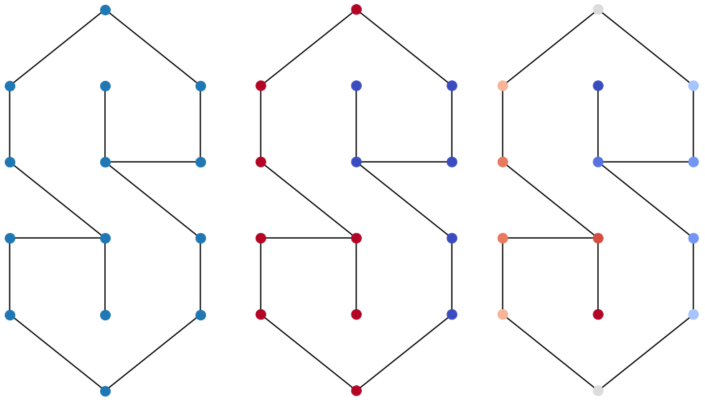


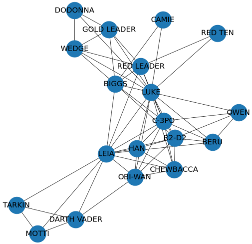


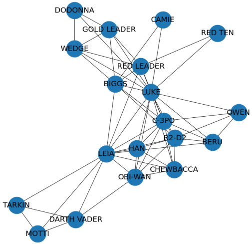


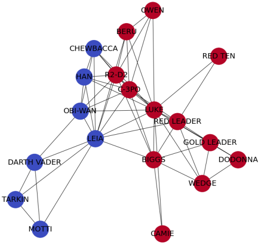


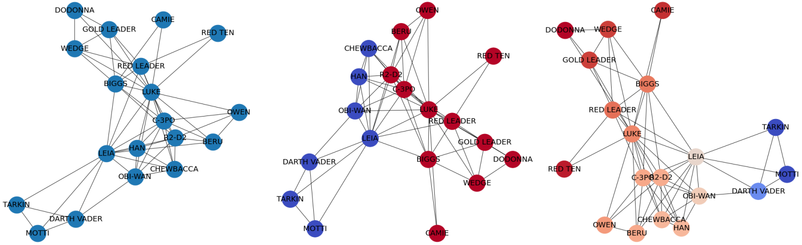


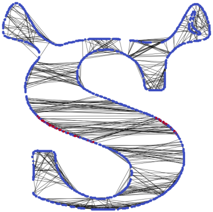


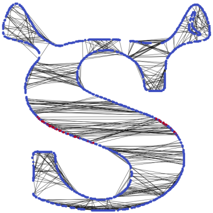

DODONNA
GOLD LEADER
CAMIE
RED TEN
WEDGE
RED LEADER
BIGGS
LUKE
OWEN
C-3PO
R2-D2
BERU
LEIA
HAN
CHEWBACCA
OBI-WAN
TARKIN
DARTH VADER
MOTTI

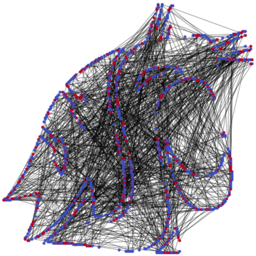

DODONNA
CAMIE
GOLD LEADER
RED TEN
WEDGE
RED LEADER
BIGGS
LUKE
OWEN
C-3PO
R2-D2
BERU
LEIA
HAN
CHEWBACCA
OBI-WAN
TARKIN
DARTH VADER
MOTTI

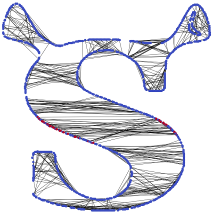

OWEN
BERU
CHEWBACCA
RED TEN
HAN
R2-D2
C-3PO
OBI-WAN
LUKE
RED LEADER
LEIA
GOLD LEADER
DARTH VADER
BIGGS
DODONNA
WEDGE
TARKIN
MOTTI
CAMIE

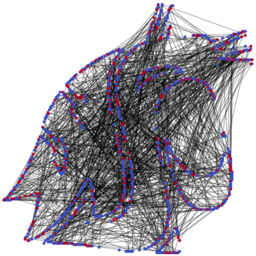

DODONNA
GOLD LEADER
CAMIE
RED TEN
WEDGE
RED LEADER
BIGGS
LUKE
OWEN
C-3PO
R2-D2
BERU
LEIA
HAN
CHEWBACCA
OBI-WAN
TARKIN
DARTH VADER
MOTTI
OWEN
BERU
CHEWBACCA
RED TEN
HAN
R2-D2
C-3PO
OBI-WAN
LUKE
RED LEADER
LEIA
GOLD LEADER
DARTH VADER
BIGGS
DODONNA
WEDGE
TARKIN
MOTTI
CAMIE
CAMIE
DODONNA
WEDGE
GOLD LEADER
BIGGS
RED LEADER
TARKIN
LUKE
LEIA
RED TEN
MOTTI
C-3PO
R2-D2
DARTH VADER
OBI-WAN
CHEWBACCA
OWEN
HAN
BERU

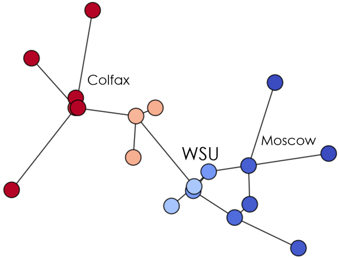


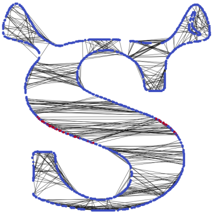


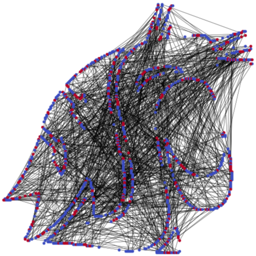


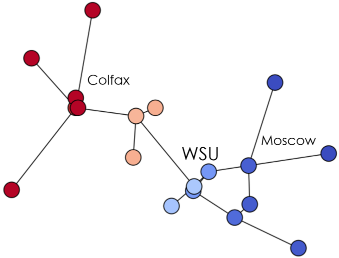


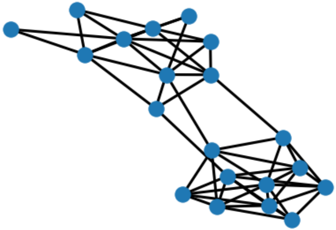


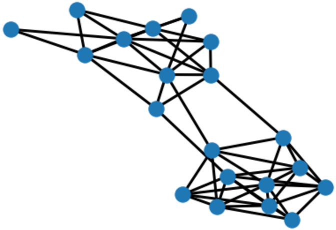

Colfax
WSU
Moscow

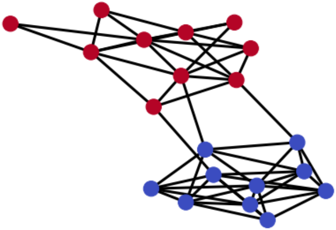


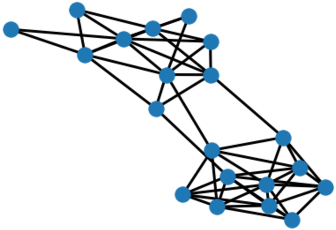


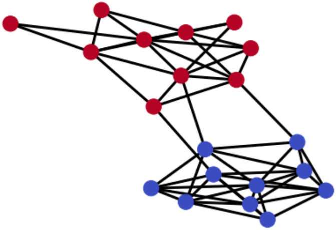

Colfax
WSU
Moscow

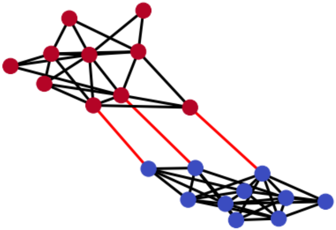


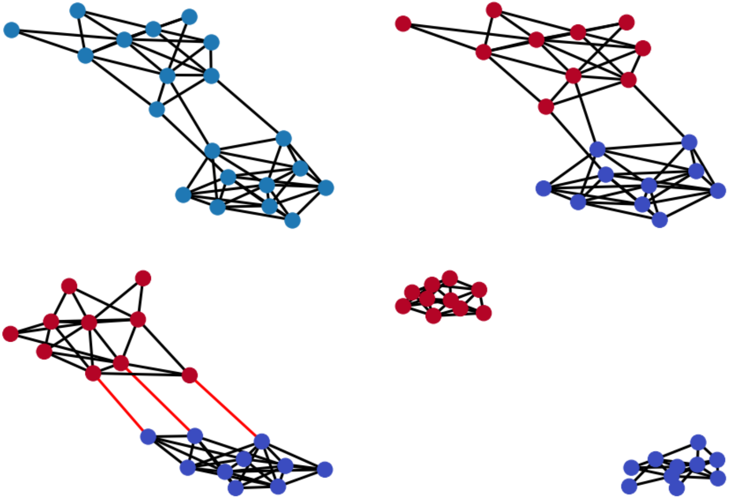










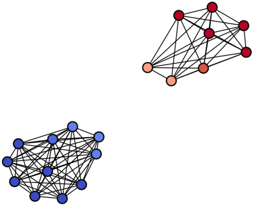


Question

Given the two connected components of a Fiedler-partitioned graph, can we determine what set of edges are missing using the entries of the Fiedler vector?

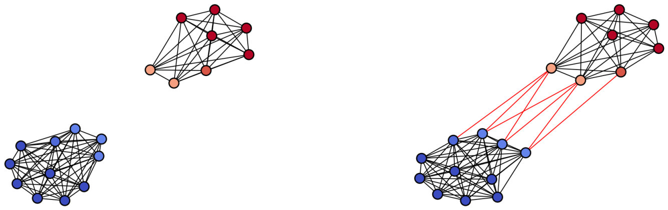
Question

Given the two connected components of a Fiedler-partitioned graph, can we determine what set of edges are missing using the entries of the Fiedler vector?



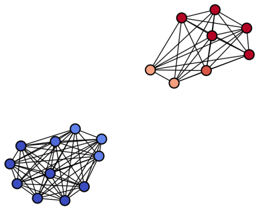
Question

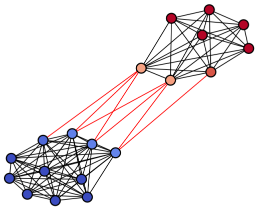
Given the two connected components of a Fiedler-partitioned graph, can we determine what set of edges are missing using the entries of the Fiedler vector?



Question

Given the two connected components of a Fiedler-partitioned graph, can we determine what set of edges are missing using the entries of the Fiedler vector?





Progress & Future Progress

Progress & Future Progress

- The answer is:

Progress & Future Progress

- The answer is: not always!

Progress & Future Progress

- The answer is: not always!
- When is it the case?

Progress & Future Progress

- The answer is: not always!
- When is it the case? No idea!

Progress & Future Progress

- The answer is: not always!
- When is it the case? No idea!
- For any class of graphs you can name, we can construct the Fiedler vector. But, for any Fiedler vector we do not know if we can construct the graph.

Progress & Future Progress

- The answer is: not always!
- When is it the case? No idea!
- For any class of graphs you can name, we can construct the Fiedler vector. But, for any Fiedler vector we do not know if we can construct the graph.

Equitable Spectral Partitions

Equitable Spectral Partitions

Motivation:

Equitable Spectral Partitions

Motivation:

- We like maps with balance and compactness!

Equitable Spectral Partitions

Motivation:

- We like maps with balance and compactness!
- How can we draw maps that have balance and compactness?

Equitable Spectral Partitions

Motivation:

- We like maps with balance and compactness!
- How can we draw maps that have balance and compactness?
- Potentially with spectral partitioning!

Equitable Spectral Partitions

Motivation:

- We like maps with balance and compactness!
- How can we draw maps that have balance and compactness?
- Potentially with spectral partitioning!

Intuition:

Equitable Spectral Partitions

Motivation:

- We like maps with balance and compactness!
- How can we draw maps that have balance and compactness?
- Potentially with spectral partitioning!

Intuition:

- There is a bottleneck in the graph G

Equitable Spectral Partitions

Motivation:

- We like maps with balance and compactness!
- How can we draw maps that have balance and compactness?
- Potentially with spectral partitioning!

Intuition:

- There is a bottleneck in the graph G
- Using the Fiedler vector, finding edges between oppositely signed nodes helps us find bottlenecks

Equitable Spectral Partitions

Motivation:

- We like maps with balance and compactness!
- How can we draw maps that have balance and compactness?
- Potentially with spectral partitioning!

Intuition:

- There is a bottleneck in the graph G
- Using the Fiedler vector, finding edges between oppositely signed nodes helps us find bottlenecks
- Let's use this to our advantage!

Spectral Partitioning Algorithm

Spectral Partitioning Algorithm

The Process:

Spectral Partitioning Algorithm

The Process:

- Take a random graph cut into k-parts

Spectral Partitioning Algorithm

The Process:

- Take a random graph cut into k-parts
- Randomly select 2 adjacent parts to form a subgraph

Spectral Partitioning Algorithm

The Process:

- Take a random graph cut into k-parts
- Randomly select 2 adjacent parts to form a subgraph
- Reassign nodes in this subgraph using the Fiedler vector

Spectral Partitioning Algorithm

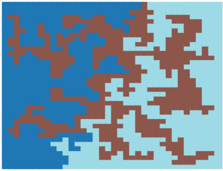
The Process:

- Take a random graph cut into k-parts
- Randomly select 2 adjacent parts to form a subgraph
- Reassign nodes in this subgraph using the Fiedler vector
- Repeat on newly obtained graph

Spectral Partitioning Algorithm

The Process:

- Take a random graph cut into k-parts
- Randomly select 2 adjacent parts to form a subgraph
- Reassign nodes in this subgraph using the Fiedler vector
- Repeat on newly obtained graph



Spectral Partitioning Algorithm

The Process:

- Take a random graph cut into k-parts
- Randomly select 2 adjacent parts to form a subgraph
- Reassign nodes in this subgraph using the Fiedler vector
- Repeat on newly obtained graph

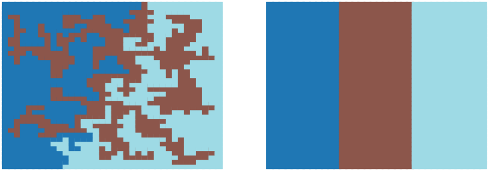


Figure 7: Convergence to a "Nice" Partition of the Algorithm After 10,000 Steps on a 36 by 36 Grid Graph

Main Questions

Main Questions

Question

- Does this always converge?

Main Questions

Question

- Does this always converge?
- What happens if we choose different classes of graphs?

Main Questions

Question

- Does this always converge?
- What happens if we choose different classes of graphs?
- Does this also converge using other spectral clustering techniques?

Main Questions

Question

- Does this always converge?
- What happens if we choose different classes of graphs?
- Does this also converge using other spectral clustering techniques?
- In what context is it appropriate to use these types of algorithms while drawing maps?

Conclusion

Conclusion

- Math is wherever we put it!

Rabbit Holes

Rabbit Holes

1. Networks & Data Science:

Rabbit Holes

1. Networks & Data Science:
   1. https://networkrepository.com/
   2. "Network Science" by Barabási
      https://networksciencebook.com/
2. Gerrymandering:

Rabbit Holes

1. Networks & Data Science:
   1. https://networkrepository.com/
   2. "Network Science" by Barabási https://networksciencebook.com/
2. Gerrymandering:
   1. https://redistrictingdatahub.org/
   2. https://mggg.org/districtr
   3. "Political Geometry" by Duchin & Walch https://mggg.org/gerrybook.html

Questions?

Questions?

Thanks!

garrett.kepler@wsu.edu