Network Applications of Discrete Curvature and Ricci Flow

Jie Gao

Stony Brook University
http://www.cs.sunysb.edu/∼jgao

Spring Workshop on DCG, April 20th, 2017

Wireless Networks

- Smart buildings, smart cities, smart communities: lots of wireless devices embedded in the physical space.

Wireless Networks

- Smart buildings, smart cities, smart communities: lots of wireless devices embedded in the physical space.
- Each node communicates with other nodes within communication range.

Wireless Networks

- Smart buildings, smart cities, smart communities: lots of wireless devices embedded in the physical space.
- Each node communicates with other nodes within communication range.
- A discrete, rough approximation of a manifold.

Wireless Networks

- Smart buildings, smart cities, smart communities: lots of wireless devices embedded in the physical space.
- Each node communicates with other nodes within communication range.
- A discrete, rough approximation of a manifold.

Mobile Networks

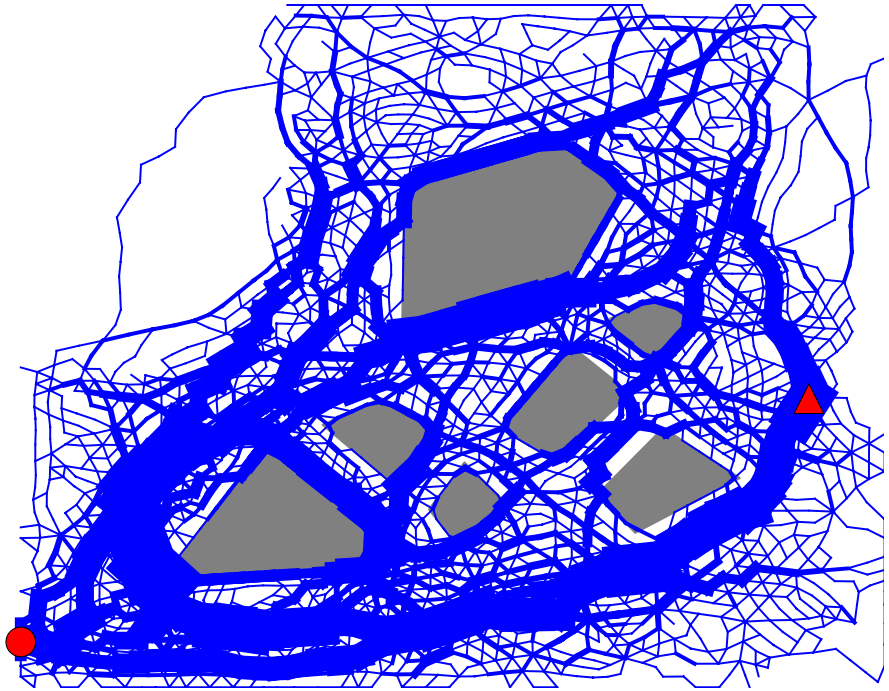
- Mobile devices roaming around.

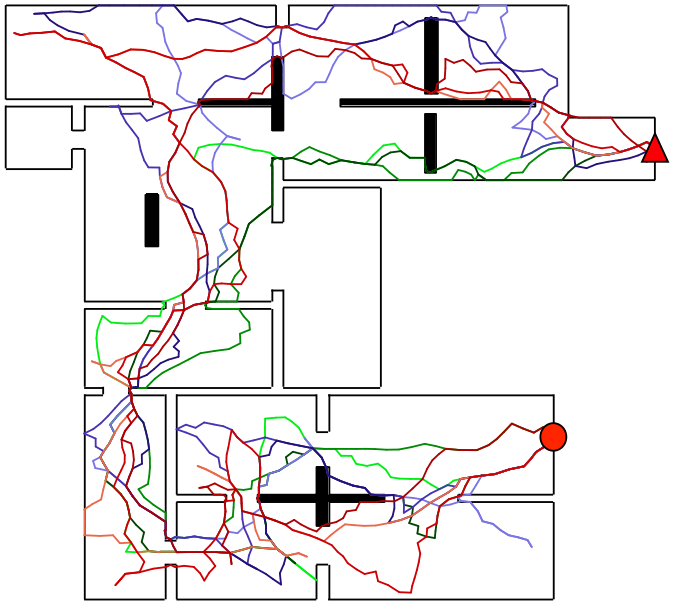
Mobile Networks

- Mobile devices roaming around.
- Vehicles trajectories, human motion behaviors.

Mobile Networks

- Mobile devices roaming around.
- Vehicles trajectories, human motion behaviors.





Social Networks

- Characterizing the social interactions.

Social Networks

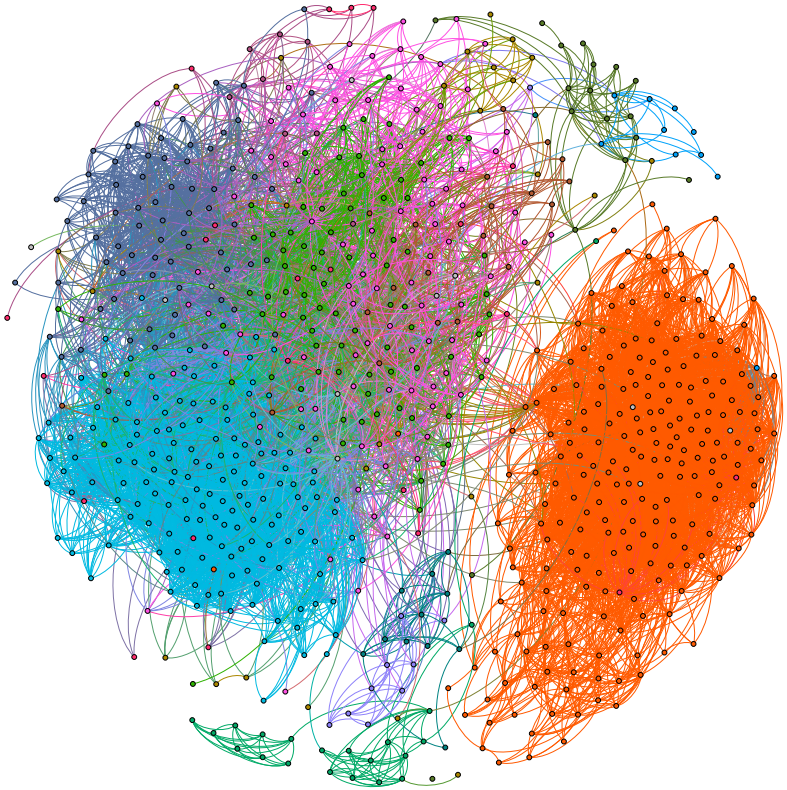
- Characterizing the social interactions.
- No geometric embedding.

Social Networks

- Characterizing the social interactions.
- No geometric embedding.
- Small world phenomena, power law degree distribution, community structures.

Social Networks

- Characterizing the social interactions.
- No geometric embedding.
- Small world phenomena, power law degree distribution, community structures.



Outline

- Past work on wireless networks: Ricci flow on triangulations.

Outline

- Past work on wireless networks: Ricci flow on triangulations.
- Recent work on Internet & Social networks: discrete graph curvatures.

Routing in A Distributed Network

Goal: deliver a message from the source to destination.

Routing in A Distributed Network

Goal: deliver a message from the source to destination.
Internet: Each router keeps a routing table (destination, next hop).

Routing in A Distributed Network

Goal: deliver a message from the source to destination.
Internet: Each router keeps a routing table (destination, next hop).
Challenges for the wireless setting:

- Each node has limited stoarge/communication capaity.

Routing in A Distributed Network

Goal: deliver a message from the source to destination.
Internet: Each router keeps a routing table (destination, next hop).
Challenges for the wireless setting:

- Each node has limited stoarge/communication capaity.
- Wireless links are unreliable.

Routing in A Distributed Network

Goal: deliver a message from the source to destination.
Internet: Each router keeps a routing table (destination, next hop).
Challenges for the wireless setting:

- Each node has limited stoarge/communication capaity.
- Wireless links are unreliable.
- The domain geometry can be complicated, with holes.

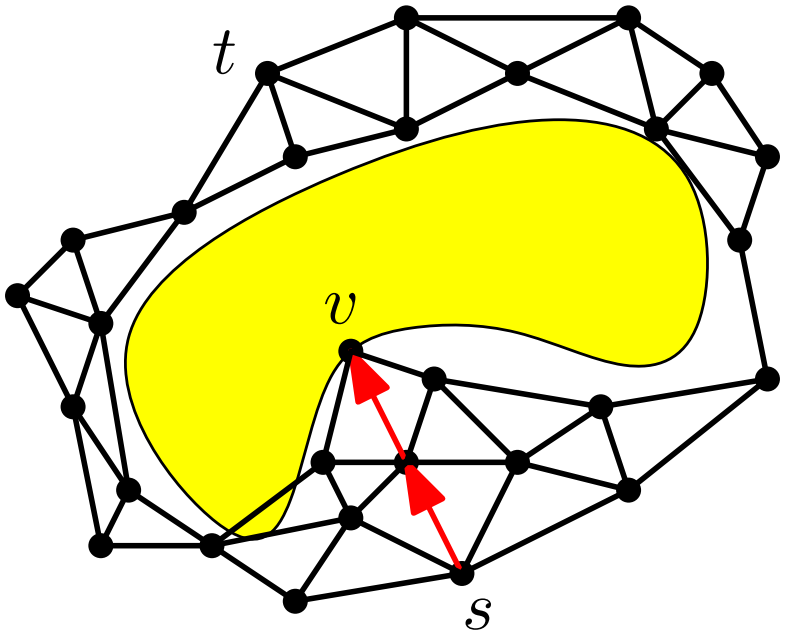
Routing in A Distributed Network

Goal: deliver a message from the source to destination.
Internet: Each router keeps a routing table (destination, next hop).
Challenges for the wireless setting:

- Each node has limited stoarge/communication capaity.
- Wireless links are unreliable.
- The domain geometry can be complicated, with holes.

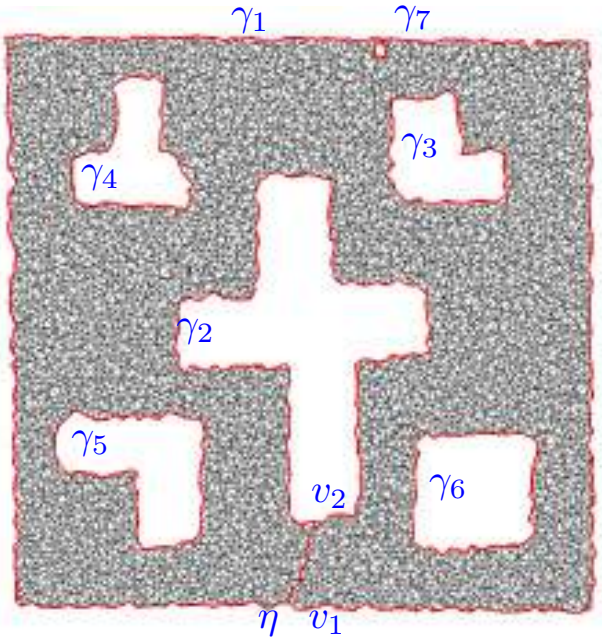
Geographical Greedy Routing

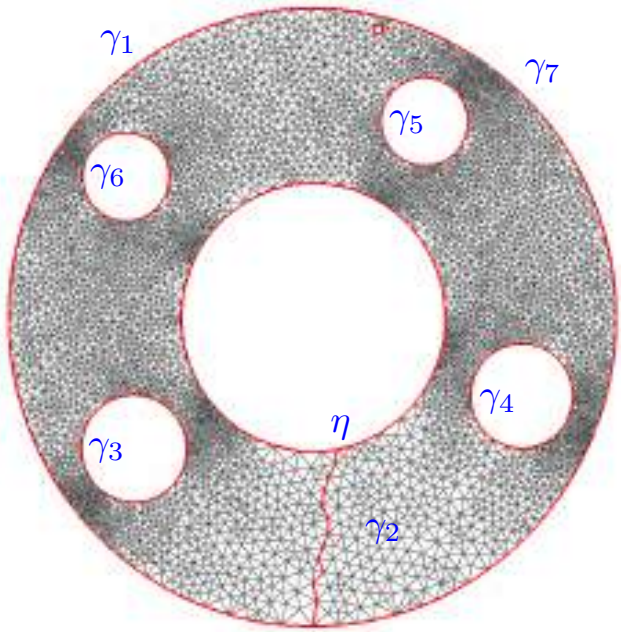
A message is sent to the neighbor closest to the destination.



Greedy Routing with Virtual Coordinates

Mapping the network into a 'circular domain' (all boundaries are circles). $\rightarrow$ Greedy routing has guaranteed delivery.



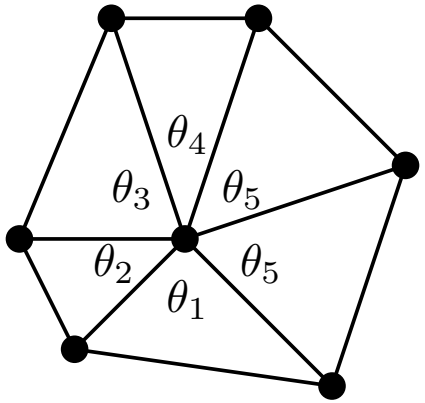


[Sarkar, Yin, G, Luo, Gu 2009]

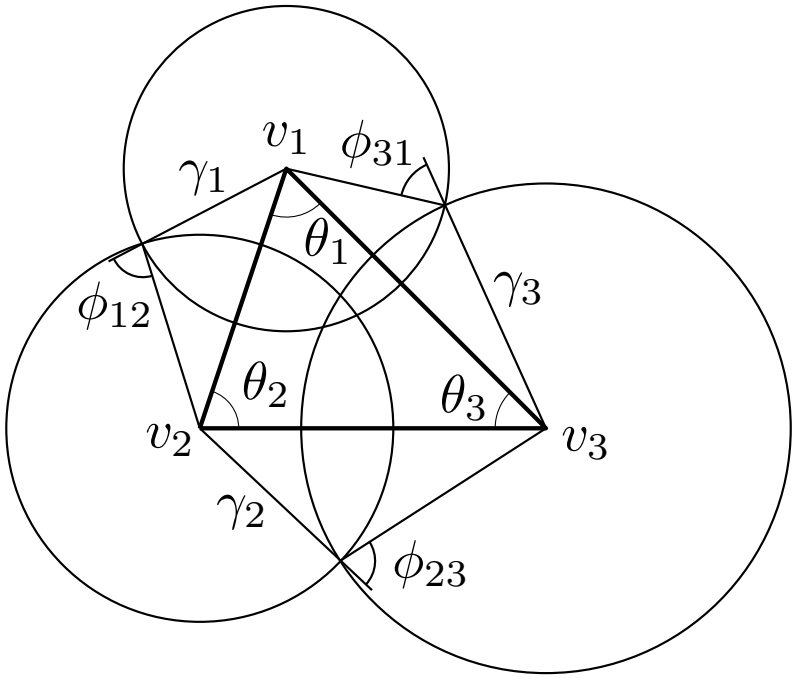
Discrete Gaussian Curvature on a Triangulation

Discrete Gaussian Curvature defined by

$$2\pi - \sum \theta_i$$



Circle Packing Metric



Ricci flow: change the circle radius to meet the target curvature (zero in the interior, and $2\pi/K$ on a hole with K nodes).

Circular Domain is Not Unique

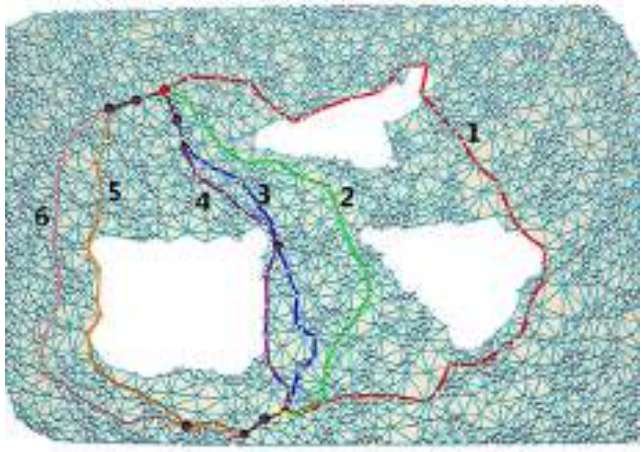
Apply a mobius transformation (which maps circles to circles).

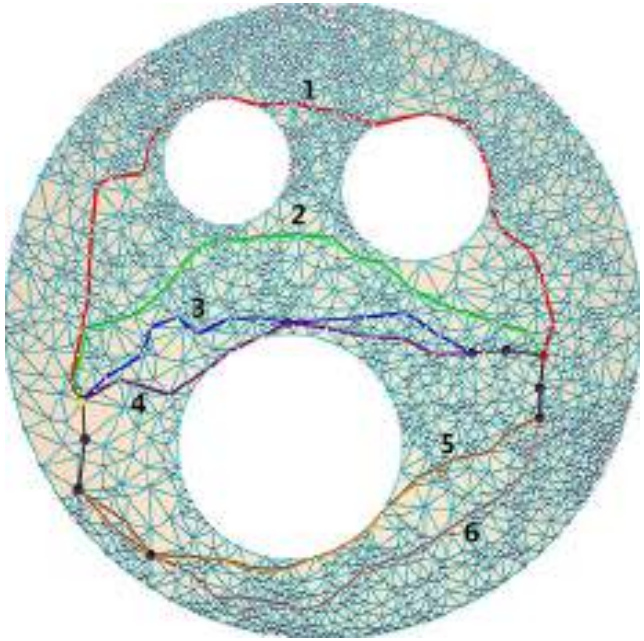
$$m(z) = \frac{\alpha z + \beta}{\gamma z + \delta}$$

Multi-Path Routing

Run greedy routing with different mobius transformation $\rightarrow$ a different path.

- Rapid response to failures.
- Multi-path routing with higher throughput.

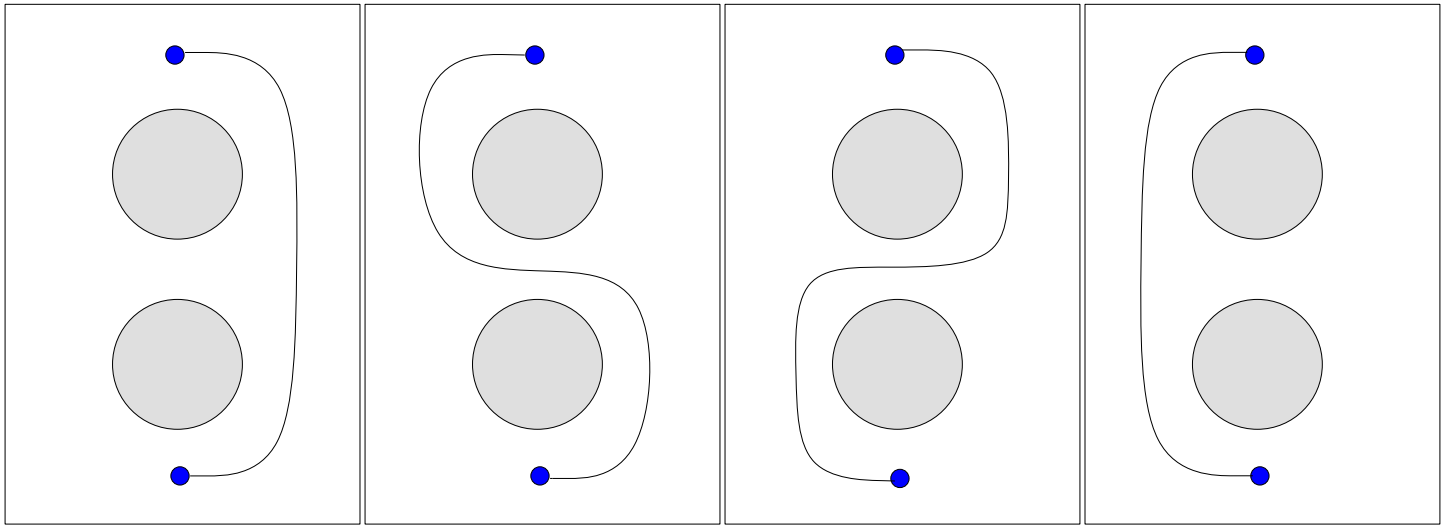




[Jiang, Ban, Goswami, Zeng, G, Gu 2011]

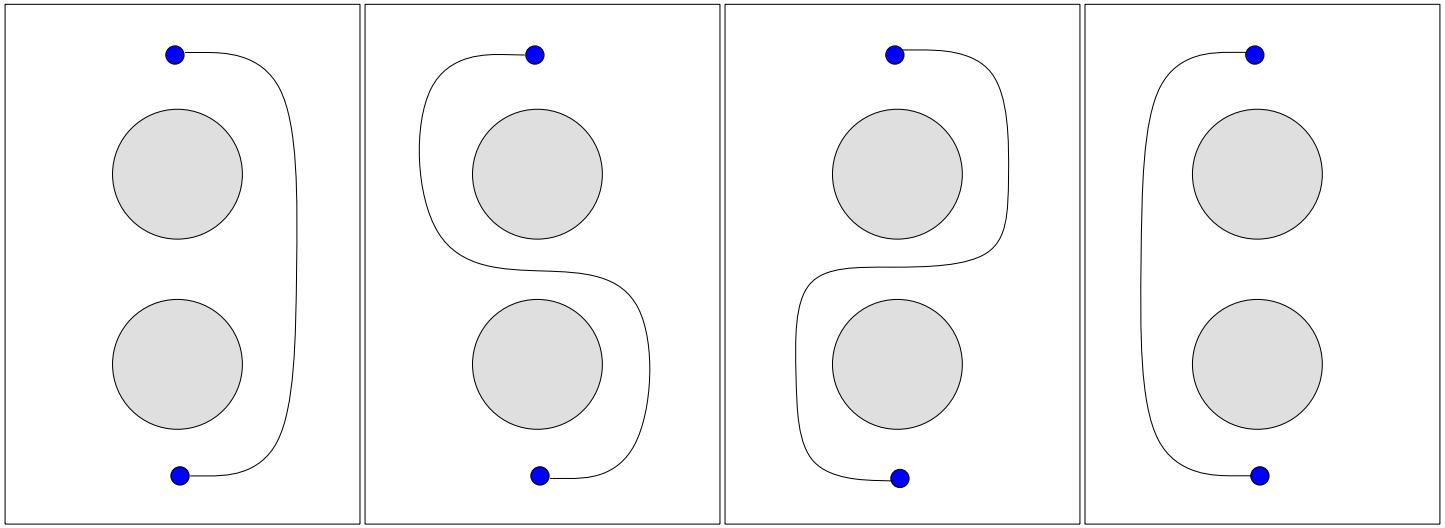
Homotopic Routing

Can we find a path with the specified homotopic type?



Homotopic Routing

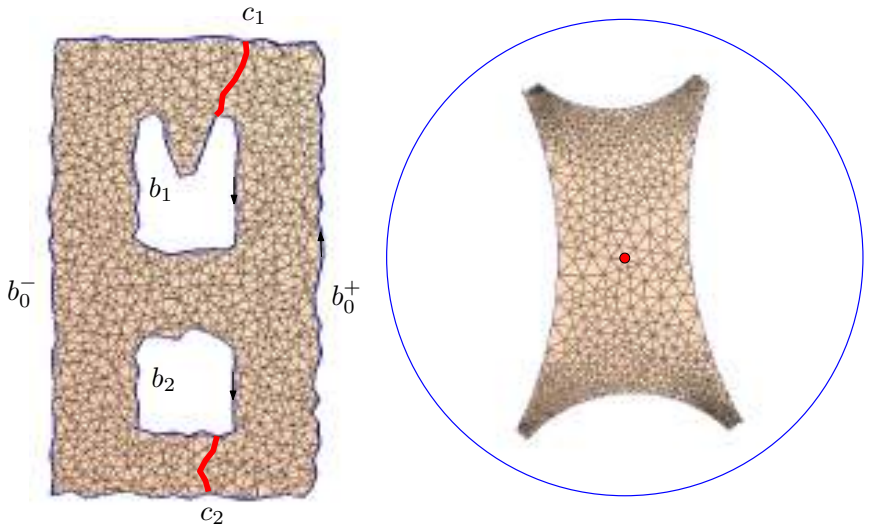
Can we find a path with the specified homotopic type?



Our idea: use universal covering space and hyperbolic embedding.

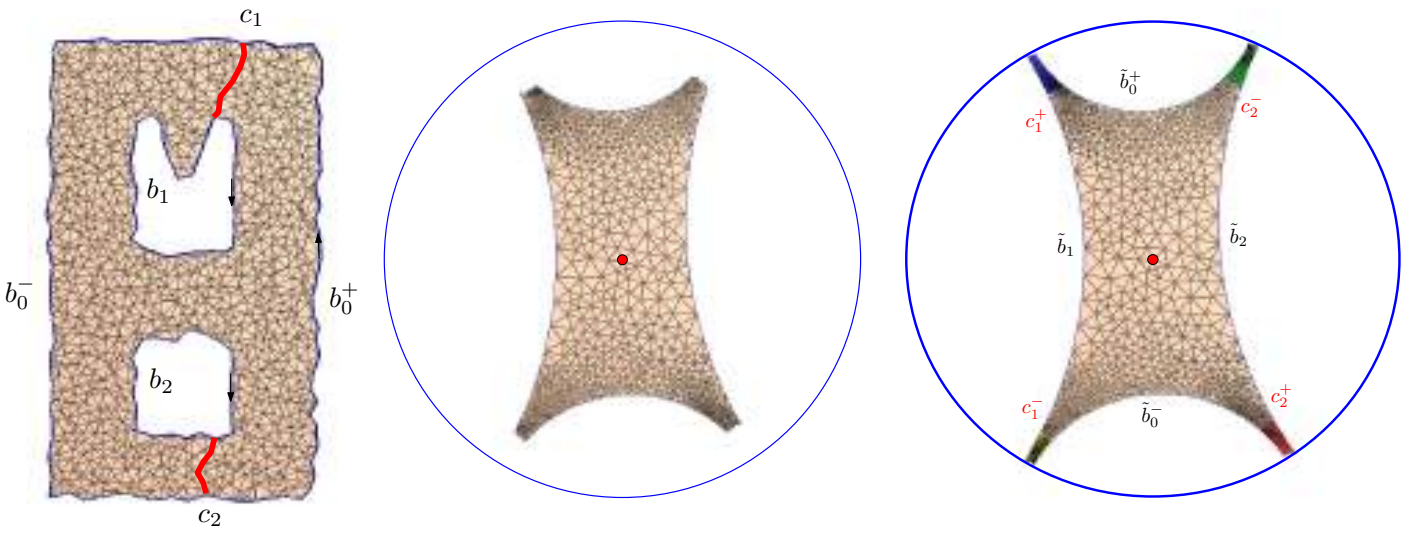
Homotopic Routing

- Cut a multi-hole domain open, embed it to a convex polygon in the hyperbolic plane.
- Greedy routing (using hyperbolic distance) guarantees delivery.



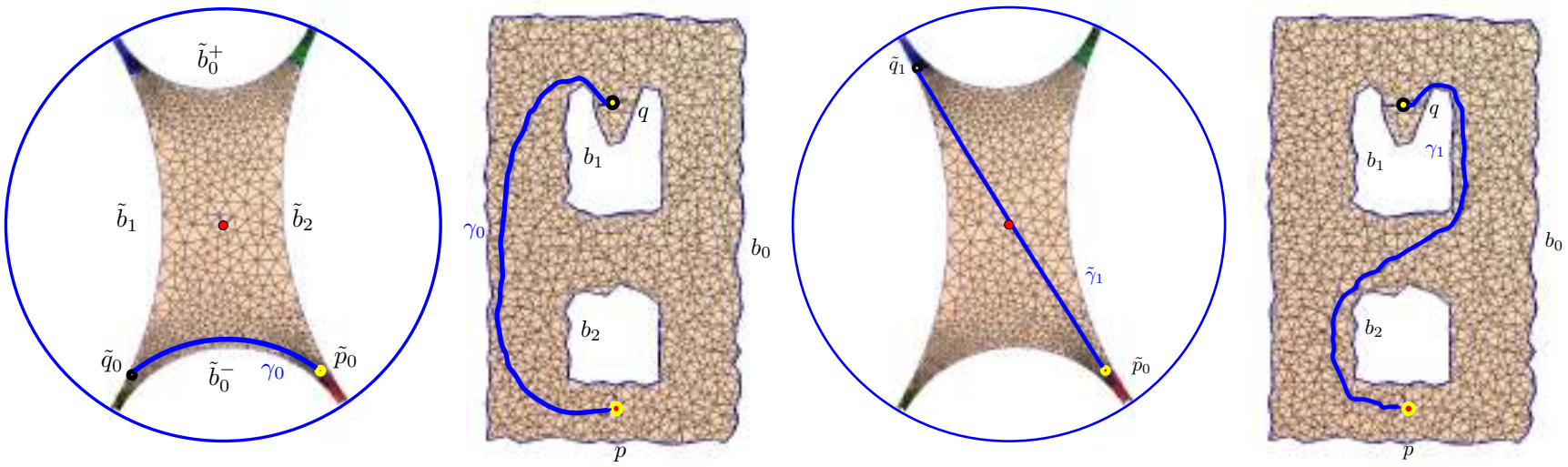
Homotopic Routing

- Use Mobius transformation to embed copies that could fill up the domain.



Homotopic Routing

- Greedy routing to the destination in different copies gives routes of different homotopy types.



[Zeng, Sarkar, Luo, Gu, G 2010]

Greedy Routing with Virtual Coordinates

- Guaranteed delivery. ✓

Greedy Routing with Virtual Coordinates

- Guaranteed delivery. ✓
- Better resilience to sudden failures in the network.✓

Greedy Routing with Virtual Coordinates

- Guaranteed delivery. ✓
- Better resilience to sudden failures in the network. ✓
- Routes of specified homotopy types. ✓

Greedy Routing with Virtual Coordinates

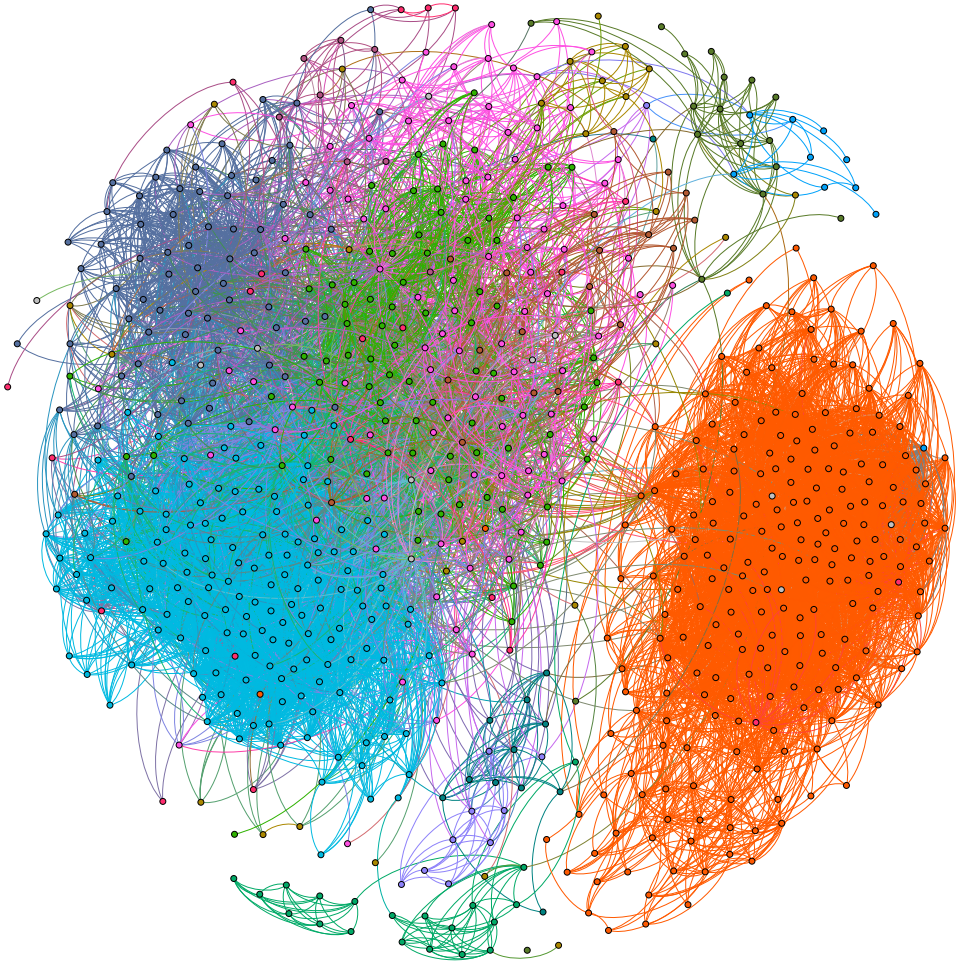
- Guaranteed delivery. ✓
- Better resilience to sudden failures in the network. ✓
- Routes of specified homotopy types. ✓
- Load balancing. ✓

Greedy Routing with Virtual Coordinates

- Guaranteed delivery. ✓
- Better resilience to sudden failures in the network.✓
- Routes of specified homotopy types. ✓
- Load balancing. ✓

Until we are tired of this setting..

What about other graphs?

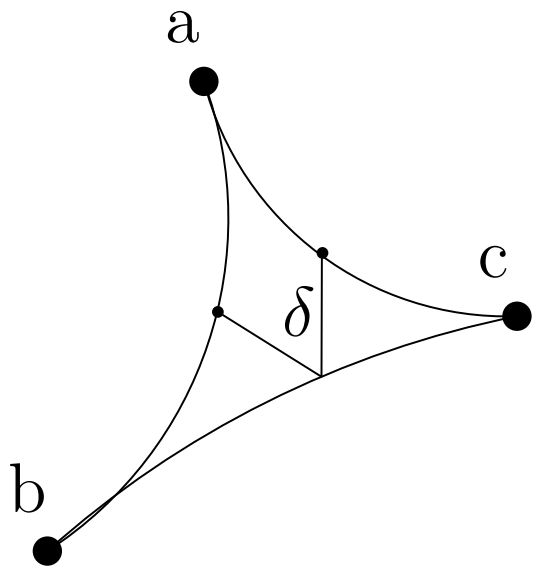


Coarse Geometry: Gromov's Curvature

It has been discovered (Baryshnikov, Narayan & Saniee) that the Internet topology has **negative curvature**, defined by Gromov's **thin triangle property**.

Coarse Geometry: Gromov's Curvature

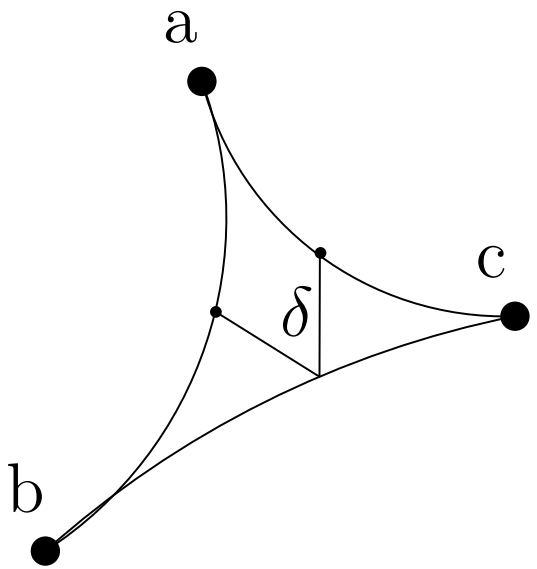
It has been discovered (Baryshnikov, Narayan & Saniee) that the Internet topology has **negative curvature**, defined by Gromov's **thin triangle property**.
For any triple a, b, c, the minimum distance from any point on one shortest path to the union of the other two shortest paths is $\leq \delta$.

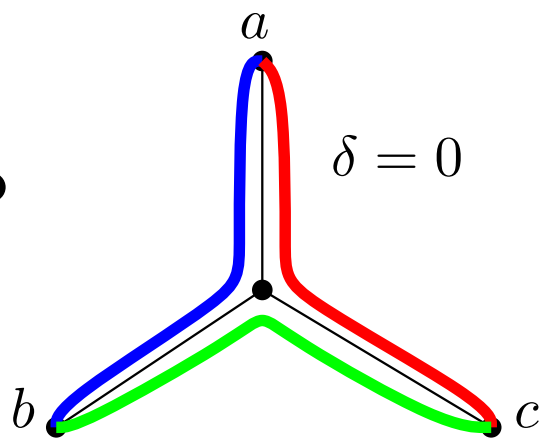


Coarse Geometry: Gromov's Curvature

It has been discovered (Baryshnikov, Narayan & Saniee) that the Internet topology has **negative curvature**, defined by Gromov's **thin triangle property**.

For any triple a, b, c, the minimum distance from any point on one shortest path to the union of the other two shortest paths is $\leq \delta$.





Internet is Negatively Curved

A variety of data sets, both AS-level and router level topologies, show δ-hyperbolicity for small constant δ (2 or 3).

Internet is Negatively Curved

A variety of data sets, both AS-level and router level topologies, show δ-hyperbolicity for small constant δ (2 or 3).

- Tree-like;

Internet is Negatively Curved

A variety of data sets, both AS-level and router level topologies, show δ-hyperbolicity for small constant δ (2 or 3).

- Tree-like;
- There is a 'core' in which all shortest path visit.

Internet is Negatively Curved

A variety of data sets, both AS-level and router level topologies, show δ-hyperbolicity for small constant δ (2 or 3).

- Tree-like;
- There is a 'core' in which all shortest path visit.
- Congestion inside the core is high;

Internet is Negatively Curved

A variety of data sets, both AS-level and router level topologies, show δ-hyperbolicity for small constant δ (2 or 3).

- Tree-like;
- There is a 'core' in which all shortest path visit.
- Congestion inside the core is high;
- But one can use preprocessing to speed up shortest path queries.

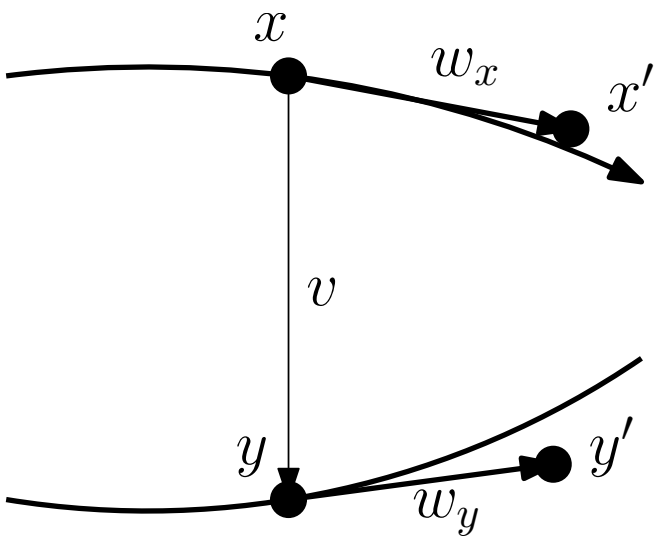
Global v.s. Local Curvature

Gromov hyperbolicity is a *global* measure. Here we examine *local* curvatures:

- Which edges are negatively curved?
- Local curvature and network congestion?
- Local curvature and node centrality?

Sectional Curvature in Geometry

Consider a tangent vector $v = xy$ and another tangent vector w_x at x. Transport w_x along v to be a tangent vector w_y at y.
If $|x'y'| < |xy|$, then sectional curvature is positive.



Ricci curvature: averaging over all directions w.

Discrete Ricci Curvature

Take the analog: for an edge xy, consider the distances from x**'s neighbors** to y**'s neighbors** and compare it with the length of xy.

Discrete Ricci Curvature

Take the analog: for an edge xy, consider the distances from x**'s neighbors** to y**'s neighbors** and compare it with the length of xy.

- Issue: how to match x's neighbors to y's neighbors?

Discrete Ricci Curvature

Take the analog: for an edge xy, consider the distances from x**'s neighbors** to y**'s neighbors** and compare it with the length of xy.

- Issue: how to match x's neighbors to y's neighbors?
- Assign uniform distribution μ_1, μ_2 on x' and y's neighbors.
- Use optimal transportation distance (earth-mover distance) from μ_1 to μ_2: the matching that minimizes the total transport distance.

Discrete Ricci Curvature

Definition (Ollivier, Lin & Yau)

Let (X, d) be a metric space and let m_1, m_2 be two probability measures on X. For any two distinct points $x, y \in X$, the (Ollivier-) Ricci curvature along xy is defined as

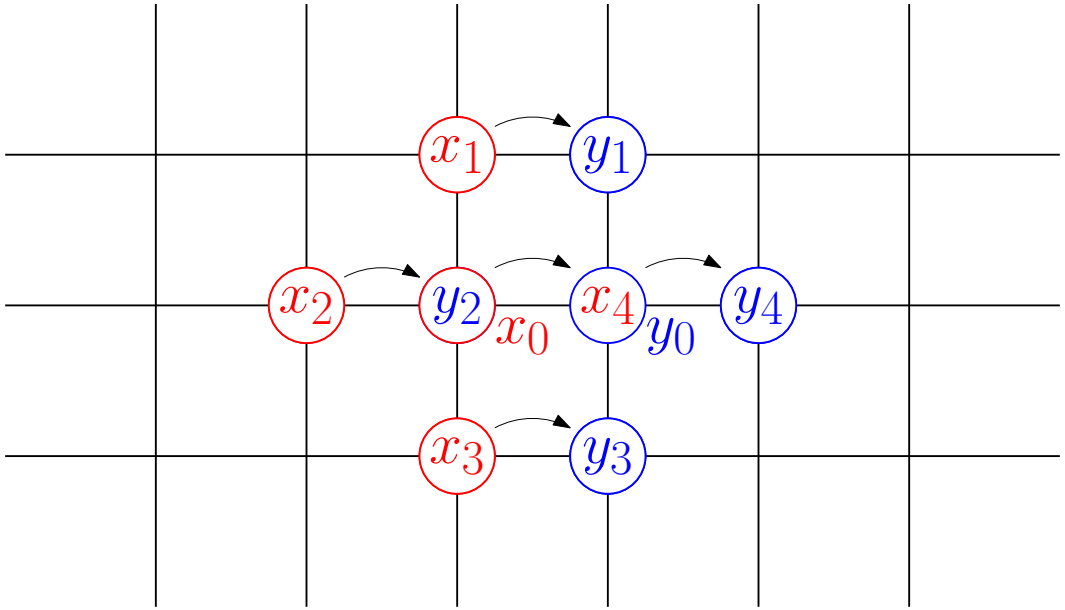
$$\kappa(x, y) := 1 - \frac{W_1(m_x, m_y)}{d(x, y)},$$

where m_x (m_y) is a probability distribution defined on x (y) and its neighbors, $W_1(\mu_1, \mu_2)$ is the L_1 **optimal transportation distance** between two probability measure μ_1 and μ_2 on X:

$$W_1(\mu_1, \mu_2) := \inf_{\psi \in \Pi(\mu_1, \mu_2)} \int_{(u,v)} d(u, v) d\psi(u, v)$$

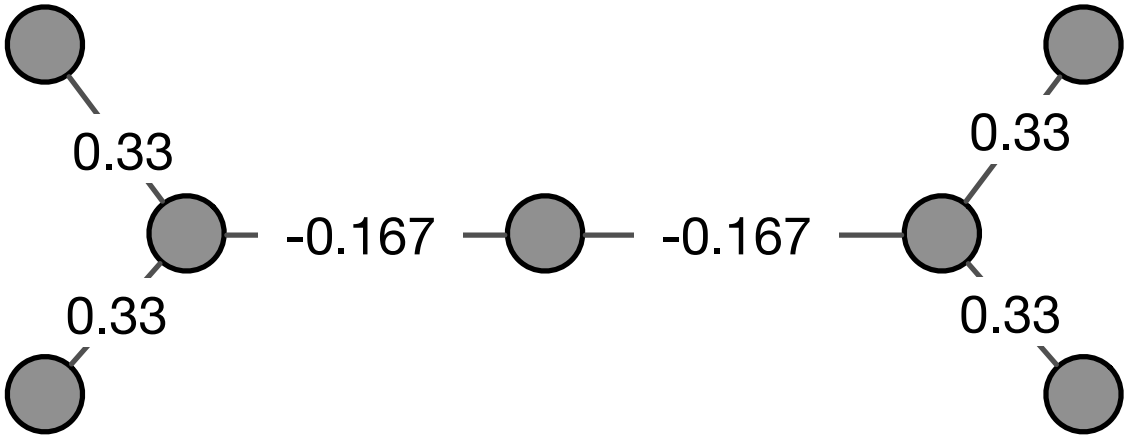
Examples

Zero curvature: 2D grid.



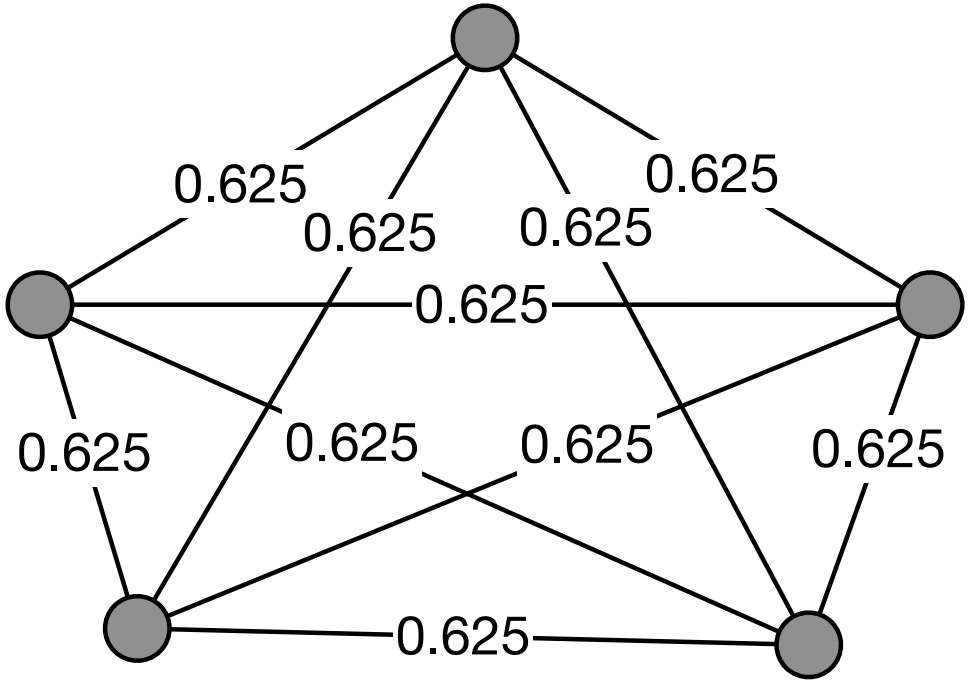
Examples

Negative curvature: tree: $\kappa(x, y) = 1/d_x + 1/d_y - 1$, d_x is degree of x.



Examples

Positive curvature: complete graph.



Data Sets

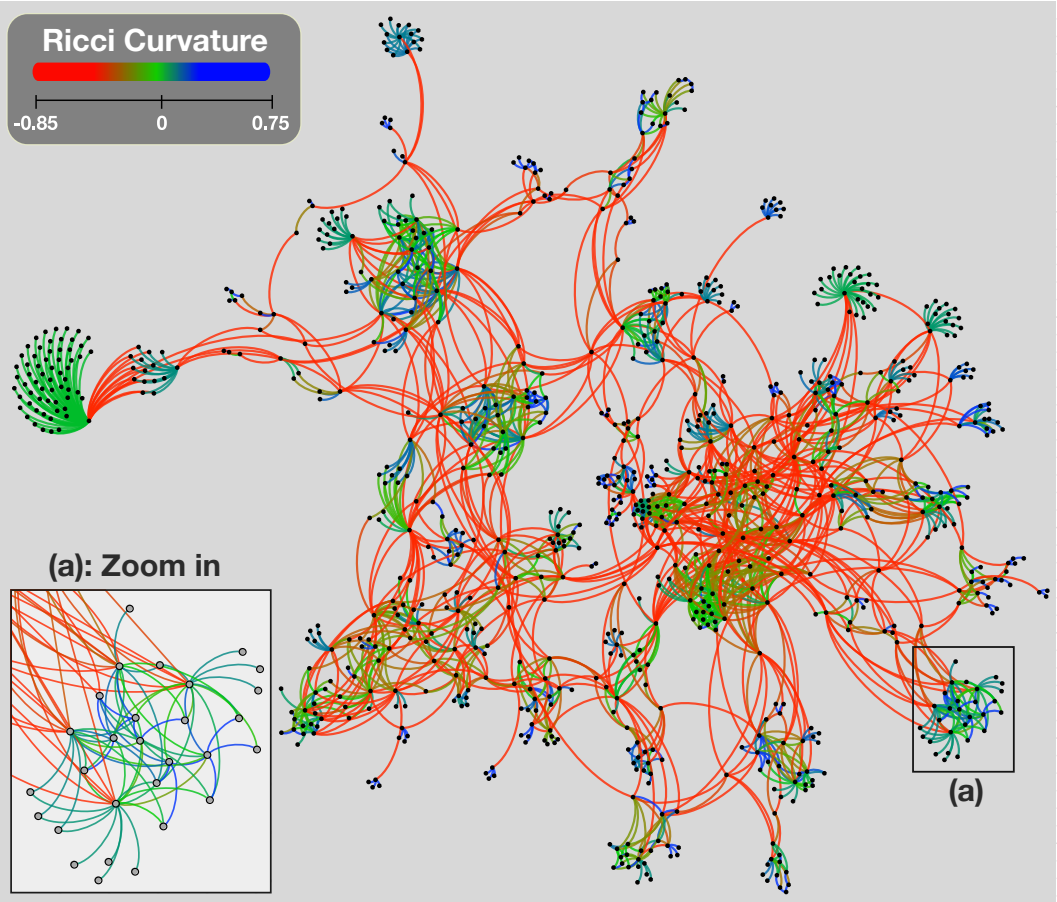
Data Set	Node	Edge	MaxDeg	AvgDeg	Diam	Mean SPL
RouteView	6474	12572	1458	3.88	9	3.71
Gnutella	6301	20777	97	6.59	9	4.64
PGP	10680	24340	205	4.56	24	7.48
Power grid	4941	6594	19	2.67	46	18.99

RocketFuel data set: router level network.

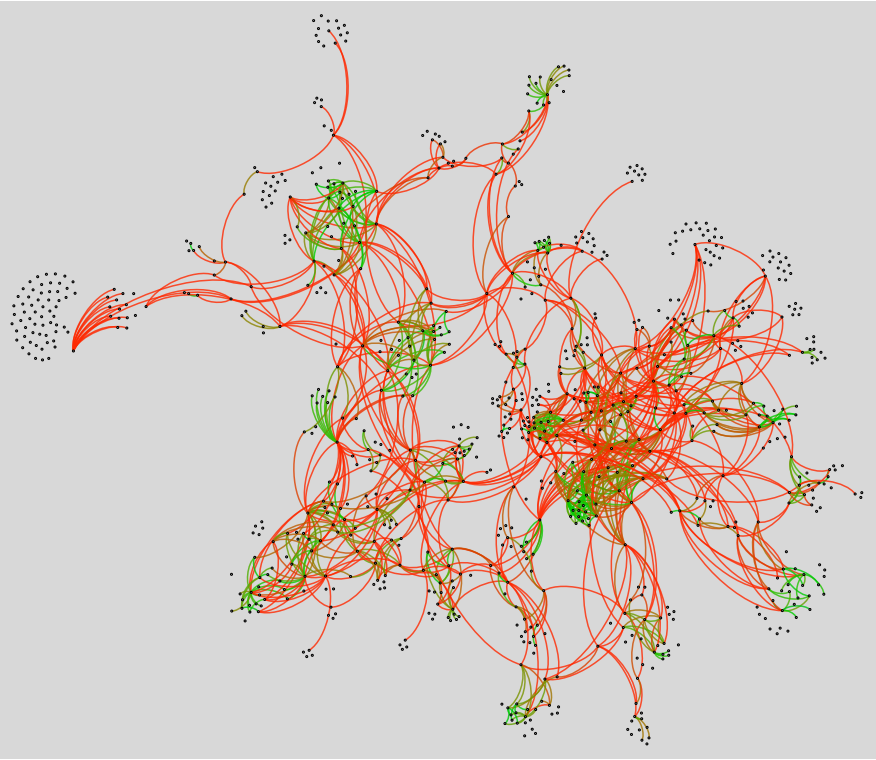
RochetFuel	Node	Edge	MaxDeg	AvgDeg	Diam	Mean SPL
AS: 7018	10145	14166	78	2.04	11	6.95
AS: 3257	843	1156	90	2.23	14	5.27
AS: 3967	895	2071	75	4.63	13	5.94
AS: 1221	2998	3789	106	2.53	12	5.53

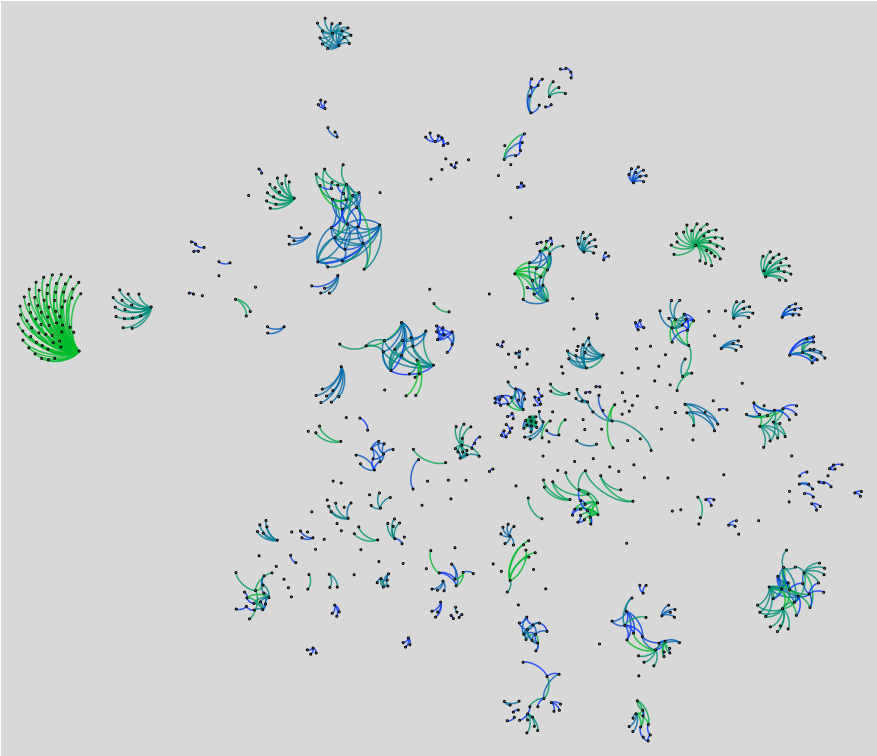
Curvature Distribution

Negatively curved edges are like "backbones", maintaining the connectivity of clusters, in which edges are mostly positively curved.

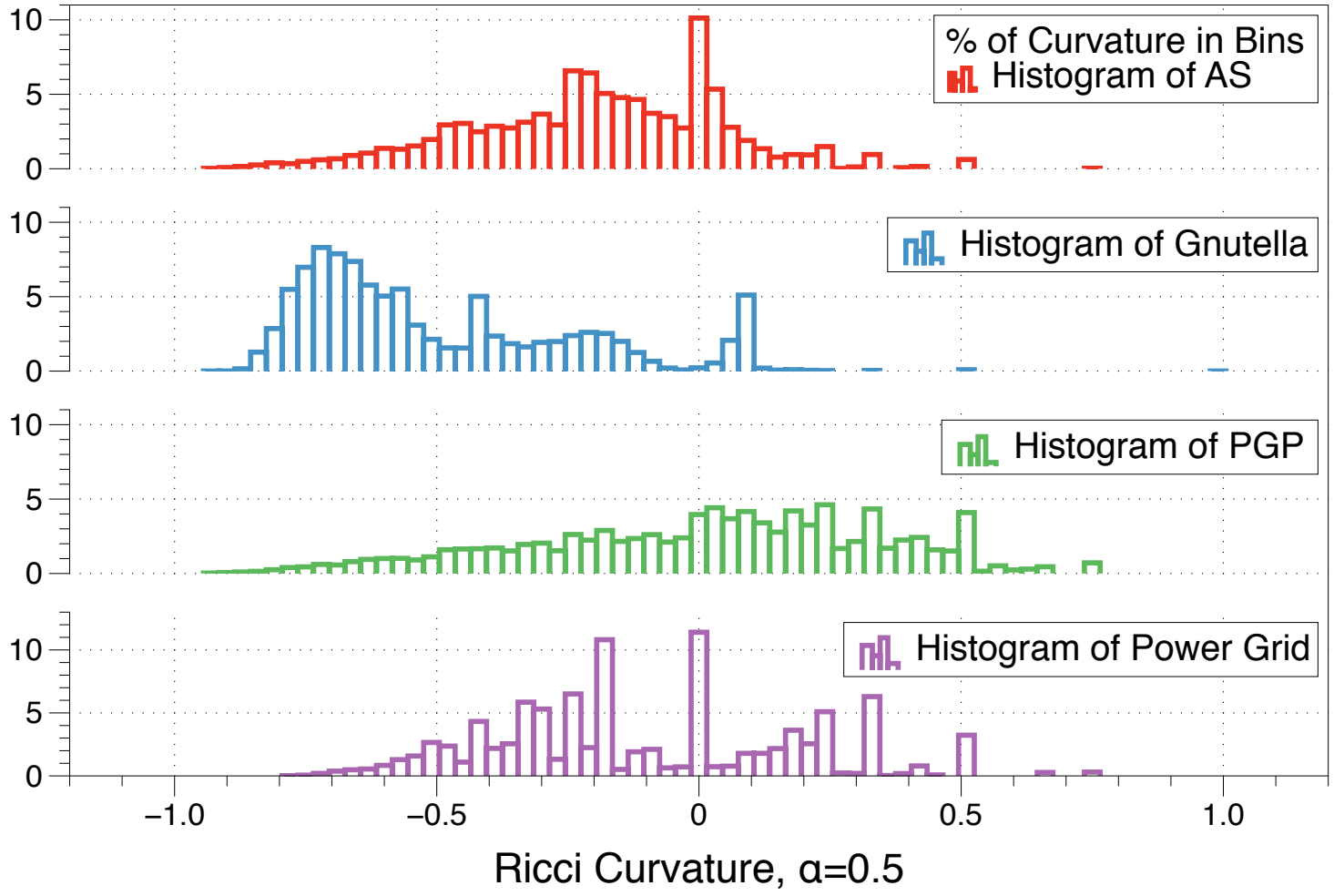


Curvature Distribution



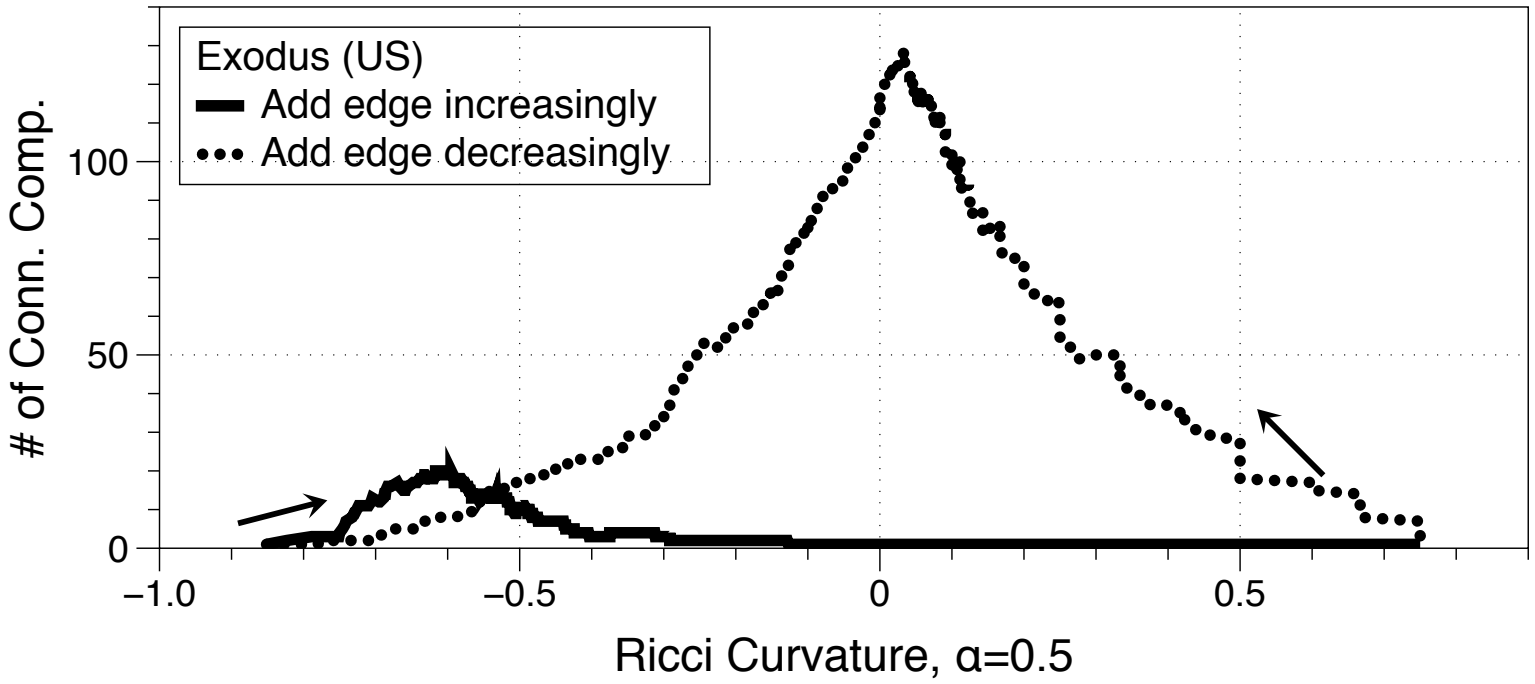


Curvature Distribution



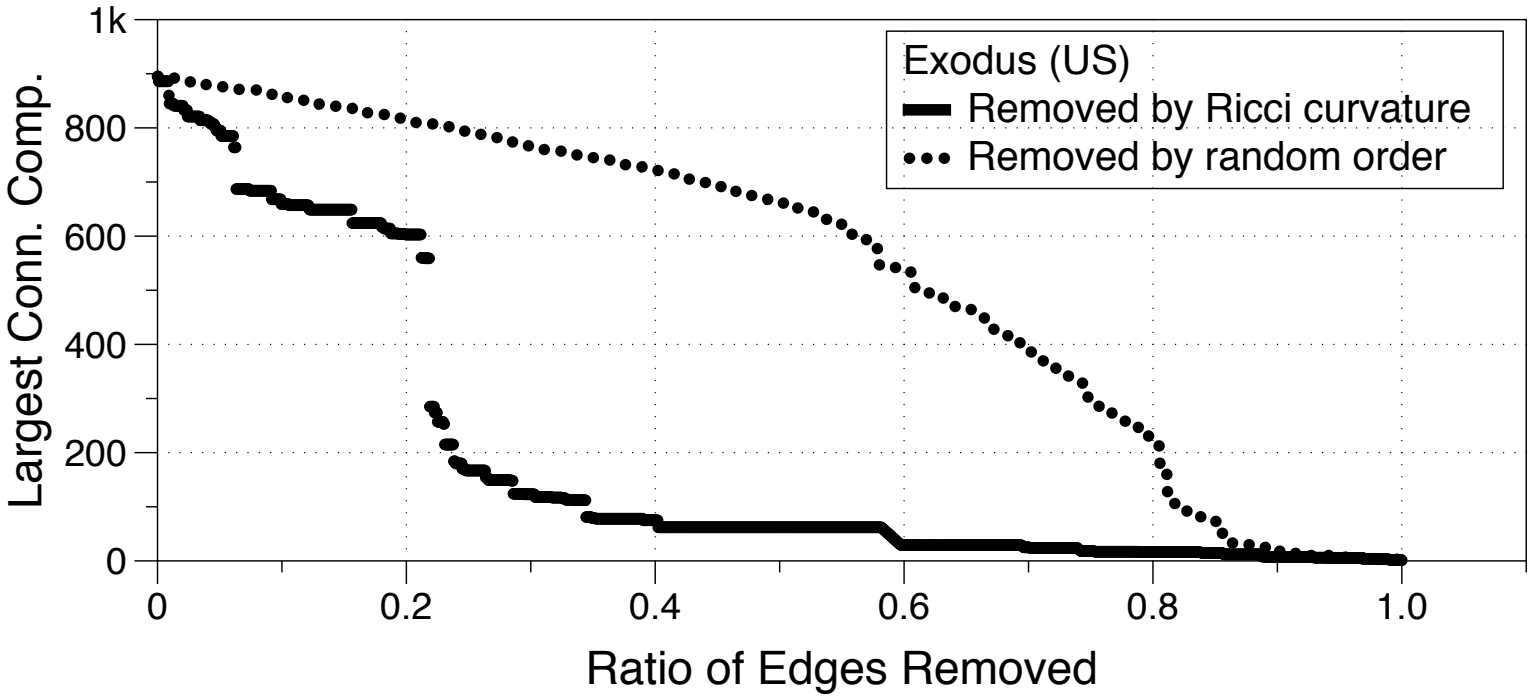
Network Connectivity

Negatively curved edges are well connected. Adding edges with increasing/decreasing curvature: few/many connected components.



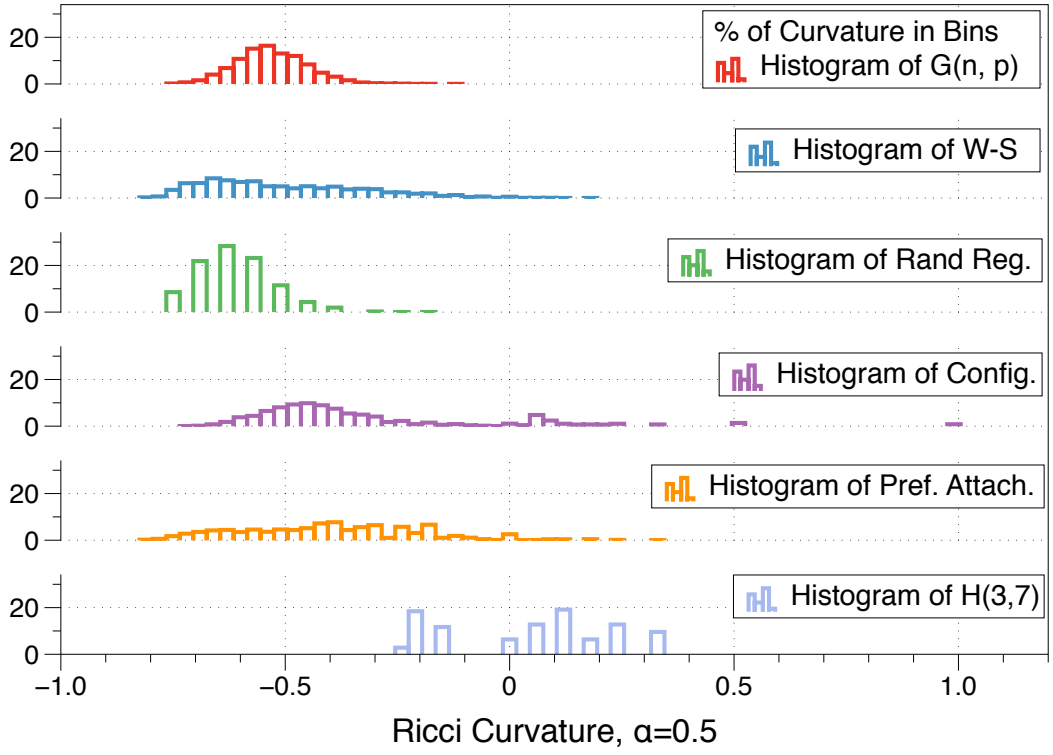
Robustness v.s. Vulnerabilty

Removing edges with increasing curvature: size of largest connected component drops quickly.



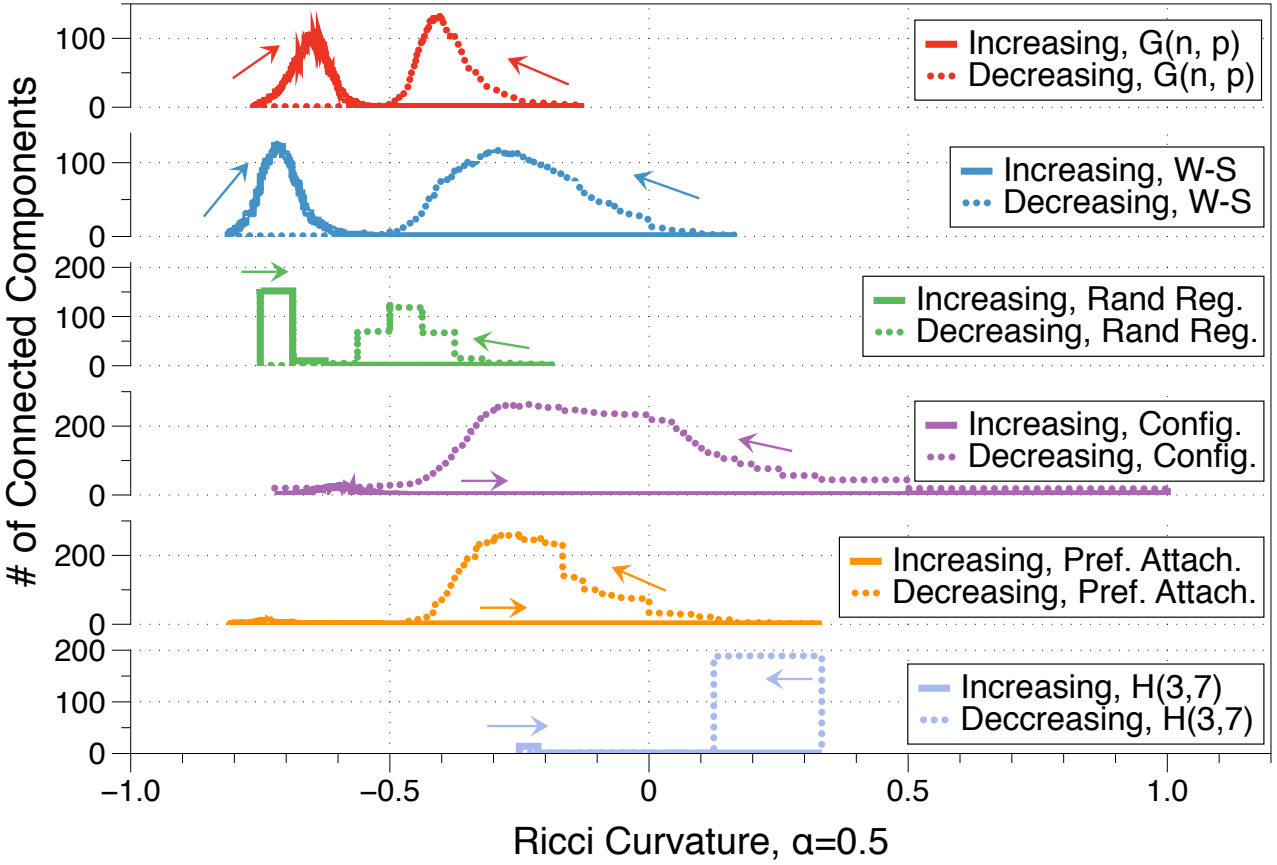
Model Networks

Erdos Renyi Graph, Watts Strogatz Graph, Random Regular Graph, Configuration Model, Preferential Attachment Model, Hyperbolic Grid $H(3,7)$



Network Connectivity Behavior

Small world property is not relevant; graph hyperbolicity and power law degree distribution appear to be more relevant.

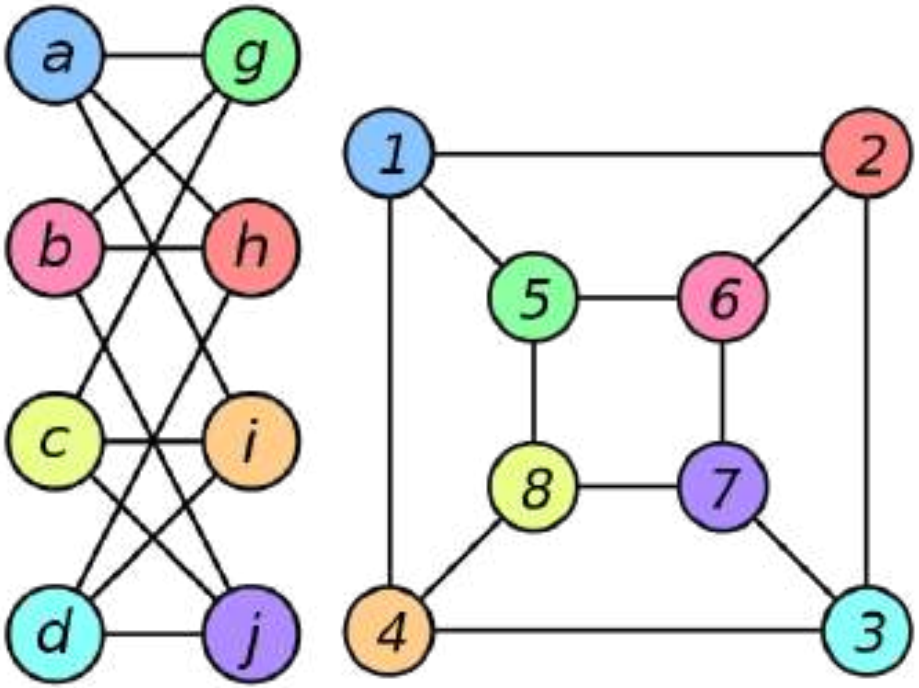


Graph Isomorphism

Given a pair of graphs G_1, G_2, find a one-to-one correspondence of the vertices in G_1 to vertices in G_2 such that (u, v) is an edge in G_1 if and only if their corresponding nodes $f(u), f(v)$ are connected in G_2.

Graph Isomorphism

Given a pair of graphs G_1, G_2, find a one-to-one correspondence of the vertices in G_1 to vertices in G_2 such that (u, v) is an edge in G_1 if and only if their corresponding nodes $f(u), f(v)$ are connected in G_2.



Graph Isomorphism

One of the most fundamental problems in theoretical computer science.

- In NP.

Graph Isomorphism

One of the most fundamental problems in theoretical computer science.

- In NP.
- Nov 2015/Jan 2017, László Babai claimed quasi-polynomial time algorithm: $O(\exp(\log^{O(1)} n))$.

Graph Isomorphism

One of the most fundamental problems in theoretical computer science.

- In NP.
- Nov 2015/Jan 2017, László Babai claimed quasi-polynomial time algorithm: $O(\exp(\log^{O(1)} n))$.
- Many practical algorithms: e.g., NAUTY.

Graph Isomorphism

One of the most fundamental problems in theoretical computer science.

- In NP.
- Nov 2015/Jan 2017, László Babai claimed quasi-polynomial time algorithm: $O(\exp(\log^{O(1)} n))$.
- Many practical algorithms: e.g., NAUTY.
- Subgraph isomorphism is NP-complete.

Graph Isomorphism

One of the most fundamental problems in theoretical computer science.

- In NP.
- Nov 2015/Jan 2017, László Babai claimed quasi-polynomial time algorithm: $O(\exp(\log^{O(1)} n))$.
- Many practical algorithms: e.g., NAUTY.
- Subgraph isomorphism is NP-complete.
- Approximate graph isomorphism: find the best correspondence between vertices in G_1 and G_2.

Applications of Graph Isomorphim

- Understanding network evolution.

Applications of Graph Isomorphim

- Understanding network evolution.
- Match two social networks on the same population – de-anonymization.

Quantify the ‘Position’ of a Node

Select well positioned nodes as *landmarks* and define the position of a node wrt landmarks.

Quantify the ‘Position’ of a Node

Select well positioned nodes as *landmarks* and define the position of a node wrt landmarks.

Q: What distance to use?

Quantify the 'Position' of a Node

Select well positioned nodes as *landmarks* and define the position of a node wrt landmarks.

Q: What distance to use?

- Internet: Measure the delay;
- Social Networks: measure the tie strength; – Trouble: time-consuming, difficult.

Quantify the 'Position' of a Node

Select well positioned nodes as *landmarks* and define the position of a node wrt landmarks.

Q: What distance to use?

- Internet: Measure the delay;
- Social Networks: measure the tie strength; – Trouble: time-consuming, difficult.
- Count # hops to these landmarks – Trouble: small world property;

Quantify the 'Position' of a Node

Select well positioned nodes as *landmarks* and define the position of a node wrt landmarks.

Q: What distance to use?

- Internet: Measure the delay;
- Social Networks: measure the tie strength; – Trouble: time-consuming, difficult.
- Count # hops to these landmarks – Trouble: small world property;

Our Idea: Edge Weights Generated by Ricci flow

Given a graph G in which $d(x, y)$ is the weight of the edge xy and $\kappa(x, y)$ is the discrete Ricci curvature, we run
/* Ricci flow

$$d_{i+1}(x, y) = d_i(x, y) - \varepsilon \cdot \kappa_i(x, y) \cdot d_i(x, y)$$

/* Normalization

$$d_i(x, y) \leftarrow d_i(x, y) \cdot \frac{|E|}{\sum_{\overline{xy} \in E} d_i(x, y)}$$

Until convergence.

Our Idea: Edge Weights Generated by Ricci flow

Given a graph G in which $d(x, y)$ is the weight of the edge xy and $\kappa(x, y)$ is the discrete Ricci curvature, we run
/* Ricci flow

$$d_{i+1}(x, y) = d_i(x, y) - \varepsilon \cdot \kappa_i(x, y) \cdot d_i(x, y)$$

/* Normalization

$$d_i(x, y) \leftarrow d_i(x, y) \cdot \frac{|E|}{\sum_{\overline{xy} \in E} d_i(x, y)}$$

Until convergence.
Ricci Flow Metric: Shortest path metric with edge weights computed above.

[Ni, Lin, G, Gu under submission]

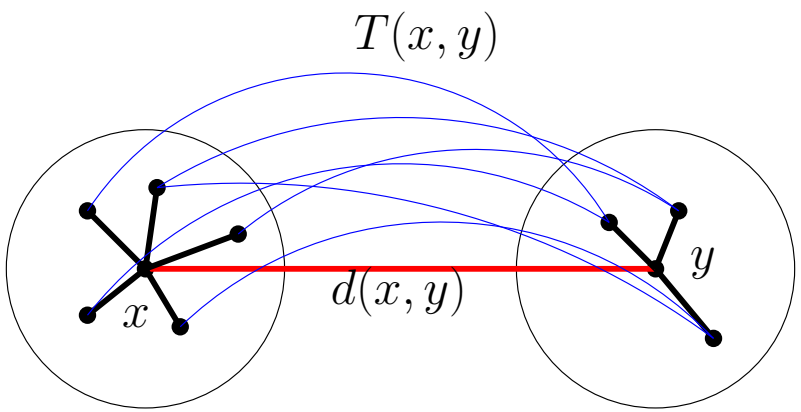
Ricci Flow Metric

$$(d(x,y) - \kappa(x,y) \cdot d(x,y)) \cdot N \approx d(x,y)$$

Ricci Flow Metric

$$(d(x,y) - \kappa(x,y) \cdot d(x,y)) \cdot N \approx d(x,y)$$

$$\frac{T(x,y)}{d(x,y)} \approx \frac{1}{N}.$$



If (x,y) is removed, there are likely alternative paths from x to y that give similar distance. $\rightarrow$ Much better robustness under edge insertion/deletion.

Compute the Correspondence

- Identify k landmarks $\ell_1, \cdots, \ell_k$ on G_1 and their correspondences in G_2.

Compute the Correspondence

- Identify k landmarks $\ell_1, \cdots, \ell_k$ on G_1 and their correspondences in G_2.
- Run discrete Ricci flow on G_1, G_2 respectively.

Compute the Correspondence

- Identify k landmarks $\ell_1, \cdots, \ell_k$ on G_1 and their correspondences in G_2.
- Run discrete Ricci flow on G_1, G_2 respectively.
- Each node u computes the distance vectors $[d(u, \ell_1), \cdots, d(u, \ell_k)]$.

Compute the Correspondence

- Identify k landmarks $\ell_1, \cdots, \ell_k$ on G_1 and their correspondences in G_2.
- Run discrete Ricci flow on G_1, G_2 respectively.
- Each node u computes the distance vectors $[d(u, \ell_1), \cdots, d(u, \ell_k)]$.
- Form a bipartite graph H with X / Y as vertices in G_1/G_2.

Compute the Correspondence

- Identify k landmarks $\ell_1, \cdots, \ell_k$ on G_1 and their correspondences in G_2.
- Run discrete Ricci flow on G_1, G_2 respectively.
- Each node u computes the distance vectors $[d(u, \ell_1), \cdots, d(u, \ell_k)]$.
- Form a bipartite graph H with X / Y as vertices in G_1/G_2.
- The edge weight $(u \in G_1, u' \in G_2)$ is the ℓ_2 norm of their distance vectors.

Compute the Correspondence

- Identify k landmarks $\ell_1, \cdots, \ell_k$ on G_1 and their correspondences in G_2.
- Run discrete Ricci flow on G_1, G_2 respectively.
- Each node u computes the distance vectors $[d(u, \ell_1), \cdots, d(u, \ell_k)]$.
- Form a bipartite graph H with X / Y as vertices in G_1/G_2.
- The edge weight ($u \in G_1, u' \in G_2$) is the ℓ_2 norm of their distance vectors.
- Run min-cost matching on H.

Evaluations

- Edge weights generated by Ricci flow;
- Spectral embedding;
- Spring embedding;
- Hop count.

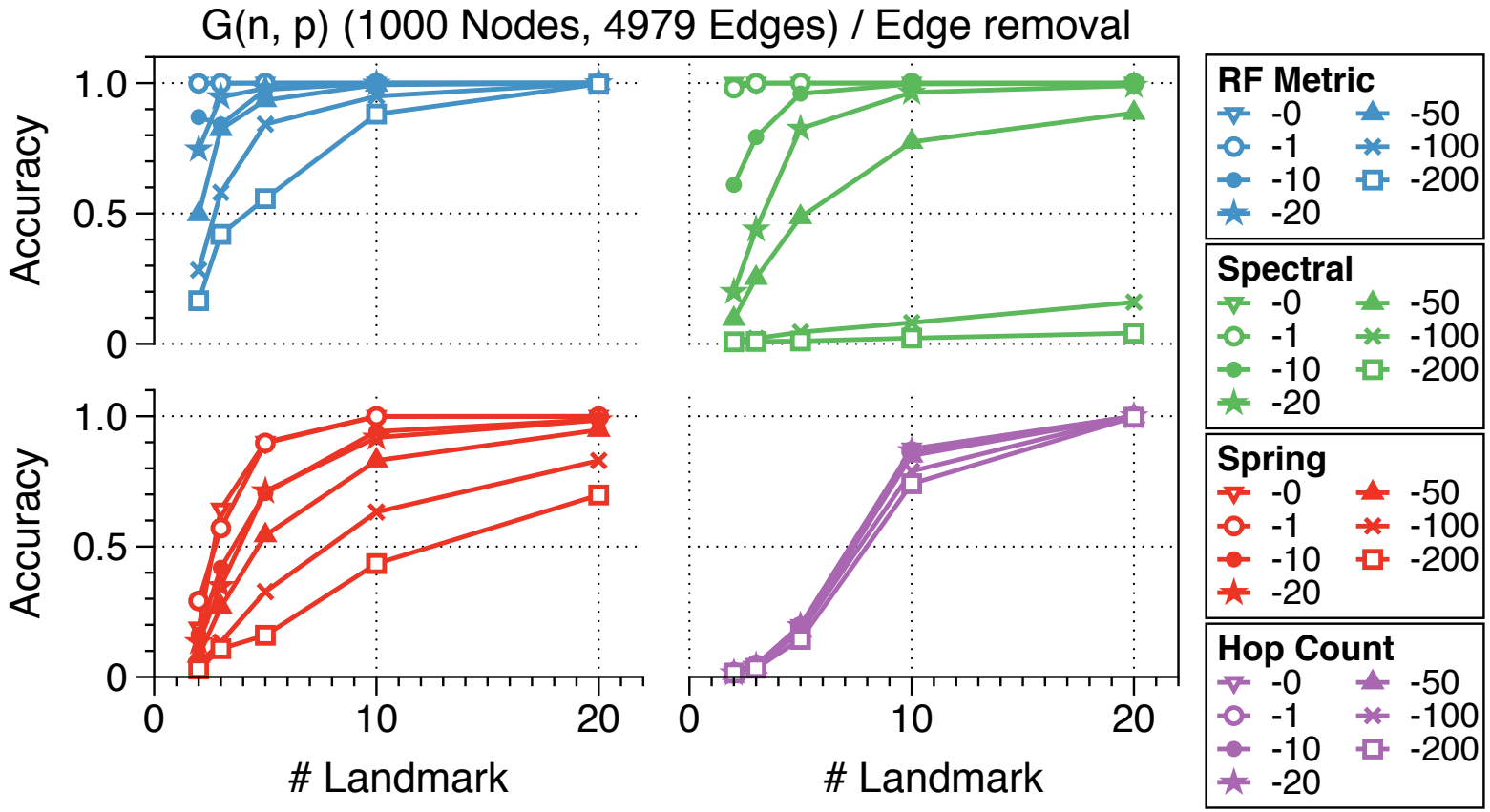
Evaluations

- Edge weights generated by Ricci flow;
- Spectral embedding;
- Spring embedding;
- Hop count.

- Model graphs: random regular graphs, Kleinberg's small world graphs, Erdos-Renyi graphs, preferential attachent graphs.
- Internet AS graphs.
- Email graphs.
- Protein-protein interaction graph.

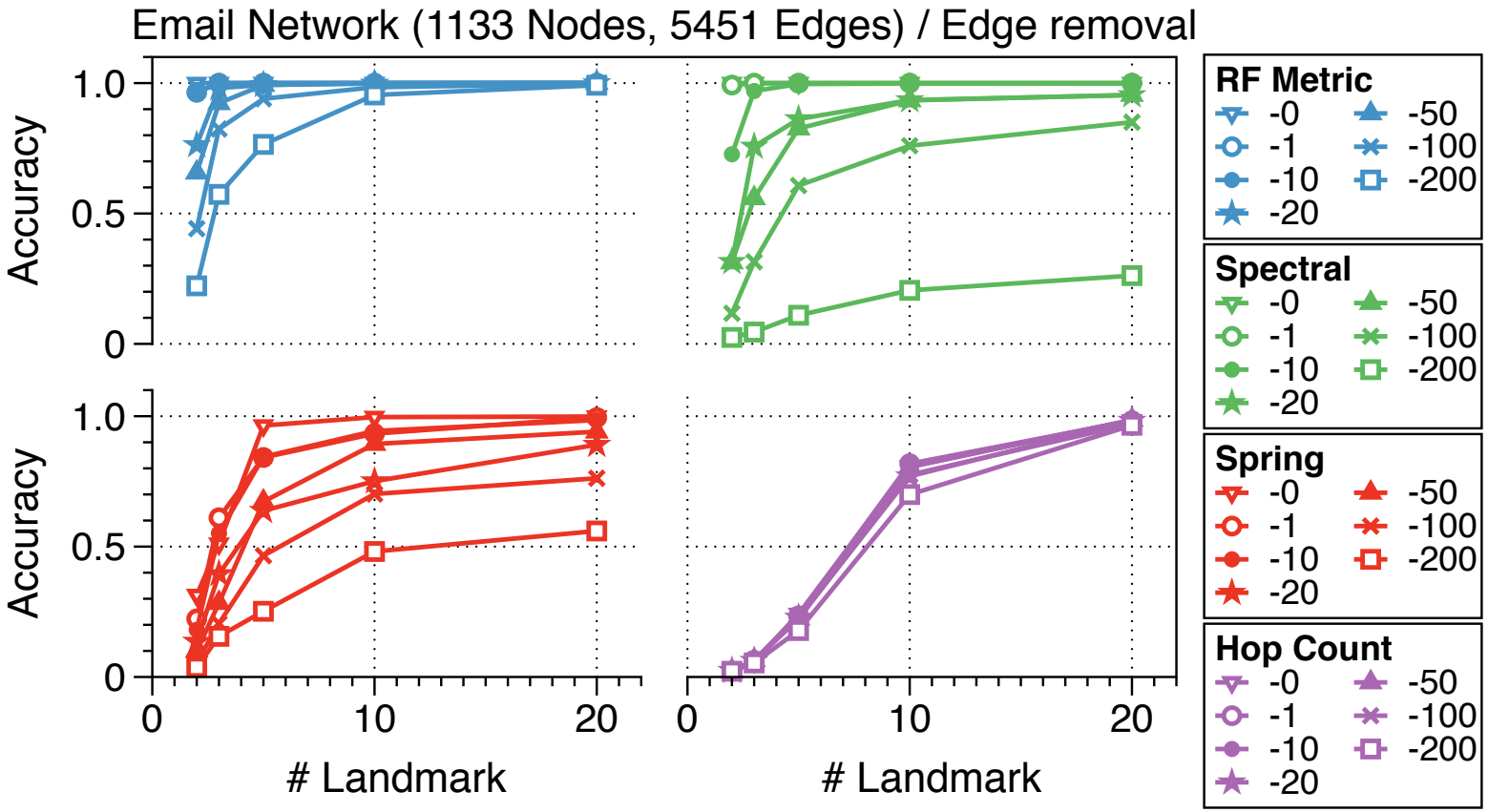
$G(n, p)$

Randomly remove some edges.



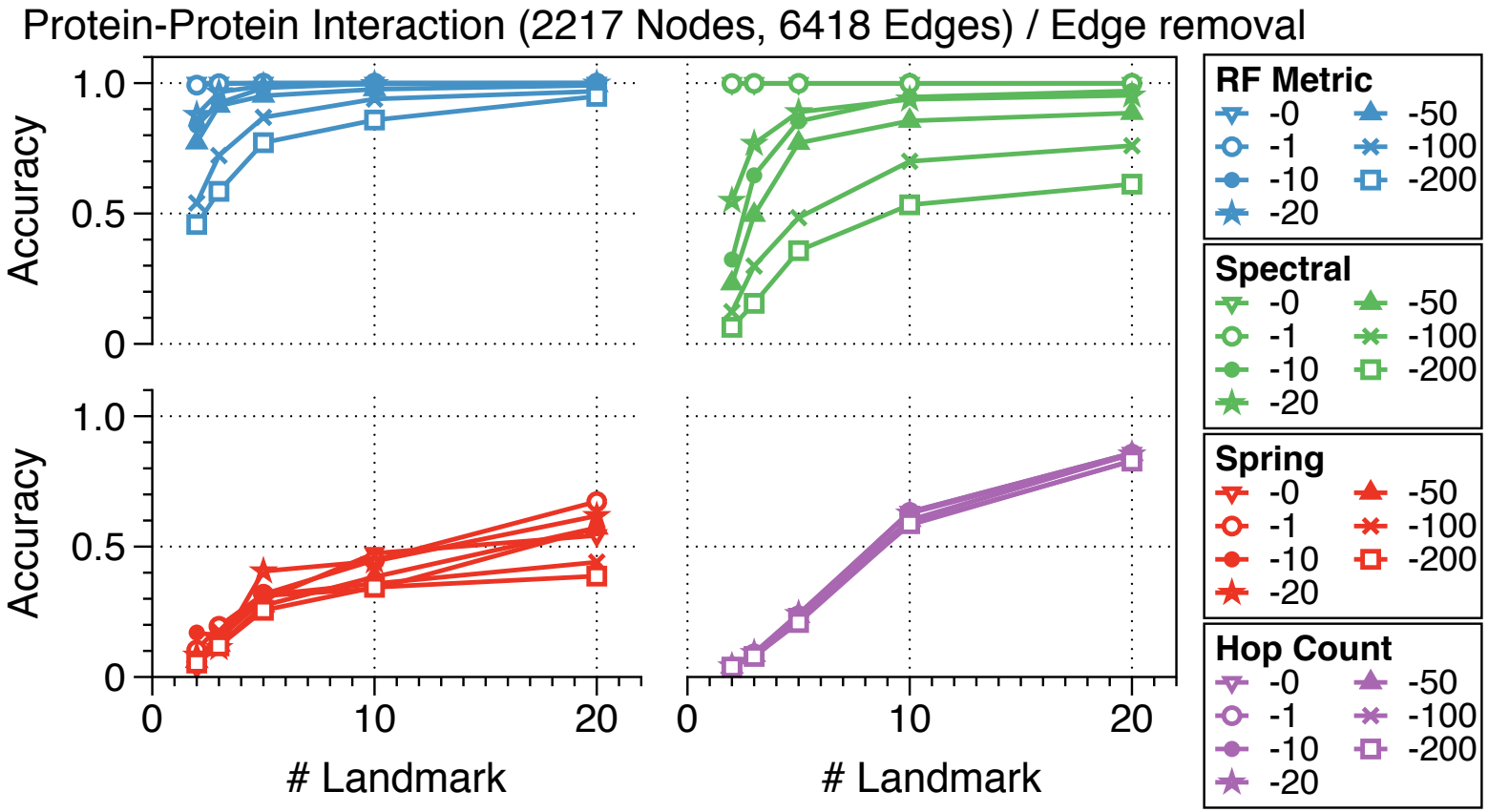
Email networks

Randomly remove some edges.



Protein-Protein networks

Randomly remove some edges.



Conclusion and On-going Work

Geometric approaches for network analysis has great potential.

Conclusion and On-going Work

Geometric approaches for network analysis has great potential.

- Speed up the computation.
- Find good approximation.
- More applications.

Acknowledgement

- David Gu (Stony Brook), Feng Luo (Rutgers), Emil Saucan (Technion).
- Xiaomeng Ban, Mayank Goswami (CUNY), Yu-Yao Lin, Siming Li, Ruirui Jiang, Chien-Chun Ni, Rik Sarkar (Edinburgh), Rui Shi, Xiaotian Yin, Xiaokang Yu, Wei Zeng (Florida).

- https://www3.cs.stonybrook.edu/∼jgao
- Questions and comments?