Initial evaluation of OSVV

<u>Classes of Graphs</u>:

- GM (Guattery-Miller) graph where eigenvector methods fail.
- PLAN - Expanders with planted bisections - where LR is known to fail
- WING - finite element mesh
- RND - Random Geometric Graph
- Random geometric graph with random edges added

	GM100.6	PLAN5	PLAN6	WING	RND-A	A1.12	A3.14	A6.13	A9.10
OSVV-100.100.10	**0.016**	**0.500**	0.778	0.026	0.037	0.134	0.358	0.714	1.102
OSVV-10.10.10	**0.016**	**0.500**	0.781	0.027	0.037	0.131	0.362	**0.707**	1.095
OSVV-1.0.10	**0.016**	0.746	0.793	0.027	0.037	0.131	0.379	1.000	1.141
METISR	**0.016**	**0.500**	0.785	0.027	0.037	0.143	0.372	0.725	**1.060**
LR	**0.016**	1.420	1.475	0.027	0.037	**0.125**	0.405	0.758	1.140
SPECFLOW	0.020	**0.500**	**0.709**	**0.026**	**0.037**	0.143	**0.348**	0.734	1.146
METIS	0.026	0.763	0.801	0.030	0.048	0.180	0.463	0.842	1.123
SPECTRAL	0.020	0.597	0.856	0.032	0.056	0.328	0.654	1.000	1.761

Fig. 2. The best score found by multiple tries (see caption of Figure 3) of each algorithm. First and 2nd-place for each graph are highlighted in red and blue respectively. Scores are given to 3 decimal digits. OSVV parameters are described as OSVV $-\eta.init.s$

	GM100.6	PLAN5	PLAN6	WING	RND-A	A1.12	A3.14	A6.13	A9.10
OSVV-100.100.10	**713.8**	**367.0**	650.0	8166.6	1955.6	955.8	735.4	1315.5	1012.9
OSVV-10.10.10	**363.1**	303.9	437.0	2802.5	880.8	101.1	369.9	**485.4**	850.7
OSVV-1.0.10	**425.6**	2075.0	3030.0	4201.0	601.5	116.6	441.0	85.3	422.8
METISR	**104.9**	**681.5**	699.6	1049.4	109.6	110.7	189.3	283.6	**327.8**
LR	**187.2**	659.8	657.5	8521.1	442.6	**509.2**	699.0	1173.2	1637.4
SPECFLOW	209.3	**636.2**	**580.7**	**4887.3**	**688.0**	639.2	**641.5**	723.6	798.2
METIS	0.01	0.06	0.07	0.09	0.01	0.01	0.02	0.02	0.03
SPECTRAL	7.1	3.2	3.3	51.5	9.0	1.4	3.1	2.3	2.5

Fig. 3. Total run time in seconds for OSVV $- \eta.init.s$ (10 tries). METISR (10000 tries). LR (10 tries). SPECFLOW (Eigensolver $-$ 1000 flow roundings). METIS (1 try). SPECTRAL (Eigensolver + 3 sweep roundings).

Connections with boosting

Iterative nature of "fast ARV" algorithms can be done with cut-matching game

- Cut player - choose bisection (to make game last long)
- Matching player - choose matching to add to *G*, i.e., *G'*=*G*+*M*
- Game stop when *G'* is an expander

Connections b/w game theory, online learning, & boosting

- Freund and Schapire (1996), Warmuth et al (2008)

Online algorithms: practice follows theory quite closely

- **Question**: can this be used as a model to understand statistical properties implicit in approximation algorithms more generally?

Other applications of spectral and flow

Recall: graph partitioning was a "hydrogen atom"

- For studying spectral/flow/etc relaxations to combinatorial problems

- Much of this "spectral" and "flow" structure inherited by approximations to other optimization problem

Spectral: NCut, k-means, Transductive Learning, Modularity relaxations, (esp, in ML), etc.

Flow: Lots of graph approximation algorithms, (in TCS)

Another application of similar ideas: Finding *dense* sub-graphs

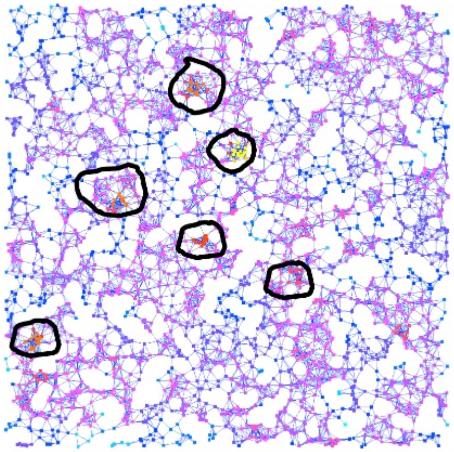
Andersen and Chellapilla (2009), Andersen (2008), Charikar (2000), Kannan and Vinay (1999), GGR (1998), Goldberg (1984), etc.

Definition: Given $G = (V, E)$, an undirected graph, define the *density* $f(S)$ of $S \subset V$ to be

$$f(S) = \frac{|E(S, \bar{S})|}{|S|}.$$

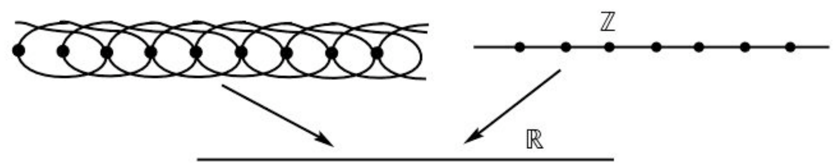
Given $G = (V, E)$, a directed bipartite graph, define the *density* $d(S, T)$ of induced subgraph (S, T) to be

$$d(S,T) = \frac{|E(S,T)|}{\sqrt{|S|}\sqrt{|T|}}.$$



- Optimize f(S) with max-flow or parametric **flow**.
- Greedy approx algorithms optimize f(S) and d(S,T).
- Global/Local **spectral** algs approximate d(S,T) - more amenable to spectral algorithms.

Also, tradeoff dense versus isolated sub-graphs. (Lang and Andersen 2007).



What is the *shape* of a graph?

Can we generalize the following intuition to general graphs:

- A 2D grid or well-shaped mesh "looks like" a 2D plane*
- A random geometric graph "looks like" a 2D plane
- An expander "looks like" a clique or complete graph or a point.

The basic idea:

- If a graph embeds well in another metric space, then it "looks like" that metric space**!

*A "planar graph" is typically a very different combinatorial thing.

**Gromov (1987); Linial, London, & Rabinovich (1985); ISOMAP, LLE, LE, … (2001)

What is the *shape* of a space?

A *long* history:

• Euclid (BC): R^n lengths, angles, dot products, etc come from his Fifth Parallel Lines Postulate

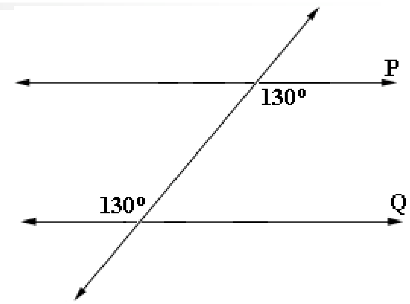
• Bolyai, Lobachevsky etc. (1830s): formulate consistent geometries with other fifth postulates

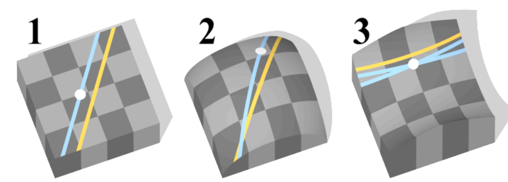
• Riemann (1850s): work on manifolds and curvature more generally

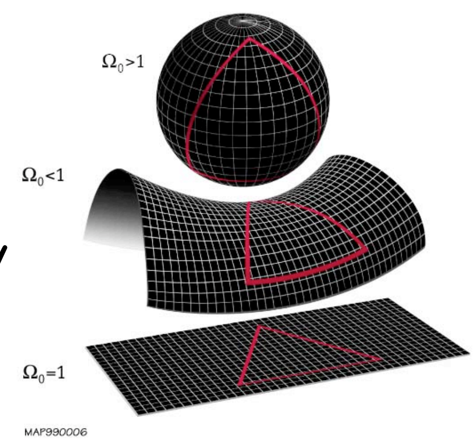
• Einstein (1910s): applications to curvature properties of physical spacetime

• Gromov (1980s): *discrete* curvature and hyperbolicity

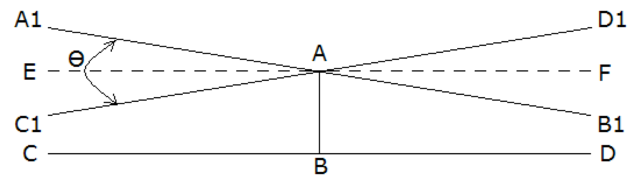
• 1990s and 2000s: applications of network curvature in routing, visualization, embedding, etc.







Hyperbolic Spaces

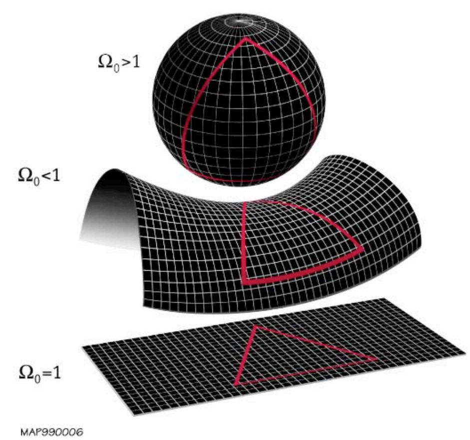


Lobachevsky and Bolyai constructed hyperbolic space - (between a point and a line, there are many "parallel" lines) - Euclid's fifth postulate is independent of the others!

A d-dimensional metric space which is homogeneous and isotropic (looks the same at every point and in every direction) is locally identical to one of:

- Sphere
- Hyperbolic space
- Euclidean plane

} The 3 maximally symmetric geometries

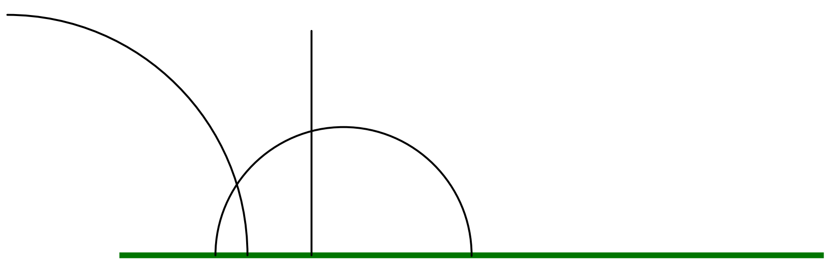


Models of the Hyperbolic Plane

UPPER HALF PLANE MODEL

- Points are $\{z : \mathrm{Im}(z) > 0\}$
- Length of a path z(t) is

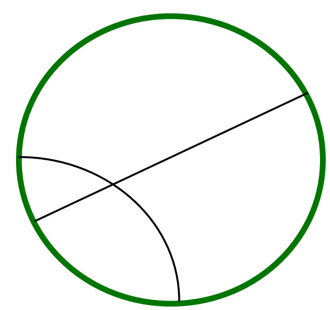
$$\int_0^1 \frac{1}{\mathrm{Im}(z)} \left\| \frac{dz}{dt} \right\| dt$$



POINCARE DISK MODEL

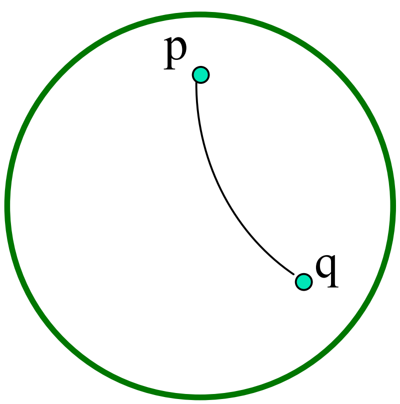
- Points are $\{z : |z| < 1\}$.
- Length of a path z(t) is

$$\int_0^1 \frac{1}{1 - \|z\|^2} \left\| \frac{dz}{dt} \right\| dt$$

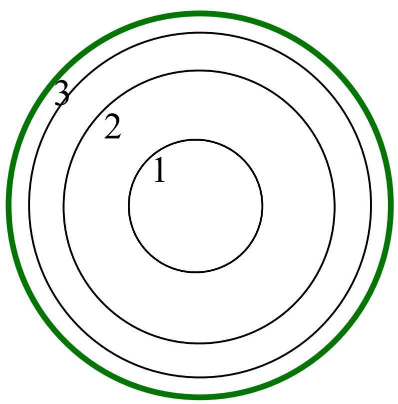


Distances in hyperbolic space

- Vectors are *longer* near the boundary.
- Shortest path from *p* to *q* bends toward the center, where vectors are shorter.
- Geodesics are circular arcs meeting the boundary at right angles.

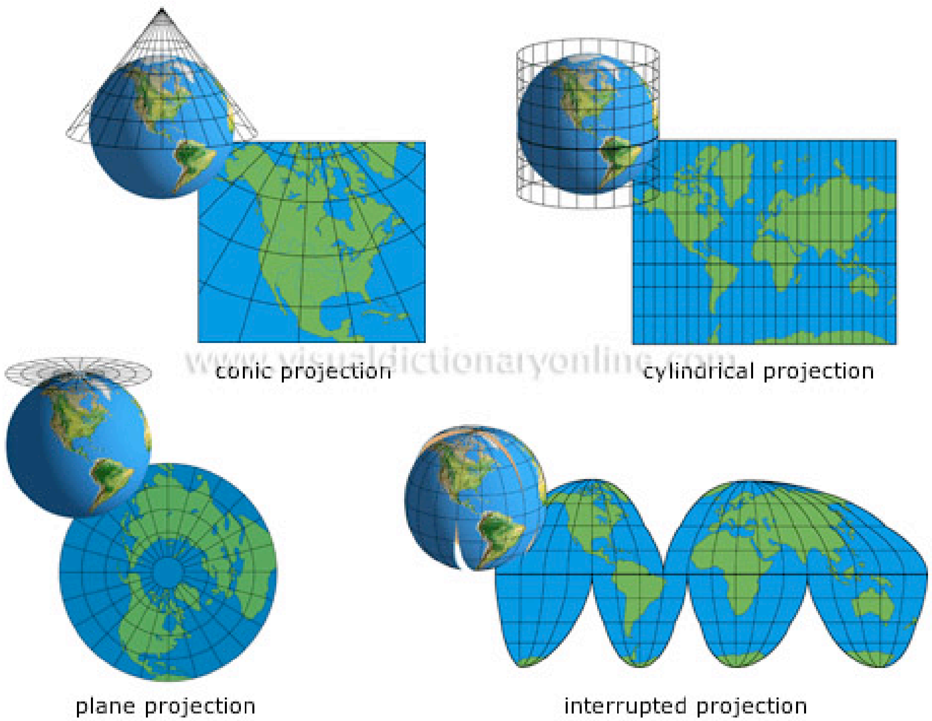


- If you draw circles of hyperbolic radius 1,2,3,… around the center of the Poincare disk, each is ≈ *e* times closer to the boundary than the previous one. Their circumferences grow exponentially!

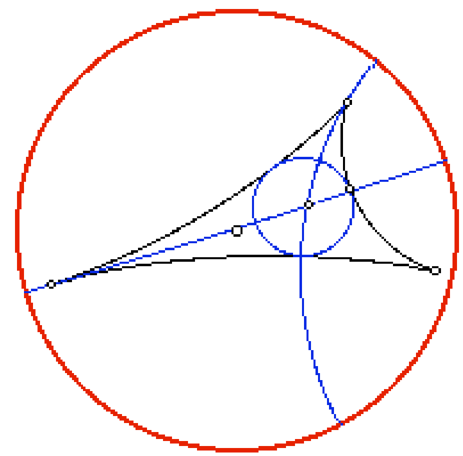


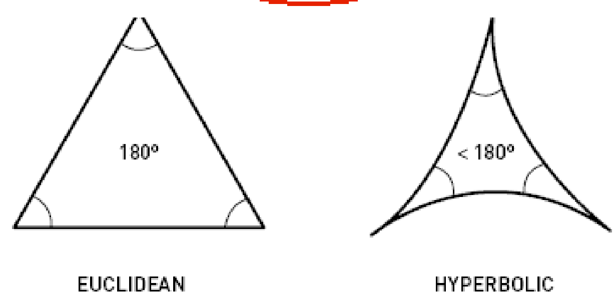
Interpreting visualizations …

Positive curvature:



Negative curvature:

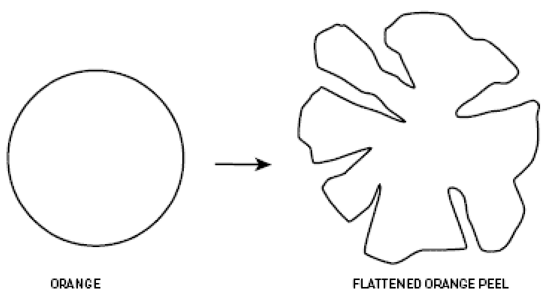




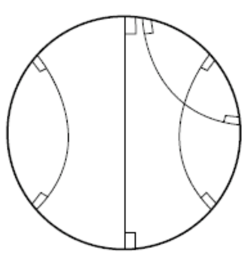
How much space is there in a space?

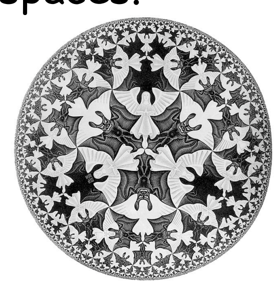
Intuitively,

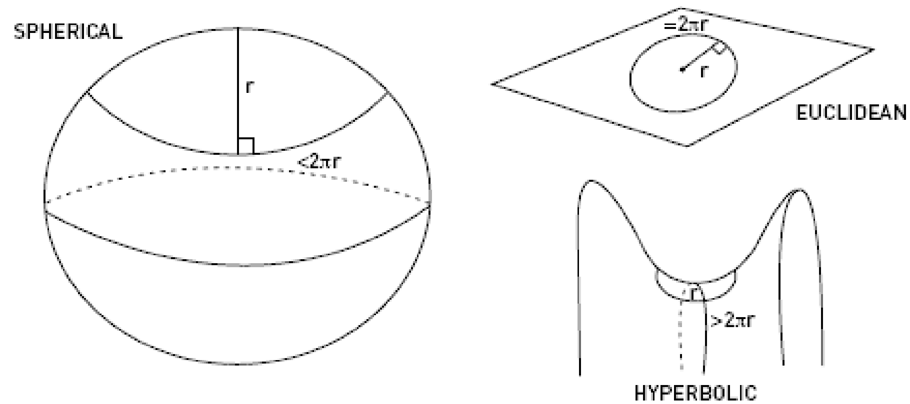
• positively-curved spaces have less space than flat spaces.



• flat spaces have less space than negatively-spaces.

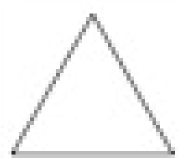
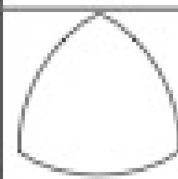
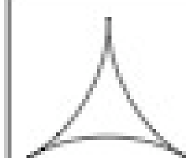






Imagine starting with a flat piece of paper and trying "cover" a sphere (you'll need to crumple it) or a saddle (you'll need to cut it to make room).

Comparison between different curvatures

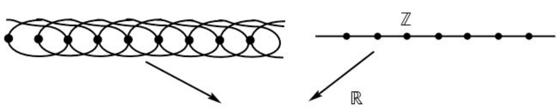
Property	Euclid.	Spherical	Hyperbolic
Curvature	0	1	-1
Parallel lines	1	0	∞
Triangles are	normal	thick	thin
Shape of triangles			
Sum of angles	π	$> \pi$	$< \pi$
Circle length	$2\pi R$	$2\pi \sin R$	$2\pi \sinh R$
Disc area	$2\pi R^2/2$	$2\pi(1 - \cos R)$	$2\pi(\cosh R - 1)$

Discrete vs. continuous

See: "Discrete Geometric Analysis," T. Sunada (2007)

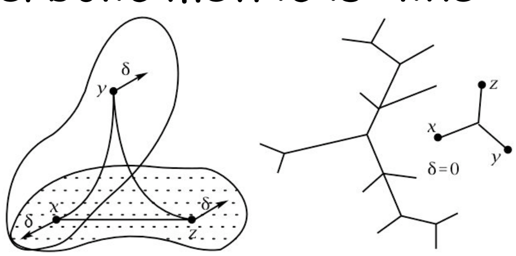
"Squint" at data with "coarse embedding"

- Line graph is "like" a line (random geometric graph is like underlying geometry).



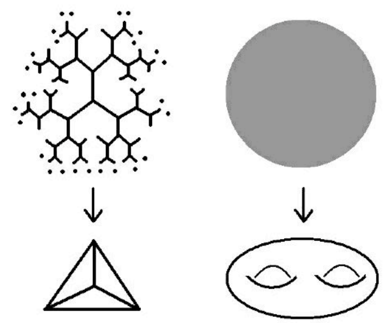
- Expander is "like" a complete graph. (Hard to visualize.)

- Hyperbolic metric is "like" tree!



A striking example of analogy

Regular tree and Poincare disc



Graph Theory	Geometry
a regular tree X	the unit disc D with the Poincaré metric
automorphism group of X	isometry group of H
a finite regular graph	a closed Riemann surface with constant negative curvature
discrete Laplacian on X	Laplacian Δ on D
paths without backtracking	geodesics
spherical functions on X	spherical functions on H
Ihara's zeta function for a finite regular graph	Selberg's zeta function for a closed Riemann surface

δ-hyperbolic metric spaces

Definition: [Gromov, 1987] A graph is δ-*hyperbolic* iff: For every 4 vertices u, v, w, and z, the larger 2 of the 3 distance sums, $d(u,v)+d(w,z)$ and $d(u,w)+d(v,z)$ and $d(u,z)+d(v,w)$, differ by at most 2δ.

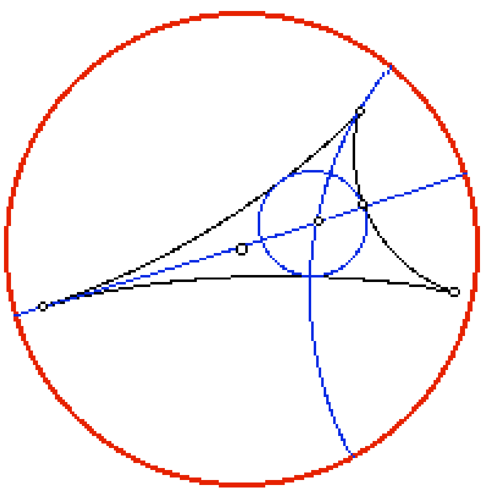
Things to note about δ-hyperbolicity:

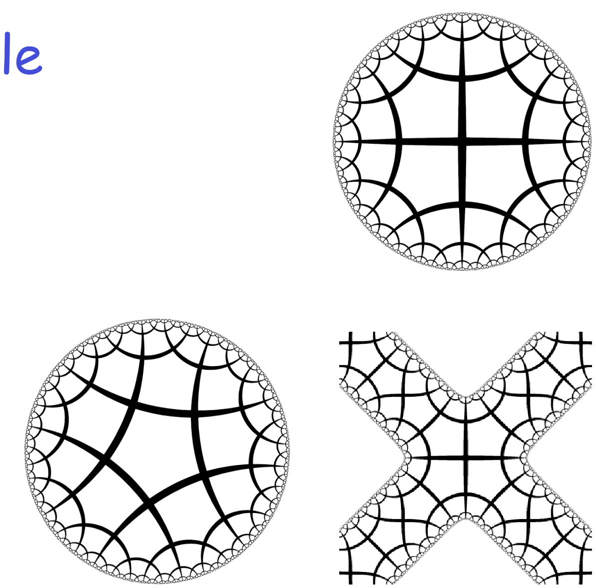
- Graph property that is both *local* (by four points) and *global* (by the distance) in the graph
- Polynomial time computable - naively in $O(n^4)$ time
- Metric space embeds into a tree iff $\delta = 0$.
- Poincare half space in R^k is δ-hyperbolic with $\delta = \log_2 3$
- Theory of δ-hyperbolic spaces generalize theory of Riemannian manifold with negative sectional curvature to metric spaces

δ-hyperbolic metric spaces, cont.

Theory of δ-hyperbolic spaces generalize theory of Riemannian manifold with negative sectional curvature to metric spaces.

- Measures *deviation from tree-ness* of a discrete space
- Equivalent definition in terms of δ-thin triangle condition:





Expanders and hyperbolicity

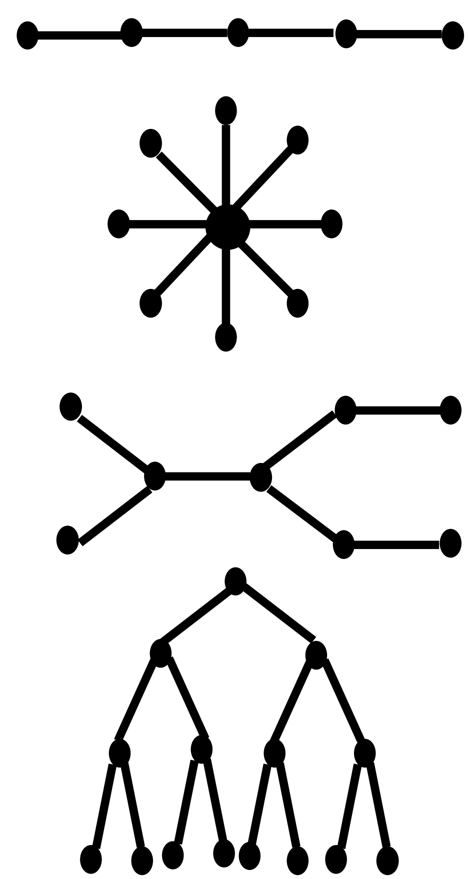
Different concepts that really are different (Benjamini 1998) :

- Constant-degree expanders - like sparsified complete graphs
- Hyperbolic metric space - like a tree-like graph

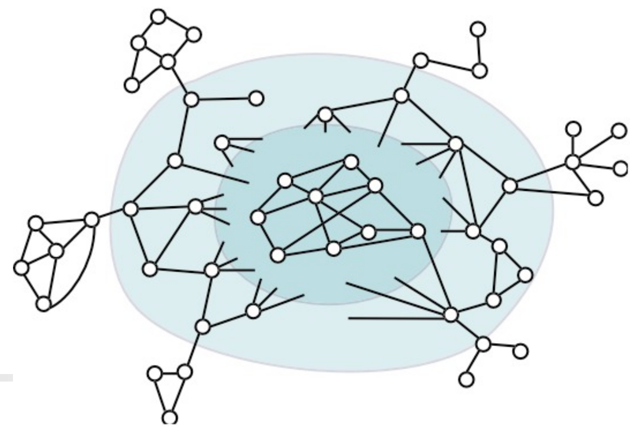
But, *degree heterogeneity enhances hyperbolicity** (so real networks will often have both properties).

*Question: Does anyone know a reference that makes these connections precise?

Trees come in all sizes and shapes:



Overview



Popular algorithmic tools with a geometric flavor

- PCA, SVD; interpretations, kernel-based extensions; algorithmic and statistical issues; and limitations

Graph algorithms and their geometric underpinnings

- Spectral, flow, multi-resolution algorithms; their implicit geometric basis; global and scalable local methods; expander-like, tree-like, and hyperbolic structure

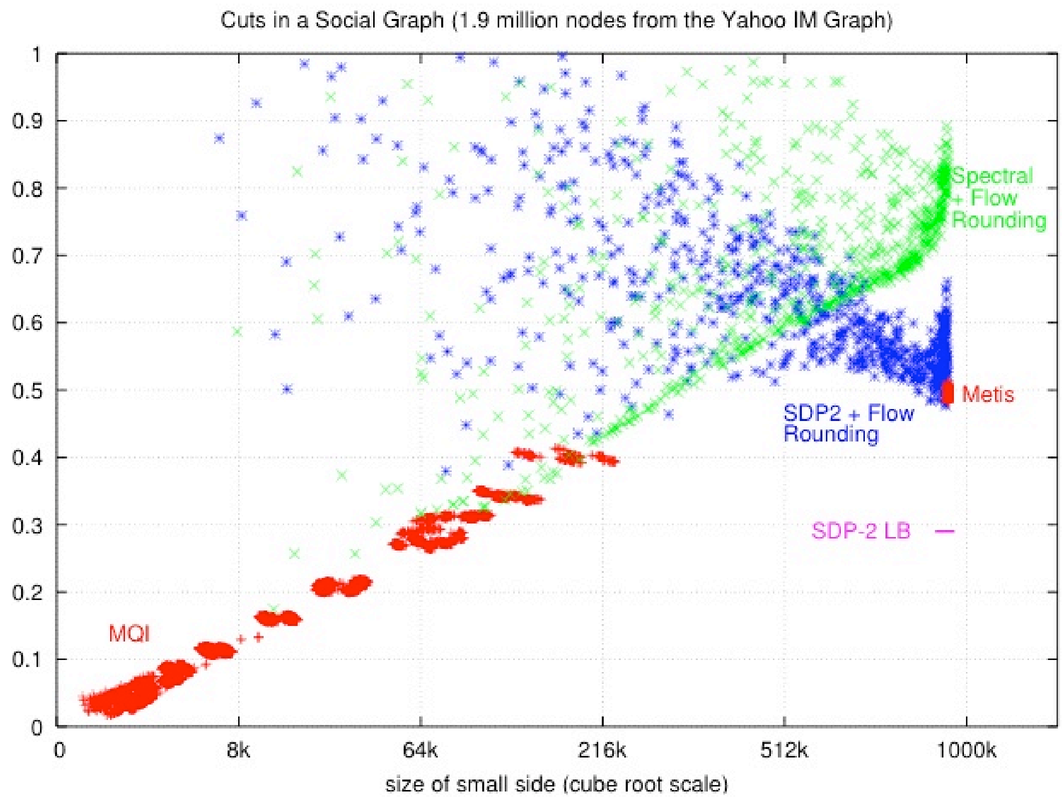
Novel insights on structure in large informatics graphs

- Successes and failures of existing models; empirical results, including "experimental" methodologies for probing network structure, taking into account algorithmic and statistical issues; implications and future directions

An awkward empirical fact

Lang (NIPS 2006), Leskovec, Lang, Dasgupta, and Mahoney (WWW 2008 & arXiv 2008)

Can we cut "internet graphs" into two pieces that are "nice" *and* "well-balanced"?



For *many* **real-world** social-and-information "power-law graphs," there is an *inverse relationship* between "cut quality" and "cut balance."

Consequences of this empirical fact

Relationship b/w small-scale structure and large-scale structure in social/information networks* is not reproduced (even qualitatively) by popular models

- This relationship governs diffusion of information, routing and decentralized search, dynamic properties, etc., etc., etc.

- This relationship also governs (implicitly) the applicability of nearly every common data analysis tool in these apps

*Probably *much* more generally--social/information networks are just so messy and counterintuitive that they provide very good methodological test cases.

Questions of interest ...

What are degree distributions, clustering coefficients, diameters, etc.?

Heavy-tailed, small-world, expander, geometry+rewiring, local-global decompositions, ...

Are there natural clusters, communities, partitions, etc.?

Concept-based clusters, link-based clusters, density-based clusters, ...

(e.g., *isolated* micro-markets with *sufficient money/clicks* with *sufficient coherence*)

How do networks grow, evolve, respond to perturbations, etc.?

Preferential attachment, copying, HOT, shrinking diameters, ...

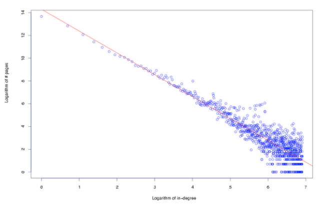
How do dynamic processes - search, diffusion, etc. - behave on networks?

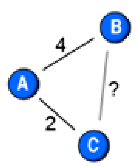
Decentralized search, undirected diffusion, cascading epidemics, ...

How best to do learning, e.g., classification, regression, ranking, etc.?

Information retrieval, machine learning, ...

Popular approaches to network analysis



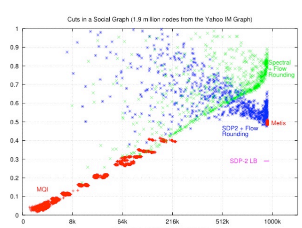


Define simple statistics (clustering coefficient, degree distribution, etc.) and fit simple models

• more complex statistics are too algorithmically complex or statistically rich

• fitting simple stats often doesn't capture what you wanted

Beyond very simple statistics:



• Density, diameter, routing, clustering, communities, …

• *Popular models often fail egregiously at reproducing more subtle properties (even when fit to simple statistics)*

Failings of "traditional" network approaches

Three recent examples of *failings* of *"small world" and "heavy tailed" approaches*:

• Algorithmic decentralized search - solving a (non-ML) problem: can we find short paths?

• Diameter and density versus time - simple dynamic property

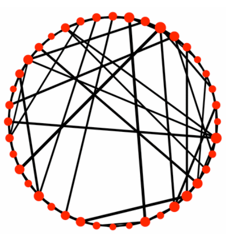
• Clustering and community structure - subtle/complex static property (used in downstream analysis)

All three examples have to do with the **coupling b/w "local" structure and "global" structure** --- *solution* goes *beyond simple statistics of traditional approaches.*

Failing 1: Search in social graphs

Milgram (1960s)

- Small world experiments - study short paths in social networks
- Individuals from Midwest forward letter to people they know to get it to an individual in Boston.



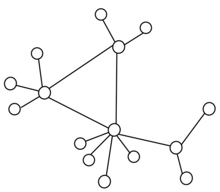
Watts and Strogatz (1998)

- "Small world" model, i.e., add random edges to an underlying local geometry, reproduces local clustering and existence of short paths

Kleinberg (2000)

- But, even Erdos-Renyi G_{np} random graphs have short paths …
- … so the existence of short paths is not so interesting
- Milgram's experiment also demonstrated people found those paths

Failing 2: Time evolving graphs



Albert and Barabasi (1999)

- "Preferential attachment" model, i.e., at each time step add a constant number of links according to a "rich-get-richer" rule
- Constant average degree, i.e., average node degree remains constant
- Diameter increases roughly logarithmically in time

Leskovec, Kleinberg, and Faloutsos (2005)

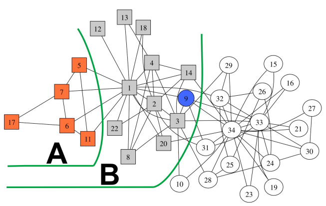
- But, *empirically*, graphs densify over time (i.e., number of edges grows superlinearly with number of nodes) and diameter shrinks over time

Failing 3:
Clustering and community structure

Sociologists (1900s)

• A "community" is any group of two or more people that is useful

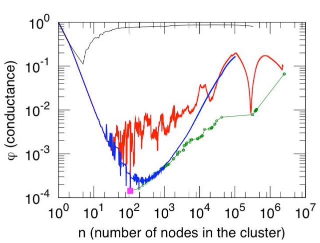
Girvan and Newman (2002,2004) and ***MANY*** *others*



• A "community" is a set of nodes "joined together in tightly-knit groups between which there are only loose connections

• Modularity becomes a popular "edge counting" metric

Leskovec, Lang, Dasgupta, and Mahoney (2008)

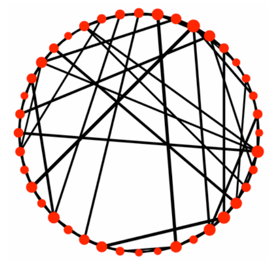


• *All* work on community detection validated on networks with good well-balanced partitions (i.e., low-dimensional and not expanders)

• **But**, *empirically*, larger clusters/communities are less-and-less cluster-like than smaller clusters (i.e., networks are expander-like)

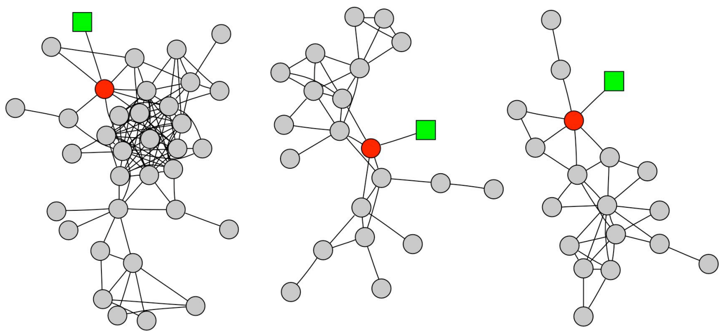
Interplay between preexisting versus generated versus implicit geometry

Preexisting geometry



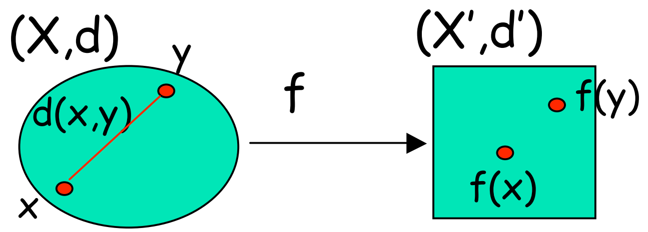
- Start with geometry and add "stuff"

Generated geometry



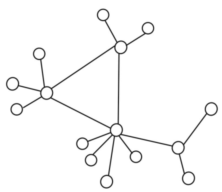
- Generative model leads to structures that are meaningfully-interpretable as geometric

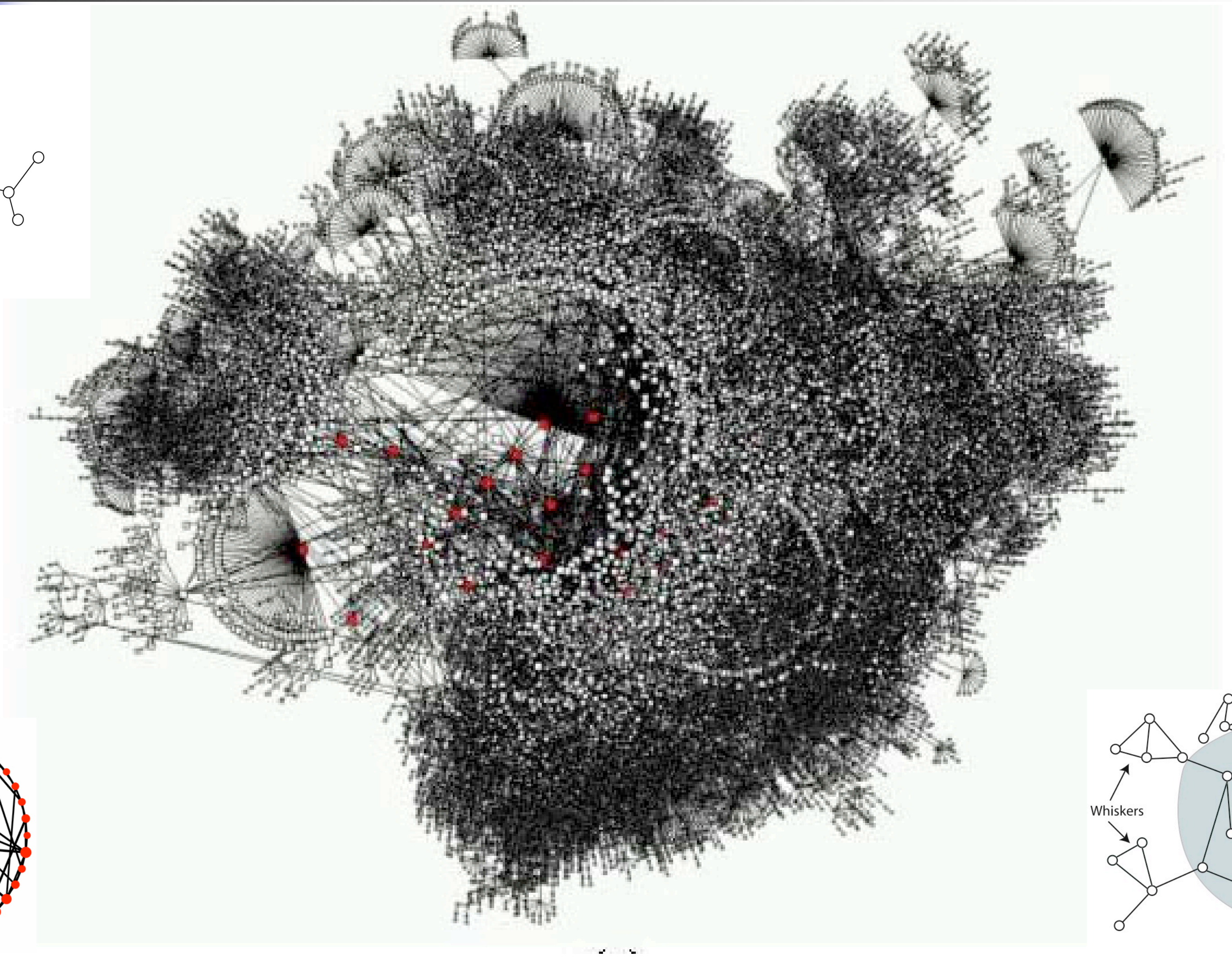
Implicitly-imposed geometry

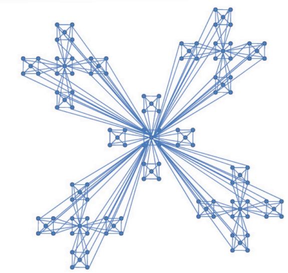


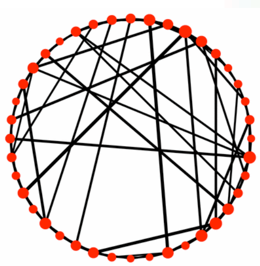
- Approximation algorithms implicitly embed the data in a metric/geometric place and then round.

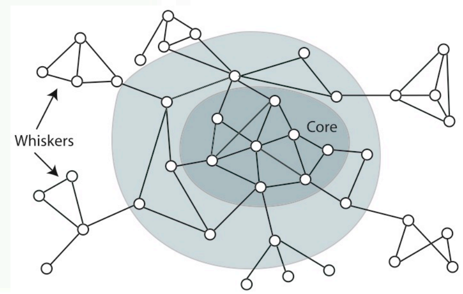
What do these networks "look" like?











Approximation algorithms as experimental probes?

Usual *modus operandi* for approximation algorithms for general problems:

• define an objective, the numerical value of which is intractable to compute

• develop approximation algorithm that returns approximation to that number

• graph achieving the approximation may be unrelated to the graph achieving the exact optimum.

But, for randomized approximation algorithms with a geometric flavor (e.g. matrix, regression, eigenvector algorithms; duality algorithms, etc):

• often can approximate the vector achieving the exact solution

• randomized algorithms compute an ensemble of answers -- the details of which depend on choices made by the algorithm

• maybe compare different approximation algorithms for the same problem.

Exptl Tools: Probing Large Networks with Approximation Algorithms

Idea: Use approximation algorithms for NP-hard graph partitioning problems as experimental probes of network structure.

Spectral - (quadratic approx) - confuses "long paths" with "deep cuts"

Multi-commodity flow - (log(n) approx) - difficulty with expanders

SDP - (sqrt(log(n)) approx) - best in theory

Metis - (multi-resolution for mesh-like graphs) - common in practice

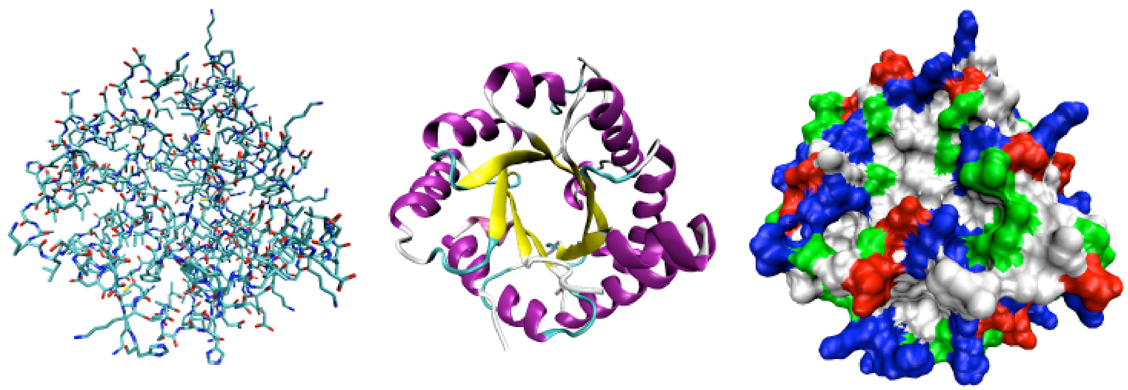
X+MQI - post-processing step on, e.g., Spectral of Metis

Metis+MQI - best conductance (empirically)

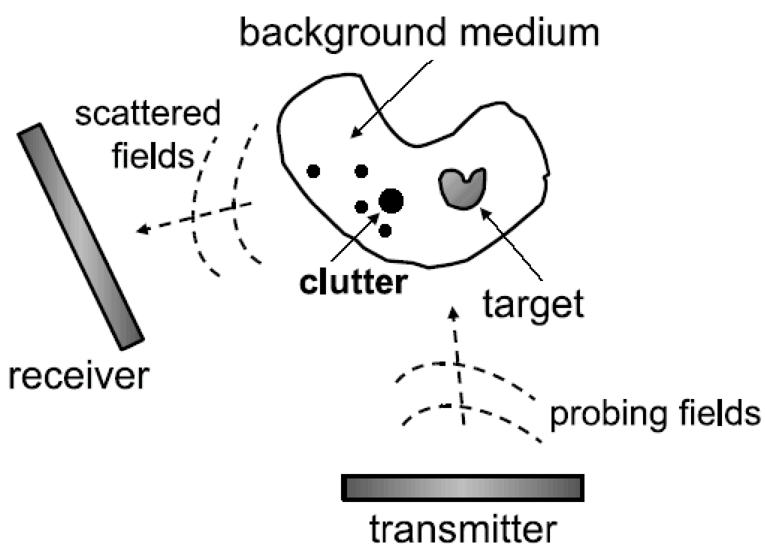
Local Spectral - connected and tighter sets (empirically, regularized communities!)

We are not interested in partitions per se, but in probing network structure.

Analogy: What does a protein look like?



Three possible representations (all-atom; backbone; and solvent-accessible surface) of the three-dimensional structure of the protein triose phosphate isomerase.



Experimental Procedure:

- Generate a bunch of output data by using the unseen object to filter a known input signal.
- Reconstruct the unseen object given the output signal and what we know about the artifactual properties of the input signal.

Experimenting with data with CS tools

- Networks as non-engineered phenomena to be studied as a natural/physical scientist would. (Jon Kleinberg 2006)
- The emergence of cyberspace and the WWW is like the discovery of a new continent. (Jim Gray 1998)
- Want Kepler's Laws of Motion for the Web. (Mike Steuerwalt 1998)

To study data "scientifically," you need

- "Experimental" data (and hopefully lots of it)
- "Experimental" tools (that do the job well)

Use approximation algorithms (*and their implicit statistical properties*) as **experimental tools**!

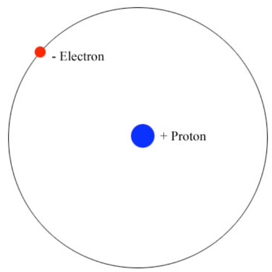
Why graph partitioning? (2 of 2)

Graph partitioning algorithms:

• tools to "experimentally probe" network structure

• "scalable" and "robust" way to explore extremely non-Euclidean structures in data

• primitive for machine learning and data analysis applications, e.g., image partitioning, semi-supervised learning, etc

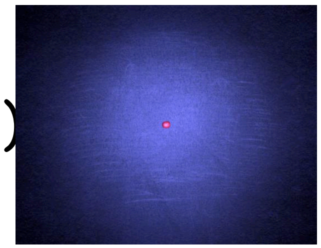


For data more generally:

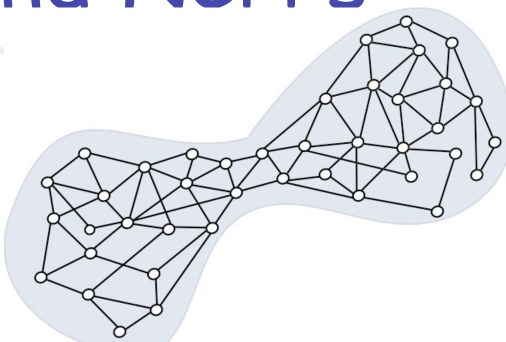
• "hydrogen atom" for theory/practice disconnect

• "hydrogen atom" for algorithmic vs statistical perspectives

• "hydrogen atom" for regularization implicit in graph algorithms (where you can't "cheat" by data preprocessing)



Communities, Conductance, and NCPPs



Let A be the adjacency matrix of G=(V,E).

The conductance ϕ of a set S of nodes is:

$$\phi(S) = \frac{\sum_{i \in S, j \notin S} A_{ij}}{\min\{A(S), A(\overline{S})\}}$$

$$A(S) = \sum_{i \in S} \sum_{j \in V} A_{ij}$$

The Network Community Profile (NCP) Plot of the graph is:

$$\Phi(k) = \min_{S \subset V, |S| = k} \phi(S)$$

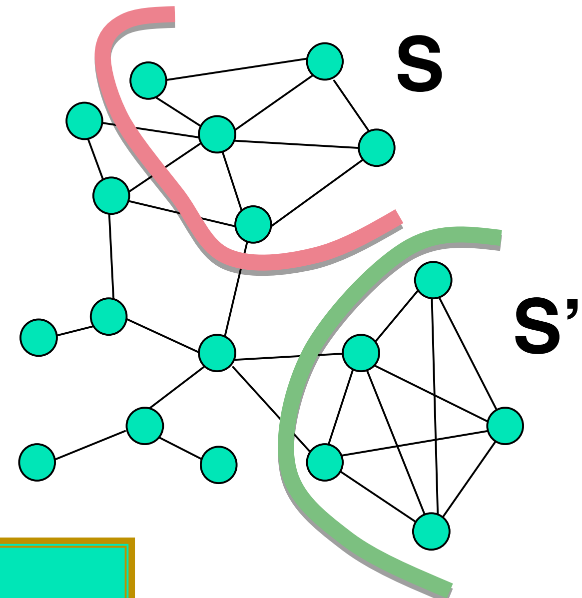
Since algorithms often have non-obvious size-dependent behavior.

*Just as conductance captures the "gestalt" notion of cluster/community quality, the NCP plot measures cluster/community quality **as a function of size**.*

NCP is intractable to compute --> use approximation algorithms!

Community Score: Conductance

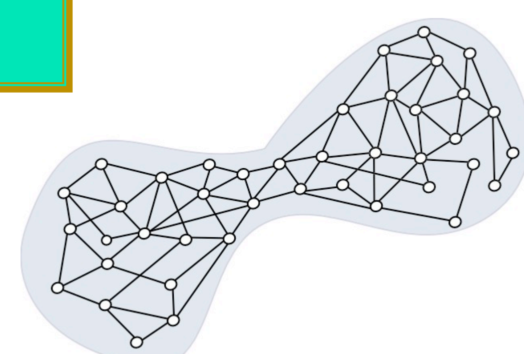
- How community like is a set of nodes?
- Need a natural intuitive measure:



- **Conductance** (normalized cut)

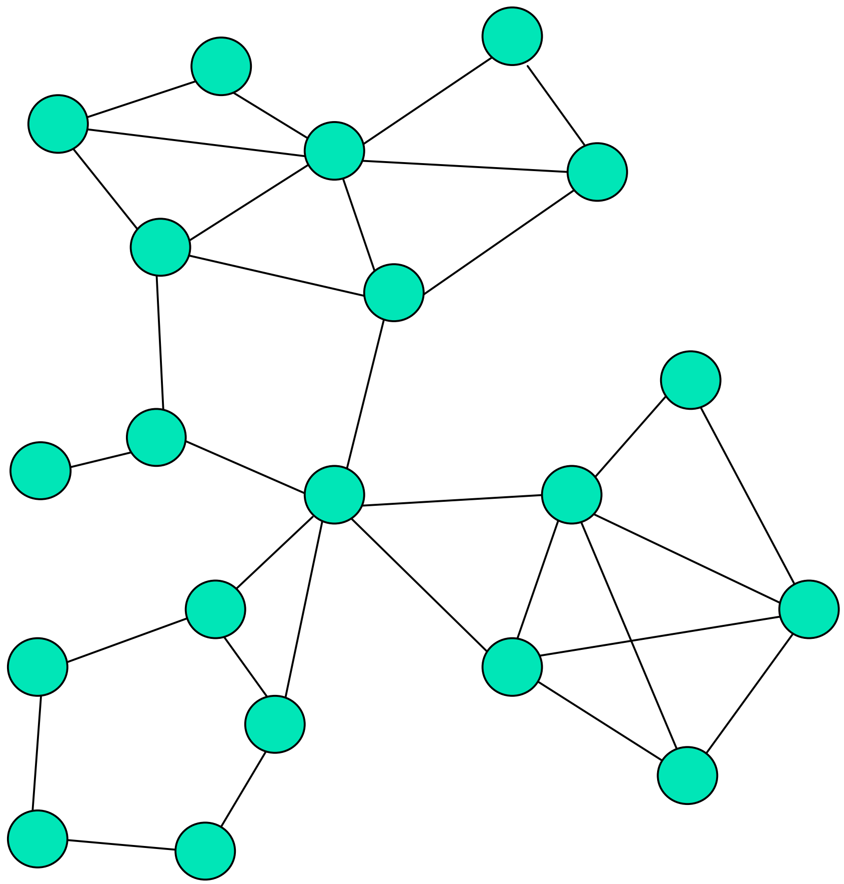
$\phi(S) \approx$ # edges cut / # edges inside

- Small $\phi(S)$ corresponds to more community-like sets of nodes



Community Score: Conductance

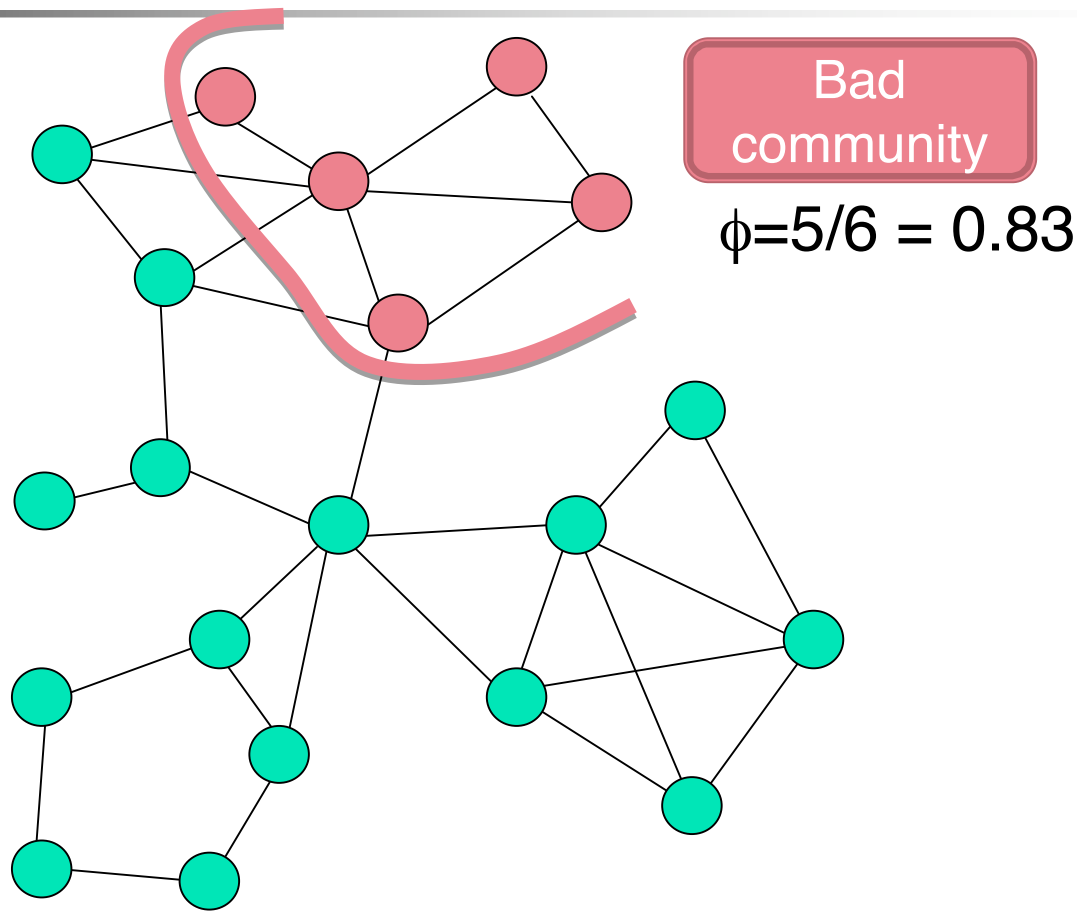
What is "best" community of 5 nodes?



Score: $\phi(S)$ **= # edges cut / # edges inside**

Community Score: Conductance

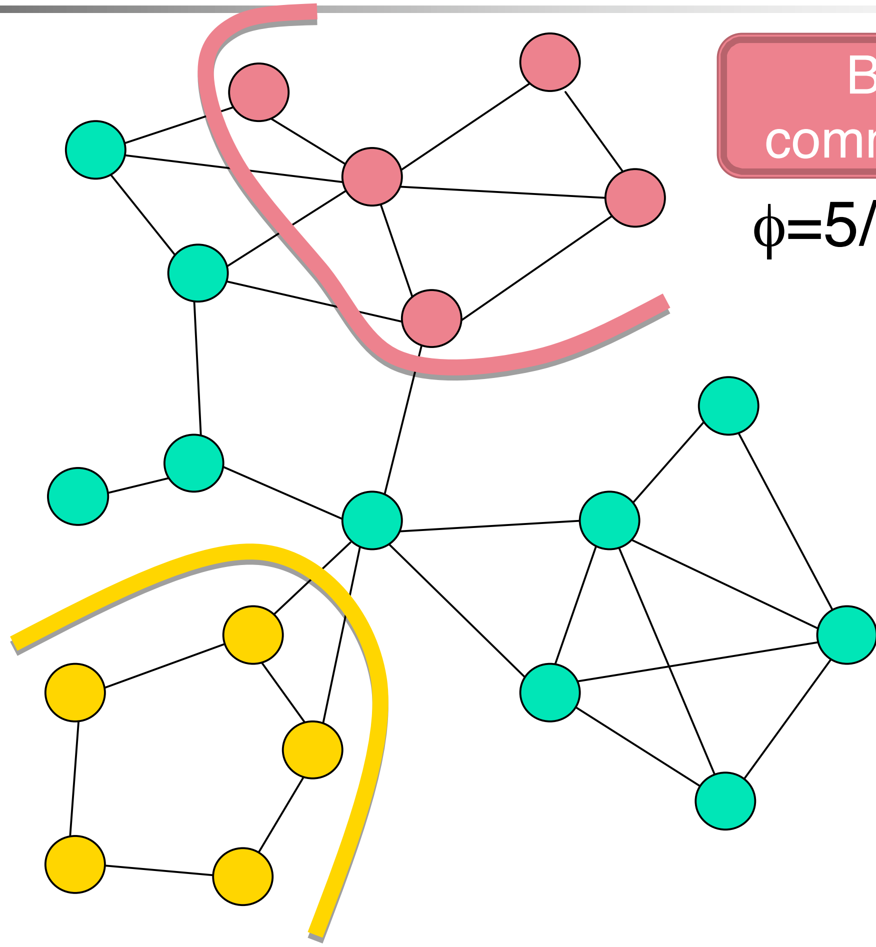
What is "best" community of 5 nodes?



Score: $\phi(S)$ = # edges cut / # edges inside

Community Score: Conductance

What is "best" community of 5 nodes?



Bad community

ϕ=5/6 = 0.83

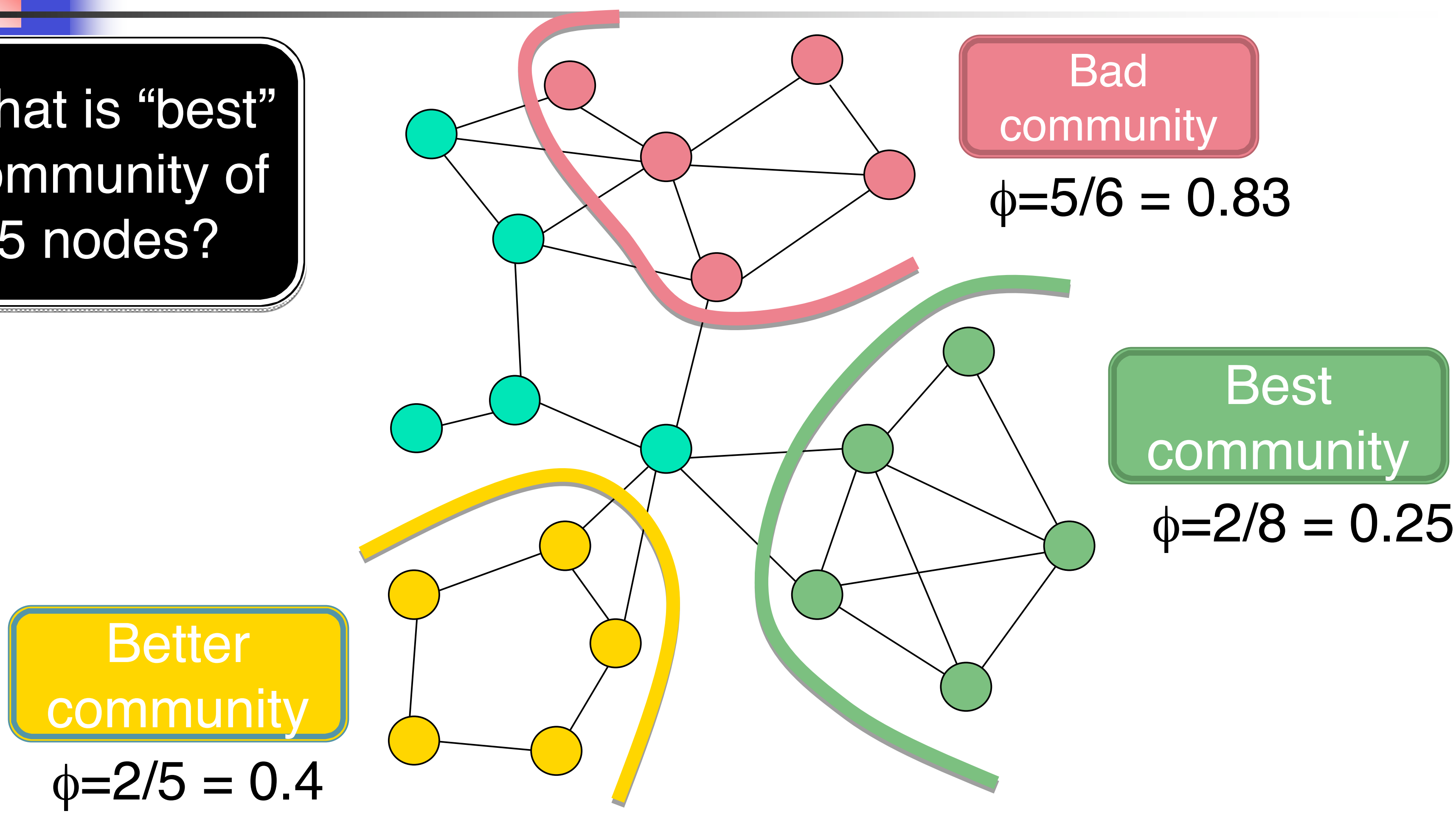
Better community

ϕ=2/5 = 0.4

Score: ϕ(S) = # edges cut / # edges inside

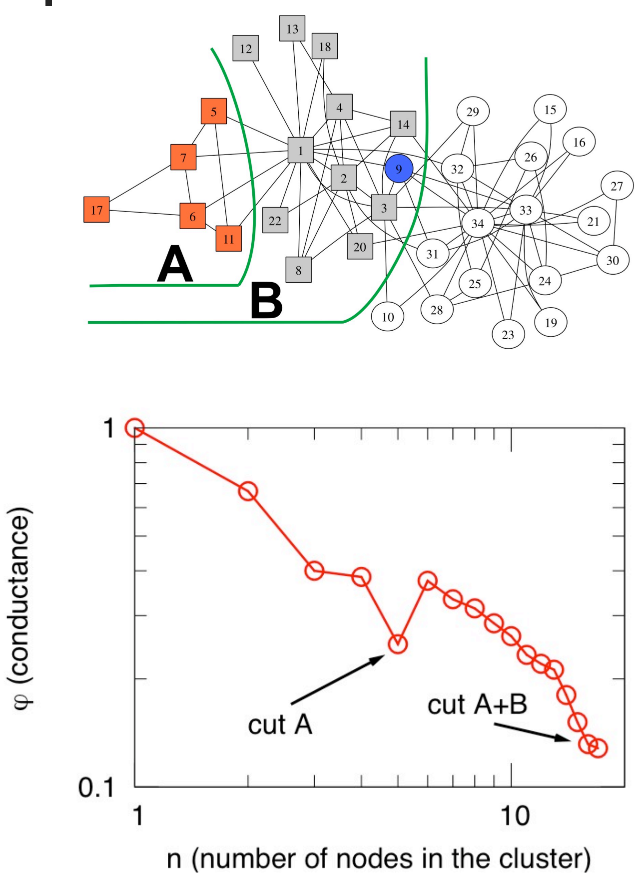
Community Score: Conductance

What is "best" community of 5 nodes?

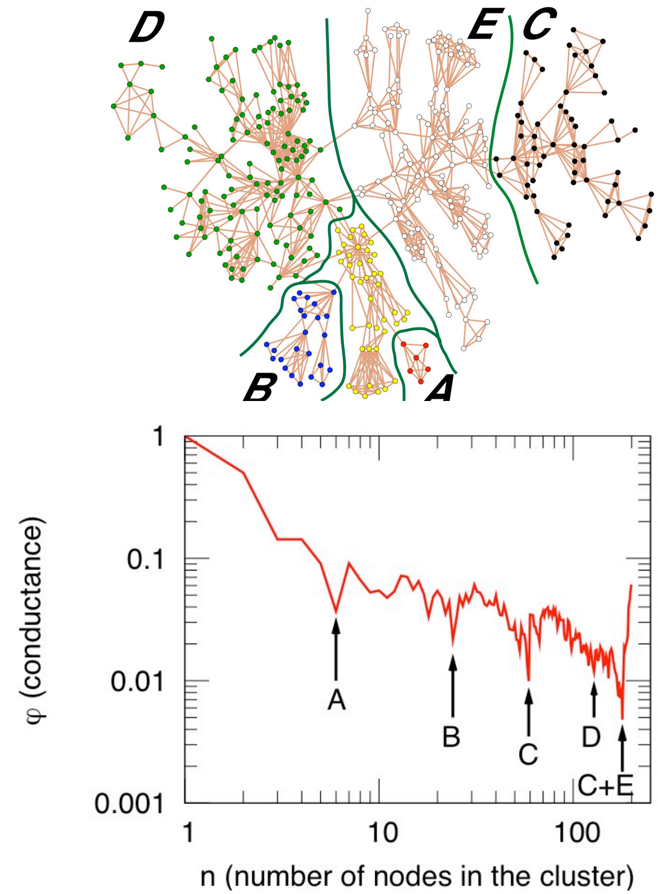


Score: ϕ(S) = # edges cut / # edges inside

Widely-studied small social networks

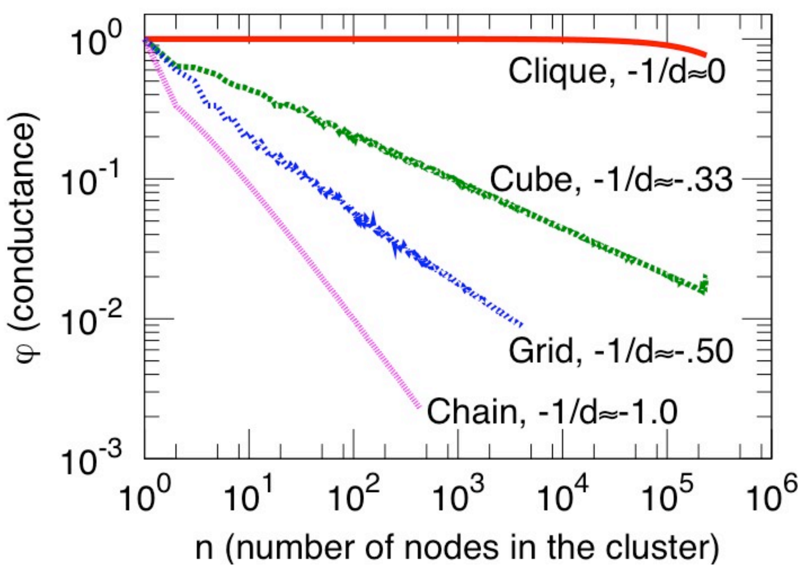


Zachary's karate club

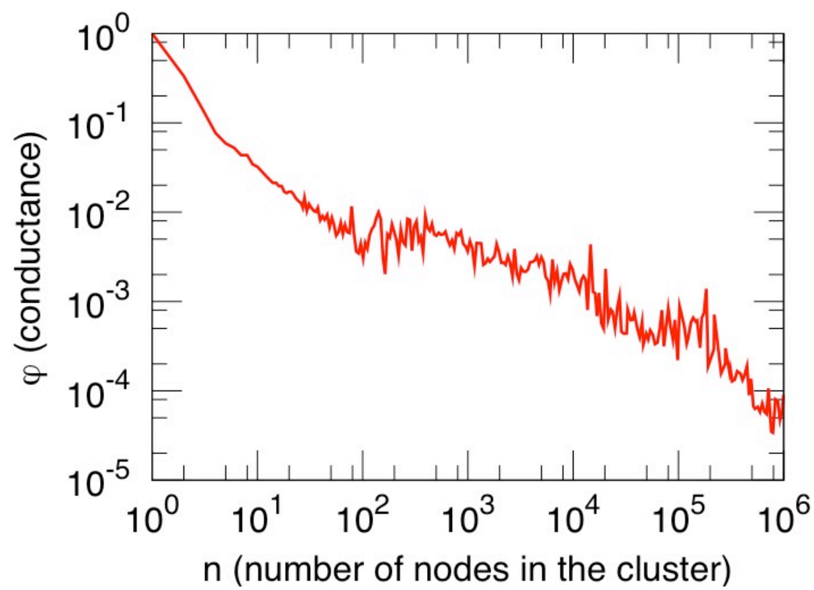


Newman's Network Science

"Low-dimensional" graphs (and expanders)



d-dimensional meshes



RoadNet-CA

Lots of Generative Models

- Preferential attachment - add edges to high-degree nodes

  (Albert and Barabasi 99, etc.)

- Copying model - add edges to neighbors of a seed node

  (Kumar et al. 00, etc.)

- Hierarchical methods - add edges based on distance in hierarchy

  (Ravasz and Barabasi 02, etc.)

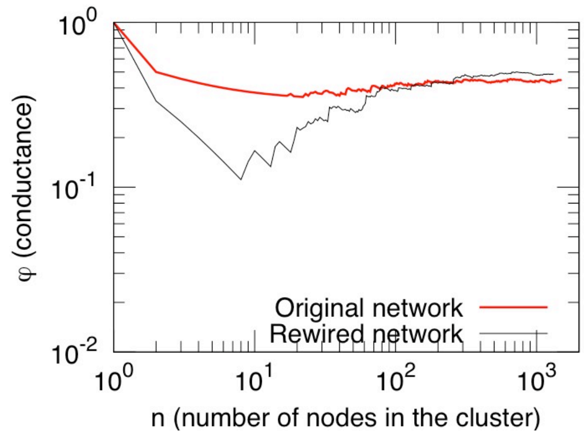
- Geometric PA and Small worlds - add edges to geometric scaffolding

  (Flaxman et al. 04; Watts and Strogatz 98; etc.)

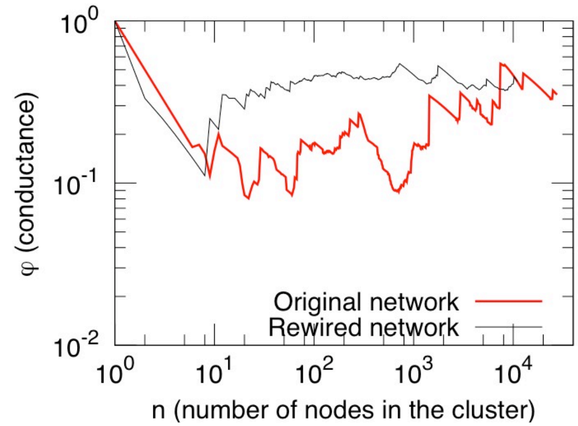
- Random/configuration models - add edges randomly

  (Molloy and Reed 98; Chung and Lu 06; etc.)

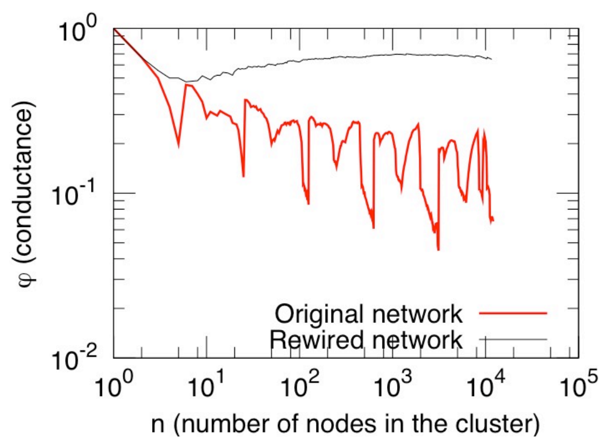
NCPP for common generative models



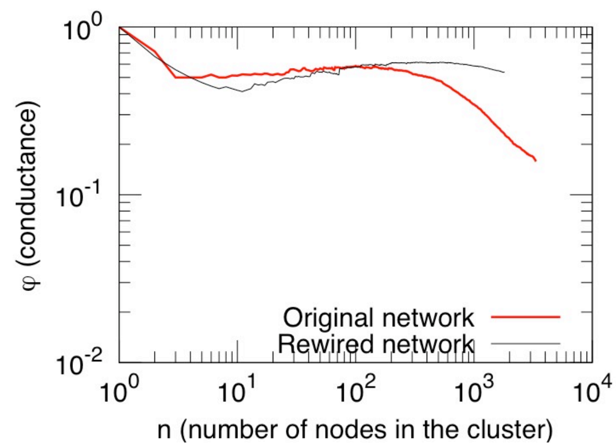
Preferential Attachment



Copying Model

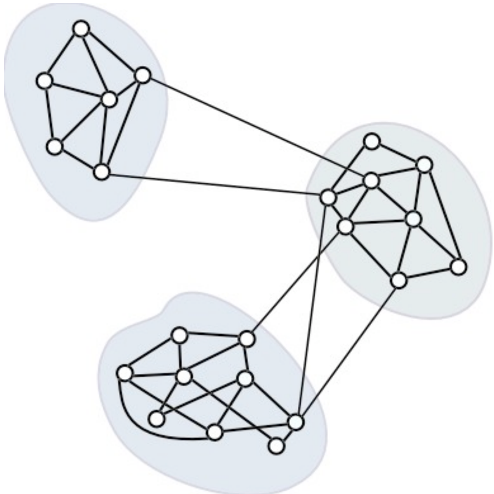


RB Hierarchical



Geometric PA

What do large networks look like?



Downward sloping NCPP

small social networks (validation)

"low-dimensional" networks (intuition)

hierarchical networks (model building)

existing generative models (incl. community models)

Natural interpretation in terms of isoperimetry

implicit in modeling with low-dimensional spaces, manifolds, k-means, etc.

Large social/information networks are very *very* different

We examined more than 70 large social and information networks

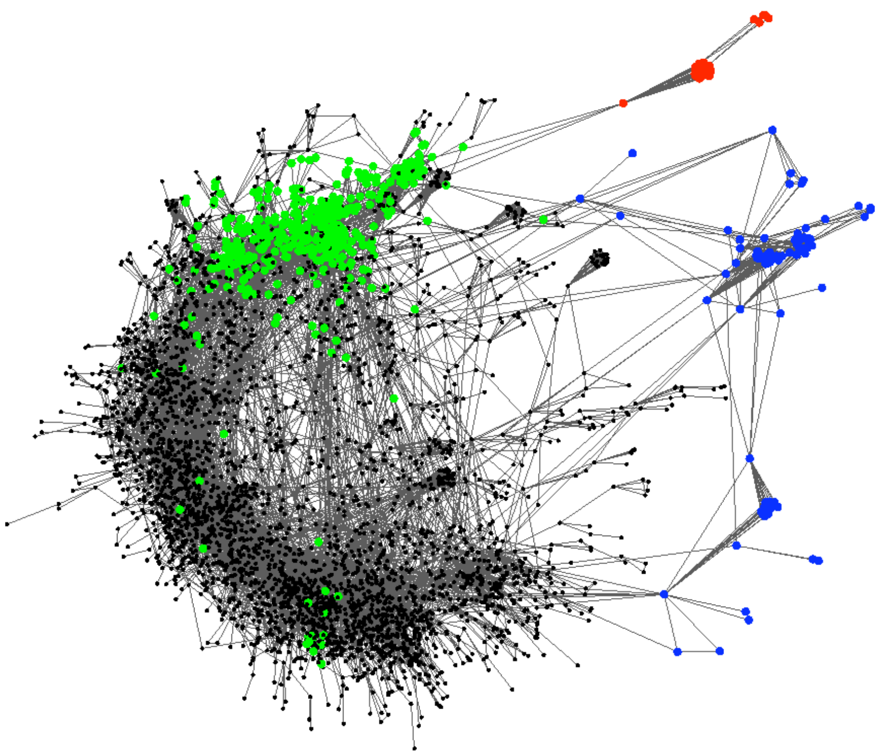
We developed principled methods to interrogate large networks

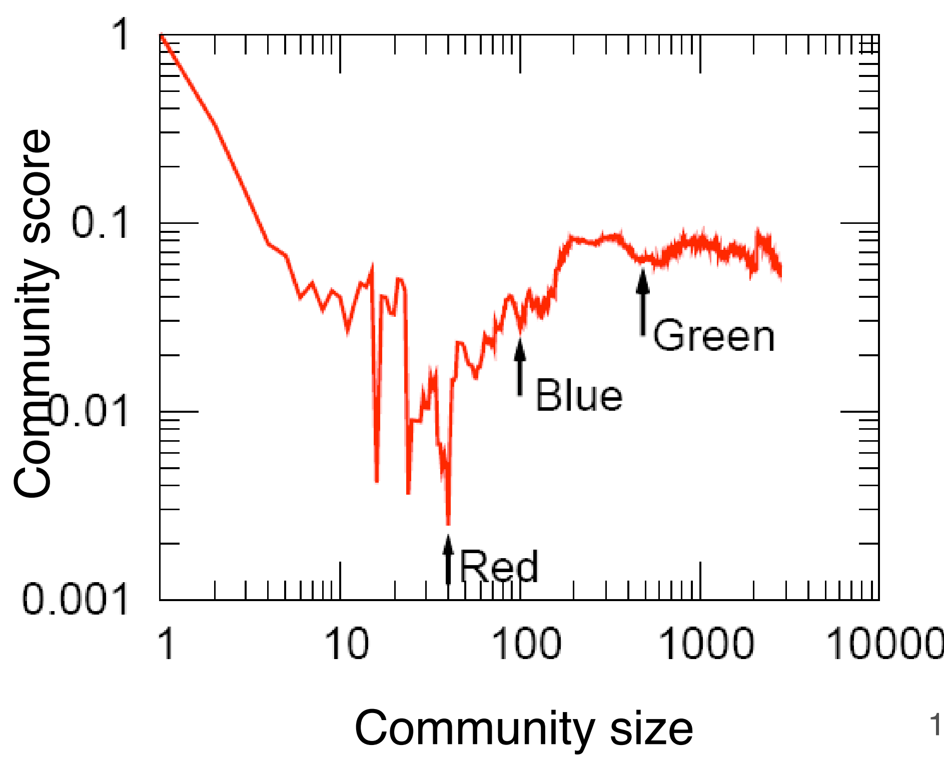
Previous community work: on small social networks (hundreds, thousands)

Typical example of our findings

Leskovec, Lang, Dasgupta, and Mahoney (WWW 2008 & arXiv 2008)

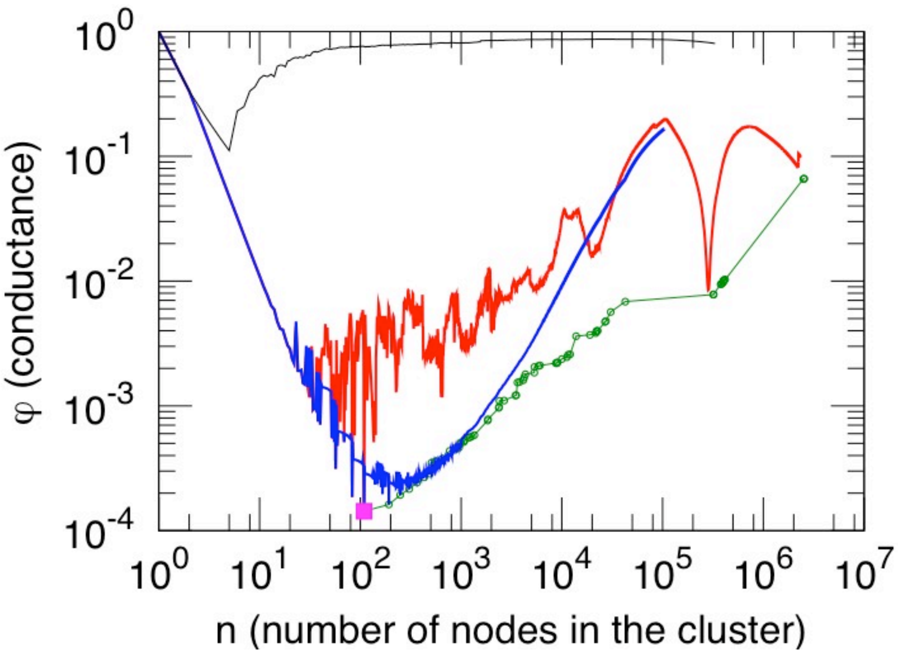
General relativity collaboration network
(4,158 nodes, 13,422 edges)



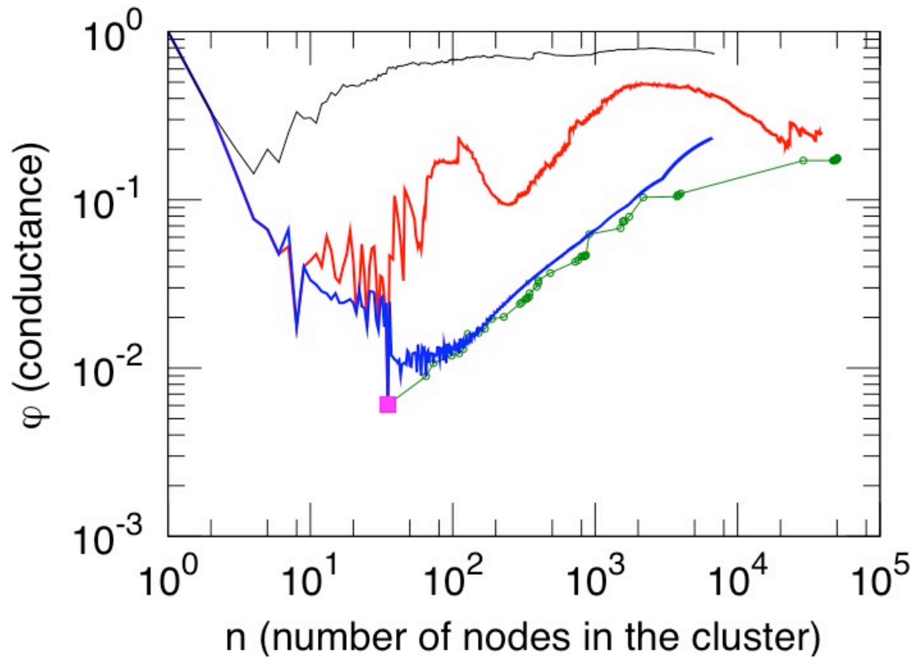


Large Social and Information Networks

Leskovec, Lang, Dasgupta, and Mahoney (WWW 2008 & arXiv 2008)



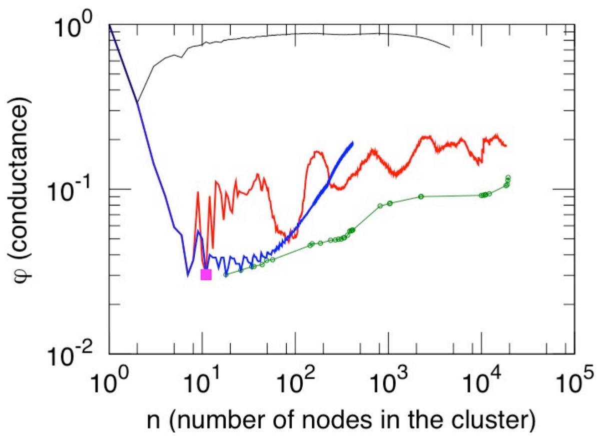
LiveJournal



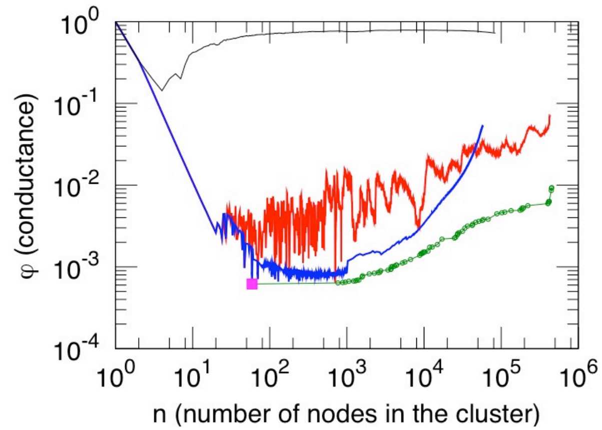
Epinions

Focus on the red curves (local spectral algorithm) - blue (Metis+Flow), green (Bag of whiskers), and black (randomly rewired network) for consistency and cross-validation.

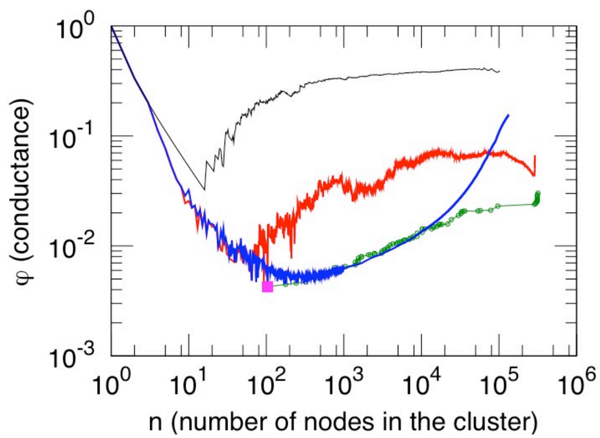
More large networks



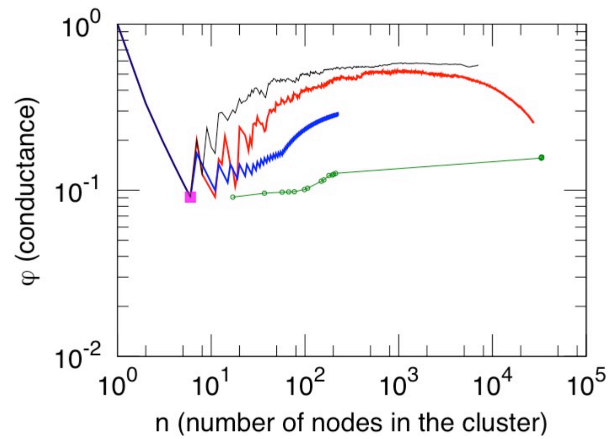
Cit-Hep-Th



Web-Google

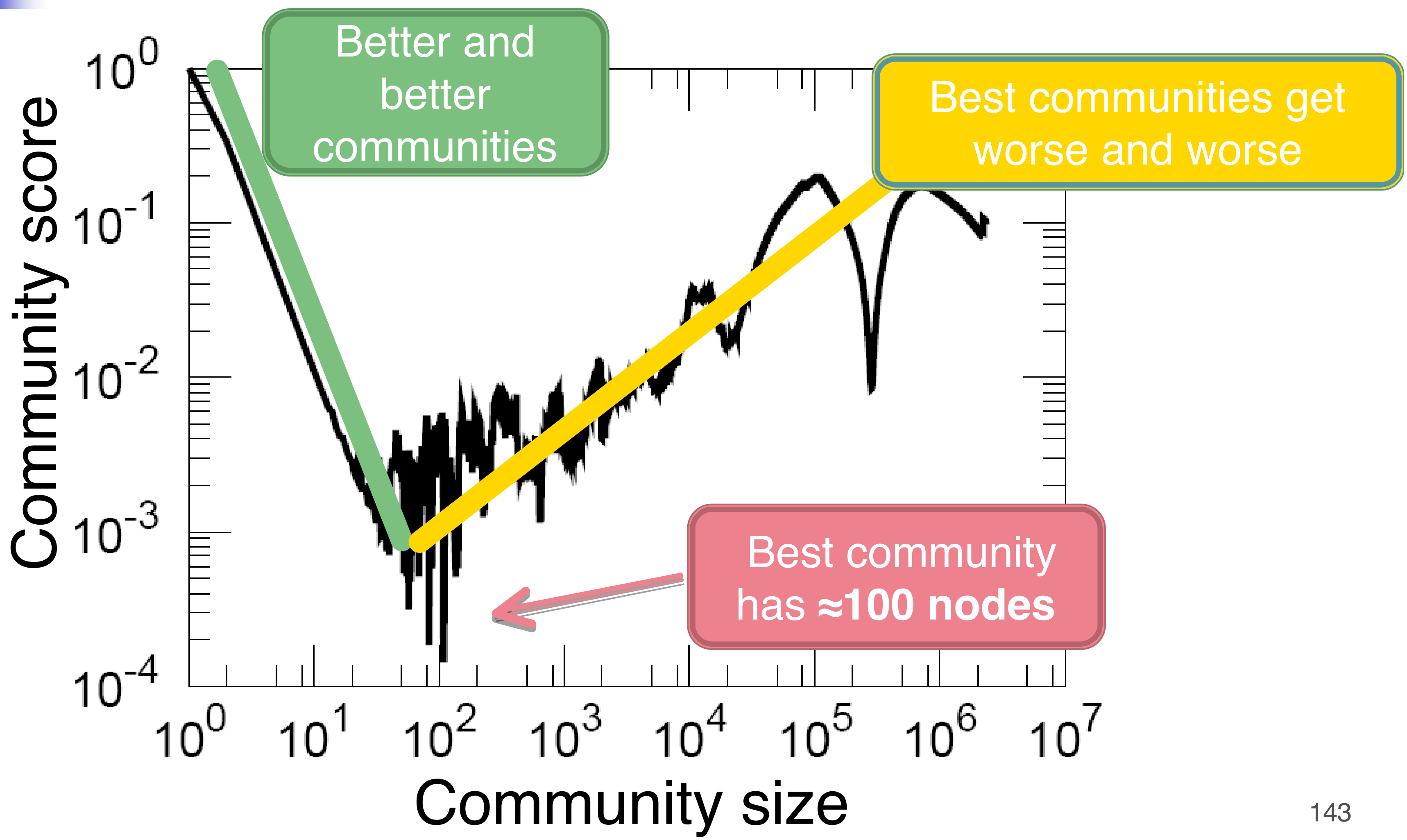


AtP-DBLP



Gnutella

NCPP: LiveJournal (N=5M, E=43M)



How do we know this plot it “correct”?

- Algorithmic Result

  Ensemble of sets returned by different algorithms are very different

  Spectral vs. flow vs. bag-of-whiskers heuristic

- Statistical Result

  Spectral method implicitly regularizes, gets more meaningful communities

- Lower Bound Result

  Spectral and SDP lower bounds for large partitions

- Structural Result

  Small barely-connected “whiskers” responsible for minimum

- Modeling Result

  Very sparse Erdos-Renyi (or PLRG wth $\beta \in (2,3)$) gets imbalanced deep cuts