Approximation Algorithms as “Experimental Probes” of Informatics Graphs

Michael W. Mahoney

Stanford University

(For more info, see:

[http:// cs.stanford.edu/people/mmahoney/](http://cs.stanford.edu/people/mmahoney/)

or Google on “Michael Mahoney”)

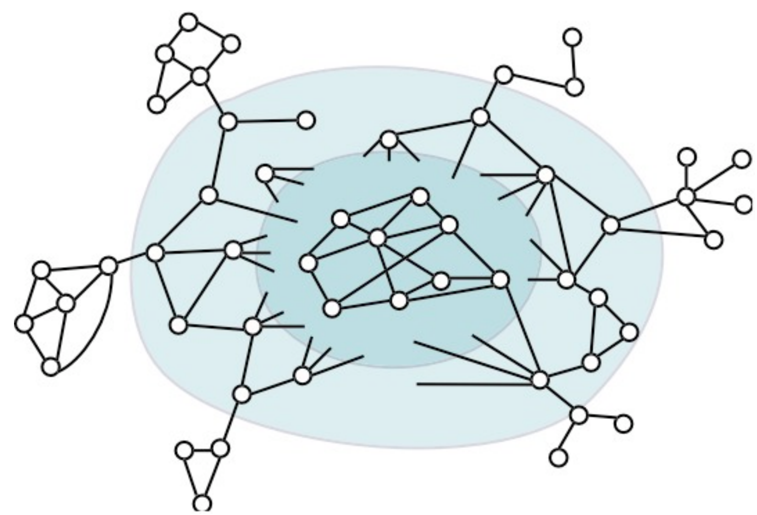
Networks and networked data

Lots of "networked" data!!

- technological networks
  - AS, power-grid, road networks
- biological networks
  - food-web, protein networks
- social networks
  - collaboration networks, friendships
- information networks
  - co-citation, blog cross-postings, advertiser-bidded phrase graphs...
- language networks
  - semantic networks...
- ...

Interaction graph model of networks:

- Nodes represent "entities"
- Edges represent "interaction" between pairs of entities



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, ...

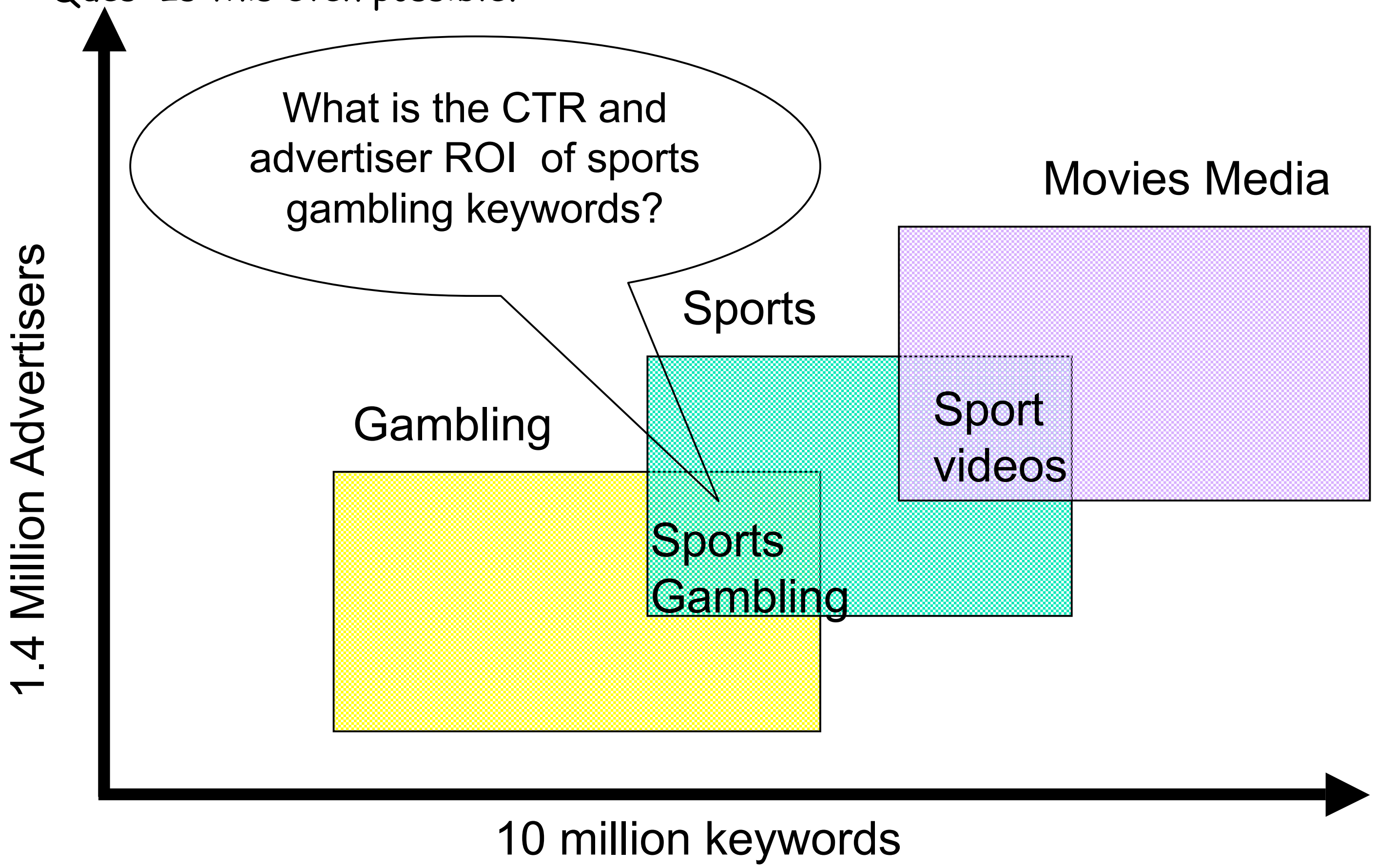
How best to do learning, e.g., classification, regression, ranking, etc.?

- Information retrieval, machine learning, ...

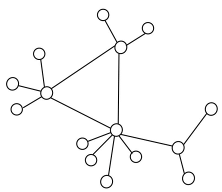
Micro-markets in sponsored search

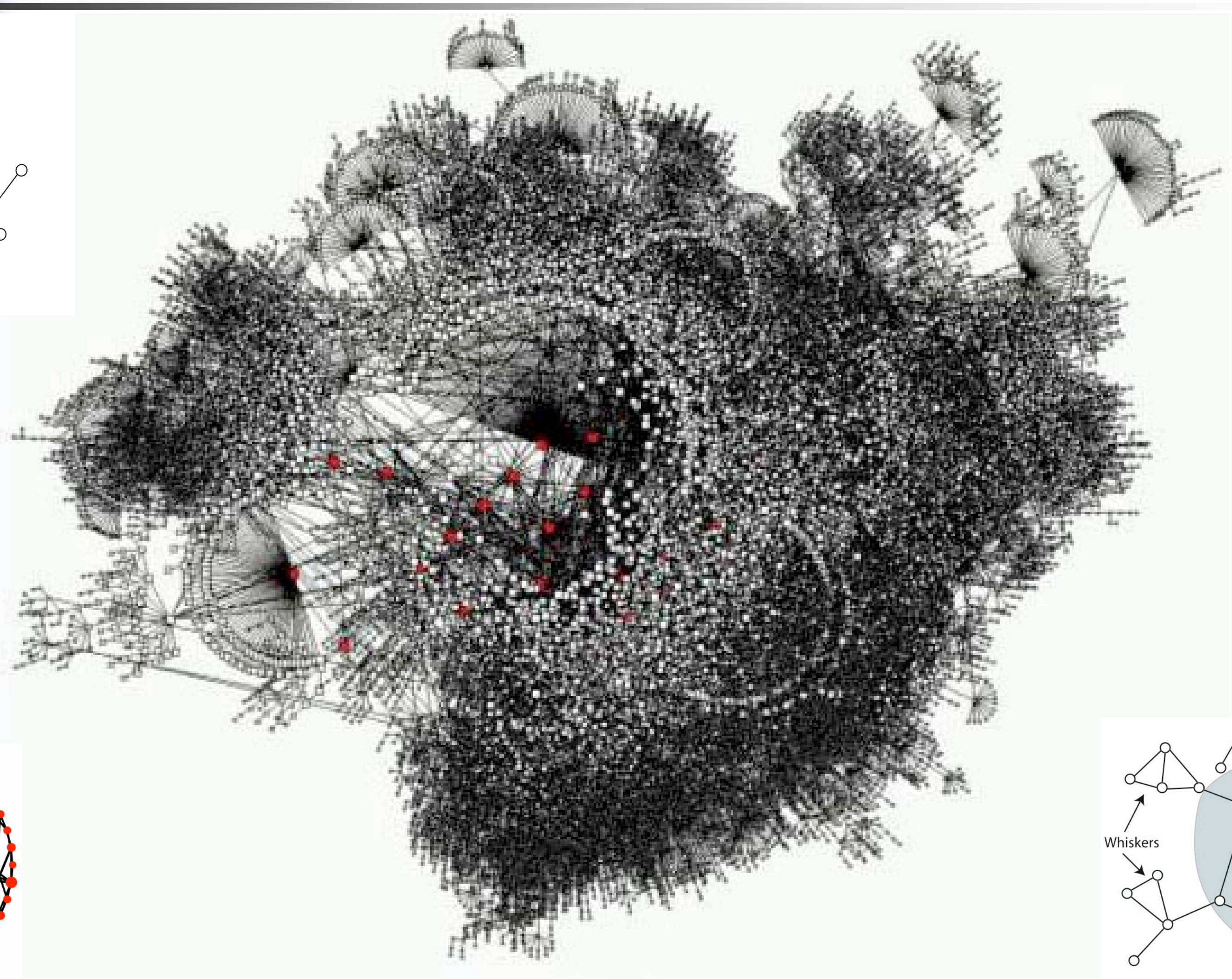
Goal: Find *isolated* markets/clusters (in an advertiser-bidded phrase bipartite graph) with *sufficient money/clicks* with *sufficient coherence.*

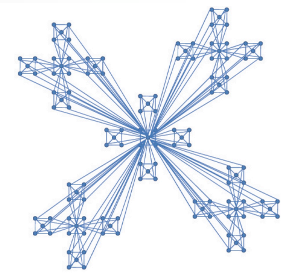
Ques: Is this even possible?

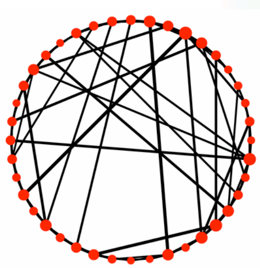


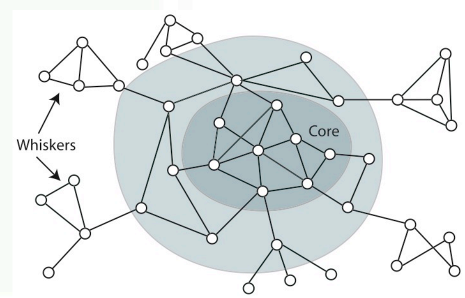
What do these networks “look” like?



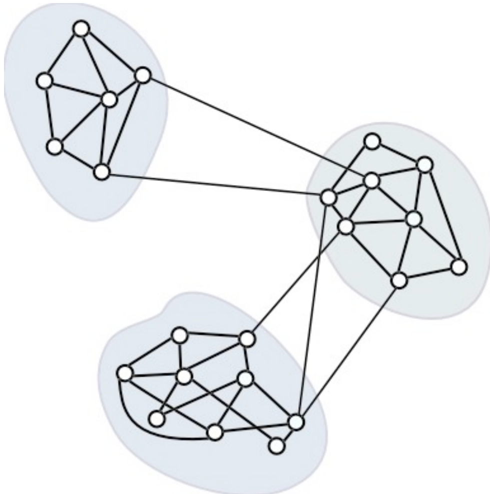








Clustering and Community Finding



- Linear (Low-rank) methods

  If Gaussian, then low-rank space is good.

- Kernel (non-linear) methods

  If low-dimensional manifold, then kernels are good

- Hierarchical methods

  Top-down and botton-up -- common in the social sciences

- Graph partitioning methods

  Define "edge counting" metric -- conductance, expansion, modularity, etc. -- in interaction graph, then optimize!

"It is a matter of common experience that communities exist in networks ... Although not precisely defined, communities are usually thought of as sets of nodes with better connections amongst its members than with the rest of the world."

Approximation algorithms as experimental probes?

The 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 algorithms, regression algorithms, 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.

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

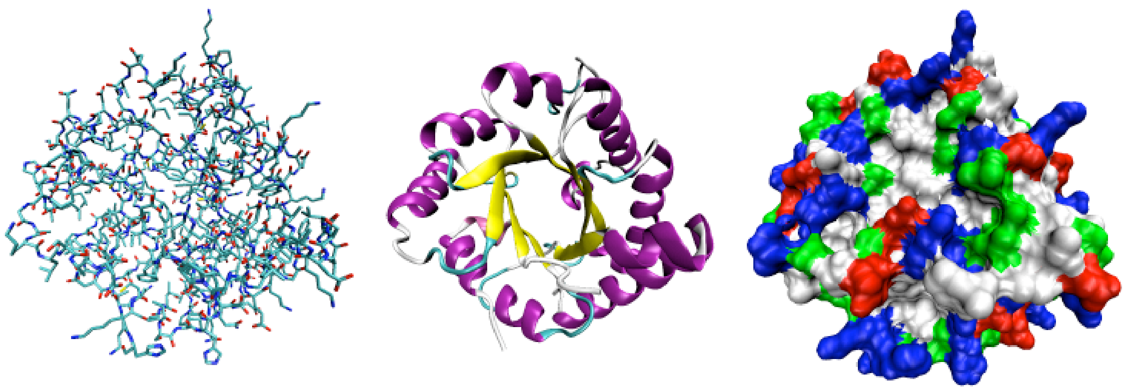
X+MQI - post-processing step on, e.g., Spectral of Metis

Metis+MQI - best conductance (empirically)

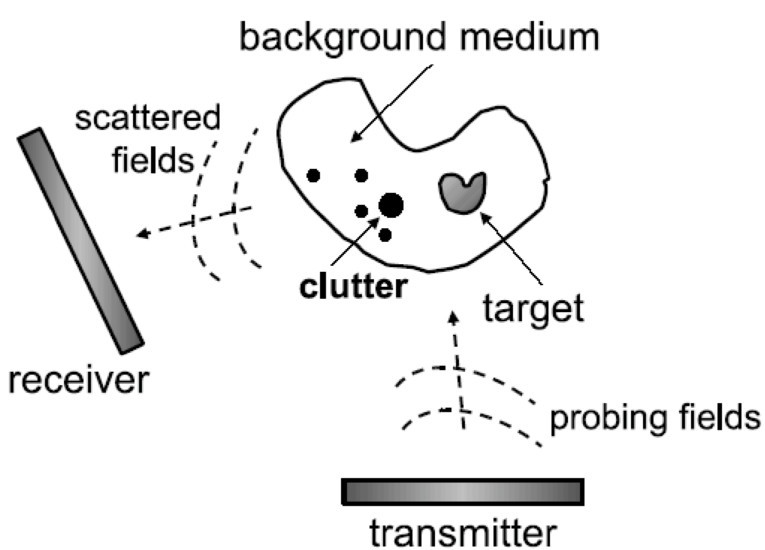
Local Spectral - connected and tighter sets (empirically, regularized communities!)

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

Analogy: What does a protein look like?



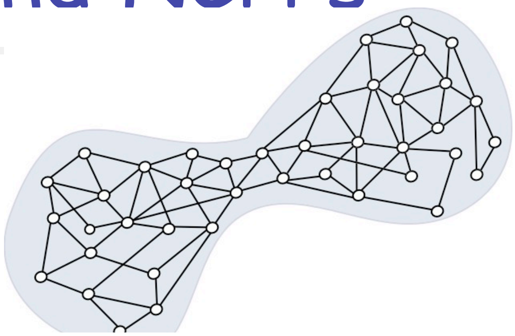
Three possible representations (all-atom; backbone; and solvent-accessible surface) of the three-dimensional structure of the protein triose phosphate isomerase.



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.

Communities, Conductance, and NCPPs



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

The conductance ϕ of a set S of nodes is:

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

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

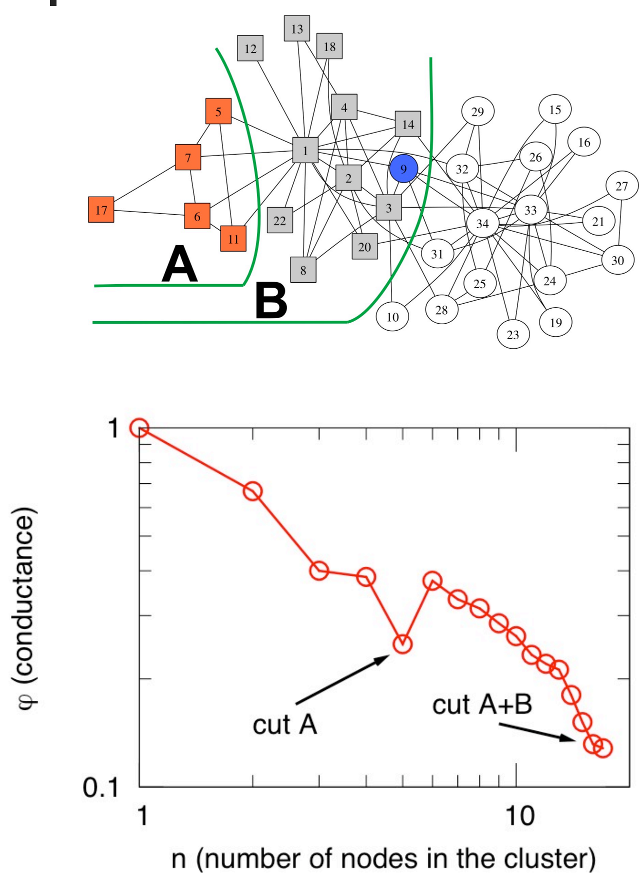
The Network Community Profile (NCP) Plot of the graph is:

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

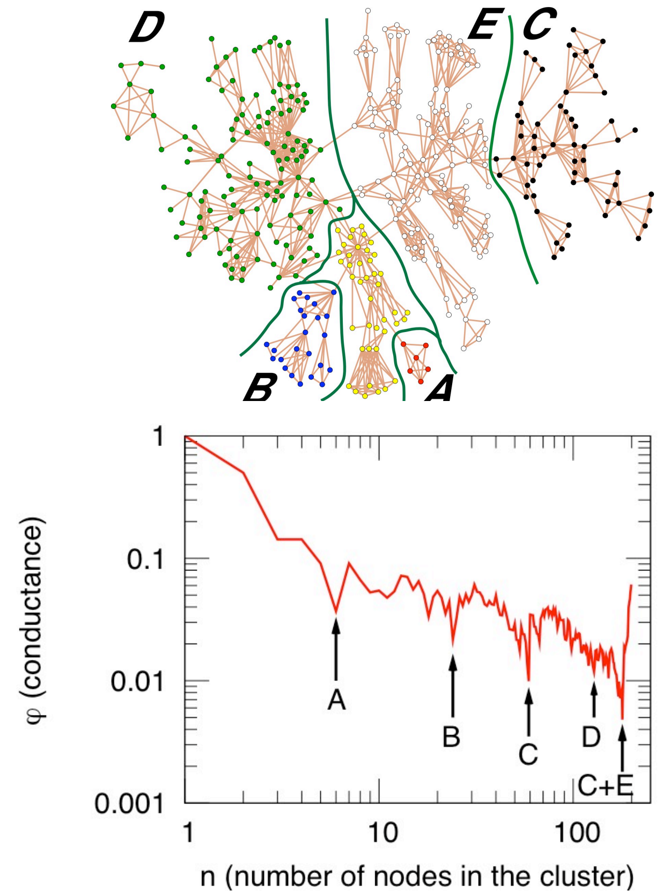
Just as conductance captures the "gestalt" notion of cluster/community quality, the NCP plot measures cluster/community quality as a function of size.

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

Widely-studied small social networks

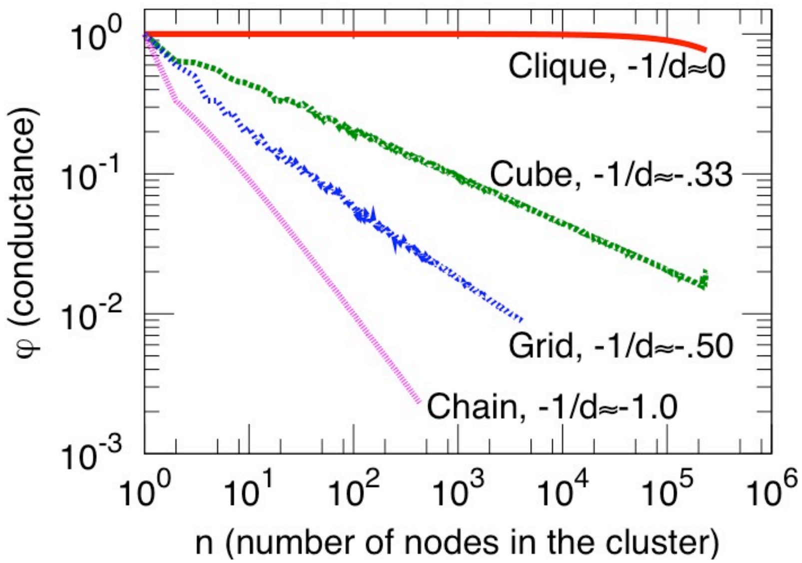


Zachary's karate club

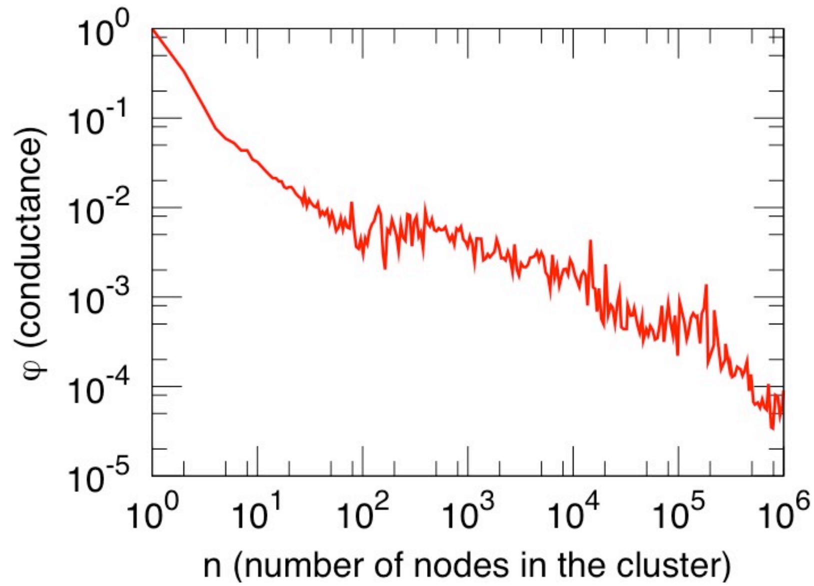


Newman's Network Science

"Low-dimensional" graphs (and expanders)

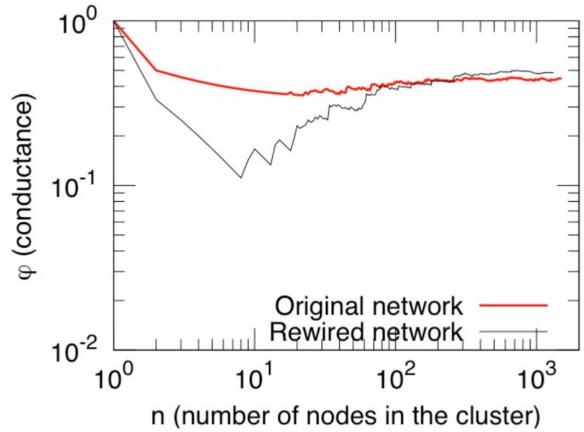


d-dimensional meshes

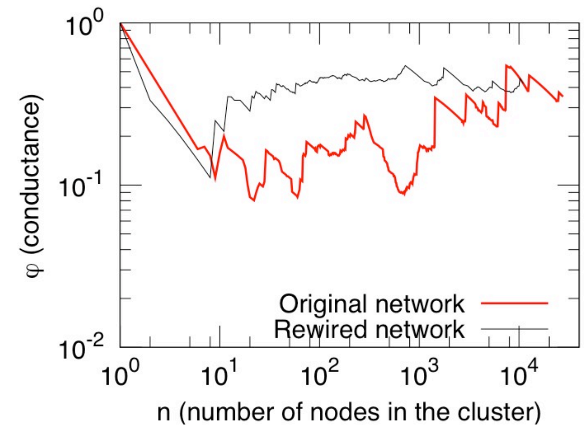


RoadNet-CA

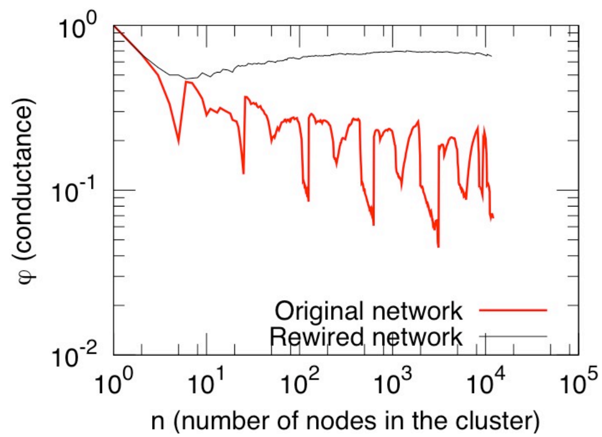
NCPP for common generative models



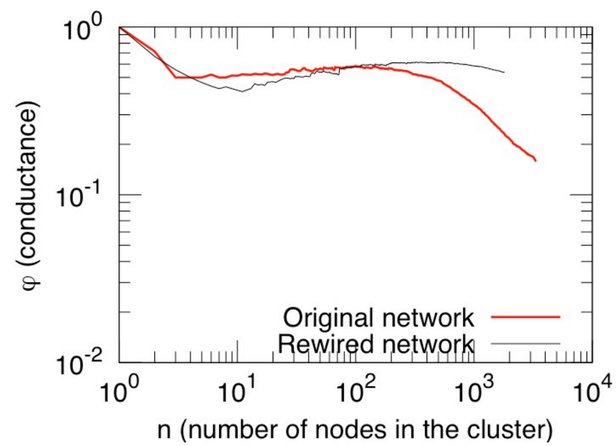
Preferential Attachment



Copying Model

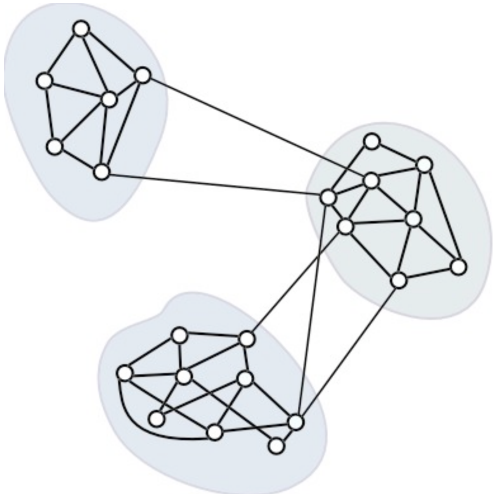


RB Hierarchical



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.

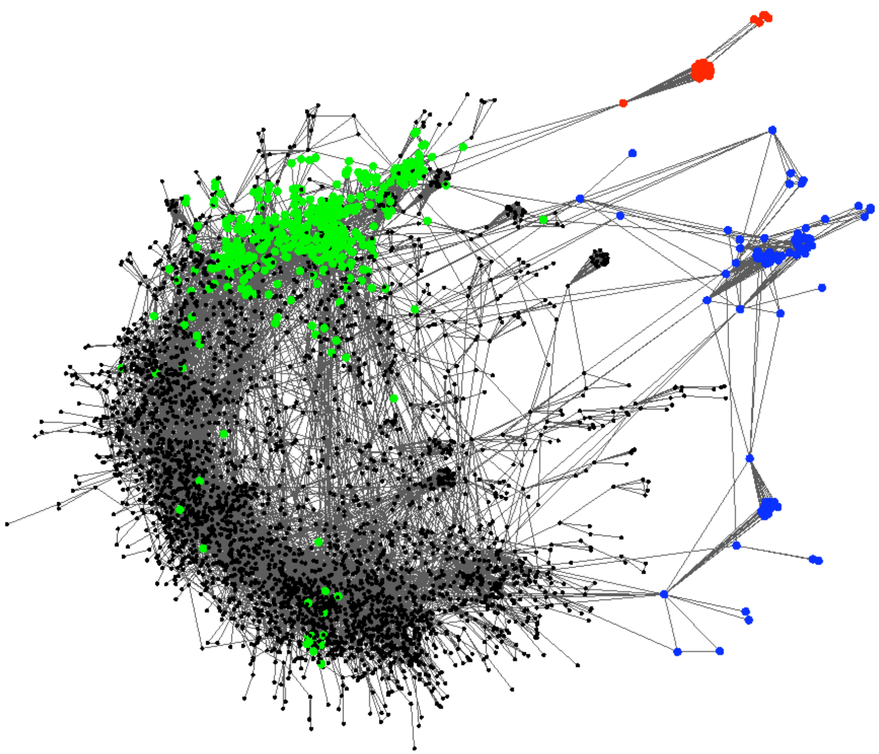
Large social/information networks are very *very* different

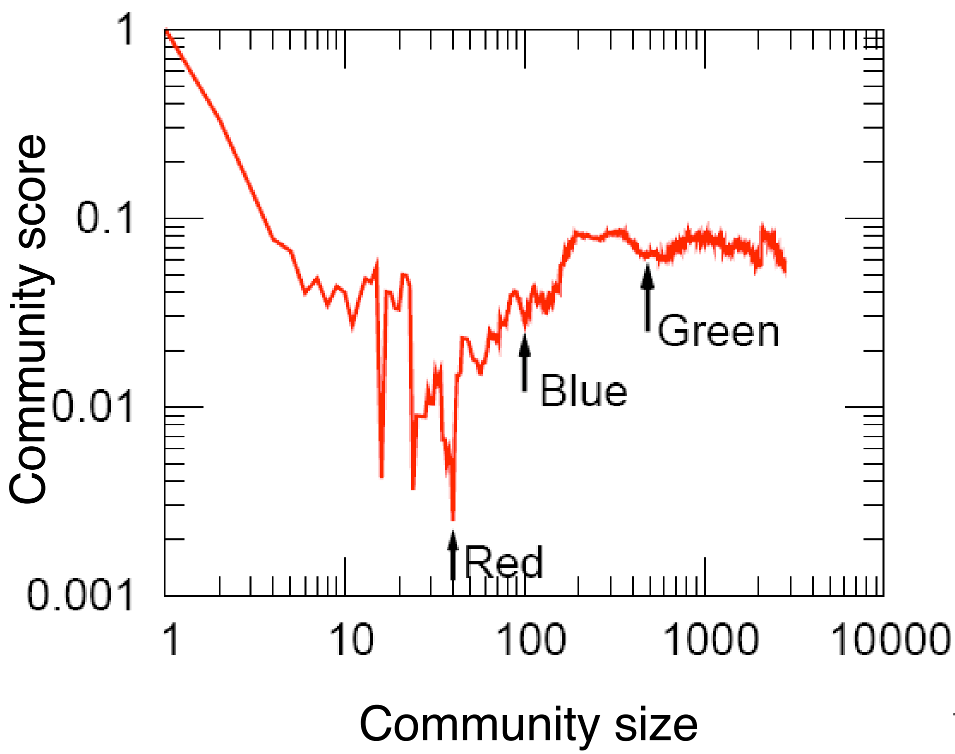
- We examined more than 70 large social and information networks
- We developed principled methods to interrogate large networks
- Previous community work: on small social networks (hundreds, thousands)

Typical example of our findings

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

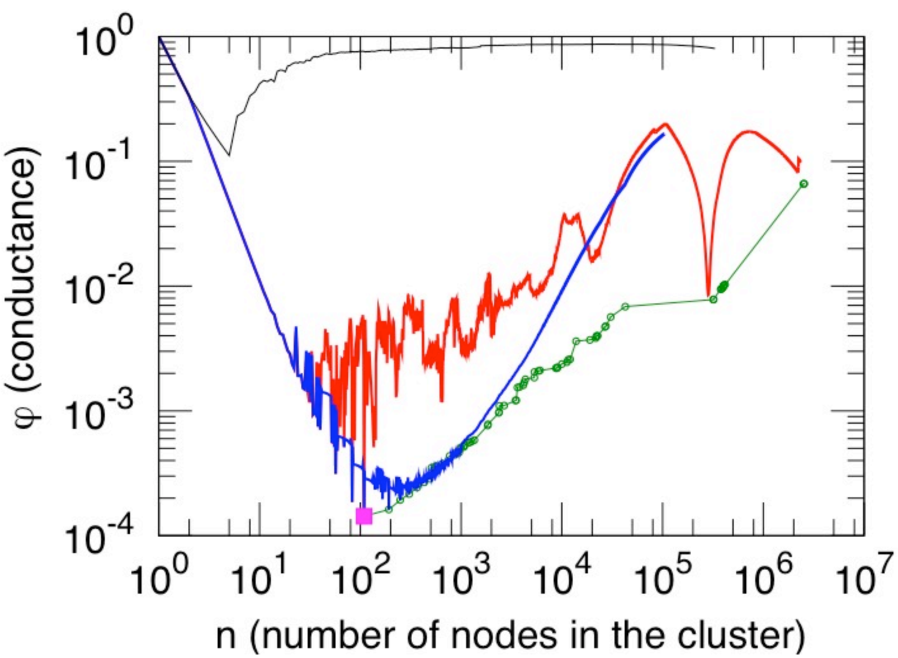
General relativity collaboration network
(4,158 nodes, 13,422 edges)



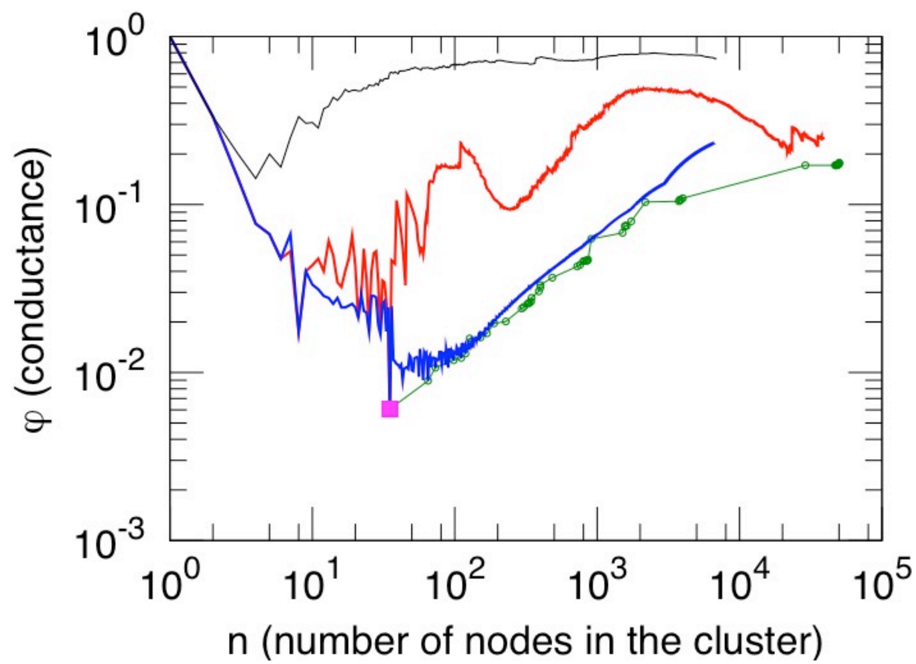


Large Social and Information Networks

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.

How do we know this plot it "correct"?

- Lower Bound Result

  Spectral and SDP lower bounds for large partitions

- Modeling Result

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

- Structural Result

  Small barely-connected "whiskers" responsible for minimum

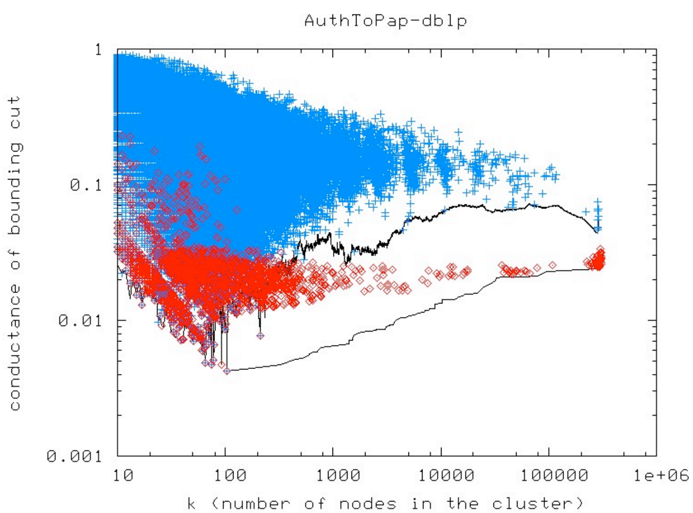
- Algorithmic Result

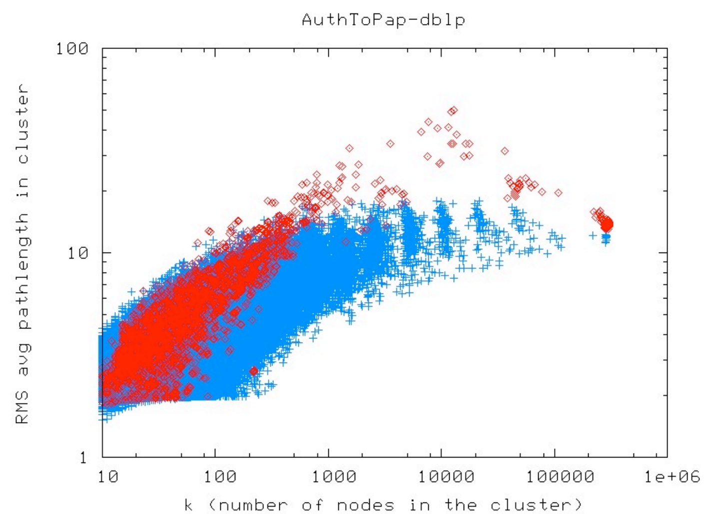
  Ensemble of sets returned by different algorithms are very different

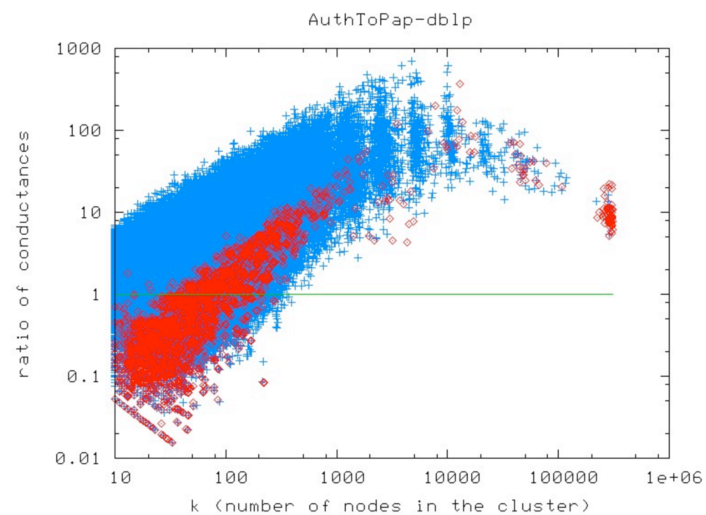
  Spectral vs. flow vs. bag-of-whiskers heuristic

  Spectral method implicitly regularizes, gets more meaningful communities

Regularized and non-regularized communities (1 of 2)



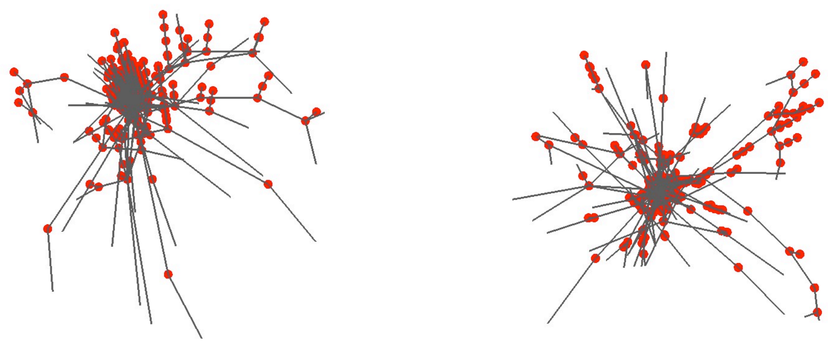




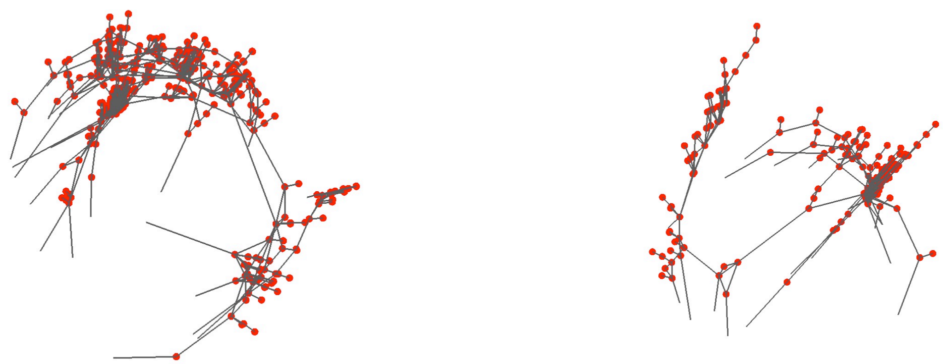
- Metis+MQI (red) gives sets with better conductance.
- Local Spectral (blue) gives tighter and more well-rounded sets.

Regularized and non-regularized communities (2 of 2)

Two ca. 500 node communities from Local Spectral Algorithm:

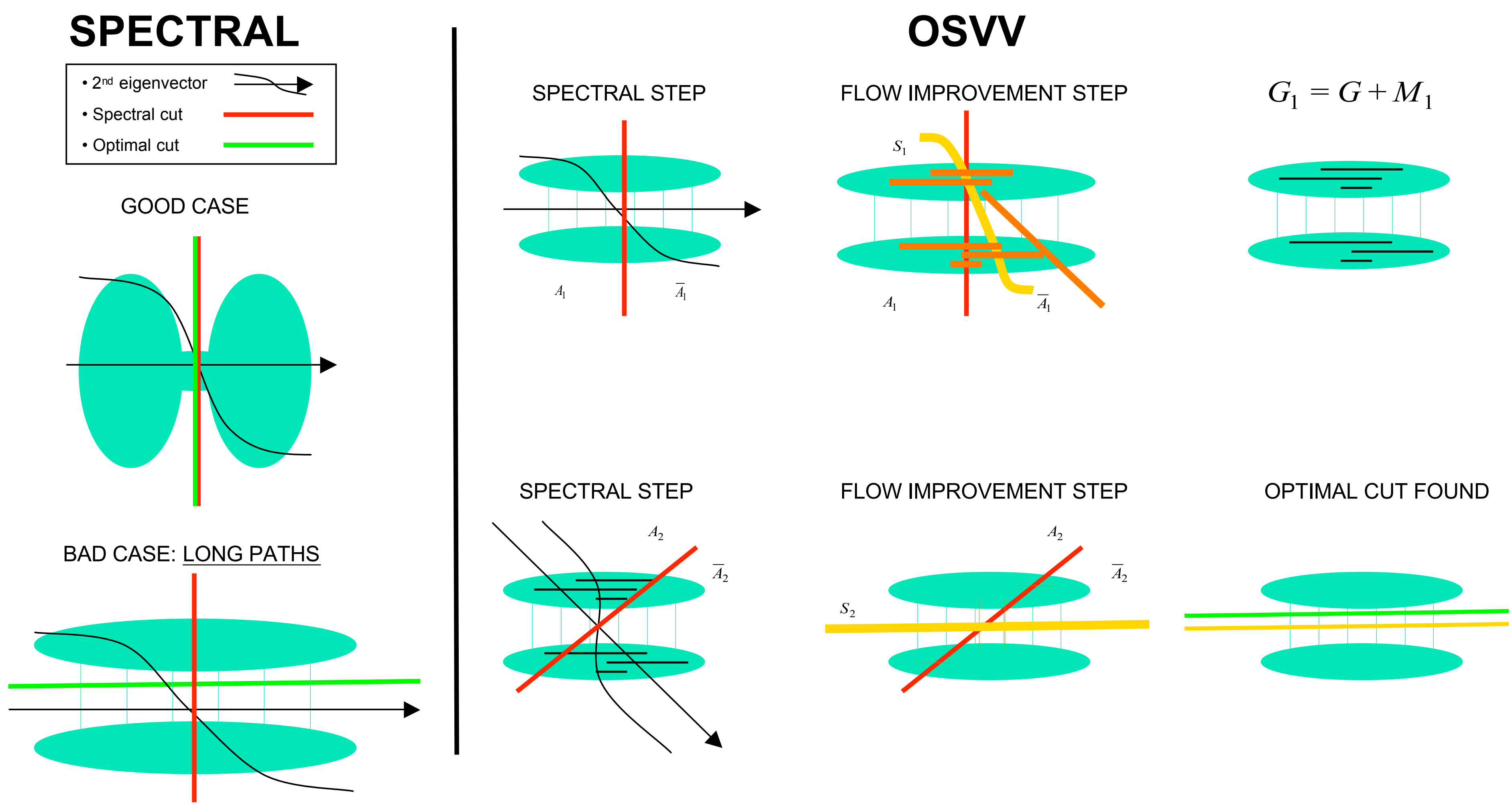


Two ca. 500 node communities from Metis+MQI:



OSVV "spectral-flow" partitioning

Orecchia, Schulman, Vazirani, and Vishnoi (2008) - variant of Arora, Rao, Vazirani (2004)



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**	1.120	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	101.4	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	1019.4	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).

Conclusions

Approximation Algorithms as Experimental Probes of Informatics Graphs

• Powerful tools to ask precise questions of large graphs

• Use statistical and regularization properties of ensembles of "approximate solution graphs" to infer properties of original network

• Community structure in real informatics graphs -- very different than small commonly-studied graphs and existing generative models