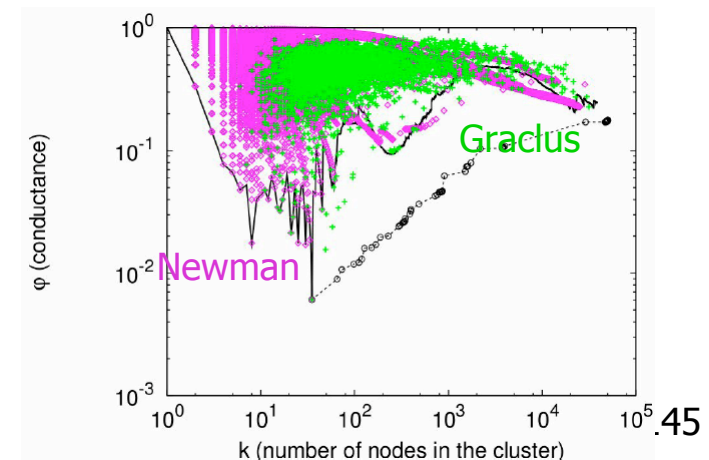
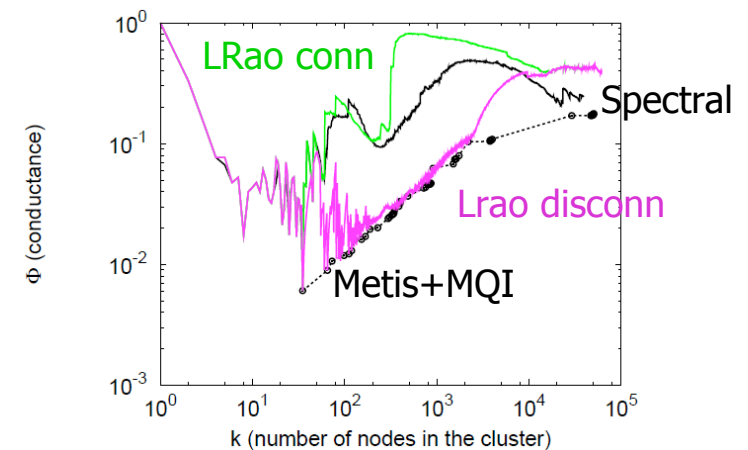


# Other clustering methods

- **LeightonRao**: based on multi-commodity flow
  - **Disconnected** clusters vs. **Connected** clusters
- **Graclus** prefers larger clusters
- **Newman's** modularity optimization similar to Local Spectral



# 12 objective functions

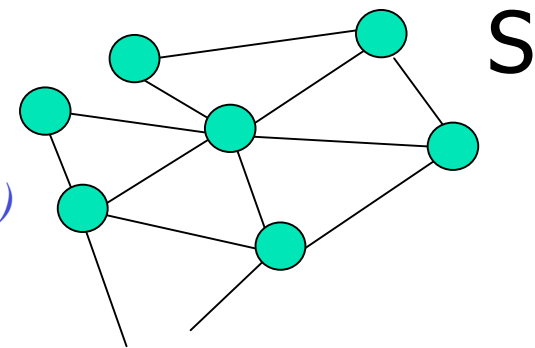
## ■ Clustering objectives:

### ■ Single-criterion:

- **Modularity**:  $m - E(m)$  (*Volume minus correction*)
- **Modularity Ratio**:  $m - E(m)$
- **Volume**:  $\sum_u d(u) = 2m + c$
- **Edges cut**:  $c$

### ■ Multi-criterion:

- **Conductance**:  $c/(2m+c)$  (*SA to Volume*)
- **Expansion**:  $c/n$
- **Density**:  $1 - m/n^2$
- **CutRatio**:  $c/n(N-n)$
- **Normalized Cut**:  $c/(2m+c) + c/2(M-m)+c$
- **Max ODF**: *max frac. of edges of a node pointing outside S*
- **Average-ODF**: *avg. frac. of edges of a node pointing outside*
- **Flake-ODF**: *frac. of nodes with more than  $\_$  edges inside*

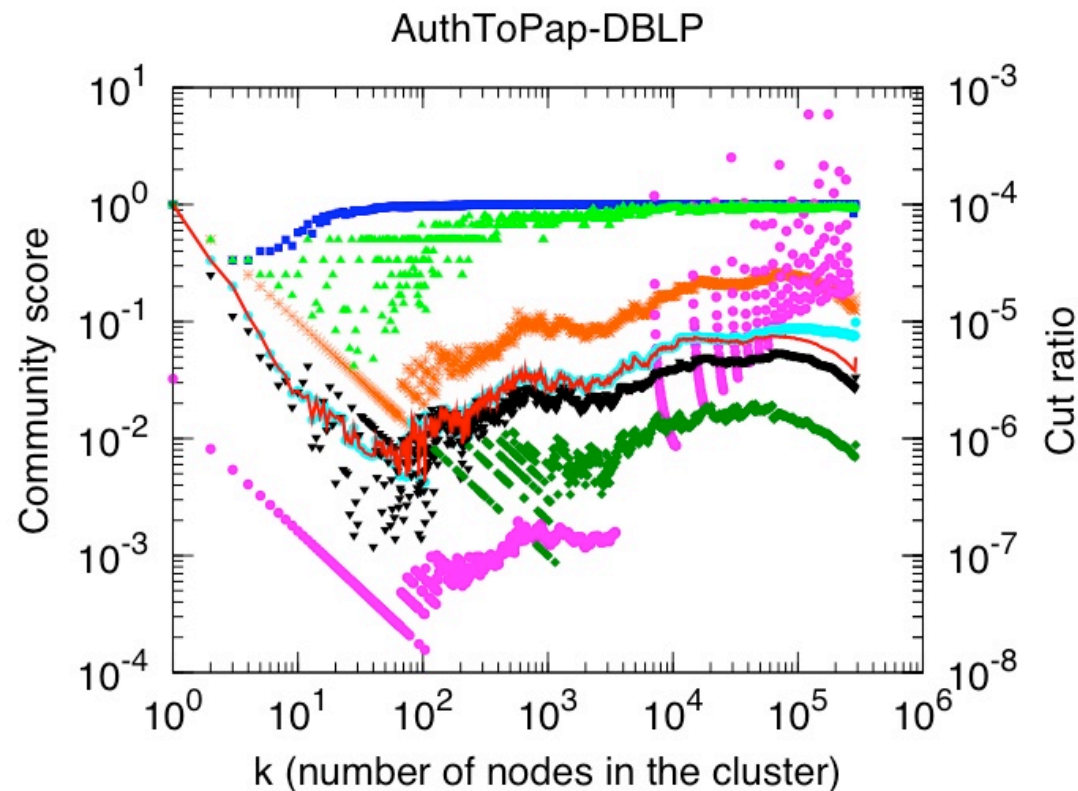


$n$ : nodes in S

$m$ : edges in S

$c$ : edges pointing  
outside S

# Multi-criterion objectives



- Qualitatively similar to conductance
- Observations:
  - Conductance, Expansion, NCut, Cut-ratio and Avg-ODF are similar
  - Max-ODF prefers smaller clusters
  - Flake-ODF prefers larger clusters
  - Internal density is bad
  - Cut-ratio has high variance

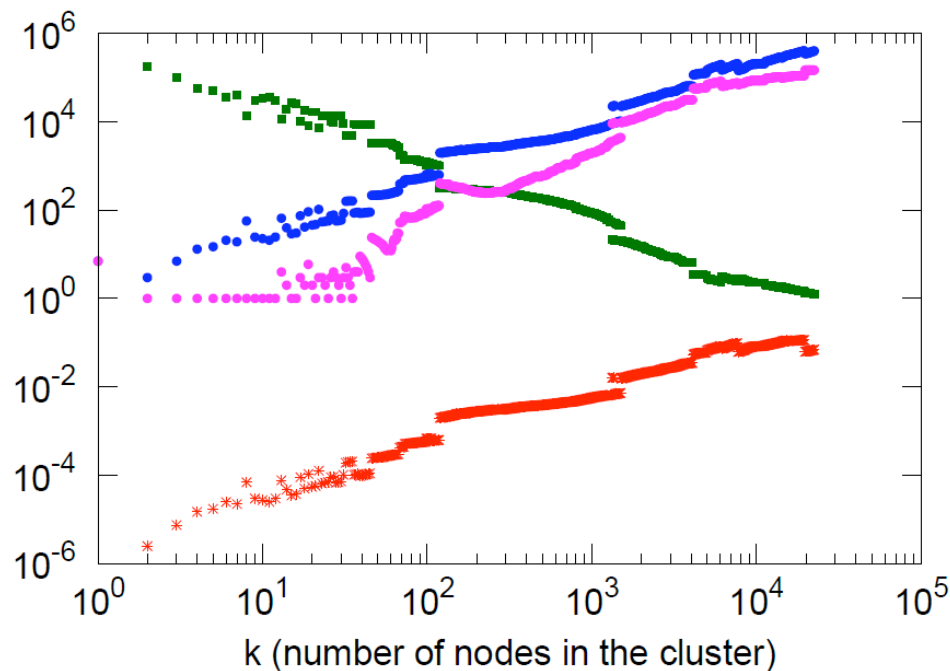
Conductance  
Expansion \*

Internal Density  
Cut Ratio

Normalized Cut  
Maximum ODF

Avg ODF  
Flake ODF

# Single-criterion objectives



Modularity

\*

Modularity Ratio

■

Volume

●

Edges cut

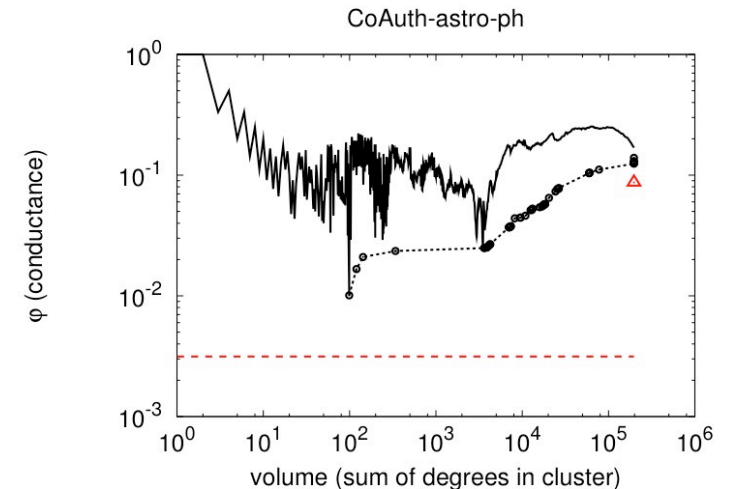
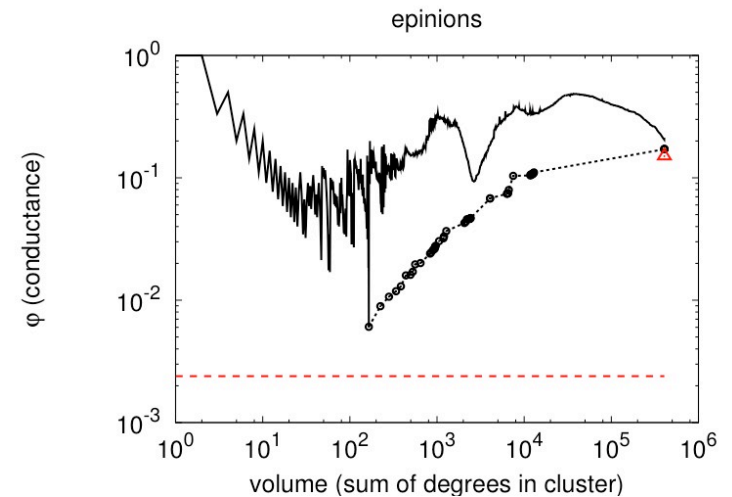
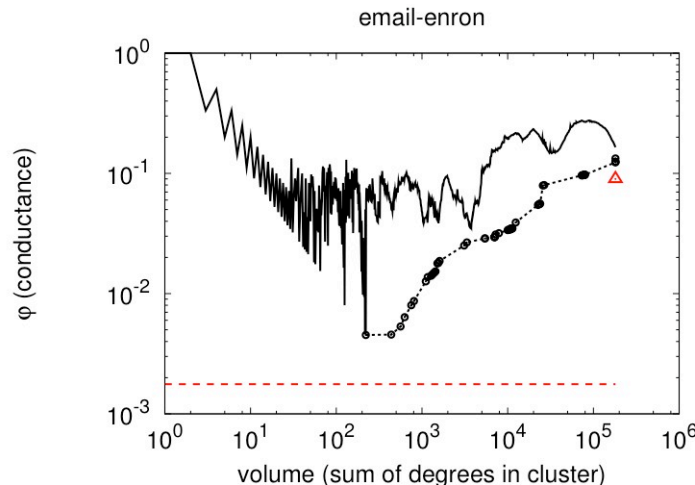
●

## Observations:

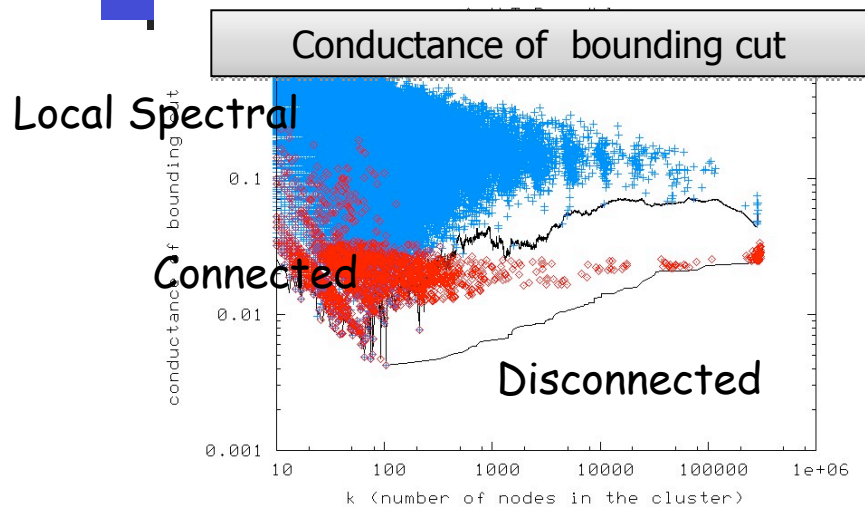
- All measures are monotonic (for rather trivial reasons)
- Modularity
  - prefers large clusters
  - Ignores small clusters
  - *Because it basically captures Volume!*

# Lower and upper bounds

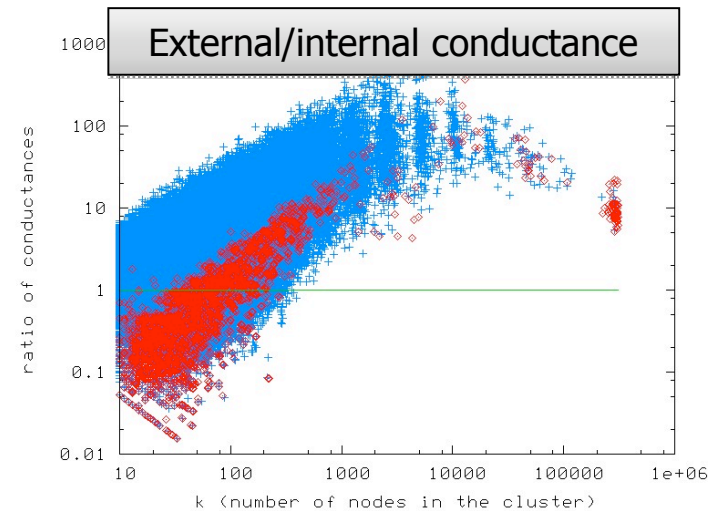
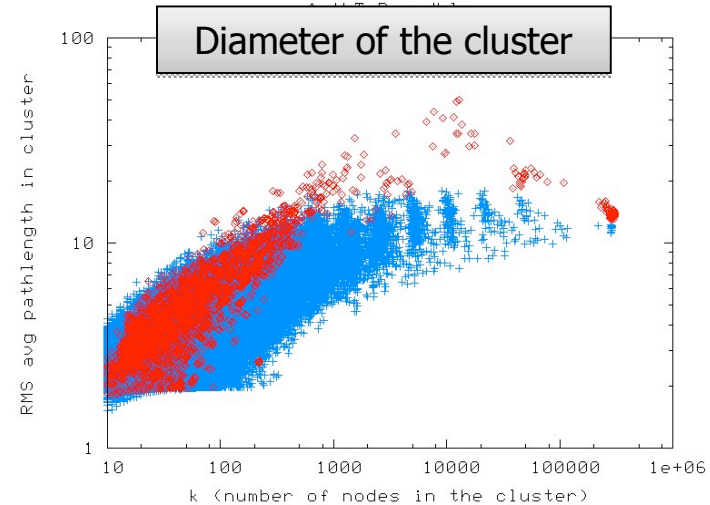
- Lower bounds on conductance can be computed from:
  - Spectral embedding (independent of balance)
  - SDP-based methods (for volume-balanced partitions)
- Algorithms find clusters close to theoretical lower bounds



## 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.



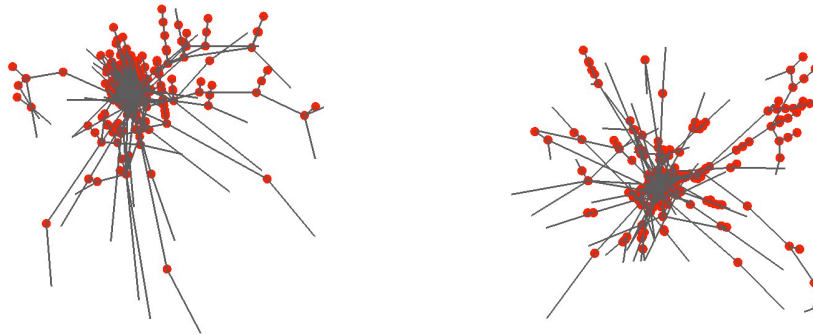
Lower is good



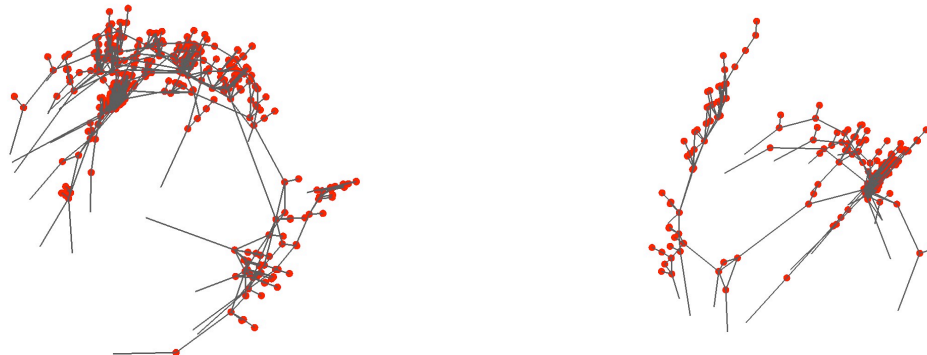
## Regularized and non-regularized communities (2 of 2)

---

Two ca. 500 node communities from Local Spectral Algorithm:



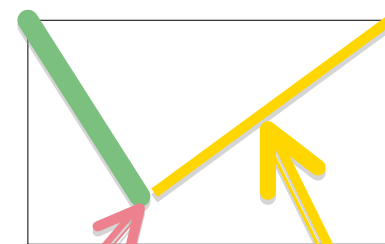
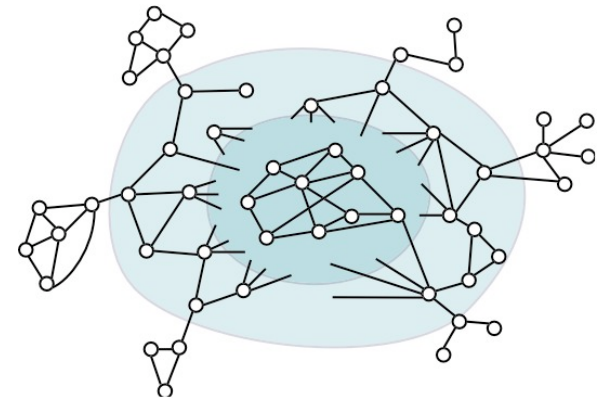
Two ca. 500 node communities from Metis+MQI:





# Interpretation: "Whiskers" and the "core" of large informatics graphs

- "Whiskers"
  - maximal sub-graph detached from network by removing a single edge
  - contains 40% of nodes and 20% of edges
- "Core"
  - the rest of the graph, i.e., the 2-edge-connected core
- Global minimum of NCPP is a whisker
- BUT, *core itself has nested whisker-core structure*



NCP plot

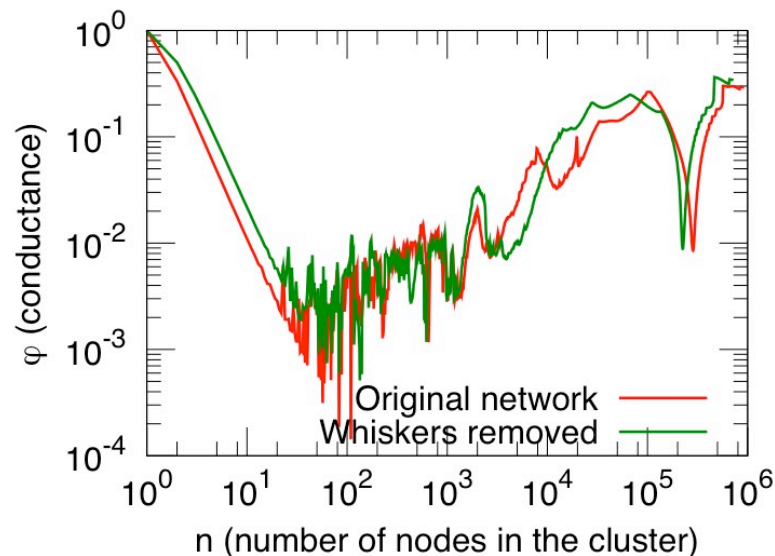
Largest  
whisker

Slope upward as  
cut into core

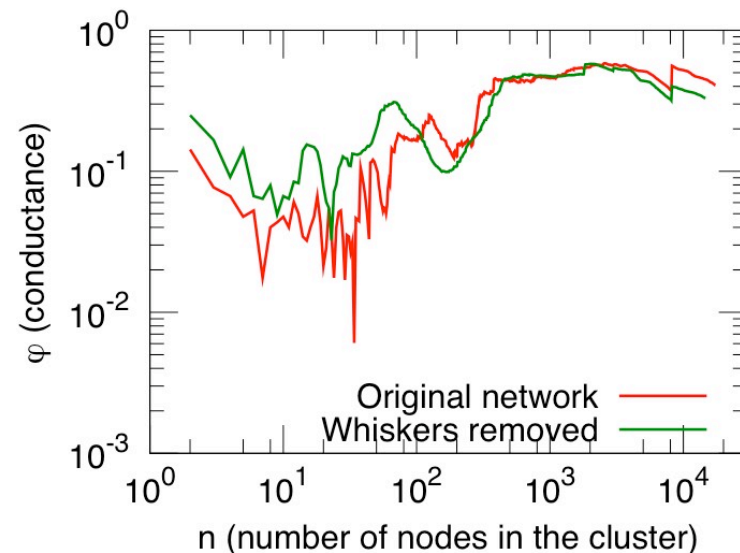


# What if the "whiskers" are removed?

Then the lowest conductance sets - the "best" communities - are "2-whiskers."  
(So, the "core" peels apart like an onion.)

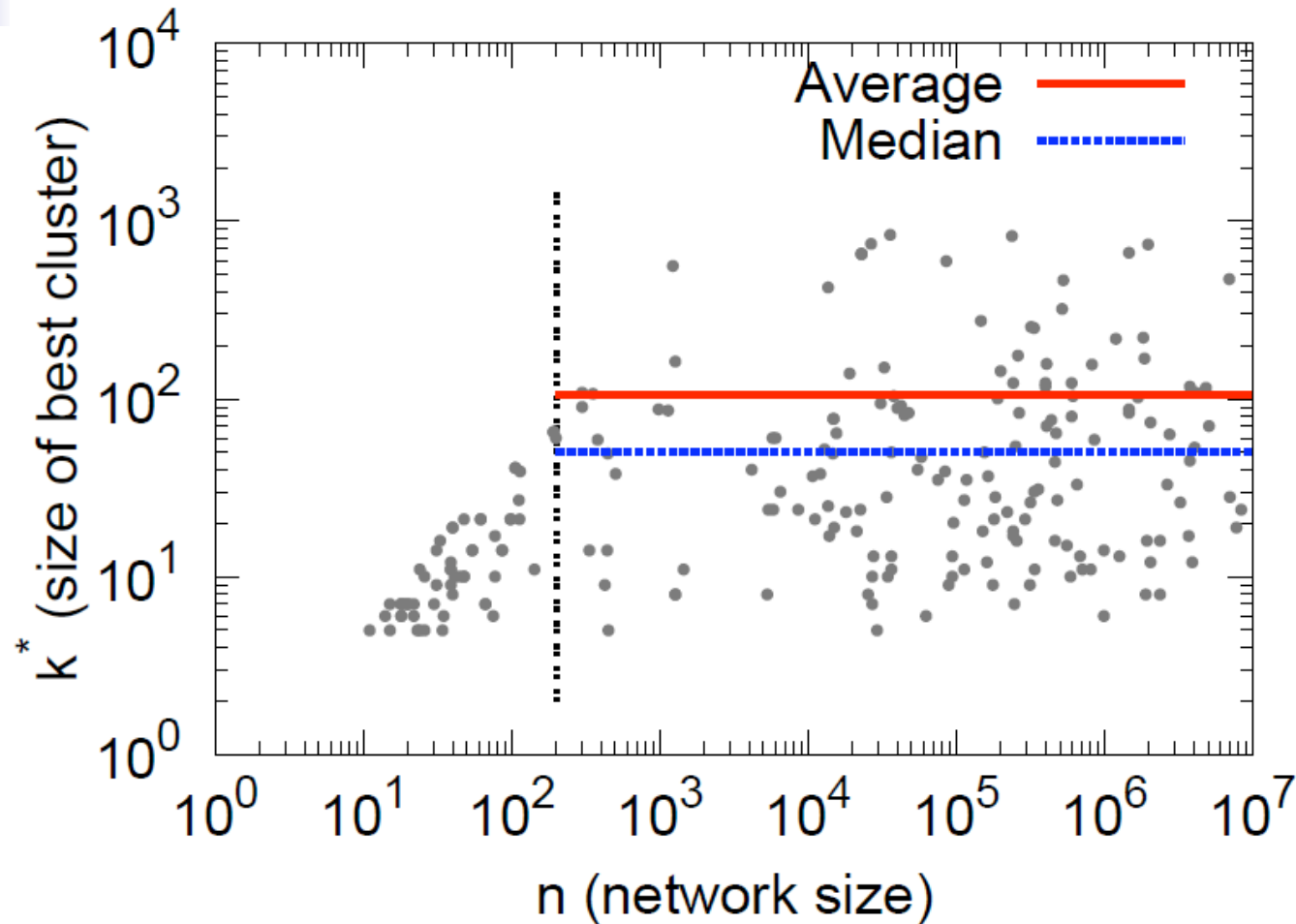


LiveJournal



Epinions

## Size of best cluster versus network size



(Each dot is a different network -- so they are roughly independent.)

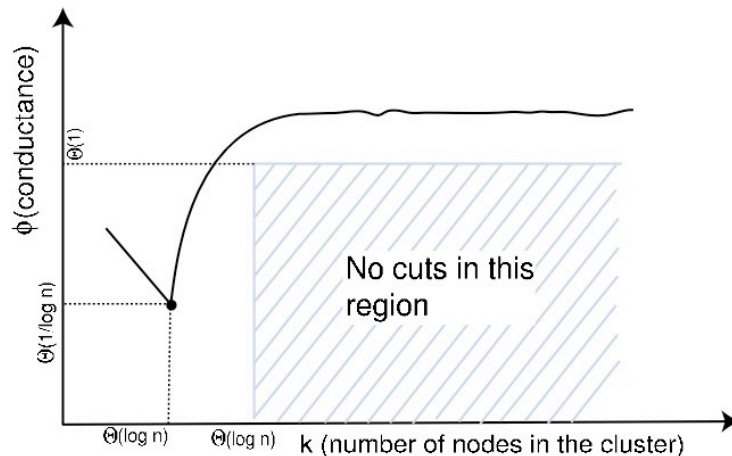
# Interpretation:

## A simple theorem on random graphs

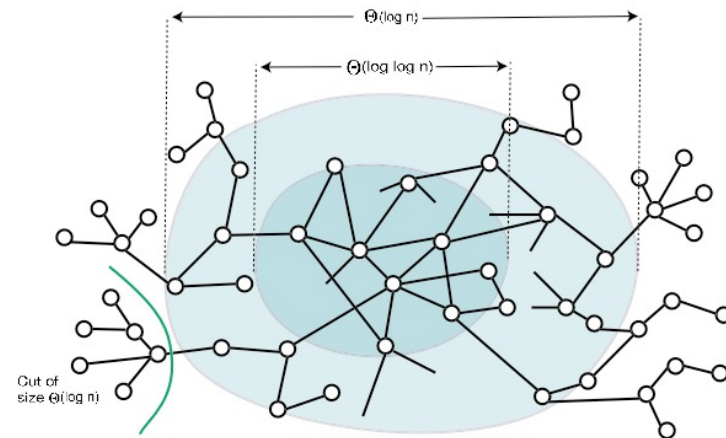
Let  $\mathbf{w} = (w_1, \dots, w_n)$ , where  
 $w_i = ci^{-1/(\beta-1)}$ ,  $\beta \in (2, 3)$ .

Connect nodes  $i$  and  $j$  w.p.

$$p_{ij} = w_i w_j / \sum_k w_k.$$



Power-law random graph with  $\beta \in (2, 3)$ .



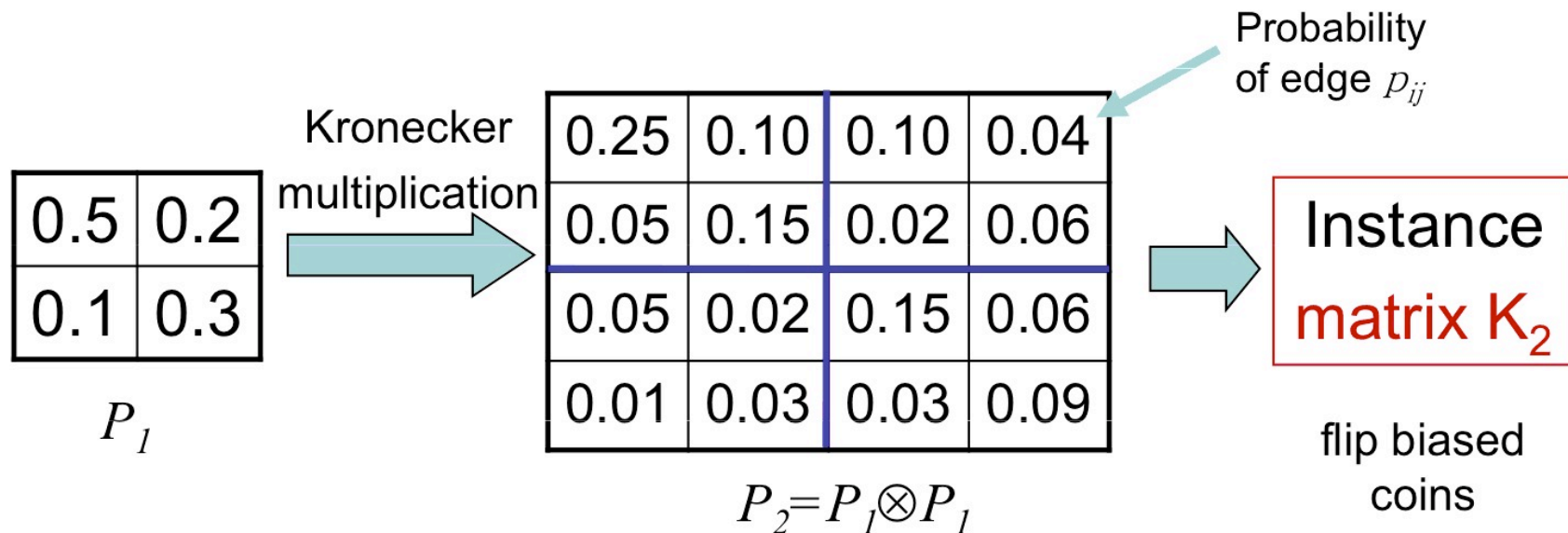
Structure of the  $G(\mathbf{w})$  model, with  $\beta \in (2, 3)$ .

- **Sparsity** (coupled with randomness) **is the issue**, not heavy-tails.
- (Power laws with  $\beta \in (2, 3)$  give us the appropriate sparsity.)

$\alpha$	$\beta$
$\beta$	$\gamma$

# Stochastic Kronecker Graphs

Leskovec, et al. (arXiv 2009); Mahdian-Xu 2007



**Deterministic** version - can reproduce HT degrees, densification power law, etc

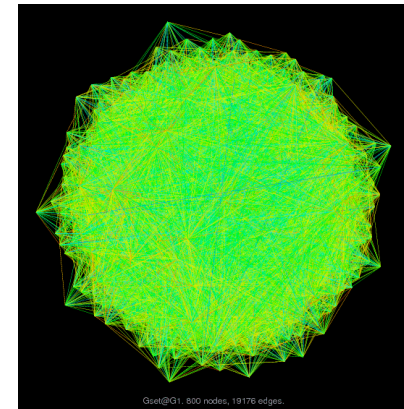
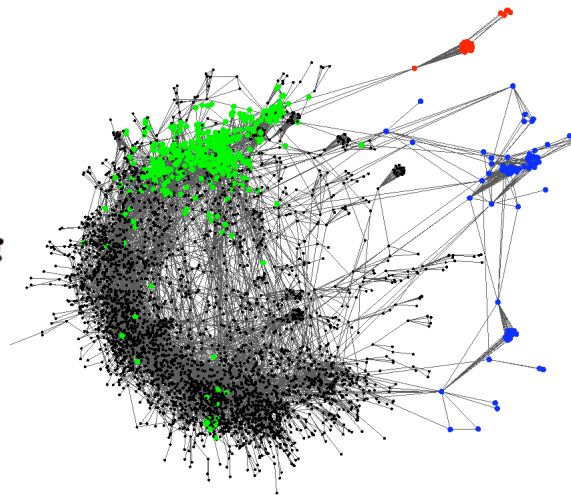
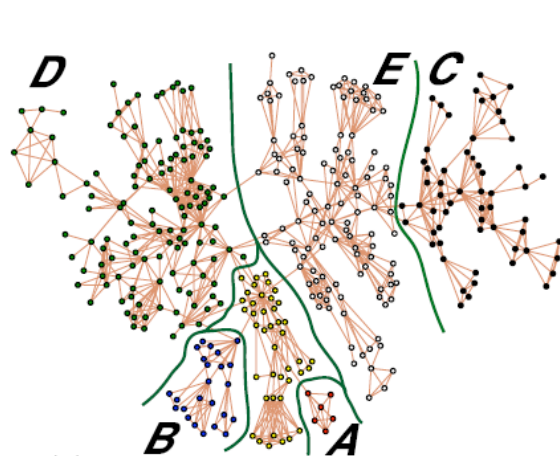
**Stochastic** version - Ass  $1 \geq \alpha \geq \beta \geq \gamma \geq 0$ . Connected iff  $\beta + \gamma > 1$  or  $\alpha = \beta = 1, \gamma = 0$ . Giant component iff  $(\alpha + \beta)(\beta + \gamma) > 1$  or  $(\alpha + \beta)(\beta + \gamma) = 1, \alpha + \beta > \beta + \gamma$

$\alpha$	$\beta$
$\beta$	$\gamma$

# Small versus Large Networks

Leskovec, et al. (arXiv 2009); Mahdian-Xu 2007

- Small and large networks are very different:  
(also, an expander)



E.g., fit these networks to Stochastic Kronecker Graph with "base"  $K=[a \ b; \ b \ c]$ :

$$K_1 = \begin{bmatrix} 0.99 & 0.17 \\ 0.17 & 0.82 \end{bmatrix}$$

$$\begin{bmatrix} 0.99 & 0.55 \\ 0.55 & 0.15 \end{bmatrix}$$

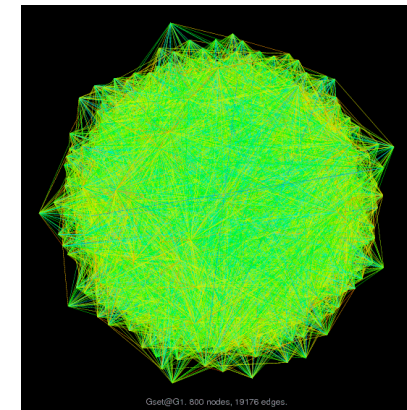
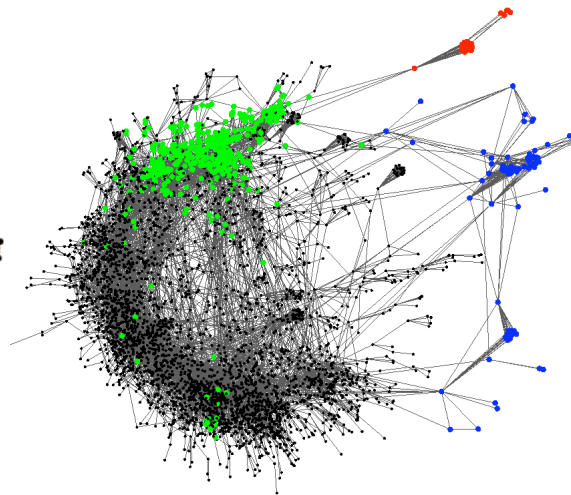
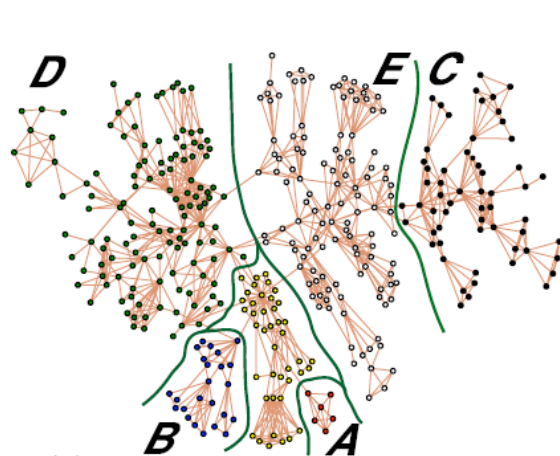
$$\begin{bmatrix} 0.2 & 0.2 \\ 0.2 & 0.2 \end{bmatrix}$$

$\alpha$	$\beta$
$\beta$	$\gamma$

# Small versus Large Networks

Leskovec, et al. (arXiv 2009); Mahdian-Xu 2007

- Small and large networks are very different:  
(also, an expander)



E.g., fit these networks to Stochastic Kronecker Graph with "base"  $K=[a \ b; \ b \ c]$ :

$$K_1 = \begin{bmatrix} \text{dark gray} & \text{light gray} \\ \text{light gray} & \text{dark gray} \end{bmatrix}$$

$$K_2 = \begin{bmatrix} \text{dark gray} & \text{light gray} \\ \text{light gray} & \text{light gray} \end{bmatrix}$$

$$K_3 = \begin{bmatrix} \text{light gray} & \text{light gray} \\ \text{light gray} & \text{light gray} \end{bmatrix}$$



# Implications: for Community Detection

- Linear (Low-rank) methods

If Gaussian, then low-rank space is good.

- Kernel (non-linear) methods

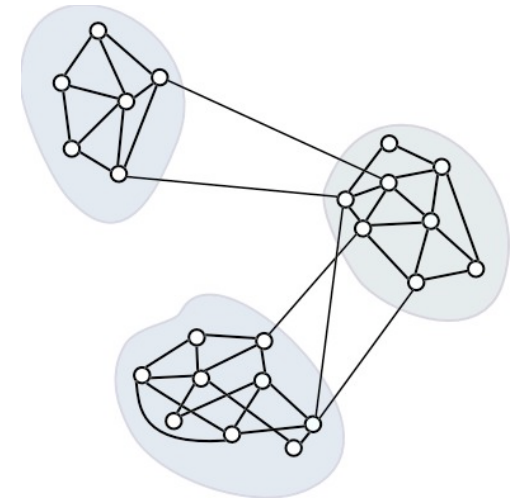
If low-dimensional manifold, then kernels are good

- Hierarchical methods

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

- Graph partitioning methods

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



*(Good and large) network communities, at least when formalized i.t.o. this bicriterion, don't really exist in these graphs!!*

*"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."*





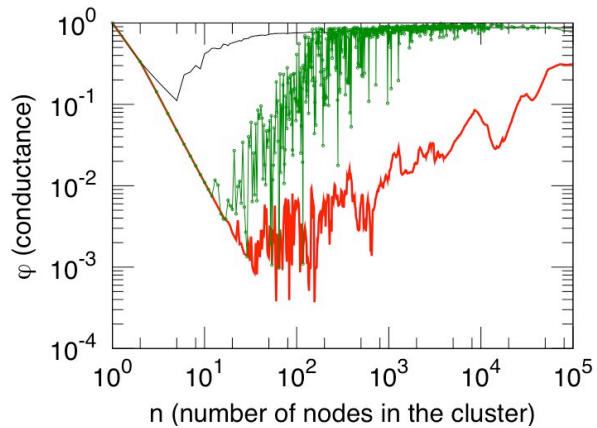
## Comparison with "Ground truth" (1 of 2)

---

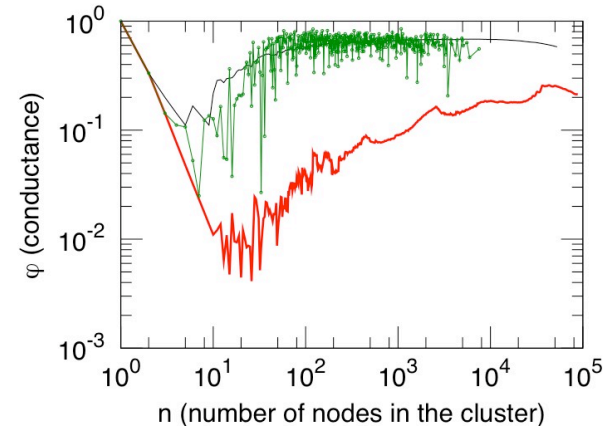
Networks with "ground truth" communities:

- LiveJournal12:
  - users create and explicitly join on-line groups
- CA-DBLP:
  - publication venues can be viewed as communities
- AmazonAllProd:
  - each item belongs to one or more hierarchically organized categories, as defined by Amazon
- AtM-IMDB:
  - countries of production and languages may be viewed as communities (thus every movie belongs to exactly one community and actors belongs to all communities to which movies in which they appeared belong)

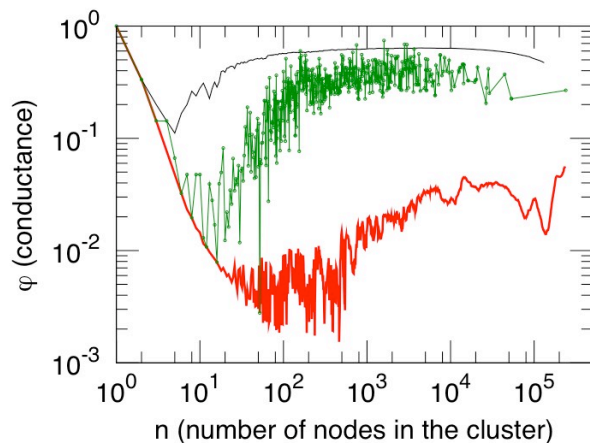
## Comparison with "Ground truth" (2 of 2)



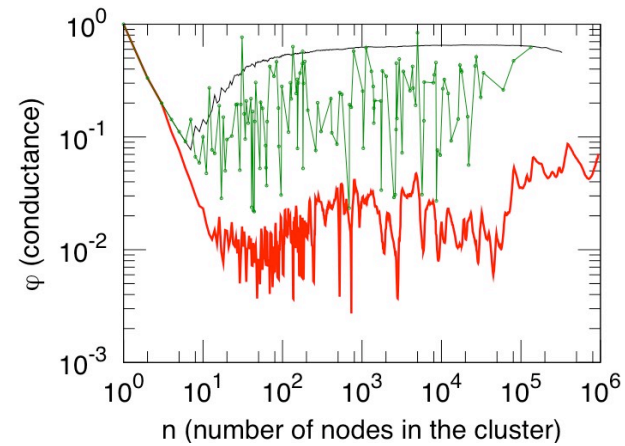
LiveJournal



CA-DBLP



AmazonAllProd



AtM-IMDB



## Miscellaneous thoughts ...

---

### Sociological work on community size (Dunbar and Allen)

- 150 individuals is maximum community size
- Military companies, on-line communities, divisions of corporations all  $\leq 150$

### Common bond vs. common identity theory

- Common bond - people are attached to individual community members
- Common identity - people are attached to the group as a whole

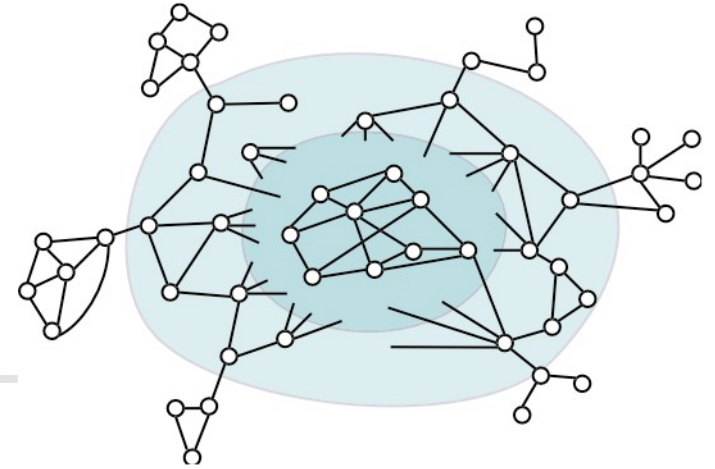
### What edges “mean” and community identification

- social networks - reasons an individual adds a link to a friend very diverse
- citation networks - links are more “expensive” and semantically uniform.



## Implications: high level

---



What is **simplest explanation** for empirical facts?

- **Extremely sparse Erdos-Renyi** reproduces qualitative NCP (i.e., deep cuts at small size scales and no deep cuts at large size scales) since:

sparsity + randomness = measure fails to concentrate

- **Power law random graphs** also reproduces qualitative NCP for analogous reason
- **Iterative forest-fire model** gives mechanism to put **local geometry** on sparse quasi-random scaffolding to get qualitative property of **relatively gradual increase of NCP**

Data are **local-structure on global-noise**, not small noise on global structure!



## Implications: high level, cont.

Remember the Stochastic Kronecker theorem:

- **Connected**, if  $b+c > 1$ :  $0.55+0.15 > 1$ . **No!**
- **Giant component**, if  $(a+b) \cdot (b+c) > 1$ :  $(0.99+0.55) \cdot (0.55+0.15) > 1$ . **Yes!**

Real graphs are in a region of parameter space analogous to *extremely sparse*  $G_{np}$ .

- Large vs small cuts, degree variability, eigenvector localization, etc.



Data are *local-structure on global-noise*, not small noise on global structure!



# Degree heterogeneity and hyperbolicity

---

Social and information networks are expander-like at large size scales, but:

- Degree heterogeneity enhances hyperbolicity

Lots of evidence:

- Scale free and internet graphs are more hyperbolic than other models, MC simulation - Jonckheere and Lohsoonthorne (2007)
- Mapping network nodes to spaces of negative curvature leads to scale-free structure - Krioukov et al (2008)
- Measurements of Internet are Gromov negatively curved - Baryshnikov (2002)
- Curvature of co-links interpreted as thematic layers in WWW - Eckmann and Moses (2002)

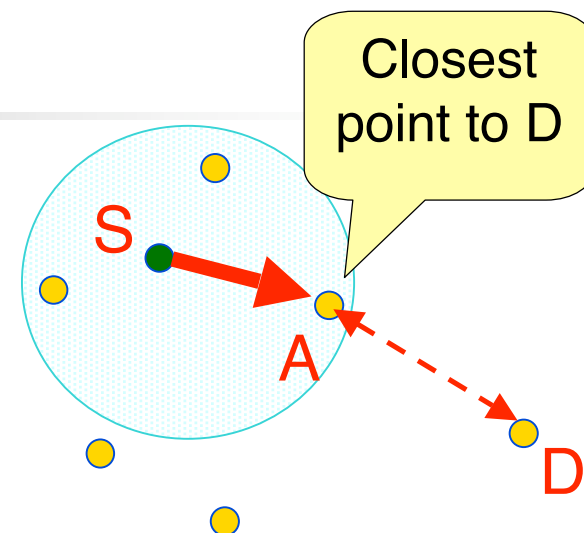
Question: Has anyone made this observation precise?

# Hyperbolic Application 1: Internet Routing

A LARGE area - lots of other work.

## Geographic routing protocols:

- A node knows (1) its location (physical or virtual coordinates), (2) its neighbors and their location, and (3) destination's location
- Forward packets to make progress to destination.



## Euclidean versus Hyperbolic embeddings:

- Use virtual coordinates (Rao et al 2004, Fonseca et al 2005)
- Hyperbolic embeddings of same dimension do better (Shavitt and Tankel (2004,2008)
- Q: Which graphs have greedy embedding in the plane? (Papadimitriou and Rataczyk 2004)
- A: Every finite graph has greedy embedding in the hyperbolic plane. (R.Kleinberg 2005)

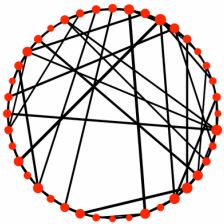


# Hyperbolic Application 2: Decentralized Search in Social Graphs



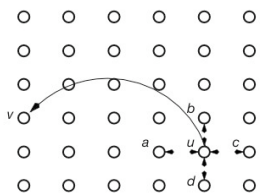
*Milgram (1960s)*

- Small world experiments - study **short paths** in social networks



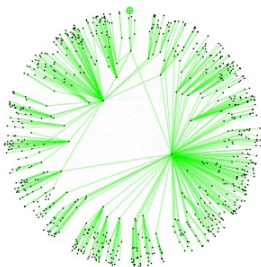
*Watts and Strogatz (1998)*

- Model that reproduce **local clustering** and **existence of short paths**



*Kleinberg (2000)*

- Model s.t. decentralized search can *find* short paths efficiently
- Careful **coupling of "local" geometric structure and "global" structure.**

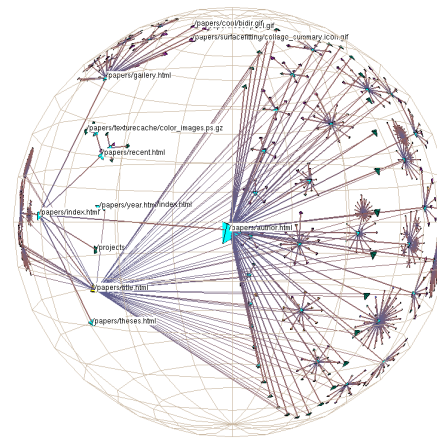
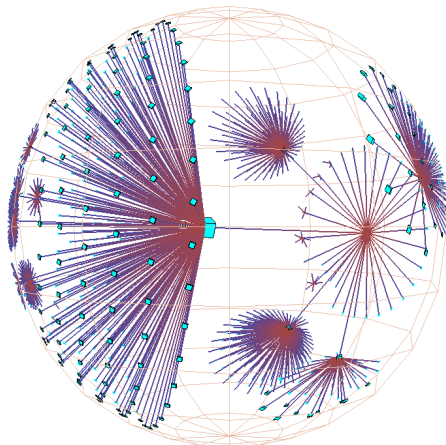
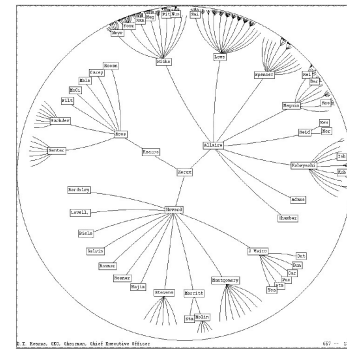
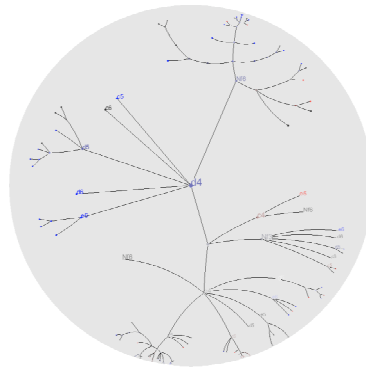
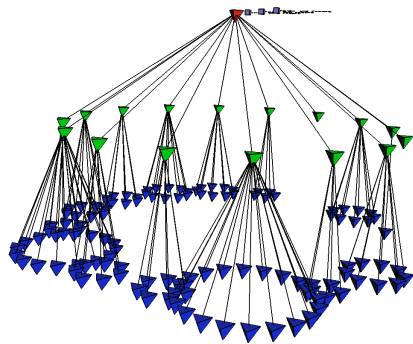


*Boguna, Krioukov, and Claffy (2008)*

- Model with degree heterogeneity for efficient decentralized search
- **Analogous local-global coupling imply embedding in hyperbolic space**

# Hyperbolic Application 3: Internet and Web Visualization

Munzner and Burchard (1995); Lamping, Rao, and Pirolli (1995); Munzner (1998)



Like the “fish-eye”  
camera lens, but  
avoids some ad-  
hoc decisions.

“There is no good way of embedding an exponentially growing tree in Euclidean space that allows us to simultaneously see both the entire structure and a closeup of a particular region. The solution is to use hyperbolic ... geometry ...” Munzner and Burchard (1995)



## "Routing" versus "diffusion" metrics

---

Consider two classes of "distances" between nodes:

- "Diffusion-type" distance - related to (spectral methods and) diffusion or commute times
- "Geodesic-type" distance - related to (flow-based methods and) routing or shortest paths

Question 1: Which is better? More useful? (As a function of the type of graph)?

Question 2: Given that a process goes from A to B with one of those processes, how does the path compare with the other process?

# Routing versus diffusions, cont\*.

## Low Dimensional Graphs

- Diffusions are discriminative and useful
- Flows and geodesics are too sensitive

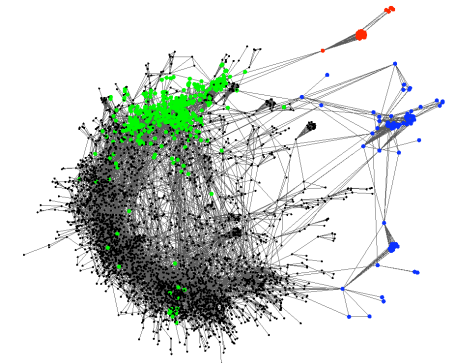
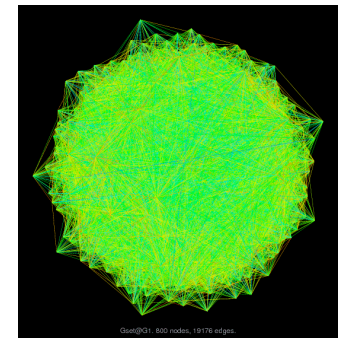
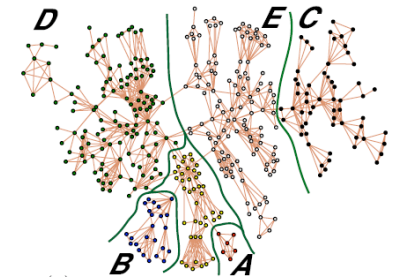
## Expander-like Graphs

- Diffusions not discriminative or useful
- Multicommodity flow and geodesics useful?

## Hyperbolic Graphs

- Diffusion path and routing path are the same.

\*Question: Does anyone know of a formalization of this intuition?



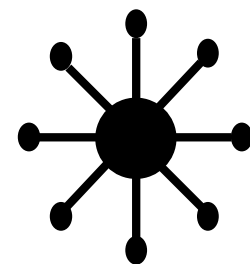
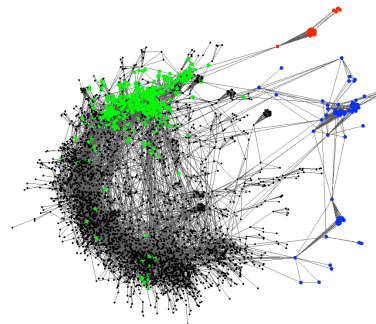
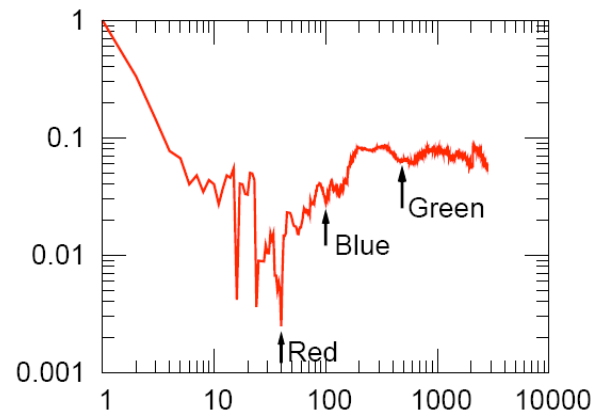
# Hyperbolic Application 4: Clustering and Community Structure

Hyperbolic properties at large size scales:

- (Degree-weighted) expansion at large size-scales
- Degree heterogeneity

*Local pockets of structure on hyperbolic scaffolding.*

- (Traditionally-conceptualized) communities get worse and worse as they get larger and larger



$\alpha$	$\beta$
$\beta$	$\gamma$

 = 

0.99	0.55
0.55	0.15



# Implications for Data Analysis and ML

---

Principled and scalable **algorithmic exploratory analysis tools**:

- spectral vs. flow vs. combinations; local vs. global vs. improvement; etc.

Doing **inference directly on data graphs**, and machine **learning in complex data environments**:

- don't do inference on feature vectors with hyperplanes in a vector space
- need methods to do it in high-variability, *only approximately* low-dimensional, tree-like or expander-like environments.

**Implicit regularization via approximate computation**:

- spectral vs. flow vs. combinations; local vs. global vs. improvement; etc.



## Data Application 1:

# Approximate eigenvector computation

---

Many uses of Linear Algebra in ML and Data Analysis involve *approximate* computations

- Power Method, Truncated Power Method, HeatKernel, Truncated Random Walk, PageRank, Truncated PageRank, Diffusion Kernels, TrustRank, etc.
- Often they come with a “generative story,” e.g., random web surfer, teleportation preferences, drunk walkers, etc.

What are these procedures *actually* computing?

- E.g., what optimization problem is 3 steps of Power Method solving?
- Important to know if we really want to “scale up”





# Implicit Regularization

---

**Regularization:** A general method for computing “smoother” or “nicer” or “more regular” solutions - useful for inference, etc.

**Recall:** Regularization is usually *implemented* by adding “regularization penalty” and optimizing the new objective.

$$\hat{x} = \operatorname{argmin}_x f(x) + \lambda g(x)$$

**Empirical Observation:** Heuristics, e.g., binning, early-stopping, etc. often implicitly perform regularization.

**Question:** Can approximate computation\* *implicitly* lead to more regular solutions? If so, can we exploit this algorithmically?

\*Here, consider approximate eigenvector computation. But, can it be done with graph algorithms?



## Two versions of spectral partitioning

---

**VP:**

$$\begin{array}{ll}\min. & x^T L_G x \\ \text{s.t.} & x^T L_{K_n} x = 1 \\ & \langle x, 1 \rangle_D = 0\end{array}$$



**R-VP:**

$$\begin{array}{ll}\min. & x^T L_G x + \lambda f(x) \\ \text{s.t.} & \text{constraints}\end{array}$$



**SDP:**

$$\begin{array}{ll}\min. & L_G \circ X \\ \text{s.t.} & L_{K_n} \circ X = 1 \\ & X \succeq 0\end{array}$$



**R-SDP:**

$$\begin{array}{ll}\min. & L_G \circ X + \lambda F(X) \\ \text{s.t.} & \text{constraints}\end{array}$$



# Views of approximate spectral methods

---

Three common procedures (L=Laplacian, and M=r.w. matrix):

- **Heat Kernel:**  $H_t = \exp(-tL) = \sum_{k=0}^{\infty} \frac{(-t)^k}{k!} L^k$
- **PageRank:**  $\pi(\gamma, s) = \gamma s + (1 - \gamma) M \pi(\gamma, s)$   
 $R_\gamma = \gamma (I - (1 - \gamma) M)^{-1}$
- **q-step Lazy Random Walk:**  $W_\alpha^q = (\alpha I + (1 - \alpha) M)^q$

Ques: Do these “*approximation* procedures” *exactly* optimizing some regularized objective?



## A simple theorem

---

$$\begin{aligned} (\mathbf{F}, \eta)\text{-SDP} \quad & \min \quad L \bullet X + \frac{1}{\eta} \cdot F(X) \\ & \text{s.t.} \quad I \bullet X = 1 \\ & \quad \quad X \succeq 0 \end{aligned}$$

Modification of the usual SDP form of spectral to have regularization (but, on the matrix  $X$ , not the vector  $x$ ).

**Theorem:** Let  $G$  be a connected, weighted, undirected graph, with normalized Laplacian  $L$ . Then, the following conditions are sufficient for  $X^*$  to be an optimal solution to  $(\mathbf{F}, \eta)\text{-SDP}$ .

- $X^* = (\nabla F)^{-1} (\eta \cdot (\lambda^* I - L))$ , for some  $\lambda^* \in \mathbb{R}$ ,
- $I \bullet X^* = 1$ ,
- $X^* \succeq 0$ .



## Three simple corollaries

---

$$F_H(X) = \text{Tr}(X \log X) - \text{Tr}(X) \text{ (i.e., generalized entropy)}$$

gives scaled *Heat Kernel matrix*, with  $t = \eta$

$$F_D(X) = -\log \det(X) \text{ (i.e., Log-determinant)}$$

gives scaled *PageRank matrix*, with  $t \sim \eta$

$$F_p(X) = (1/p) \|X\|_p^p \text{ (i.e., matrix p-norm, for } p > 1)$$

gives *Truncated Lazy Random Walk*, with  $\lambda \sim \eta$

These "approximation procedures" compute regularized versions of the Fiedler vector!



## Large-scale applications

---

*A lot of work on large-scale data already implicitly uses these ideas:*

- Fuxman, Tsaparas, Achan, and Agrawal (2008): random walks on query-click for automatic keyword generation
- Najork, Gallapudi, and Panigraphy (2009): carefully “whittling down” neighborhood graph makes SALSA faster and better
- Lu, Tsaparas, Ntoulas, and Polanyi (2010): test which page-rank-like implicit regularization models are most consistent with data

**Question:** Can we formalize this to understand *when it succeeds and when it fails?*



## Data Application 2: Classification in ML

---

### Supervised binary classification

- **Observe**  $(X,Y) \in (X,Y) = (\mathbb{R}^n, \{-1,+1\})$  sampled from unknown distribution  $P$
- **Construct classifier**  $\alpha: X \rightarrow Y$  (drawn from some family  $\Lambda$ , e.g., hyper-planes) after seeing  $k$  samples from unknown  $P$

Question: How big must  $k$  be to get good prediction, i.e., low error?

- **Risk**:  $R(\alpha)$  = probability that  $\alpha$  misclassifies a random data point
- **Empirical Risk**:  $R_{\text{emp}}(\alpha)$  = risk on observed data

Ways to bound  $|R(\alpha) - R_{\text{emp}}(\alpha)|$  over all  $\alpha \in \Lambda$

- **VC dimension**: distribution-independent; typical method
- **Annealed entropy**: distribution-dependent; but can get much finer bounds





## Unfortunately ...

---

Sample complexity of *dstbn-free learning* typically depends on the *ambient dimension* to which the data to be classified belongs

- E.g.,  $\Omega(d)$  for learning half-spaces in  $\mathbb{R}^d$ .

*Very unsatisfactory* for *formally* high-dimensional data

- *approximately low-dimensional environments* (e.g., close to manifolds, empirical signatures of low-dimensionality, etc.)
- *high-variability environments* (e.g., heavy-tailed data, sparse data, pre-asymptotic sampling regime, etc.)

**Ques:** Can *distribution-dependent tools* give improved learning bounds for data with *more realistic sparsity and noise*?



# Annealed entropy

---

**Definition (Annealed Entropy):** Let  $\mathcal{P}$  be a probability measure on  $\mathcal{H}$ . Given a set  $\Lambda$  of decision rules and a set of points  $Z = \{z_1, \dots, z_\ell\} \subset \mathcal{H}$ , let  $N^\Lambda(z_1, \dots, z_\ell)$  be the number of ways of labeling  $\{z_1, \dots, z_\ell\}$  into positive and negative samples. Then,

$$H_{ann}^\Lambda(k) := \ln E_{\mathcal{P} \times k} N^\Lambda(z_1, \dots, z_k)$$

is the *annealed entropy* of the classifier  $\Lambda$  with respect to  $\mathcal{P}$ .

**Theorem:** Given the above notation, the inequality

$$\text{Prob} \left[ \sup_{\alpha \in \Lambda} \frac{R(\alpha) - R_{emp}(\alpha, \ell)}{\sqrt{R(\alpha)}} > \epsilon \right] < 4 \exp \left( \left( \frac{H_{ann}^\Lambda(2\ell)}{\ell} - \frac{\epsilon^2}{4} \right) \ell \right)$$

holds true, for any number of samples  $\ell$  and for any error parameter  $\epsilon$ .



# "Toward" learning on informatics graphs

---

*Dimension-independent* sample complexity bounds for

- *High-variability environments*
  - probability that a feature is nonzero decays as power law
  - magnitude of feature values decays as a power law
- *Approximately low-dimensional environments*
  - when have bounds on the covering number in a metric space
  - when use diffusion-based spectral kernels

Bound  $H_{\text{ann}}$  to get exact or gap-tolerant classification

**Note:** "toward" since we still learning in a vector space, not *directly* on the graph

# Eigenvector localization ...

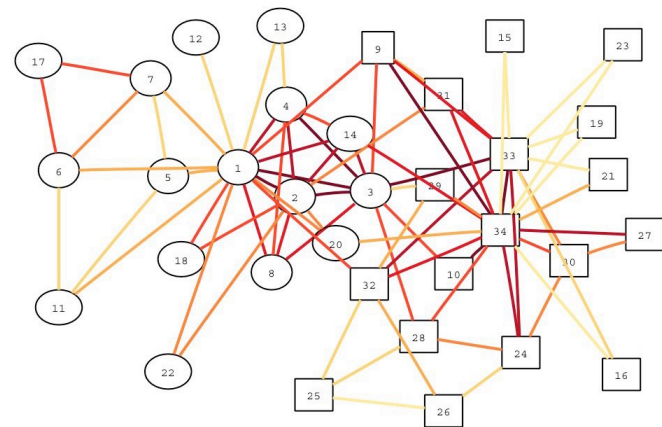
Let  $\{f_i\}_{i=1}^n$  be the eigenfunctions of the normalized Laplacian of  $\mathcal{L}_G$  and let  $\{\lambda_i\}_{i=1}^n$  be the corresponding eigenvalues. Then, **Diffusion Maps** is:

$$\Phi : v \mapsto (\lambda_0^k f_0(v), \dots, \lambda_n^k f_n(v)),$$

and **Laplacian Eigenmaps** is the special case of this feature map when  $k = 0$ .

## When do eigenvectors localize?

- High degree nodes.
- Articulation/boundary points.
- Points that “stick out” a lot.
- Sparse random graphs



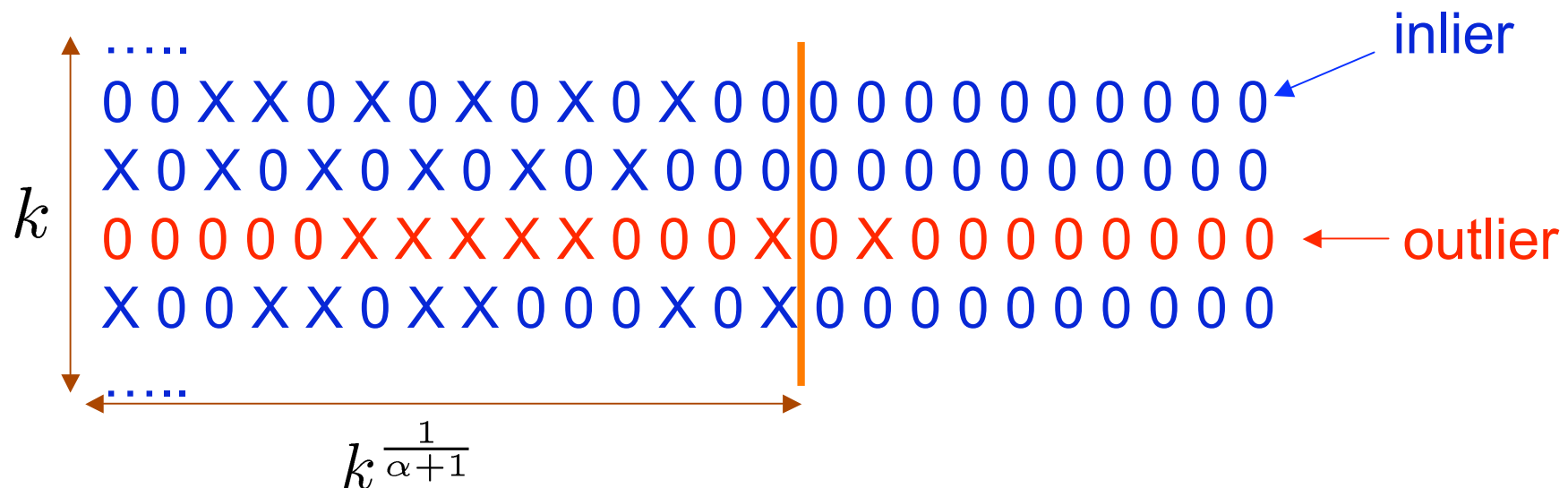
This is seen in many data sets when eigen-methods are chosen for algorithmic, and not statistical, reasons.

# Exact learning with a heavy-tail model

Mahoney and Narayanan (2009,2010)

**Heavy-tailed model:** Let  $\mathcal{P}$  be a probability distribution in  $R^d$ . Suppose  $\mathcal{P}[x_i \neq 0] \leq Ci^{-\alpha}$  for some absolute constant  $C > 0$ , with  $\alpha > 1$ .

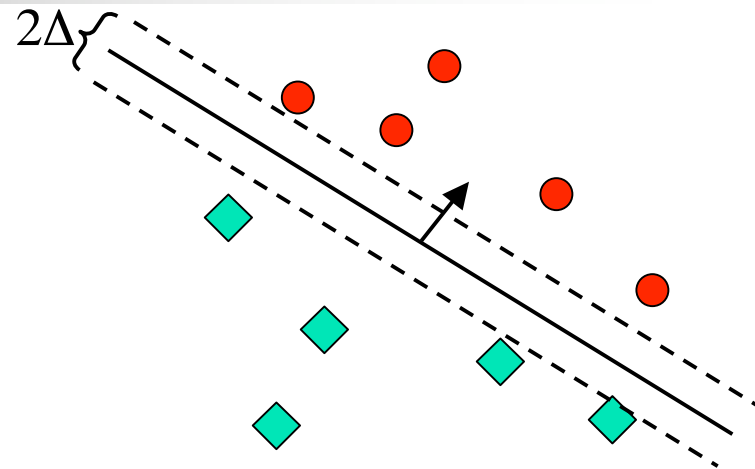
**Theorem:** In this model,  $H_{ann}^\Lambda(\ell) \leq \left( \frac{C}{\alpha-1} \ell^{\frac{1}{\alpha}} + 1 \right) \ln(\ell)$ . Thus, need only  $\ell = \tilde{O} \left( \left( \frac{C \ln(\delta^{-1})}{\epsilon^2} \right)^{\frac{\alpha+1}{\alpha}} \right)$  samples, independent of (possibly infinite)  $d$ .



# Gap-tolerant classification

Mahoney and Narayanan (2009,2010)

**Def:** A *gap-tolerant classifier* consists of an oriented hyper-plane and a margin of thickness  $\Delta$  around it. Points outside the margin are labeled  $\pm 1$ ; points inside the margin are simply declared “correct.”



Only the expectation of the norm needs to be bounded! Particular elements can behave poorly!

**Theorem:** Let  $\mathcal{P}$  be a probability measure on a Hilbert space  $\mathcal{H}$ , and let  $\Delta > 0$ . If  $E_{\mathcal{P}} \|x\|^2 = r^2 < \infty$ , then the annealed entropy of gap-tolerant classifiers in  $\mathcal{H}$ , where the gap is  $\Delta$ , is

$$H_{ann}^{\Delta}(\ell) \leq \left( \ell^{\frac{1}{2}} \left( \frac{r}{\Delta} \right) + 1 \right) (1 + \ln(\ell + 1)).$$

so can get dimension-independent bounds!



# Large-margin classification with very "outlying" data points

Mahoney and Narayanan (2009,2010)

Apps to dimension-independent large-margin learning:

- with **spectral kernels**, e.g. **Diffusion Maps kernel** underlying manifold-based methods, on **arbitrary graphs**
- with **heavy-tailed data**, e.g., when the **magnitude of the elements** of the feature vector decay in a **heavy-tailed** manner

Technical notes:

- new proof bounding VC-dim of gap-tolerant classifiers in Hilbert space generalizes to **Banach spaces** - useful if dot products & kernels too limiting
- Ques: *Can we control aggregate effect of "outliers" in other data models?*
- Ques: *Can we learn if measure never concentrates?*





## Data application 2, more generally ...

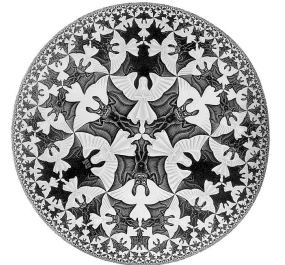
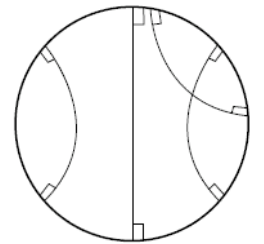
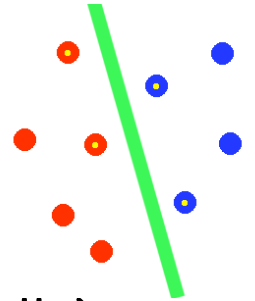
---

Machine learning in environments more general than  $\mathbb{R}^n$  or RKHS?

- On expander-like or hyperbolic structures (locally/globally)
- On other classes of metric spaces, while exploiting metric/geometric structure for learning?
- How do ideas like margin, etc. generalize?

*Learn directly on graph* (non-vector/matrix) data

- i.e., don't filter through vector space, but perform capacity control, etc directly on graph
- don't assume  $m, n, p \rightarrow \infty$  in a nice way





## Conclusions (1 of 4): General Observations

---

Network data are often **very/extremely large**:

- Premium on fast/scalable algorithms
- (Good - lots of algorithms; Bad - they often return meaningless answers.)

Network data are often **very/extremely sparse**:

- Premium on statistical regularization
- (Good - lots of regularization methods; Bad - they work on vectors, not graphs.)
- BTW, this implies “landmark point methods” often inappropriate

Networks have **complex, nonlinear, adversarial structure**

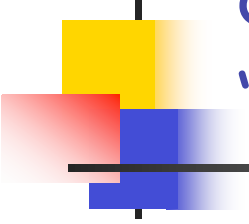
- Structures don't exist in small (e.g., thousands of nodes) networks
- Need tools to explore things we can't visualize
- Big difference between “analyst apps” and “next-user-interaction apps”



## Conclusions (2 of 4): General Observations

---

- Algorithmic primitives to “probe” networks locally and globally
- Infer properties of original network from statistical and regularization properties of ensembles of approximate solutions
- Real informatics graphs -- very different than small commonly-studied graphs and existing generative models
- Tools promising for coupling local properties (often low-dimensional) and global properties (often expander-like)
- Tools promising to study pre-existing geometry versus generated geometry - recall  $geometry \approx inference$
- Validation is difficult - if you have a clean validation and/or a pretty picture, you're looking at unrealistic network data!



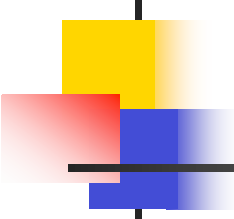
## Conclusion (3 of 4) : "Structure" and "randomness" in large informatics graphs

---

High-level observations to formalize:

- There **do not exist** a "small" number of linear components that capture "most" of the variance/information in the data.
- There **do not exist** "nice" manifolds that describe the data well.
- There is "locally linear" structure or geometry on small size scales that **does not propagate** to global/large size scales.
- At large size scales, the "true" geometry is more "hyperbolic" or "tree-like" or "expander-like".

**Important:** *even if you do not care about communities, conductance, hyperbolicity, etc., these empirical facts place **very severe constraints** on the **types of models** and **types of analysis tools** that are appropriate.*



## Conclusion (4 of 4): Geometric Network Analysis Tools?

---

- **Approximation algorithms have geometry hidden somewhere**  
Spectral methods, LP methods, tree methods, metric embeddings
- **Local Spectral Methods**  
Identify geometry at multiple nodes at multiple size scales  
No need to assume local geometries are on a global manifold
- **Approximate Computation as Implicit Regularization**  
Approximate solutions are better than exact solutions  
Especially relevant for extremely sparse/noisy networks  
Use this to regularize and do inference directly on network?
- **Methodological test case**  
Good “hydrogen atom” for development of algorithmic and statistical tools for probing graph data more generally