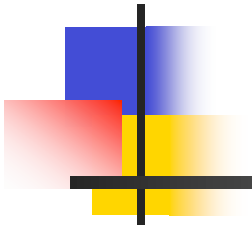
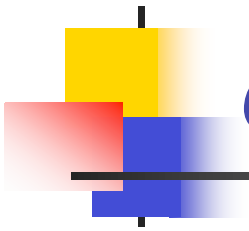


Machine Learning and Linear Algebra of Large Informatics Graphs

Michael W. Mahoney

Stanford University

(For more info, see:
http:// cs.stanford.edu/people/mmahoney/
or Google on "Michael Mahoney")



Outline

A Bit of History of ML and LA

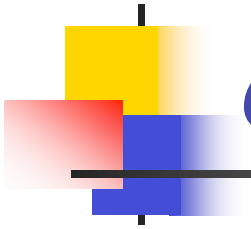
- Role of data, noise, randomization, and recently-popular algorithms

Large Informatics Graphs

- Characterize small-scale and large-scale clustering structure
- Provides novel perspectives on matrix and graph algorithms

New Machine Learning and New Linear Algebra

- Optimization view of "local" version of spectral partitioning
- Regularized optimization perspective on: PageRank, HeatKernel, and Truncated Iterated Random Walk
- Beyond VC bounds: Learning in high-variability environments



Outline

A Bit of History of ML and LA

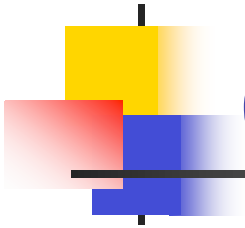
- Role of data, noise, randomization, and recently-popular algorithms

Large Informatics Graphs

- Characterize small-scale and large-scale clustering structure
- Provides novel perspectives on matrix and graph algorithms

New Machine Learning and New Linear Algebra

- Optimization view of "local" version of spectral partitioning
- Regularized optimization perspective on: PageRank, HeatKernel, and Truncated Iterated Random Walk
- Beyond VC bounds: Learning in high-variability environments



(Biased) History of NLA

≤ 1940s: Prehistory

- Close connections to data analysis, noise, statistics, randomization

1950s: Computers

- Banish randomness & downgrade data (except in scientific computing)

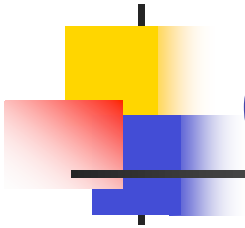
1980s: NLA comes of age - high-quality codes

- QR, SVD, spectral graph partitioning, etc. (written for HPC)

1990s: Lots of new DATA

- LSI, PageRank, NCuts, etc., etc., etc. used in ML and Data Analysis

2000s: New problems force new approaches ...



(Biased) History of ML

≤ 1940s: Prehistory

- Do statistical data analysis "by hand"; the "computers" were people

1960s: Beginnings of ML

- Artificial intelligence, neural networks, perceptron, etc.

1980s: Combinatorial foundations for ML

- VC theory, PAC learning, etc.

1990s: Connections to Vector Space ideas

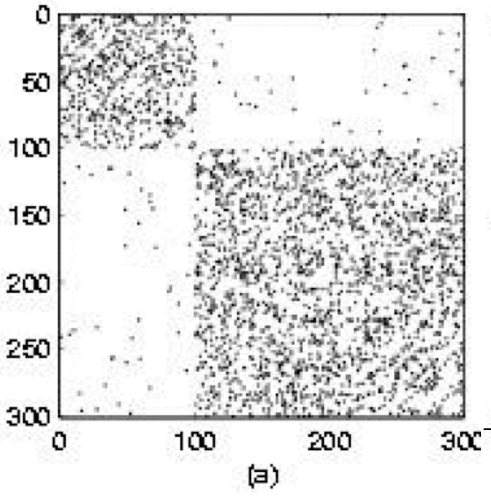
- Kernels, manifold-based methods, Normalized Cuts, etc.

2000s: New problems force new approaches ...

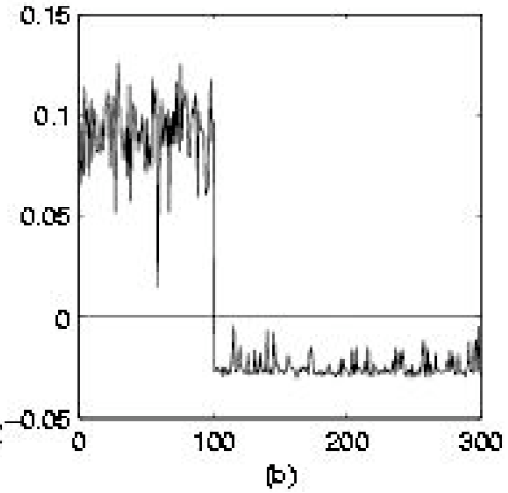
Spectral Partitioning and NCuts

$$\begin{aligned} \text{minimize} \quad & x^T L_G x \\ \text{s.t.} \quad & \langle x, x \rangle_D = 1 \\ & \langle x, 1 \rangle_D = 0 \end{aligned}$$

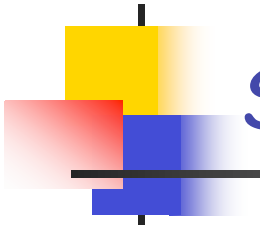
Adjacency matrix



Eigenvector q_2



- Solvable via eigenvalue problem
- Bounds via Cheeger's inequality
- Used in parallel scientific computing, Computer Vision (called Normalized Cuts), and Machine Learning
- Connections between graph Laplacian and manifold Laplacian
- *But, what if there are not "good well-balanced" cuts (as in "low-dim" data)?*



Spectral Ranking and PageRank

Vigna (TR - 2010)

PageRank - the "damped spectral ranking of normalized adjacency matrix of web graph"

Long history of similar "ranking" ideas - Seely 1949; Wei 1952; Katz 1953; Hubbell 1965; etc.; etc.; etc.

Potential Surprises:

- When computed, *approximate it* with the Power Method (Ugh?)
- Of minimal importance in today's ranking functions (Ugh?)
- Connections to Fiedler vector, clustering, and data partitioning.

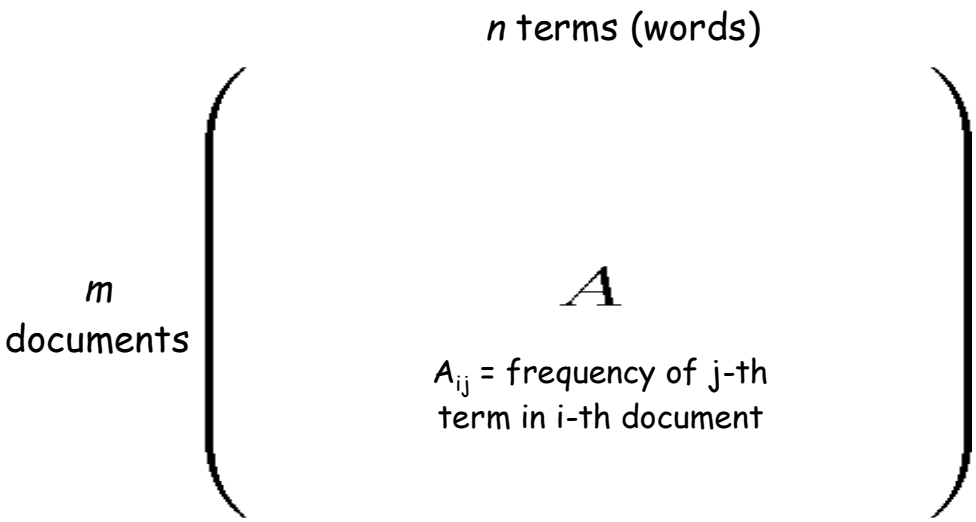
LSI: A_k for document-term "graphs"

(Berry, Dumais, and O'Brien '92)

Best rank-k approx to A.

Latent Semantic Indexing (LSI)

Replace A by A_k; apply clustering/classification algorithms on A_k.



Pros

- Less storage for small *k*.

O(km+kn) vs. O(mn)

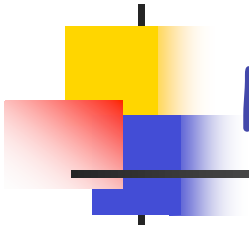
- Improved performance.

Documents are represented in a "concept" space.

Cons

- A_k destroys sparsity.
- Interpretation is difficult.
- Choosing a good *k* is tough.

- Can interpret document corpus in terms of k topics.
- Or think of this as just selecting one model from a parameterized class of models!



Problem 1: SVD & "heavy-tailed" data

Theorem: (Mihail and Papadimitriou, 2002)

The largest eigenvalues of the adjacency matrix of a graph with power-law distributed degrees are also power-law distributed.

- I.e., heterogeneity (e.g., heavy-tails over degrees) plus noise (e.g., random graph) implies heavy tail over eigenvalues.
- Idea: 10 components may give 10% of mass/information, but to get 20%, you need 100, and to get 30% you need 1000, etc; i.e., no scale at which you get most of the information
- No "latent" semantics without preprocessing.

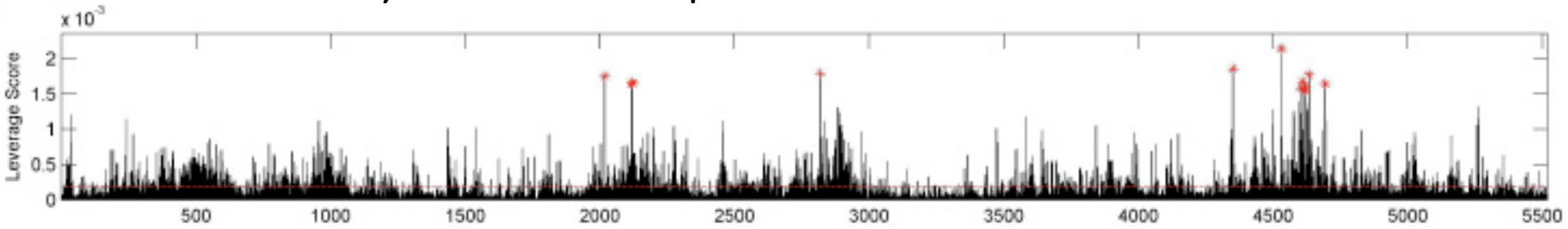
Problem 2: SVD & "high-leverage" data

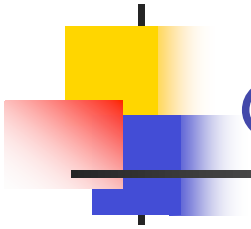
Given an m x n matrix A and rank parameter k:

- How localized, or coherent, are the (left) singular vectors?
- Let $\rho_i = (P_{Uk})_{ii} = ||U_k^{(i)}||_2$ (where U_k is any o.n. basis spanning that space)

These "statistical leverage scores" quantify which rows have the most influence/leverage on low-rank fit

- Essential for "bridging the gap" between NLA and TCS-- and making TCS randomized algorithms numerically-implementable
- Often *very* non-uniform in practice



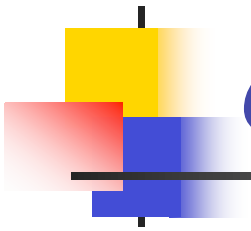


Q: Why do SVD-based methods work at all?

Given that the "assumptions" underlying its use (approximately low-rank and no high-leverage data points) are so manifestly violated.

A: Low-rank spaces are *very* structured places.

- If "all models are wrong, but some are useful," those that are useful have "capacity control."
- Low-rank structure is implicitly capacity control -- like bound on VC dimension of hyperplanes
- Diffusions and L2 methods "aggregate" information in very particular way (with associated plusses *and* minusses)
- Not so with multi-linearity, non-negativity, sparsity, graphs, etc.



Outline

A Bit of History of ML and LA

- Role of data, noise, randomization, and recently-popular algorithms

Large Informatics Graphs

- Characterize small-scale and large-scale clustering structure
- Provides novel perspectives on matrix and graph algorithms

New Machine Learning and New Linear Algebra

- Optimization view of "local" version of spectral partitioning
- Regularized optimization perspective on: PageRank, HeatKernel, and Truncated Iterated Random Walk
- Beyond VC bounds: Learning in high-variability environments

Networks and networked data

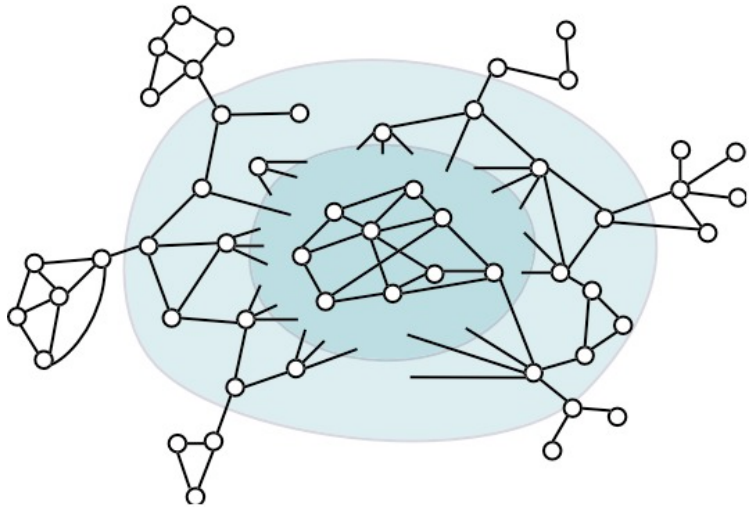
Lots of "networked" data!!

- 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



Large Social and Information Networks

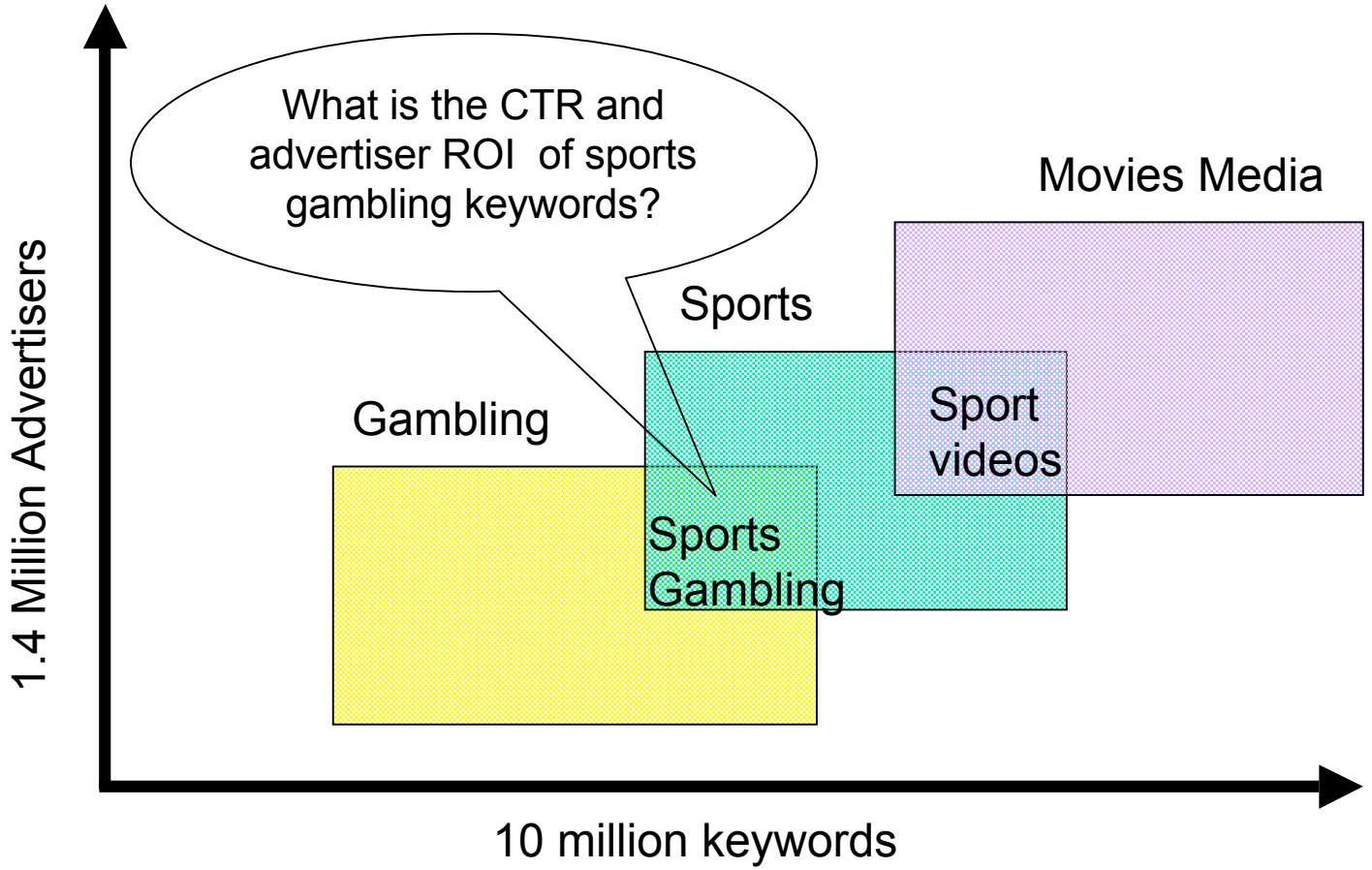
• Social nets	Nodes	Edges	Description
LIVEJOURNAL	4,843,953	42,845,684	Blog friendships [4]
EPINIONS	75,877	405,739	Who-trusts-whom [35]
FLICKR	404,733	2,110,078	Photo sharing [21]
DELICIOUS	147,567	301,921	Collaborative tagging
CA-DBLP	317,080	1,049,866	Co-authorship (CA) [4]
CA-COND-MAT	21,363	91,286	CA cond-mat [25]
• Information networks			
CIT-HEP-TH	27,400	352,021	hep-th citations [13]
BLOG-POSTS	437,305	565,072	Blog post links [28]
• Web graphs			
WEB-GOOGLE	855,802	4,291,352	Web graph Google
WEB-WT10G	1,458,316	6,225,033	TREC WT10G web
• Bipartite affiliation (authors-to-papers) networks			
ATP-DBLP	615,678	944,456	DBLP [25]
ATP-ASTRO-PH	54,498	131,123	Arxiv `astro-ph` [25]
• Internet networks			
AS	6,474	12,572	Autonomous systems
GNUTELLA	62,561	147,878	P2P network [36]

Table 1: Some of the network datasets we studied.

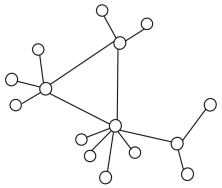
Micro-markets in sponsored search

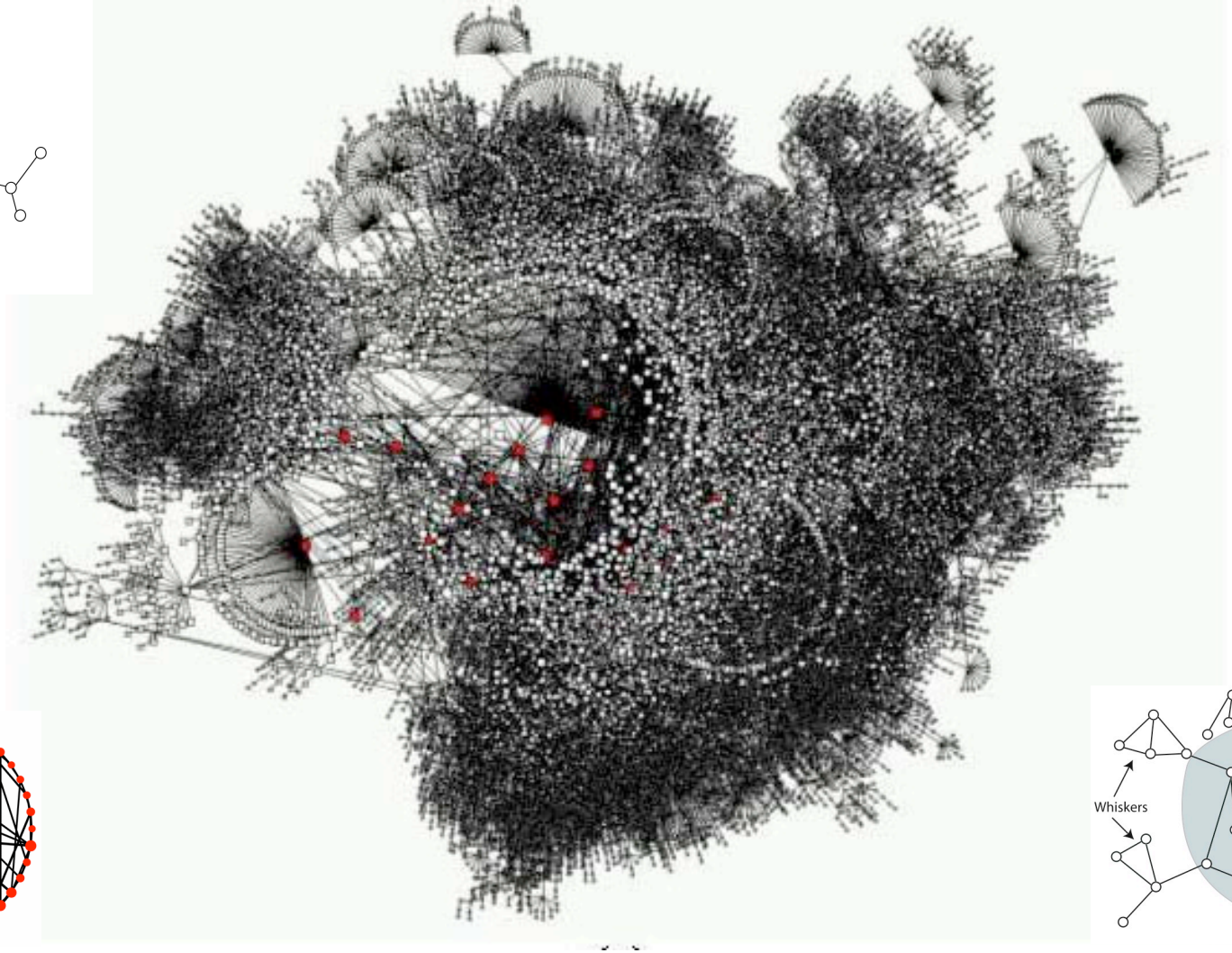
Goal: Find *isolated* markets/clusters with *sufficient money/clicks* with *sufficient coherence*.

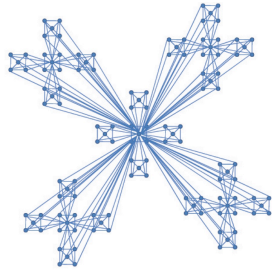
Ques: Is this even possible?

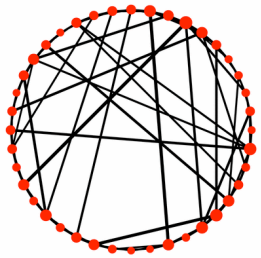


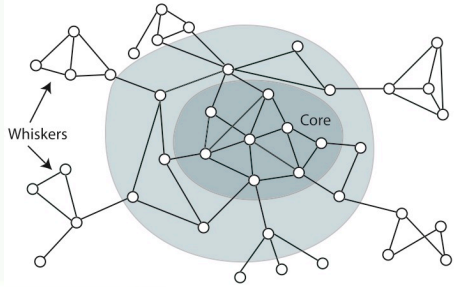
What do these networks "look" like?



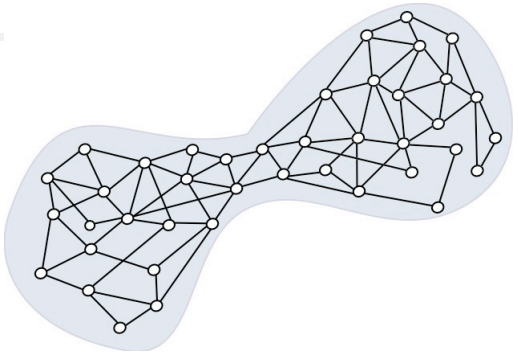








Communities, Conductance, and NCPPs



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:

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

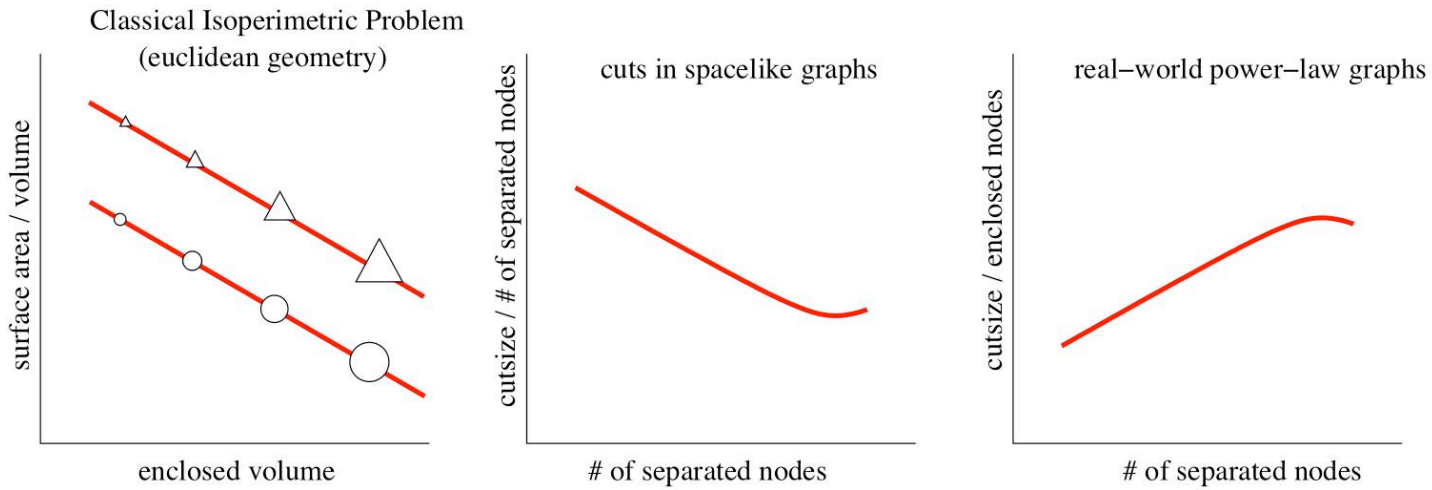
Just as conductance captures a Surface-Area-To-Volume notion

- *the NCP captures a Size-Resolved Surface-Area-To-Volume notion.*

Why worry about both criteria?

• Some graphs (e.g., "space-like" graphs, finite element meshes, road networks, random geometric graphs) cut quality and cut balance "work together"

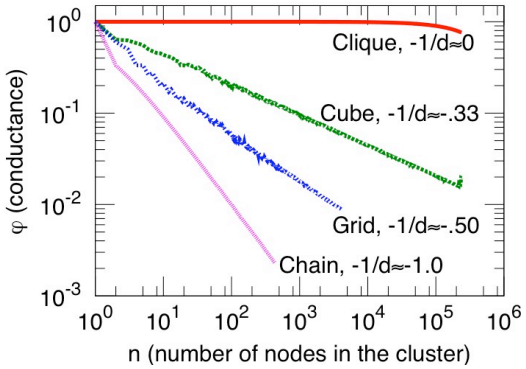
Tradeoff between cut quality and balance



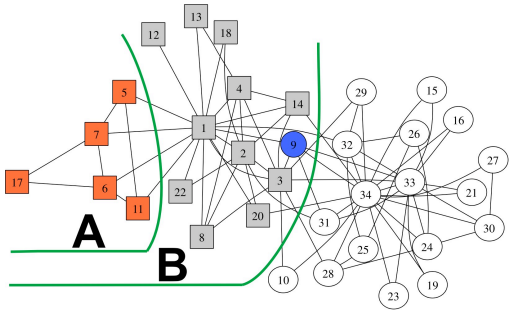
• For other classes of graphs (e.g., informatics graphs, as we will see) there is a "tradeoff," i.e., better cuts lead to worse balance

• For still other graphs (e.g., expanders) there are no good cuts of any size

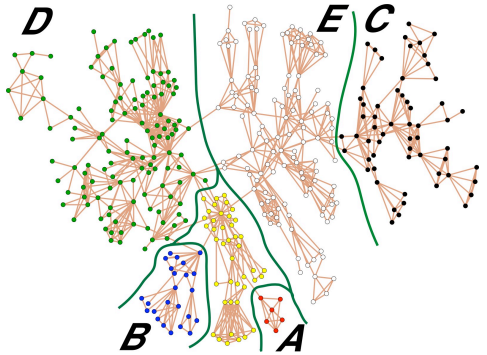
Widely-studied small social networks, "low-dimensional" graphs, and expanders



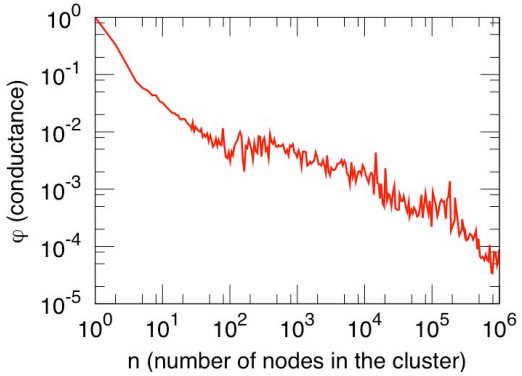
d-dimensional meshes



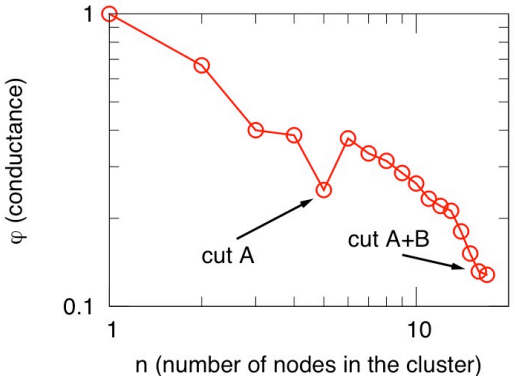
Zachary's karate club

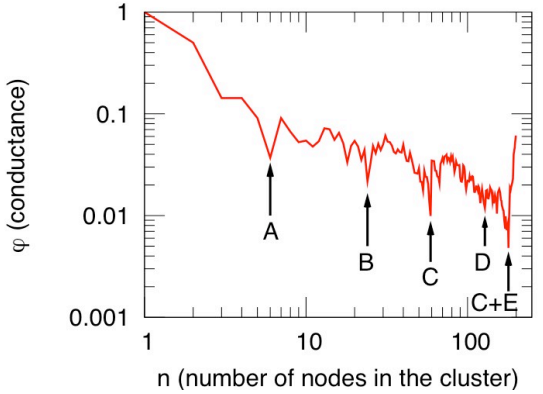


Newman's Network Science

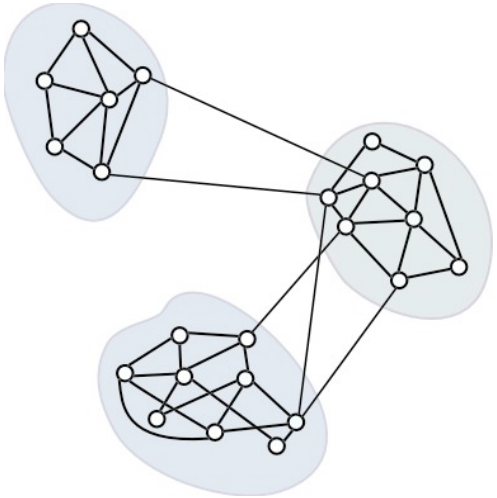


RoadNet-CA





What do large networks look like?



Downward sloping NCPP

small social networks (validation)

"low-dimensional" networks (intuition)

hierarchical networks (model building)

Natural interpretation in terms of isoperimetry

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

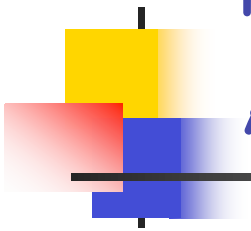
Large social/information networks are very *very* different

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)

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

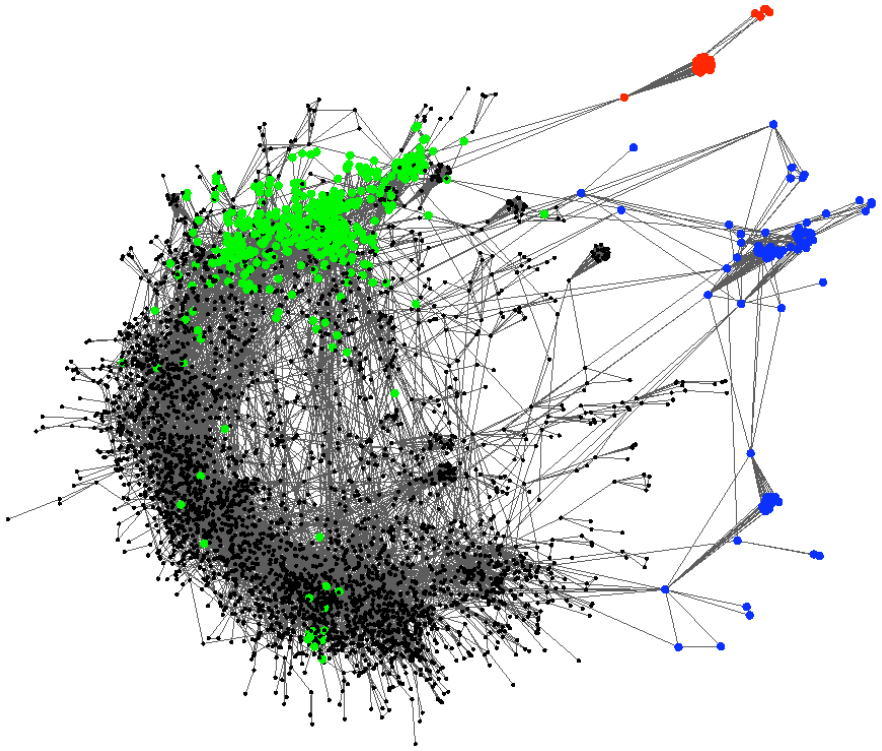
Metis+MQI - best conductance (empirically)

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

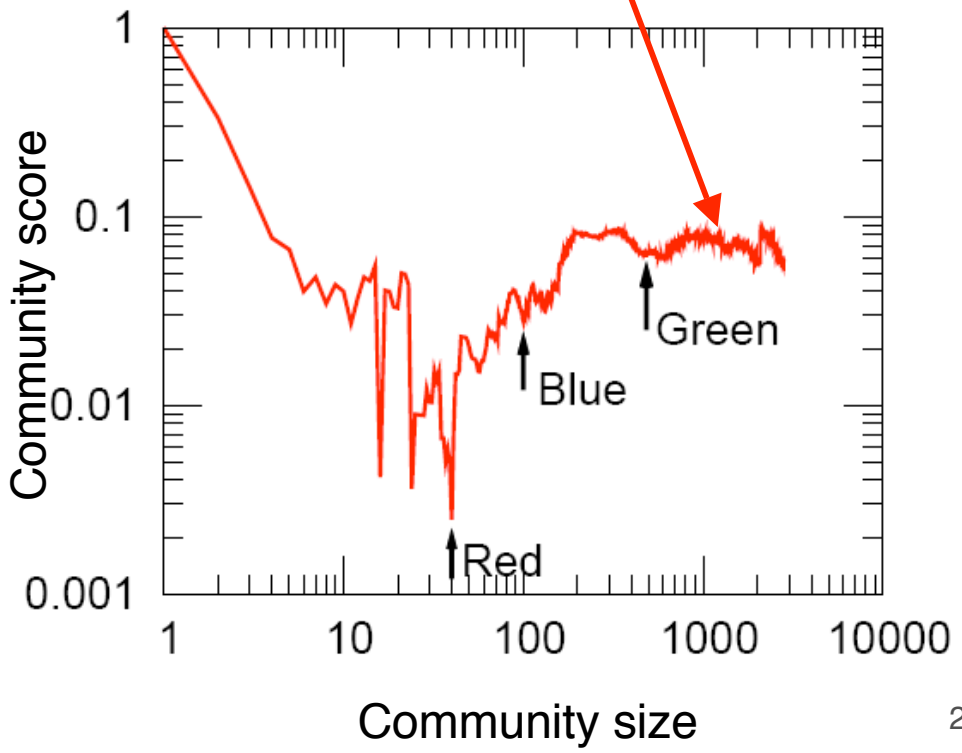
- *We exploit the "statistical" properties implicit in "worst case" algorithms.*

Typical example of our findings

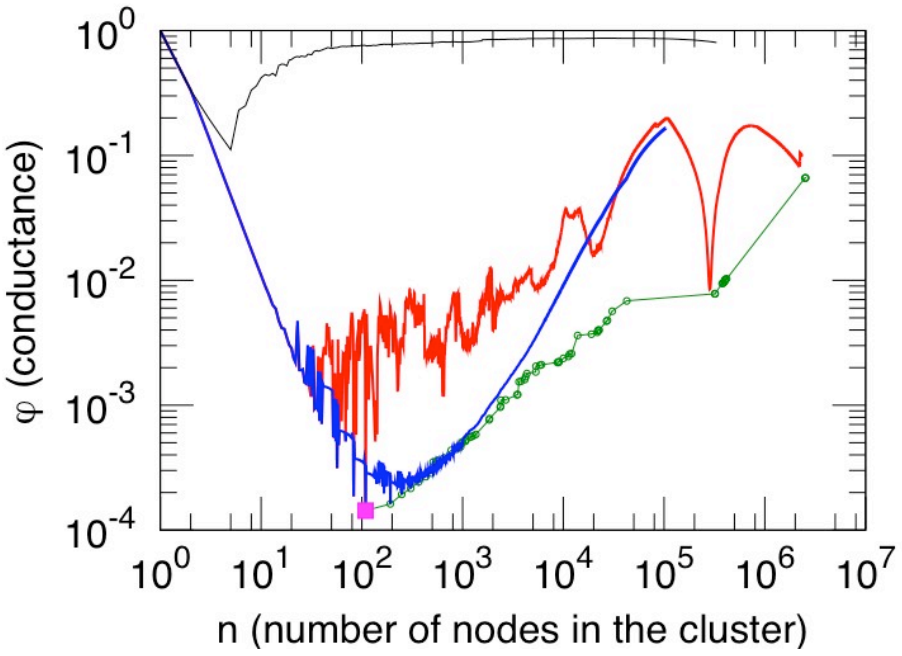
General relativity collaboration network
(4,158 nodes, 13,422 edges)



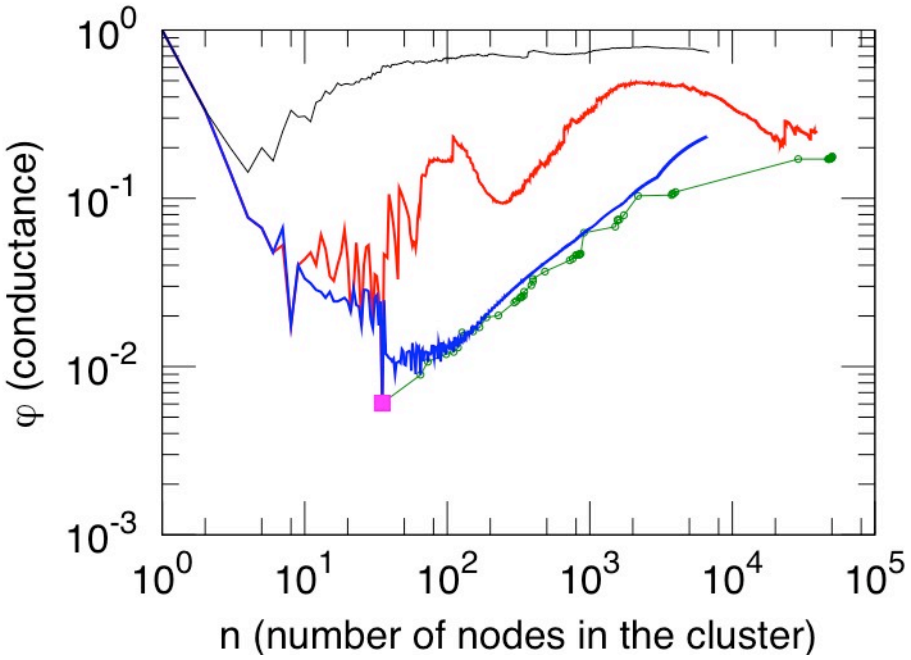
Data are expander-like
at large size scales !!!



Large Social and Information Networks



LiveJournal

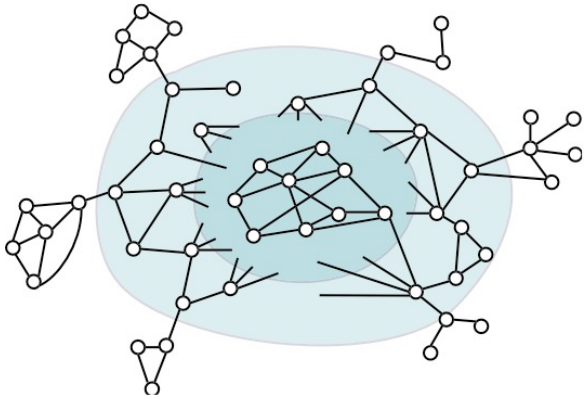


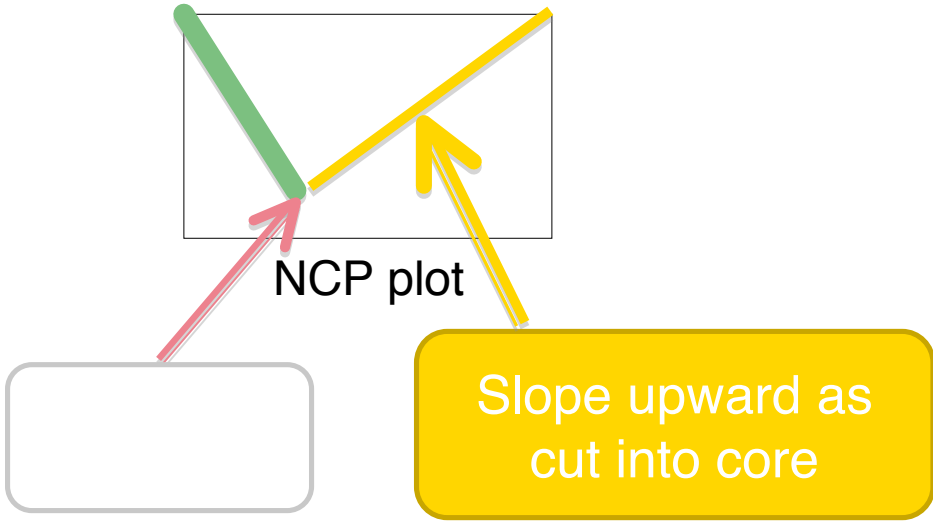
Epinions

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.

"Whiskers" and the "core"

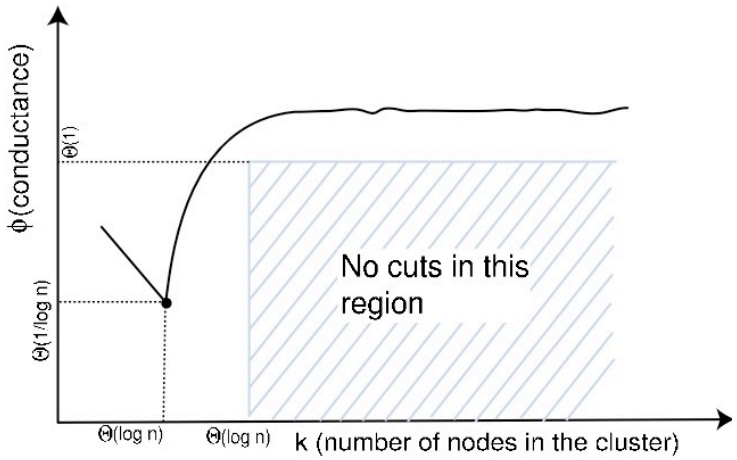
- "Whiskers"
  - maximal sub-graph detached from network by removing a single edge
  - contains 40% of nodes and 20% of edges
- "Core"
  - the rest of the graph, i.e., the 2-edge-connected core
- Global minimum of NCPP is a whisker
- *And, the core has a core-peripehery structure, recursively …*



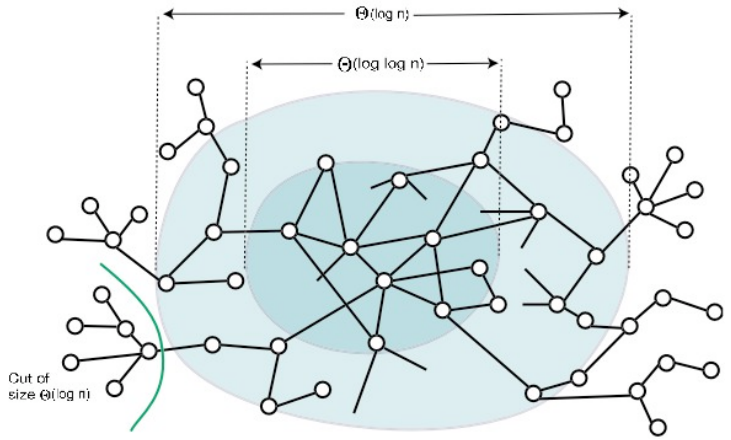


A simple theorem on random graphs

Let $\mathbf{w} = (w_1, \ldots, w_n)$, where
$w_i = ci^{-1/(\beta-1)}, \quad \beta \in (2,3)$.
Connect nodes i and j w.p.
$p_{ij} = w_i w_j / \sum_k w_k$.



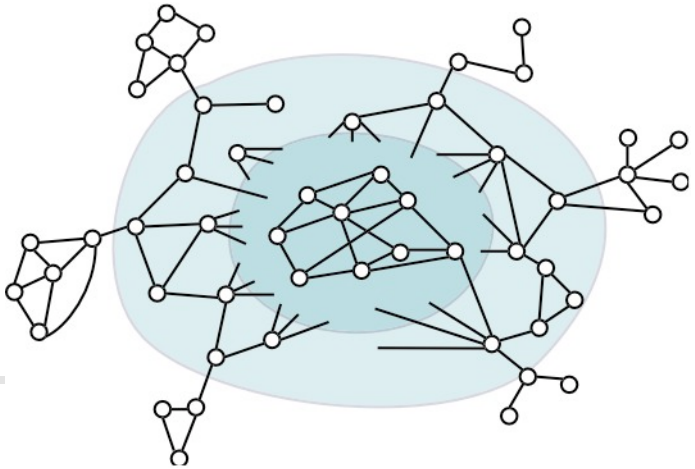
Power-law random graph with β ε (2,3).



Structure of the G(w) model, with β ε (2,3).

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

Implications: high level



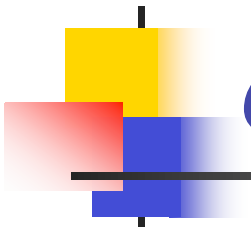
What is simplest explanation for empirical facts?

• *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

Think of the data as: local-structure on global-noise; not small noise on global structure!



Outline

A Bit of History of ML and LA

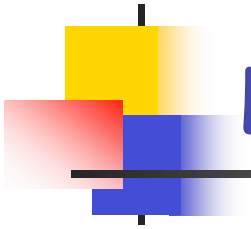
- Role of data, noise, randomization, and recently-popular algorithms

Large Informatics Graphs

- Characterize small-scale and large-scale clustering structure
- Provides novel perspectives on matrix and graph algorithms

New Machine Learning and New Linear Algebra

- Optimization view of "local" version of spectral partitioning
- Regularized optimization perspective on: PageRank, HeatKernel, and Truncated Iterated Random Walk
- Beyond VC bounds: Learning in high-variability environments



Lessons learned …

… on local and global clustering properties of messy data:

• Often good clusters "near" particular nodes, but no good meaningful global clusters.

… on approximate computation and implicit regularization:

• Approximation algorithms (Truncated Power Method, Approx PageRank, etc.) are very useful; but what do they actually compute?

… on learning and inference in high-variability data:

• Assumptions underlying common methods, e.g., VC dimension bounds, eigenvector delocalization, etc. often manifestly violated.

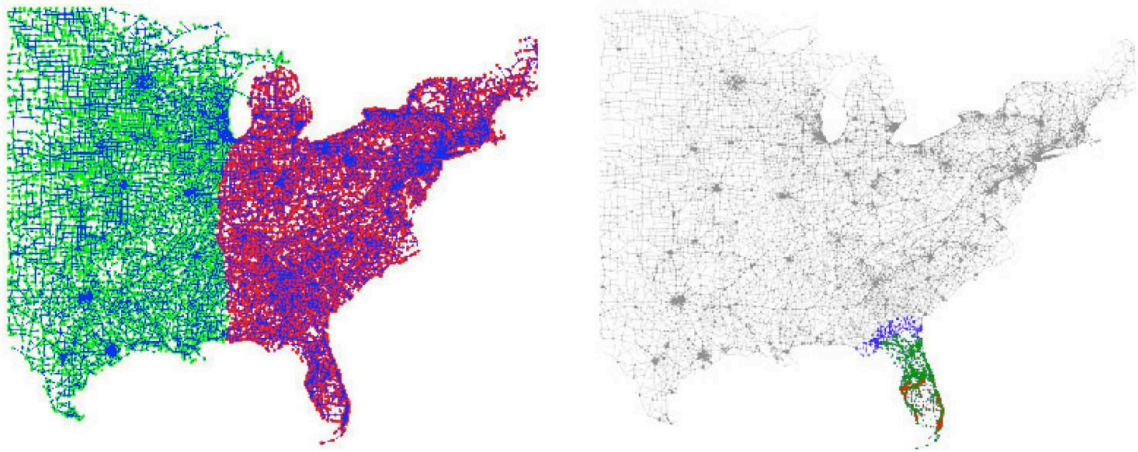
New ML and LA (1 of 3): Local spectral optimization methods

Local spectral methods - provably-good local version of global spectral

ST04: truncated "local" random walks to compute locally-biased cut

ACL06: approximate locally-biased PageRank vector computations

Chung08: approximate heat-kernel computation to get a vector



Q: Can we write these procedures as optimization programs?

Recall spectral graph partitioning

The basic optimization problem:

$$\begin{aligned} \text{minimize} \quad & x^T L_G x \\ \text{s.t.} \quad & \langle x, x \rangle_D = 1 \\ & \langle x, 1 \rangle_D = 0 \end{aligned}$$

- Relaxation of:

$$\phi(G) = \min_{S \subset V} \frac{E(S, \bar{S})}{Vol(S)Vol(\bar{S})}$$

- Solvable via the eigenvalue problem:

$$\mathcal{L}_G y = \lambda_2(G) y$$

- Sweep cut of second eigenvector yields:

$$\lambda_2(G)/2 \leq \phi(G) \leq \sqrt{8\lambda_2(G)}$$

Also recall Mihail's sweep cut for a general test vector:

Thm.[Mihail] Let x be such that $< x, 1 >_D = 0$. Then there is a cut along x that satisfies $\frac{x^T L_G x}{x^T D x} \geq \phi^2(S)/8$.

Geometric correlation and generalized PageRank vectors

Given a cut T, define the vector:

$$s_T := \sqrt{\frac{\mathrm{vol}(T)\mathrm{vol}(\bar{T})}{2m}} \left(\frac{1_T}{\mathrm{vol}(T)} - \frac{1_{\bar{T}}}{\mathrm{vol}(\bar{T})} \right)$$

Can use this to define a geometric notion of correlation between cuts:

$$< s_T, 1 >_D = 0$$

$$< s_T, s_T >_D = 1$$

$$< s_T, s_U >_D = K(T, U)$$

Defn. Given a graph $G = (V, E)$, a number $\alpha \in (-\infty, \lambda_2(G))$ and any vector $s \in R^n$, $s \perp_D 1$, a ***Generalized Personalized PageRank (GPPR)*** vector is any vector of the form

$$p_{\alpha,s} := (L_G - \alpha L_{K_n})^+ Ds.$$

- PageRank: a spectral ranking method (regularized version of second eigenvector of L_G)
- Personalized: s is nonuniform; & generalized: teleportation parameter α can be negative.

Local spectral partitioning *ansatz*

Mahoney, Orecchia, and Vishnoi (2010)

Primal program:

$$\begin{aligned} \text{minimize} \quad & x^T L_G x \\ \text{s.t.} \quad & < x, x >_D = 1 \\ & < x, s >_D^2 \geq \kappa \end{aligned}$$

Dual program:

$$\begin{aligned} \max \quad & \alpha - \beta(1-\kappa) \\ \text{s.t.} \quad & L_G \succeq \alpha L_{K_n} - \beta \left(\frac{L_{K_T}}{\text{vol}(\bar{T})} + \frac{L_{K_{\bar{T}}}}{\text{vol}(T)} \right) \\ & \beta \geq 0 \end{aligned}$$

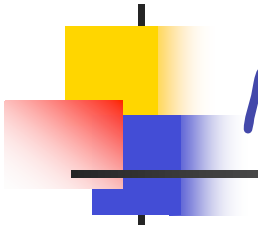
Interpretation:

- Find a cut well-correlated with the seed vector s.
- If s is a single node, this relax:

$$\min_{S \subset V, s \in S, |S| \leq 1/k} \frac{E(S, \bar{S})}{Vol(S) Vol(\bar{S})}$$

Interpretation:

- Embedding a combination of scaled complete graph K_n and complete graphs T and $\bar{T}$ (K_T and $K_{\bar{T}}$) - where the latter encourage cuts near $(T, \bar{T})$.



Main results (1 of 2)

Mahoney, Orecchia, and Vishnoi (2010)

Theorem: If x^* is an optimal solution to LocalSpectral, it is a GPPR vector for parameter α, and it can be computed as the solution to a set of linear equations.

Proof:

(1) Relax non-convex problem to convex SDP

(2) Strong duality holds for this SDP

(3) Solution to SDP is rank one (from comp. slack.)

(4) Rank one solution is GPPR vector.

Main results (2 of 2)

Mahoney, Orecchia, and Vishnoi (2010)

Theorem: If x^* is optimal solution to LocalSpect(G,s,κ), one can find a cut of conductance $\leq 8\lambda(G,s,\kappa)$ in time $O(n \lg n)$ with sweep cut of x^*.

Upper bound, as usual from sweep cut & Cheeger.

Theorem: Let s be seed vector and κ correlation parameter. For all sets of nodes T s.t. $\kappa' := \langle s, s_T \rangle_D^2$, we have: $\phi(T) \geq \lambda(G,s,\kappa)$ if $\kappa \leq \kappa'$, and $\phi(T) \geq (\kappa'/\kappa)\lambda(G,s,\kappa)$ if $\kappa' \leq \kappa$.

Lower bound: Spectral version of flow-improvement algs.

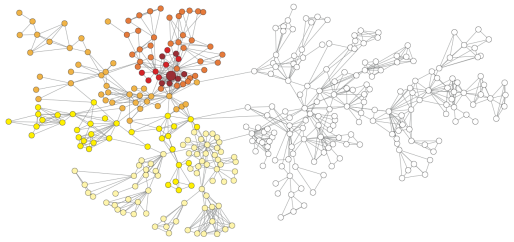
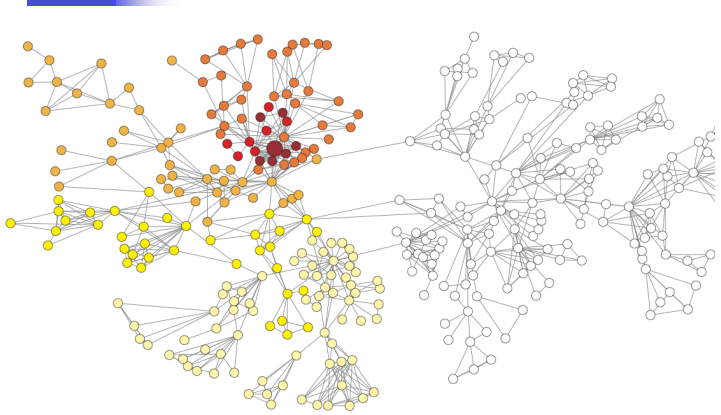
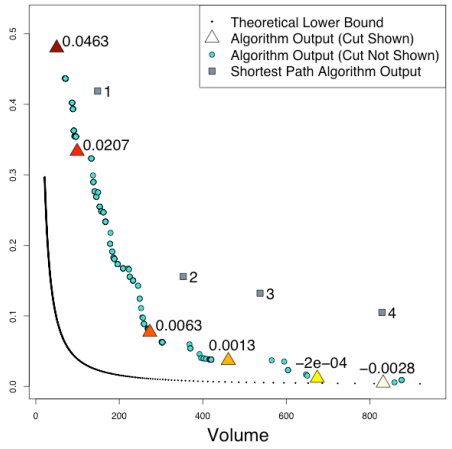
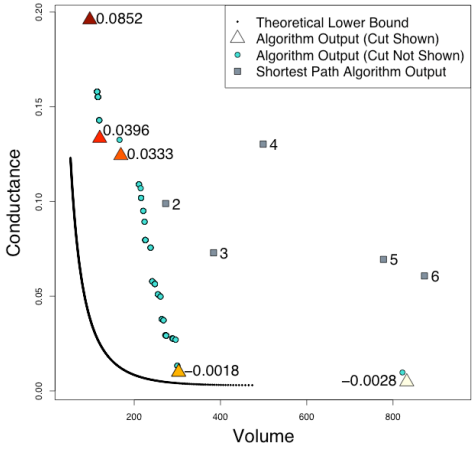


Illustration on small graphs



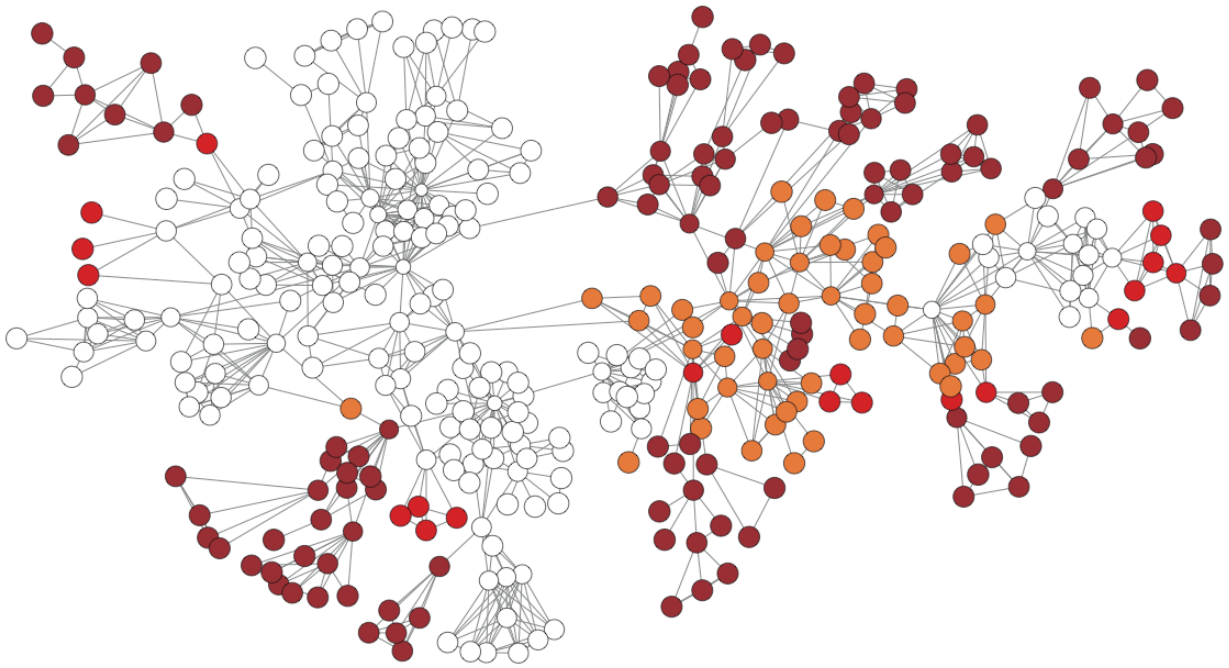


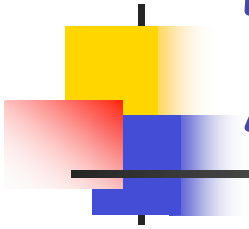


- Similar results if we do local random walks, truncated PageRank, and heat kernel diffusions.
- Often, it finds "worse" quality but "nicer" partitions than flow-improve methods. (Tradeoff we'll see later.)

Illustration with general seeds

- Seed vector doesn't need to correspond to cuts.
- It could be any vector on the nodes, e.g., can find a cut "near" low-degree vertices with $s_i = -(d_i - d_{av})$, $i \varepsilon [n]$.





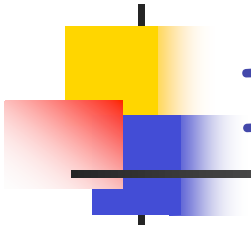
New ML and LA (2 of 3): Approximate eigenvector computation

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.

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

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?

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: $\pi(\gamma, s) = \gamma s + (1-\gamma) M \pi(\gamma, s)$

$$R_\gamma = \gamma \left(I - (1-\gamma) M\right)^{-1}$$

- q-step Lazy Random Walk: $W_\alpha^q = (\alpha I + (1-\alpha) M)^q$

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

Two versions of spectral partitioning

VP: $\longleftrightarrow$ **SDP:**

$$\begin{aligned} \text{min.} \quad & x^T L_G x \\ \text{s.t.} \quad & x^T L_{K_n} x = 1 \\ & < x, 1 >_D = 0 \end{aligned} \qquad \begin{aligned} \text{min.} \quad & L_G \circ X \\ \text{s.t.} \quad & L_{K_n} \circ X = 1 \\ & X \succeq 0 \end{aligned}$$

R-VP: **R-SDP:**

$$\begin{aligned} \text{min.} \quad & x^T L_G x + \lambda f(x) \\ \text{s.t.} \quad & constraints \end{aligned} \qquad \begin{aligned} \text{min.} \quad & L_G \circ X + \lambda F(X) \\ \text{s.t.} \quad & constraints \end{aligned}$$

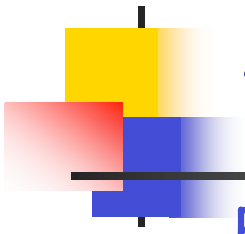
A simple theorem

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

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 $X^\star$ to be an optimal solution to (F,η)-SDP.

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



Three simple corollaries

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

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

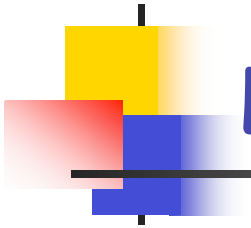
$F_D(X) = -\log\det(X)$ (i.e., Log-determinant)

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

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

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

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

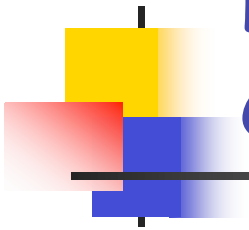


Large-scale applications

A lot of work on large-scale data already *implicitly* uses variants of 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, *for either matrix and/or graph approximation algorithms*?



New ML and LA (3 of 3):
Classification in high-variability environments

Supervised binary classification

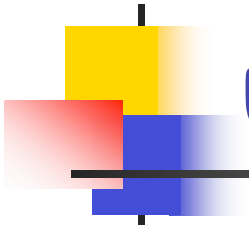
- Observe (X,Y) ε *(X,Y)* = (R^n , {-1,+1}) sampled from unknown distribution *P*
- Construct classifier α:*X->Y* (drawn from some family Λ, e.g., hyper-planes) after seeing *k* samples from unknown *P*

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

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

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

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



Unfortunately …

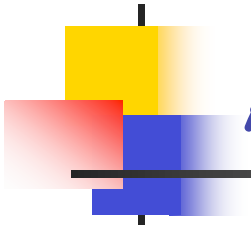
Sample complexity of dstbn-free learning typically depends on the ambient dimension to which the data to be classified belongs

- E.g., $\Omega(d)$ for learning half-spaces in 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 Λ of decision rules and a set of points $Z = \{z_1, \ldots, z_\ell\} \subset \mathcal{H}$, let $N^\Lambda(z_1, \ldots, z_\ell)$ be the number of ways of labeling $\{z_1, \ldots, z_\ell\}$ into positive and negative samples. Then,

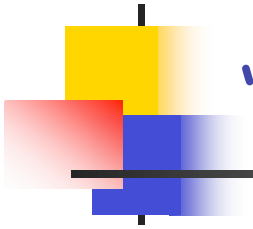
$$H^\Lambda_{ann}(k) := \ln E_{\mathcal{P}^{\times k}} N^\Lambda(z_1, \ldots, z_k)$$

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

Theorem: Given the above notation, the inequality

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

holds true, for any number of samples ℓ and for any error parameter ϵ.



"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_{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 …

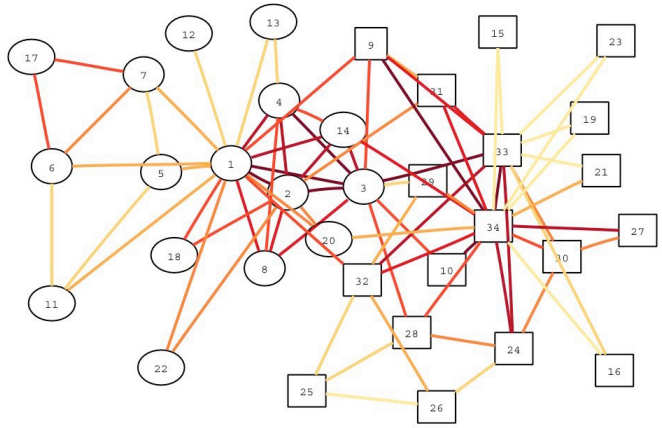
Let $\{f_i\}_{i=1}^n$ be the eigenfunctions of the normalized Laplacian of $\mathcal{L}_G$ and let $\{\lambda_i\}_{i=1}^n$ be the corresponding eigenvalues. Then, **Diffusion Maps** is:

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

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



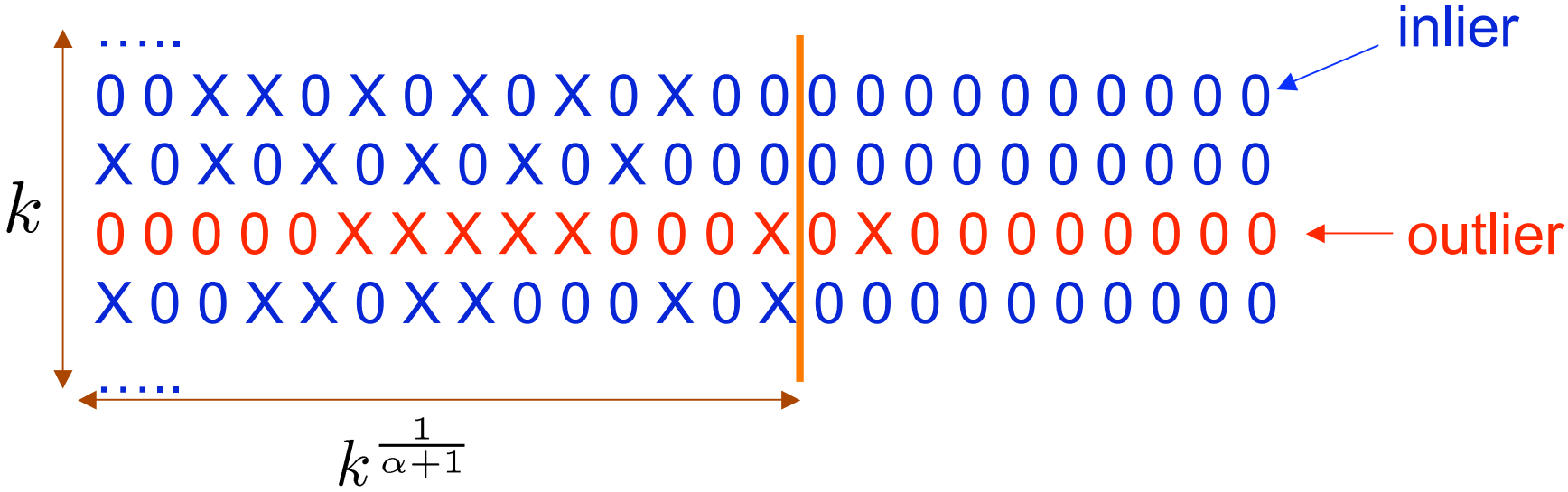
This is seen in many data sets when eigen-methods are chosen for algorithmic, and not statistical, reasons.

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 $\alpha > 1$.

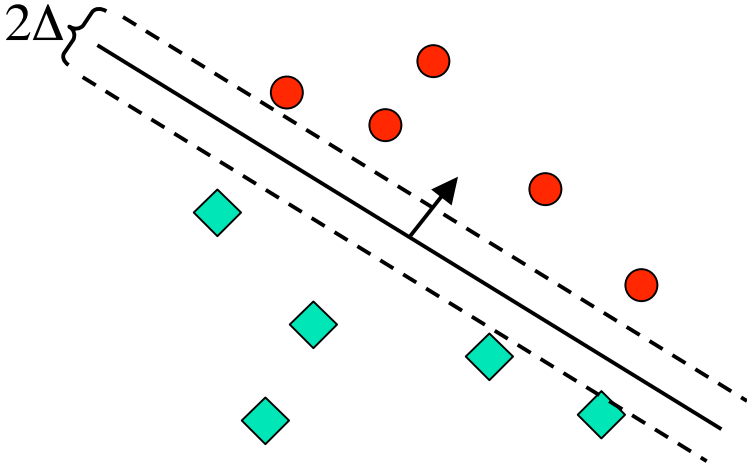
Theorem: In this model, $H^{\Lambda}_{ann}(\ell) \leq \left(\frac{C}{\alpha-1}\ell^{\frac{1}{\alpha}} + 1\right)\ln(\ell)$. Thus, need only $\ell = \tilde{O}\left(\left(\frac{C\ln(\delta^{-1})}{\epsilon^2}\right)^{\frac{\alpha+1}{\alpha}}\right)$ samples, independent of (possibly infinite) d.



Gap-tolerant classification

Mahoney and Narayanan (2009,2010)

Def: *A gap-tolerant classifier* consists of an oriented hyper-plane and a margin of thickness Δ around it. Points outside the margin are labeled ±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 then the annealed entropy of gap-tolerant classifiers in $\mathcal{H}$, where the gap is Δ, is

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

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

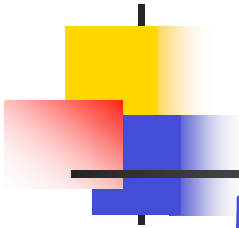
• with **heavy-tailed data**, e.g., when the magnitude of the elements of the feature vector decay in a heavy-tailed manner

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?*



Conclusions

Large informatics graphs

• Important in theory -- starkly illustrate that many common assumptions are inappropriate, so a good "hydrogen atom" for method development -- as well as important in practice

Local pockets of structure on global noise

• Implication for clustering and community detection, & implications for the use of common ML and DA tools

Several examples of new directions for ML and DA

• Principled algorithmic tools for local versus global exploration

• Approximate computation and implicit regularization

• Learning in high-variability environments