Geometric Tools for Identifying Structure in Large Social and Information Networks

Michael W. Mahoney
Stanford University

ICASSP - May 2011

(For more info, see: http://cs.stanford.edu/people/mmahoney/ or Google on "Michael Mahoney")
Lots of “networked data” out there!

• Technological and communication networks
  - AS, power-grid, road networks
• Biological and genetic networks
  - food-web, protein networks
• Social and information networks
  - collaboration networks, friendships; co-citation, blog cross-postings, advertiser-bidded phrase graphs ...
• Financial and economic networks
  - encoding purchase information, financial transactions, etc.
• Language networks
  - semantic networks ...
• Data-derived “similarity networks”
  - recently popular in, e.g., “manifold” learning
• ...

Sponsored ("paid") Search
Text-based ads driven by user query
Sponsored Search Problems

Keyword-advertiser graph:
- provide new ads
- maximize CTR, RPS, advertiser ROI

Motivating cluster-related problems:
- **Marketplace depth broadening:**
  - find new advertisers for a particular query/submarket
- **Query recommender system:**
  - suggest to advertisers new queries that have high probability of clicks
- **Contextual query broadening:**
  - broaden the user’s query using other context information
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 is the CTR and advertiser ROI of sports gambling keywords?
How people think about networks

“Interaction graph” model of networks:

- **Nodes** represent “entities”
- **Edges** represent “interaction” between pairs of entities

Graphs are combinatorial, not obviously-geometric

- Strength: powerful framework for analyzing *algorithmic complexity*
- Drawback: geometry used for learning and *statistical inference*
How people think about networks

A schematic illustration ...

... of hierarchical clusters?

Some evidence for micro-markets in sponsored search?
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, ...
What do these networks “look” like?
Popular approaches to large network data

Heavy-tails and power laws (at large size-scales):

• extreme heterogeneity in local environments, e.g., as captured by degree distribution, and relatively unstructured otherwise

• basis for preferential attachment models, optimization-based models, power-law random graphs, etc.

Local clustering/structure (at small size-scales):

• local environments of nodes have structure, e.g., captures with clustering coefficient, that is meaningfully “geometric”

• basis for small world models that start with global “geometry” and add random edges to get small diameter and preserve local “geometry”
Popular approaches to data more generally

Use geometric data analysis tools:
- **Low-rank methods** - very popular and flexible
- **Manifold methods** - use other distances, e.g., diffusions or nearest neighbors, to find “curved” low-dimensional spaces

These geometric data analysis tools:
- View data as a point cloud in $\mathbb{R}^n$, i.e., *each of the m data points is a vector in $\mathbb{R}^n$*
- Based on SVD, a basic vector space structural result
- **Geometry gives a lot** -- scalability, robustness, capacity control, basis for inference, etc.
Can these approaches be combined?

These approaches are very different:
- network is a single data point---not a collection of feature vectors
drawn from a distribution, and not really a matrix
- can’t easily let m or n (number of data points or features) go to
infinity---so nearly every such theorem fails to apply

Can associate matrix with a graph and vice versa, but:
- often do more damage than good
- questions asked tend to be very different
- graphs are really combinatorial things*

*But graph geodesic distance is a metric, and metric embeddings give fast algorithms!
Modeling data as matrices and graphs

In computer science:
- data are typically **discrete**, e.g., graphs
- focus is on **fast algorithms** for the given data set

In statistics*:
- data are typically **continuous**, e.g., vectors
- focus is on **inferring something** about the world

*very broadly-defined!
What do the data “look like” (if you 
squint at them)?

A “hot dog”? (or pancake that embeds well in low dimensions)

A “tree”? (or tree-like hyperbolic structure)

A “point”? (or clique-like or expander-like structure)
Squint at the data graph ...

Say we want to find a “best fit” of the adjacency matrix to:

<table>
<thead>
<tr>
<th></th>
<th>α</th>
<th>β</th>
</tr>
</thead>
<tbody>
<tr>
<td>α</td>
<td></td>
<td>β</td>
</tr>
<tr>
<td>β</td>
<td>γ</td>
<td></td>
</tr>
</tbody>
</table>

What does the data “look like”? How big are $\alpha$, $\beta$, $\gamma$?

$\alpha \approx \gamma \gg \beta$

low-dimensional

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

core-periphery

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

expander or $K_n$

$\beta \gg \alpha \approx \gamma$

bipartite graph
What is an expander?
What is an expander?

**Def:** an *expander* is a “sparse” graph that does not have any “good” partitions into two or more pieces.

- E.g., a not-extremely-sparse random graph

Who cares?

- Expanders are metric spaces that are *least* like “low-dimensional” metric spaces
- Very important in TCS for algorithm design
- Large social and information are expanders ...
Overview

Popular algorithmic tools with a geometric flavor

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

Graph algorithms and their geometric underpinnings

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

Novel insights on structure in large informatics graphs

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

Popular algorithmic tools with a geometric flavor

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

Graph algorithms and their geometric underpinnings

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

Novel insights on structure in large informatics graphs

- Successes and failures of existing models; empirical results, including “experimental” methodologies for probing network structure, taking into account algorithmic and statistical issues; implications and future directions
The Singular Value Decomposition (SVD)

The formal definition:

Given any $m \times n$ matrix $A$, one can decompose it as:

$$
\begin{pmatrix}
A \\
m \times \rho
\end{pmatrix} = 
\begin{pmatrix}
U \\
\rho \times \rho
\end{pmatrix} \cdot 
\begin{pmatrix}
\Sigma \\
\rho \times \rho
\end{pmatrix} \cdot 
\begin{pmatrix}
V \\
\rho \times n
\end{pmatrix}^T
$$

$\rho$: rank of $A$

$U$ ($V$): orthogonal matrix containing the left (right) singular vectors of $A$.

$\Sigma$: diagonal matrix containing $\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_\rho$, the singular values of $A$.

SVD is the “the Rolls-Royce and the Swiss Army Knife of Numerical Linear Algebra.”

* Dianne O’Leary, MMDS 2006
Rank-\(k\) approximations (\(A_k\))

Truncate the SVD at the top-\(k\) terms:

\[
\begin{pmatrix}
A_k
\end{pmatrix}
= 
\begin{pmatrix}
U_k
\end{pmatrix}
\cdot
\begin{pmatrix}
\Sigma_k
\end{pmatrix}
\cdot
\begin{pmatrix}
V_k^T
\end{pmatrix}
\]

\(U_k\) (\(V_k\)): orthogonal matrix containing the top \(k\) left (right) singular vectors of \(A\).

\(\Sigma_k\): diagonal matrix containing the top \(k\) singular values of \(A\).

**Important:** Keeping top \(k\) singular vectors provides “best” rank-\(k\) approximation to \(A\) (w.r.t. Frobenius norm, spectral norm, etc.):

\[
A_k = \text{argmin}\{ ||A-X||_{2,F} : \text{rank}(X) \leq k \}.
\]
Blue circles are $m$ data points in a 2-D space.

The SVD of the $m$-by-$2$ matrix of the data will return ...

$V^{(1)}$: 1st (right) singular vector: direction of maximal variance, 

$\sigma_1$: how much of data variance is explained by the first singular vector.

$V^{(2)}$: 2nd (right) singular vector: direction of maximal variance, after removing projection of the data along first singular vector.

$\sigma_2$: measures how much of the data variance is explained by the second singular vector.
Latent Semantic Indexing (LSI)

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

$$ n \text{ terms (words)} $$

$$ m \text{ documents} $$

$$ A_{ij} = \text{frequency of } j\text{-th term in } i\text{-th document} $$

Pros

- Less storage for small $k$.
- Improved performance.
- Documents are represented in a "concept" space.

Cons

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

- Sometimes people interpret document corpus in terms of $k$ topics when use this.
- Better to think of this as just selecting one model from a parameterized class of models!
LSI/SVD and 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.
- Intuitive 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.
Interpreting the SVD - be very careful

Reification

• assigning a “physical reality” to large singular directions
• invalid in general

Just because “If the data are ‘nice’ then SVD is appropriate” does NOT imply converse.

Mahoney and Drineas (PNAS, 2009)
Interpretation: Centrality

Centrality (of a vertex) - measures relative importance of a vertices in a graph

- **degree centrality** - number of links incident upon a node
- **betweenness centrality** - high for vertices that occur on many shortest paths
- **closeness centrality** - mean geodesic distance between a vertex and other reachable nodes
- **eigenvector centrality** - connections to high-degree nodes are more important, and so on iteratively (a “spectral ranking” measure)

**Motivation and behavior on nice graphs is clear -- but what do they actually compute on non-nice graphs?**
Eigen-methods in ML and data analysis

Eigen-tools appear (explicitly or implicitly) in many data analysis and machine learning tools:

• Latent semantic indexing
• PCA and MDS
• Manifold-based ML methods
• Diffusion-based methods
• k-means clustering
• Spectral partitioning and spectral ranking
Overview

Popular algorithmic tools with a geometric flavor

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

Graph algorithms and their geometric underpinnings

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

Novel insights on structure in large informatics graphs

• Successes and failures of existing models; empirical results, including “experimental” methodologies for probing network structure, taking into account algorithmic and statistical issues; implications and future directions
Graph partitioning

A family of combinatorial optimization problems - want to partition a graph’s nodes into two sets s.t.:

• Not much edge weight across the cut (cut quality)
• Both sides contain a lot of nodes

Several standard formulations:

• Graph bisection (minimum cut with 50-50 balance)
• $\beta$-balanced bisection (minimum cut with 70-30 balance)
• $\text{cutsize}/\min\{|A|,|B|\}$, or $\text{cutsize}/(|A||B|)$ (expansion)
• $\text{cutsize}/\min\{\text{Vol}(A),\text{Vol}(B)\}$, or $\text{cutsize}/(\text{Vol}(A)\text{Vol}(B))$ (conductance or N-Cuts)

All of these formalizations of the bi-criterion are NP-hard!
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
Why graph partitioning?

**Graph partitioning algorithms:**
- capture a qualitative notion of connectedness
- well-studied problem in traditionally/recently both in theory and practice
- many machine learning and data analysis applications

**Don’t care about exact solution to intractable problem:**
- output of approximation algs is not something we “settle for”
- randomized/approximation algs often give “better” answers than exact solution
- nearly-linear/poly-time computation captures “qualitative existence”
The “lay of the land”

**Spectral methods** - compute eigenvectors of associated matrices

**Local improvement** - easily get trapped in local minima, but can be used to clean up other cuts

**Multi-resolution** - view (typically space-like graphs) at multiple size scales

**Flow-based methods** - single-commodity or multi-commodity version of max-flow-min-cut ideas

*comes with strong worst-case guarantees
Spectral Methods

Fiedler (1973) and Donath & Hoffman (1973)

- use eigenvectors of discrete graph Laplacian

Popular in scientific computing, parallel computing, etc. (1980s) and machine learning (2000s)

Algorithm:

1. Compute the exact/approximate eigenvector.

2. Perform “rounding”: choose the best of the n cuts defined by that eigenvector.
An “embedding” view of spectral

Use Rayleigh quotient to characterize $\lambda_1$:

$$\lambda_1 = \min_{x \perp D_1} \frac{\sum_{i \sim j} (x_i - x_j)^2}{\sum_i x_i^2 d_i}$$

But since $x \perp D_1$, this is equivalent to:

$$\frac{\lambda_1}{\text{vol}(G)} = \min_{x \perp D_1} \frac{\sum_{i \sim j} (x_i - x_j)^2}{\sum_{i,j} (x_i - x_j)^2 d_i d_j}$$

Interpretation:

- Minimize “mixing” subject to variance constraint
- Embed graph on a line and cut
- But duality not tight

Interpretation:

- Minimize “mixing” subject to “mixing” in complete graph $K_n$
- Embed graph in (scaled) $K_n$
- Duality tighter (can also see this in dual later)
Maximum flow problem

- Directed graph $G=(V,E)$.
- Source $s \in V$, sink $t \in V$.
- Capacity $c(e) \in \mathbb{Z}^+$ for each edge $e$.
- Flow: function $f: E \rightarrow \mathbb{N}$ s.t.
  - For all $e$: $f(e) \leq c(e)$
  - For all $v$, except $s$ and $t$: flow into $v =$ flow out of $v$
- Flow value: flow out of $s$
- Problem: find flow from $s$ to $t$ with maximum value

Important Variant: Multiple Sources and Multiple Sinks
An “embedding” view of flow

**Theorem:** (Bourgain)

Every n-point metric space embeds into $L_1$ with distortion $O(\log(n))$.

Flow-based algorithm to get sparsest cuts.

1. Solve LP to get distance $d: V \times V \rightarrow \mathbb{R}^+$.
2. Obtain $L_1$ embedding using Bourgain’s constructive theorem
3. Perform an appropriate “rounding.”

Thus, it boils down to an embedding and expanders are worst.
“Spectral” versus “flow”

Spectral:

• Compute an eigenvector
• “Quadratic” worst-case bounds
• Worst-case achieved -- on “long stringy” graphs
• Embeds you on a line (or complete graph)

Flow:

• Compute a LP
• $O(\log n)$ worst-case bounds
• Worst-case achieved -- on expanders
• Embeds you in $L1$

Two methods -- complementary strengths and weaknesses

• What we compute will be determined at least as much by as the approximation algorithm we use as by objective function.
Extensions of the basic ideas

**Cut improvement algorithms**

- Given an input cut, find a good one nearby or certify that none exists

**Local algorithms and locally-biased objectives**

- Run in a time depending on the size of the output and/or are biased toward input seed set of nodes

**Combining spectral and flow**

- to take advantage of their complementary strengths
Interplay between preexisting versus generated versus implicit geometry

**Preexisting geometry**
- Start with geometry and add “stuff”

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

**Implicitly-imposed geometry**
- Approximation algorithms implicitly embed the data in a metric/geometric place and then round.
What is the shape of a graph?

Can we generalize the following intuition to general graphs:

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

The basic idea:

• If a graph embeds well in another metric space, then it “looks like” that metric space**!

Hyperbolic Spaces

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

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

- Sphere
- Hyperbolic space
- Euclidean plane

The 3 maximally symmetric geometries
Comparison between different curvatures

<table>
<thead>
<tr>
<th>Property</th>
<th>Euclid.</th>
<th>Spherical</th>
<th>Hyperbolic</th>
</tr>
</thead>
<tbody>
<tr>
<td>Curvature</td>
<td>0</td>
<td>1</td>
<td>−1</td>
</tr>
<tr>
<td>Parallel lines</td>
<td>1</td>
<td>0</td>
<td>∞</td>
</tr>
<tr>
<td>Triangles are</td>
<td>normal</td>
<td>thick</td>
<td>thin</td>
</tr>
<tr>
<td>Shape of triangles</td>
<td><img src="triangle.png" alt="Triangle" /></td>
<td><img src="thick_triangle.png" alt="Thick Triangle" /></td>
<td><img src="thin_triangle.png" alt="Thin Triangle" /></td>
</tr>
<tr>
<td>Sum of angles</td>
<td>π</td>
<td>&gt; π</td>
<td>&lt; π</td>
</tr>
<tr>
<td>Circle length</td>
<td>$2\pi R$</td>
<td>$2\pi \sin R$</td>
<td>$2\pi \sinh R$</td>
</tr>
<tr>
<td>Disc area</td>
<td>$2\pi R^2 / 2$</td>
<td>$2\pi (1 - \cos R)$</td>
<td>$2\pi (\cosh R - 1)$</td>
</tr>
</tbody>
</table>
δ-hyperbolic metric spaces

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

Things to note about δ-hyperbolicity:

• Graph property that is both local (by four points) and global (by the distance) in the graph

• Polynomial time computable - naively in \( O(n^4) \) time

• Metric space embeds into a tree iff \( \delta = 0. \)

• Poincare half space in \( \mathbb{R}^k \) is \( \delta \)-hyperbolic with \( \delta = \log_2 3 \)

• Theory of \( \delta \)-hyperbolic spaces generalize theory of Riemannian manifold with negative sectional curvature to metric spaces
Expanders and hyperbolicity

Different concepts that really are different (Benjamini 1998):

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

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

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

Popular algorithmic tools with a geometric flavor

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

Graph algorithms and their geometric underpinnings

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

Novel insights on structure in large informatics graphs

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

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

For many real-world social-and-information “power-law graphs,” there is an inverse relationship between “cut quality” and “cut balance.”
Consequences of this empirical fact

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

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

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

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

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

• more complex statistics are too algorithmically complex or statistically rich
• fitting simple stats often doesn’t capture what you wanted

Beyond very simple statistics:

• Density, diameter, routing, clustering, communities, ...
• Popular models often fail egregiously at reproducing more subtle properties (even when fit to simple statistics)
Failings of “traditional” network approaches

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

- **Algorithmic decentralized search** - solving a (non-ML) problem: can we find short paths?
- **Diameter and density versus time** - simple dynamic property
- **Clustering and community structure** - subtle/complex static property (used in downstream analysis)

All three examples have to do with the coupling b/w “local” structure and “global” structure --- solution goes beyond simple statistics of traditional approaches.
Failing 1: Search in social graphs

**Milgram (1960s)**
- Small world experiments - study short paths in social networks
- Individuals from Midwest forward letter to people they know to get it to an individual in Boston.

**Watts and Strogatz (1998)**
- “Small world” model, i.e., add random edges to an underlying local geometry, reproduces local clustering and existence of short paths

- But, even Erdos-Renyi $G_{np}$ random graphs have short paths …
- … so the existence of short paths is not so interesting
- Milgram’s experiment also demonstrated people found those paths
Failing 2: Time evolving graphs

Albert and Barabasi (1999)

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

Leskovec, Kleinberg, and Faloutsos (2005)

• But, empirically, graphs densify over time (i.e., number of edges grows superlinearly with number of nodes) and diameter shrinks over time
Failing 3: Clustering and community structure

**Sociologists (1900s)**
- A “community” is any group of two or more people that is useful

**Girvan and Newman (2002,2004) and MANY others**
- A “community” is a set of nodes “joined together in tightly-knit groups between which there are only loose connections
- Modularity becomes a popular “edge counting” metric

**Leskovec, Lang, Dasgupta, and Mahoney (2008)**
- All work on community detection validated on networks with good well-balanced partitions (i.e., low-dimensional and not expanders)
- But, **empirically**, larger clusters/communities are less-and-less cluster-like than smaller clusters (i.e., networks are expander-like)
What do these networks “look” like?
Exptl Tools: Probing Large Networks with Approximation Algorithms

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

- **Spectral** - (quadratic approx) - confuses “long paths” with “deep cuts”
- **Multi-commodity flow** - (log(n) approx) - difficulty with expanders
- **SDP** - (sqrt(log(n)) approx) - best in theory
- **Metis** - (multi-resolution for mesh-like graphs) - common in practice
- **X+MQI** - post-processing step on, e.g., Spectral of Metis

**Metis+MQI** - best conductance (empirically)

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

*We are not interested in partitions per se, but in probing network structure.*
Analogy: What does a protein look like?

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

Experimental Procedure:

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

Let $A$ be the adjacency matrix of $G=(V,E)$. The conductance $\phi$ of a set $S$ of nodes is:

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

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!
Community Score: Conductance

- How community like is a set of nodes?
- Need a natural intuitive measure:
  - **Conductance** (normalized cut)
  - \( \phi(S) \approx \# \text{edges cut} / \# \text{edges inside} \)
  - Small \( \phi(S) \) corresponds to more community-like sets of nodes
Community Score: Conductance

What is “best” community of 5 nodes?

Score: $\phi(S) = \# \text{ edges cut} / \# \text{ edges inside}$
Community Score: Conductance

What is “best” community of 5 nodes?

Score: $\phi(S) = \frac{\# \text{ edges cut}}{\# \text{ edges inside}}$

$\phi = \frac{5}{6} = 0.83$
Community Score: Conductance

Better community
\[ \phi = \frac{2}{5} = 0.4 \]

What is “best” community of 5 nodes?

\[ \phi = \frac{5}{6} = 0.83 \]

Score: \[ \phi(S) = \frac{\text{# edges cut}}{\text{# edges inside}} \]
Community Score: Conductance

What is “best” community of 5 nodes?

Better community
$\phi = 2/5 = 0.4$

Best community
$\phi = 2/8 = 0.25$

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

Better community
$\phi = 5/6 = 0.83$
Widely-studied small social networks

Zachary's karate club

Newman's Network Science
“Low-dimensional” graphs (and expanders)
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)
## Large Social and Information Networks

<table>
<thead>
<tr>
<th>Social nets</th>
<th>Nodes</th>
<th>Edges</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>EPINIONS</td>
<td>75,877</td>
<td>405,739</td>
<td>Who-trusts-whom [35]</td>
</tr>
<tr>
<td>FLICKR</td>
<td>404,733</td>
<td>2,110,078</td>
<td>Photo sharing [21]</td>
</tr>
<tr>
<td>DELICIOUS</td>
<td>147,567</td>
<td>301,921</td>
<td>Collaborative tagging</td>
</tr>
<tr>
<td>CA-DBLP</td>
<td>317,080</td>
<td>1,049,866</td>
<td>Co-authorship (CA) [4]</td>
</tr>
<tr>
<td>CA-COND-MAT</td>
<td>21,363</td>
<td>91,286</td>
<td>CA cond-mat [25]</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Information networks</th>
</tr>
</thead>
<tbody>
<tr>
<td>CIT-HEP-TH</td>
</tr>
<tr>
<td>BLOG-POSTS</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Web graphs</th>
</tr>
</thead>
<tbody>
<tr>
<td>WEB-GOOGLE</td>
</tr>
<tr>
<td>WEB-WT10G</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Bipartite affiliation (authors-to-papers) networks</th>
</tr>
</thead>
<tbody>
<tr>
<td>ATP-DBLP</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Internet networks</th>
</tr>
</thead>
<tbody>
<tr>
<td>AS</td>
</tr>
<tr>
<td>GNUTELLA</td>
</tr>
</tbody>
</table>

**Table 1:** Some of the network datasets we studied.
Typical example of our findings

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

Leskovec, Lang, Dasgupta, and Mahoney (WWW 2008 & arXiv 2008)
Focus on the red curves (local spectral algorithm) - blue (Metis+Flow), green (Bag of whiskers), and black (randomly rewired network) for consistency and cross-validation.
More large networks

Cit-Hep-Th

Web-Google

AtP-DBLP

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

- Better and better communities
- Best communities get worse and worse

Graph showing community score vs. community size with a logarithmic scale.
How do we know this plot it “correct”?

• Algorithmic Result
  Ensemble of sets returned by different algorithms are very different
  Spectral vs. flow vs. bag-of-whiskers heuristic

• Statistical Result
  Spectral method implicitly regularizes, gets more meaningful communities

• Lower Bound Result
  Spectral and SDP lower bounds for large partitions

• Structural Result
  Small barely-connected “whiskers” responsible for minimum

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

![Graphs showing clustering methods comparison]
12 objective functions

- Clustering objectives:
  - Single-criterion:
    - Modularity: $m-E(m)$ \textit{(Volume minus correction)}
    - Modularity Ratio: $m-E(m)$
    - Volume: $\sum_u d(u) = 2m + c$
    - Edges cut: $c$
  - Multi-criterion:
    - Conductance: $c/(2m+c)$ \textit{(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: $\text{max frac. of edges of a node pointing outside } S$
    - Average-ODF: $\text{avg. frac. of edges of a node pointing outside}$
    - Flake-ODF: $\text{frac. of nodes with mode than } _\_ \text{ edges inside}$
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
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**
"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
- 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.)
Small versus Large Networks

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

E.g., fit these networks to Stochastic Kronecker Graph with “base” $K=\begin{bmatrix} a & b \\ b & c \end{bmatrix}$:

$$K_1 = \begin{bmatrix}
0.99 & 0.17 \\
0.17 & 0.82
\end{bmatrix} \begin{bmatrix}
0.99 & 0.55 \\
0.55 & 0.15
\end{bmatrix} \begin{bmatrix}
0.2 & 0.2 \\
0.2 & 0.2
\end{bmatrix}$$
Small and large networks are very different:
(Also, an expander)

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

$$K_1 = \begin{bmatrix} \alpha & \beta \\ \beta & \gamma \end{bmatrix}$$
Interpretation: A simple theorem on random graphs

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

Structure of the \( G(w) \) 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.)
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:
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:

  \[\text{sparsity + randomness = measure fails to concentrate}\]

• *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!*
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: 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

\[
\begin{bmatrix}
\alpha & \beta \\
\beta & \gamma \\
\end{bmatrix} =
\begin{bmatrix}
0.99 & 0.55 \\
0.55 & 0.15 \\
\end{bmatrix}
\]
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!

“*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)
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.
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:

\[
\begin{align*}
\text{minimize} & \quad x^T L_G x \\
\text{s.t.} & \quad \langle x, x \rangle_D = 1 \\
& \quad \langle x, 1 \rangle_D = 0
\end{align*}
\]

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

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

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

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

**Thm.** [Mihail] Let \( x \) be such that \( \langle x, 1 \rangle_D = 0 \). 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:

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

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

$$\langle s_T, 1 \rangle_D = 0$$
$$\langle s_T, s_T \rangle_D = 1$$
$$\langle s_T, s_U \rangle_D = K(T, U)$$

**Defn.** Given a graph $G = (V, E)$, a number $\alpha \in (-\infty, \lambda_2(G))$ and any vector $s \in \mathbb{R}^n$, $s \perp_D 1$, a Generalized Personalized PageRank ($\text{GPPR}$) 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 $\alpha$ can be negative.
Local spectral partitioning ansatz

Primal program:

\[
\begin{align*}
\text{minimize} \quad & x^T L_G x \\
\text{s.t.} \quad & \langle x, x \rangle_D = 1 \\
& \langle x, s \rangle_D \geq \kappa
\end{align*}
\]

Dual program:

\[
\begin{align*}
\text{max} \quad & \alpha - \beta (1 - \kappa) \\
\text{s.t.} \quad & L_G \geq \alpha L_{K_n} - \beta \left( \frac{L_{K_T}}{\text{vol}(T)} + \frac{L_{K_{\tilde{T}}}}{\text{vol}(\tilde{T})} \right) \\
& \beta \geq 0
\end{align*}
\]

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 \( \tilde{T} \) (\( K_T \) and \( K_{\tilde{T}} \)) - where the latter encourage cuts near \((T, \tilde{T})\).

Mahoney, Orecchia, and Vishnoi (2010)
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 $\alpha$, 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)

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

**Theorem:** Let $s$ be seed vector and $\kappa$ 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 $\kappa' \leq \kappa$.

*Upper bound, as usual from sweep cut & Cheeger.*

*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 $s_i = -(d_i - d_{av})$, $i \in [n]$. 
Many uses of Linear Algebra in ML and Data Analysis involve approximate computations


- 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.

\[ \hat{x} = \arg\min_x f(x) + \lambda g(x) \]

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:**
  \[
  \pi(\gamma, s) = \gamma s + (1 - \gamma) M \pi(\gamma, s) \\
  R_\gamma = \gamma (I - (1 - \gamma) M)^{-1}
  \]

• **q-step Lazy Random Walk:**
  \[ W^q_\alpha = (\alpha I + (1 - \alpha) M)^q \]

Ques: Do these “approximation procedures” exactly optimizing some regularized objective?
Two versions of spectral partitioning

**VP:**

\[
\begin{align*}
\text{min.} & \quad x^T L_G x \\
\text{s.t.} & \quad x^T L_{K_n} x = 1 \\
& \quad \langle x, 1 \rangle_D = 0
\end{align*}
\]

**R-VP:**

\[
\begin{align*}
\text{min.} & \quad x^T L_G x + \lambda f(x) \\
\text{s.t.} & \quad \text{constraints}
\end{align*}
\]

**SDP:**

\[
\begin{align*}
\text{min.} & \quad L_G \circ X \\
\text{s.t.} & \quad L_{K_n} \circ X = 1 \\
& \quad X \succeq 0
\end{align*}
\]

**R-SDP:**

\[
\begin{align*}
\text{min.} & \quad L_G \circ X + \lambda F(X) \\
\text{s.t.} & \quad \text{constraints}
\end{align*}
\]
A simple theorem

\[(F, \eta)\text{-SDP} \quad \min L \cdot X + \frac{1}{\eta} \cdot F(X)\]
\[
s.t. \quad I \cdot X = 1
X \succeq 0\]

**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 $(F, \eta)$-SDP.

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

$$F_H(X) = \text{Tr}(X \log X) - \text{Tr}(X)$$ (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 $t \sim \eta$

$$F_p(X) = \frac{1}{p} \|X\|_p^p$$ (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 $(X,Y) \in (X,Y) = (\mathbb{R}^n, \{-1,+1\})$ sampled from unknown distribution $P$
• Construct classifier $\alpha : X \to Y$ (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: $R_{emp}(\alpha) =$ risk on observed data

Ways to bound $| R(\alpha) - R_{emp}(\alpha) |$ 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 \( Z = \{z_1, \ldots, z_\ell\} \subset \mathcal{H} \), 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 \( \Lambda \) with respect to \( \mathcal{P} \).

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

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

holds true, for any number of samples \( \ell \) 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 $H_{ann}$ 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 $\mathbb{R}^d$. Suppose $\mathcal{P}[x_i \neq 0] \leq C i^{-\alpha}$ 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) $d$.

\begin{align*}
&\cdots \quad 0 \quad 0 \quad X \quad X \quad 0 \quad 0 \quad X \quad 0 \quad X \quad 0 \quad X \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \\
&X \quad 0 \quad X \quad 0 \quad X \quad 0 \quad X \quad 0 \quad X \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \\
&0 \quad 0 \quad 0 \quad 0 \quad 0 \quad X \quad X \quad X \quad X \quad X \quad 0 \quad 0 \quad X \quad 0 \quad X \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \\
&X \quad 0 \quad 0 \quad X \quad X \quad X \quad X \quad 0 \quad 0 \quad 0 \quad X \quad 0 \quad X \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \\
&\cdots \quad X \quad 0 \quad X \quad X \quad X \quad X \quad 0 \quad 0 \quad 0 \quad X \quad 0 \quad X \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0
\end{align*}

$k, \ k^\frac{1}{\alpha+1}$
**Gap-tolerant classification**

**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.”

**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

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

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

so can get dimension-independent bounds!
Large-margin classification with very “outlying“ data points

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 (1 of 2)

- **Geometric tools** for experimentally “probing” large social and information graphs: $\text{geometry} \approx \text{inference}$

- Tools for coupling local properties (often low-dimensional) and global properties (expander-like)

- **Real informatics graphs** -- very different than small commonly-studied graphs and existing generative models

- **New directions** for machine learning, sparse modeling, data analysis etc.
Conclusions (2 of 2)

• Validation is difficult - if you have a clean validation and/or a pretty picture, you’re looking at unrealistic network data!

• 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.