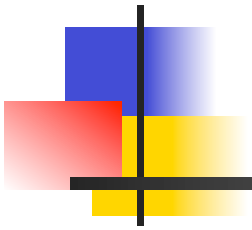


Approximate computation and implicit regularization in large-scale data analysis

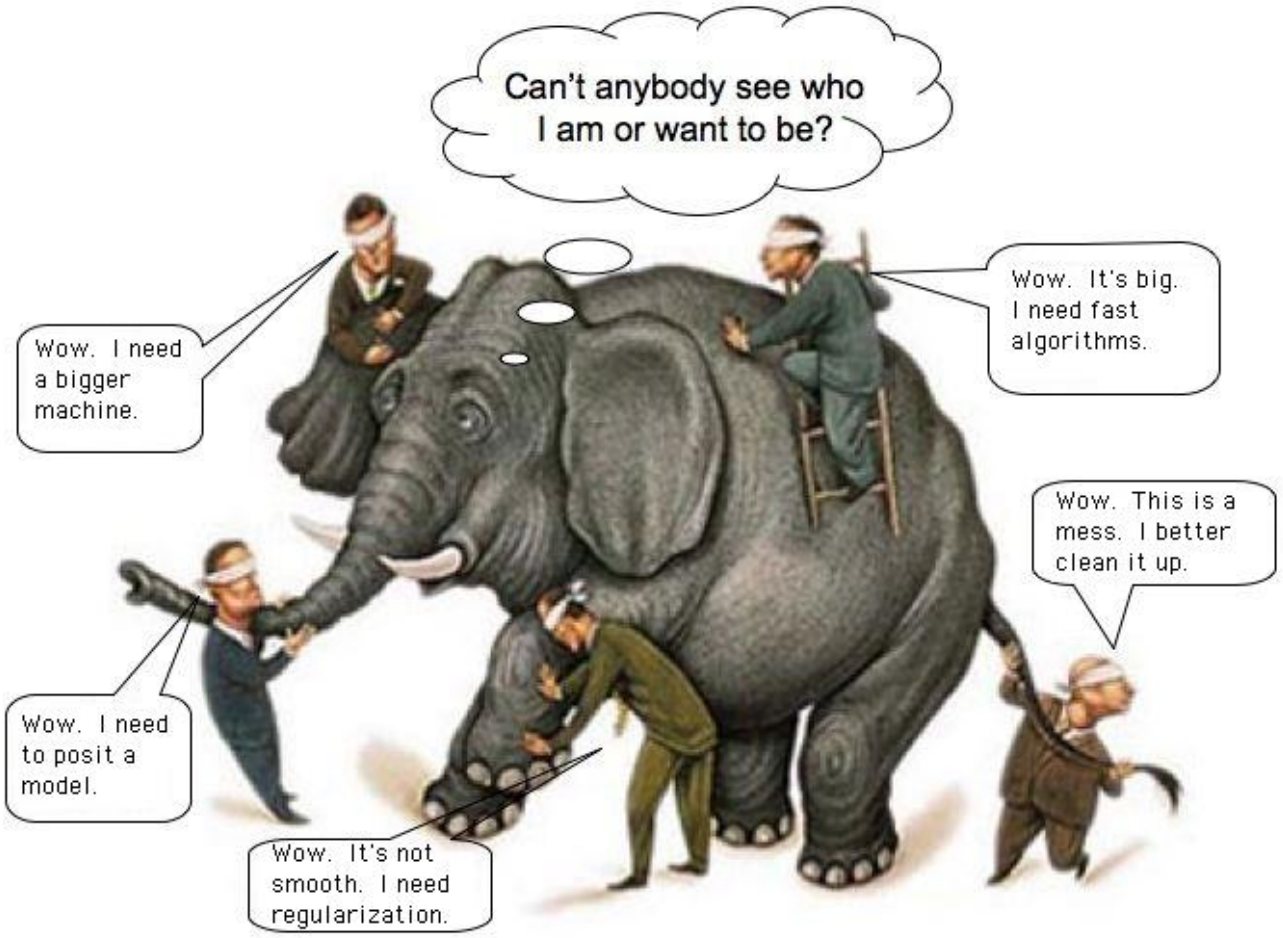
Michael W. Mahoney

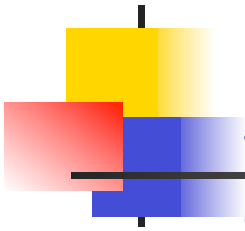
Stanford University

Jan 2013

(For more info, see: http://cs.stanford.edu/people/mmahoney)

How do we view BIG data?





Algorithmic & Statistical Perspectives ...

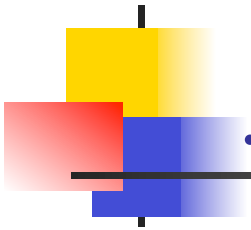
Lambert (2000)

Computer Scientists

• *Data*: are a record of everything that happened.
• *Goal*: process the data to find interesting patterns and associations.
• *Methodology*: Develop approximation algorithms under different models of data access since the goal is typically computationally hard.

Statisticians (and Natural Scientists, etc)

• *Data*: are a particular random instantiation of an underlying process describing unobserved patterns in the world.
• *Goal*: is to extract information about the world from noisy data.
• *Methodology*: Make inferences (perhaps about unseen events) by positing a model that describes the random variability of the data around the deterministic model.



... are VERY different **paradigms**

Statistics, natural sciences, scientific computing, etc:

- Problems often involve computation, but the study of computation *per se* is secondary
- Only makes sense to develop algorithms for well-posed* problems
- First, write down a model, and think about computation later

Computer science:

- Easier to study computation *per se* in discrete settings, e.g., Turing machines, logic, complexity classes
- Theory of algorithms divorces computation from data
- First, run a fast algorithm, and ask what it means later

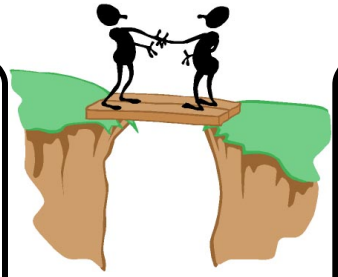
*Solution exists, is unique, and varies continuously with input data

Anecdote 1: Randomized Matrix Algorithms

Mahoney "Algorithmic and Statistical Perspectives on Large-Scale Data Analysis" (2010)
Mahoney "Randomized Algorithms for Matrices and Data" (2011)

Theoretical origins

- theoretical computer science, convex analysis, etc.
- Johnson-Lindenstrauss
- Additive-error algs
- Good worst-case analysis
- No statistical analysis
- No implementations



Practical applications

- NLA, ML, statistics, data analysis, genetics, etc
- Fast JL transform
- Relative-error algs
- Numerically-stable algs
- Good statistical properties
- Beats LAPACK & parallel-distributed implementations

How to "bridge the gap"?

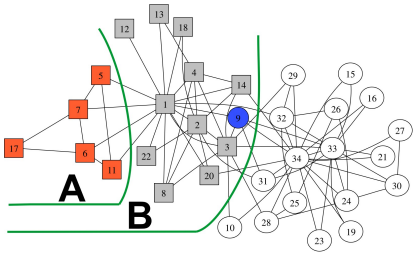
- decouple randomization from linear algebra
- importance of statistical leverage scores!

Anecdote 2: Communities in large informatics graphs

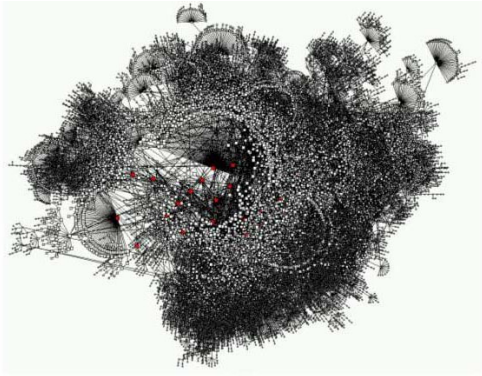
Mahoney "Algorithmic and Statistical Perspectives on Large-Scale Data Analysis" (2010)
Leskovec, Lang, Dasgupta, & Mahoney "Community Structure in Large Networks ..." (2009)

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

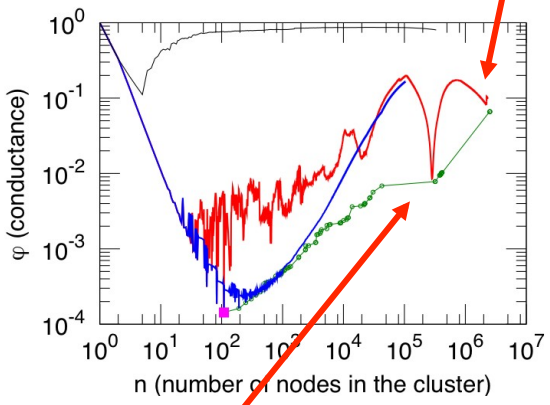
People imagine social networks to look like:



Real social networks actually look like:



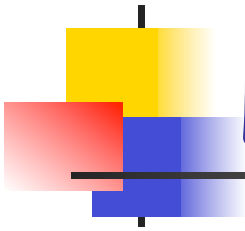
Size-resolved conductance (degree-weighted expansion) plot looks like:



There do not exist good large clusters in these graphs !!!

How do we know this plot is "correct"?

- (since computing conductance is intractable)
- Lower Bound Result; Structural Result; Modeling Result; Etc.
- Algorithmic Result (ensemble of sets returned by different approximation algorithms are very different)
- *Statistical Result* (Spectral provides more meaningful communities than flow)



Lessons from the anecdotes

We are being forced to engineer a union between two very different worldviews on what are fruitful ways to view the data

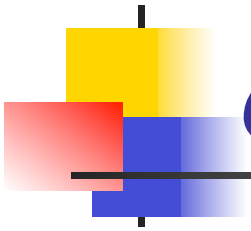
- in spite of our best efforts *not* to

Often fruitful to consider the statistical properties implicit in worst-case algorithms

- rather that *first* doing statistical modeling and *then* doing applying a computational procedure as a black box
- for both anecdotes, this was *essential* for leading to "useful theory"

How to extend these ideas to "bridge the gap" b/w the theory and practice of MMDS (Modern Massive Data Set) analysis.

- **QUESTION:** Can we identify a/the *concept at the heart of the algorithmic-statistical disconnect* and then drill-down on it?



Outline and overview

Preamble: algorithmic & statistical perspectives

General thoughts: data & algorithms, and explicit & implicit regularization

Approximate first nontrivial eigenvector of Laplacian

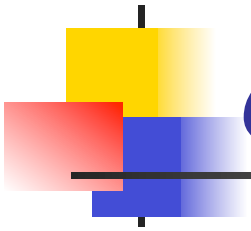
- Three diffusion-based procedures (heat kernel, PageRank, truncated lazy random walk) are *implicitly* solving a regularized optimization *exactly*!

A statistical interpretation of this result

- Analogous to Gaussian/Laplace interpretation of Ridge/Lasso regression

Spectral versus flow-based algs for graph partitioning

- Theory says each regularizes in different ways; empirical results agree!



Outline and overview

Preamble: algorithmic & statistical perspectives

General thoughts: data & algorithms, and explicit & implicit regularization

Approximate first nontrivial eigenvector of Laplacian

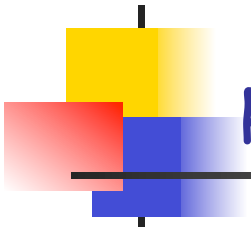
- Three diffusion-based procedures (heat kernel, PageRank, truncated lazy random walk) are *implicitly* solving a regularized optimization *exactly*!

A statistical interpretation of this result

- Analogous to Gaussian/Laplace interpretation of Ridge/Lasso regression

Spectral versus flow-based algs for graph partitioning

- Theory says each regularizes in different ways; empirical results agree!



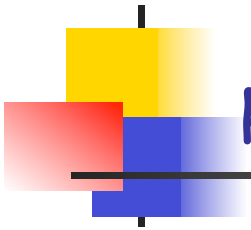
Relationship b/w algorithms and data (1 of 3)

Before the digital computer:

- Natural (and other) sciences rich source of problems, Statistics invented to solve those problems
- *Very* important notion: well-posed (well-conditioned) problem: solution exists, is unique, and is continuous w.r.t. problem parameters
- *Simply doesn't make sense to solve ill-posed problems*

Advent of the digital computer:

- Split in (yet-to-be-formed field of) "Computer Science"
- Based on application (scientific/numerical computing vs. business/ consumer applications) as well as tools (continuous math vs. discrete math)
- *Two very different perspectives on relationship b/w algorithms and data*



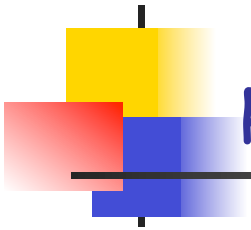
Relationship b/w algorithms and data (2 of 3)

Two-step approach for "numerical/statistical" problems

- Is problem well-posed/well-conditioned?
- If no, replace it with a well-posed problem. (Regularization!)
- If yes, design a stable algorithm.

View Algorithm A as a function f

- Given x, it tries to compute y but actually computes y*
- Forward error: $\Delta y = y^* - y$
- Backward error: smallest Δx s.t. $f(x+\Delta x) = y^*$
- Forward error ≤ Backward error * condition number
- *Backward-stable algorithm provides accurate solution to well-posed problem!*



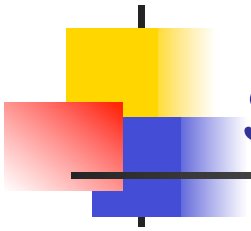
Relationship b/w algorithms and data (3 of 3)

One-step approach for study of computation, *per se*

- Concept of computability captured by 3 seemingly-different discrete processes (recursion theory, λ-calculus, Turing machine)
- Computable functions have internal structure (P vs. NP, NP-hardness, etc.)
- Problems of practical interest are "intractable" (e.g., NP-hard vs. poly(n), or $O(n^3)$ vs. $O(n \log n)$)

Modern Theory of Approximation Algorithms

- provides forward-error bounds for worst-cast input
- worst case in two senses: (1) for all possible input & (2) i.t.o. relatively-simple complexity measures, but independent of "structural parameters"
- get bounds by "relaxations" of IP to LP/SDP/etc., i.e., a "nicer" place



Statistical regularization (1 of 3)

Regularization in statistics, ML, and data analysis

- arose in integral equation theory to "solve" ill-posed problems
- computes a better or more "robust" solution, so better inference
- involves making (explicitly or implicitly) assumptions about data
- provides a trade-off between "solution quality" versus "solution niceness"
- often, heuristic approximation have regularization properties as a "side effect"
- lies at *the heart of the disconnect between the "algorithmic perspective" and the "statistical perspective"*

Statistical regularization (2 of 3)

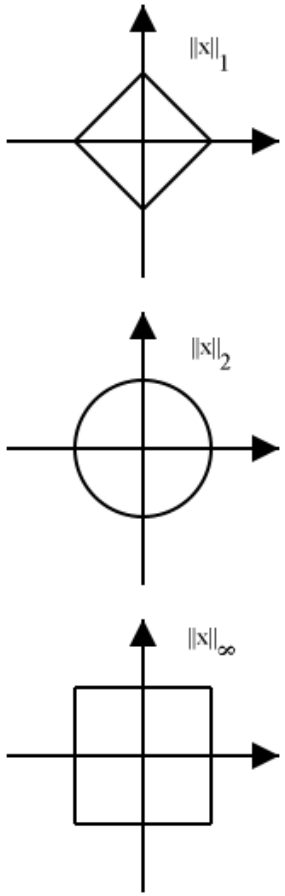
Usually *implemented* in 2 steps:

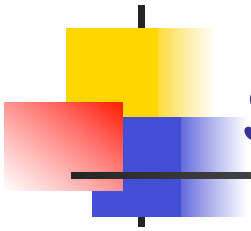
- add a norm constraint (or "geometric capacity control function") g(x) to objective function f(x)
- solve the modified optimization problem

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

Often, this is a "harder" problem, e.g., L1-regularized L2-regression

$$x' = \text{argmin}_x\ ||Ax-b||_2 + \lambda\, ||x||_1$$



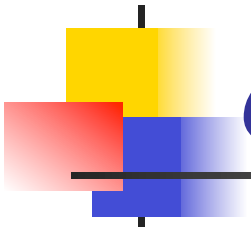


Statistical regularization (3 of 3)

Regularization is often observed as a side-effect or by-product of other design decisions

- "binning," "pruning," etc.
- "truncating" small entries to zero, "early stopping" of iterations
- approximation algorithms and heuristic approximations engineers do to implement algorithms in large-scale systems

Big question: Can we formalize the notion that/when approximate computation can *implicitly* lead to "better" or "more regular" solutions than exact computation?



Outline and overview

Preamble: algorithmic & statistical perspectives

General thoughts: data & algorithms, and explicit & implicit regularization

Approximate first nontrivial eigenvector of Laplacian

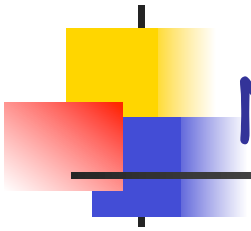
- Three diffusion-based procedures (heat kernel, PageRank, truncated lazy random walk) are *implicitly* solving a regularized optimization *exactly*!

A statistical interpretation of this result

- Analogous to Gaussian/Laplace interpretation of Ridge/Lasso regression

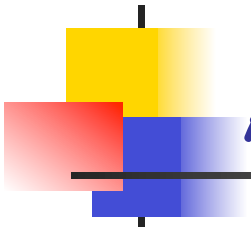
Spectral versus flow-based algs for graph partitioning

- Theory says each regularizes in different ways; empirical results agree!



Notation for weighted undirected graph

- vertex set $V = \{1, \ldots, n\}$
- edge set $E \subset V \times V$
- edge weight function $w : E \to R_+$
- degree function $d : V \to R_+$, $d(u) = \sum_v w(u, v)$
- diagonal degree matrix $D \in R^{V \times V}$, $D(v, v) = d(v)$
- combinatorial Laplacian $L_0 = D - W$
- normalized Laplacian $L = D^{-1/2} L_0 D^{-1/2}$



Approximating the top eigenvector

Basic idea: Given a Laplacian SPSD matrix A,

- Power method starts with "any" v_0, and iteratively computes

$$v_{t+1} = Av_t / ||Av_t||_2 \quad \text{->} \quad v_{\text{"1"}} \quad .$$

- Similarly for other "diffusion-based" methods
- If we truncate after (say) 3 or 10 iterations,
  - we still have some admixing from other eigen-directions
  - thus we *approximate* the exact solution!
  - do we *exactly solve* a (regularized) version of the problem?

What objective does the exact eigenvector optimize?

- Rayleigh quotient $R(A,x) = x^TAx / x^Tx$, for a *vector* x.

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:

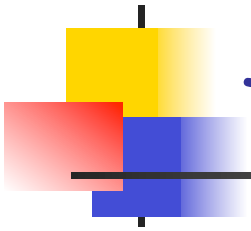
$$\pi(\gamma, s) = \gamma s + (1-\gamma) M \pi(\gamma, s)$$

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

- q-step Lazy Random Walk:

$$W_\alpha^q = (\alpha I + (1-\alpha) M)^q$$

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



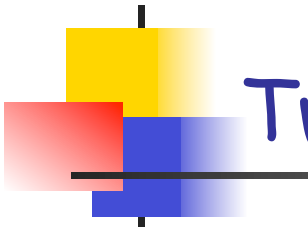
Two versions of spectral partitioning

VP:

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

R-VP:

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



Two versions of spectral partitioning

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

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

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

R-VP:

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

R-SDP:

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

A simple theorem

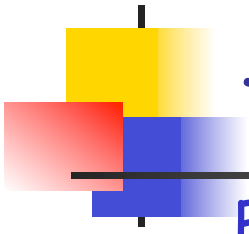
Mahoney and Orecchia (2010)

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

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)

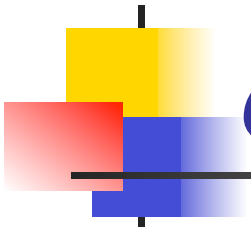
gives scaled PageRank matrix, with $t \sim \eta$

$F_p(X) = (1/p)||X||_p^p$ (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 *exactly*!

I.e., the *exactly* optimize min $L \bullet X + (1/\eta) F(X)$



Outline and overview

Preamble: algorithmic & statistical perspectives

General thoughts: data & algorithms, and explicit & implicit regularization

Approximate first nontrivial eigenvector of Laplacian

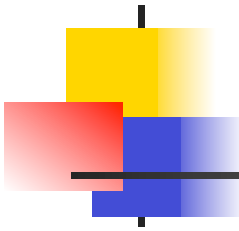
- Three diffusion-based procedures (heat kernel, PageRank, truncated lazy random walk) are *implicitly* solving a regularized optimization *exactly*!

A statistical interpretation of this result

- Analogous to Gaussian/Laplace interpretation of Ridge/Lasso regression

Spectral versus flow-based algs for graph partitioning

- Theory says each regularizes in different ways; empirical results agree!

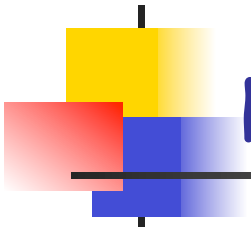


Statistical framework for regularized graph estimation

Perry and Mahoney (2011)

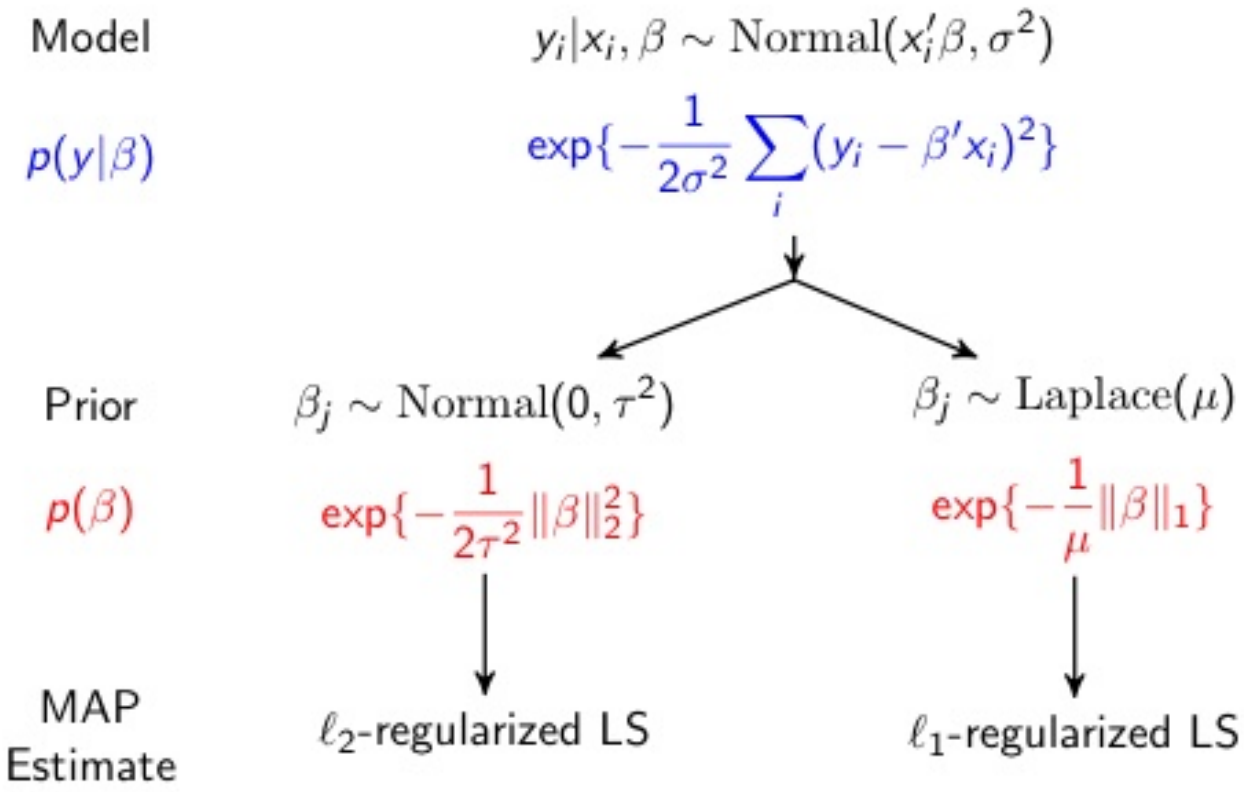
Question: What about a "statistical" interpretation of this phenomenon of *implicit regularization via approximate computation*?

- Issue 1: Best to think of the graph (e.g., Web graph) as a single data point, so what is the "ensemble"?
- Issue 2: No reason to think that "easy-to-state problems" and "easy-to-state algorithms" intersect.
- Issue 3: No reason to think that "priors" corresponding to what people actually do are particularly "nice."

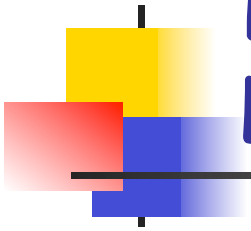


Recall regularized linear regression

- Observe n predictor-response pairs in $R^p \times R$: $(x_1, y_1), \ldots, (x_n, y_n)$
- Original problem: find β such that $\beta' x_i \approx y_i$; minimize $F(\beta) = \sum_i \|y_i - \beta' x_i\|_2^2$
- Regularized problem:
  minimize $F(\beta) + \lambda \|\beta\|_2^2$ (ridge) or
  minimize $F(\beta) + \lambda \|\beta\|_1$ (lasso)
- These can be interpreted in terms of a Gaussian prior or a Laplace prior, respectively, on the coefficient vector of the regression problem



Regularization is equivalent to "Bayesianization" in the following sense: the solution to the regularized problem is equal to the maximim *a posteriori* probability (MAP) estimate of the parameter with a prior determined by the regularization penalty.



Bayesian inference for the population Laplacian (broadly)

To apply the Bayesian formalism to the Laplacian eigenvector problem, we

- assume there exists a "population" Laplacian $\mathcal{L}$, from prior $p(\mathcal{L})$
- construe the observed/sample Laplacian as noisy version of $\mathcal{L}$, from distribution $p(L \mid \mathcal{L})$
- estimate $\hat{\mathcal{L}} = \text{argmax}_{\mathcal{L}}\{p(\mathcal{L} \mid L)\}$
- equivalently, $\hat{\mathcal{L}} = \text{argmin}_{\mathcal{L}}\{-\log p(L \mid \mathcal{L}) - \log p(\mathcal{L})\}$

In estimating $\mathcal{L}$,

- negative log of the likelihood plays the role of optimization criterion;
- negative log of prior distribution for $\mathcal{L}$ plays the role of penalty function.

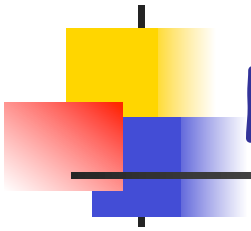
Bayesian inference for the population Laplacian (specifics)

- two parameters, m (scalar) and U (function)
- assume $\mathcal{L} \in \mathcal{X}$, where

$$\mathcal{X} = \{X : X \succeq 0,\ XD^{1/2}1 = 0,\ \mathrm{rank}(X) = n - 1\}$$

- prior $p(\mathcal{L}) \propto \exp\{-U(\mathcal{L})\}$
- model $L \sim \frac{1}{m}\mathrm{Wishart}(\mathcal{L}, m)$, i.e.

$$p(L \mid \mathcal{L}) \propto \frac{\exp\{-\frac{m}{2}\mathrm{Tr}(L\,\mathcal{L}^{+})\}}{|\mathcal{L}|^{m/2}}$$



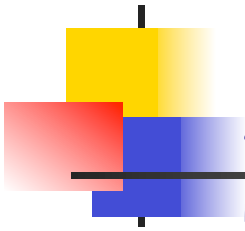
Heuristic justification for Wishart

1. $L_0 = \sum_{i=1}^{m} x_i x_i'$, where $x_i(u) = +1$, $x_i(v) = -1$, and (u, v) is the ith edge in graph.

2. Approximate distribution of x_i by $\tilde{x}_i \sim \text{Normal}(0, \mathcal{L}_0)$; first two moments of x_i and $\tilde{x}_i$ match.

3. $\sum_{i=1}^{m} \tilde{x}_i \tilde{x}_i'$ is distributd as $\text{Wishart}(\mathcal{L}_0, m)$.

4. Similar approximation holds for normalized Laplacian.



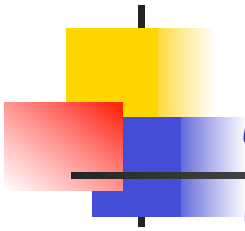
A prior related to PageRank procedure

Perry and Mahoney (2011)

Let $\mathcal{L}^+ = \tau O \Lambda O'$ be the spectral decomposition of $\mathcal{L}^+$, where $\tau = \text{Trace}(\mathcal{L}^+) \geq 0$ is a scale factor, $O \in R^{n \times n-1}$ is an orthogonal matrix, and $\Lambda = \text{diag}\big(\lambda(1), \ldots, \lambda(n-1)\big)$, where $\sum_v \lambda(v) = 1$. (Note λ is unordered.) The prior takes the form:

$$p(\mathcal{L}) \propto p(\tau) \prod_{v=1}^{n-1} \lambda(v)^{\alpha-1}$$

Note: $p(\tau)$ is unrestricted; and λ is Dirichlet distributed with shape parameter $(\alpha, \ldots, \alpha)$.



Main "Statistical" Result

Perry and Mahoney (2011)

Proposition If $\hat{\mathcal{L}}$ is the MAP estimate of $\mathcal{L}$, with $\hat{\tau} = \text{Trace}(\hat{\mathcal{L}}^{+})$ and $\hat{\Theta} = \hat{\tau}^{-1}\hat{\mathcal{L}}^{+}$, then $\hat{\Theta}$ solves the Mahoney-Orecchia regularized SDP with $G(X) = -\log|X|$ and η defined by

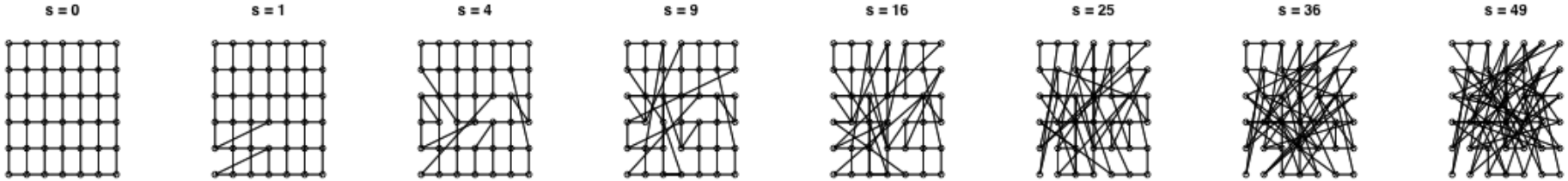
$$\eta = \frac{m\,\hat{\tau}}{m + 2\,(\alpha - 1)}.$$

That is, with this specific prior, the MAP estimate solves the regularized SDP related to the PageRank procedure.

Note: with different choices of priors, one can recover the Heat Kernel and Lazy Random Walk SDP solutions.

Empirical evaluation setup

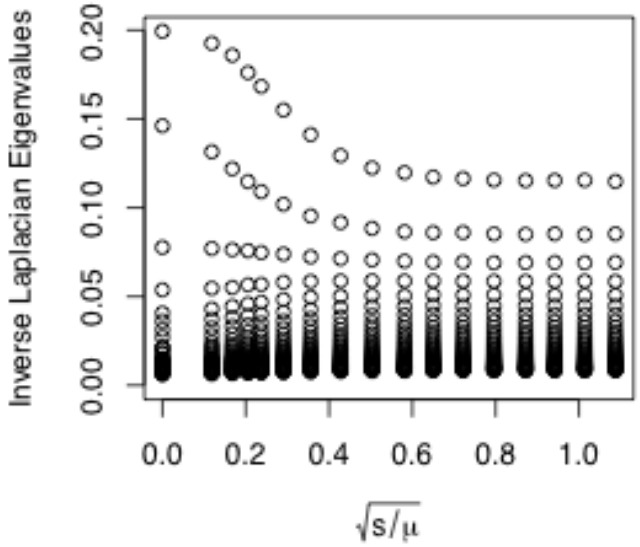
Generate a population Laplacian $\mathcal{L}$ by performing s edge swaps starting from a 2-dimensional grid with n nodes and μ edges.



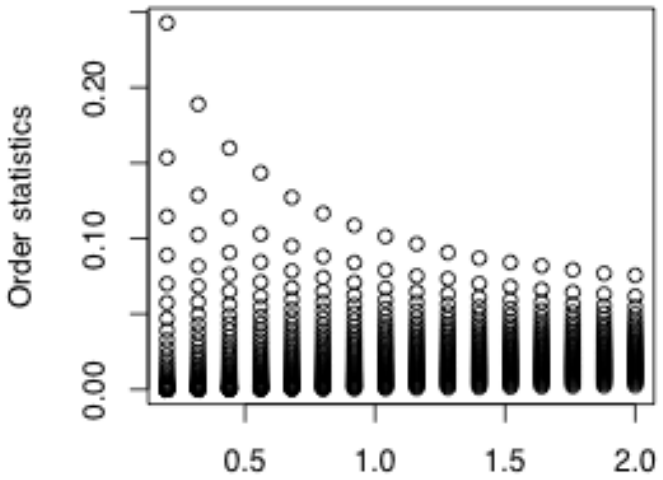
When $s = 0$ the population graph with Laplacian $\mathcal{L}$ is a low-dimensional grid; as $s \to \infty$, it becomes an expander-like random graph.

The prior vs. the simulation procedure

Perry and Mahoney (2011)

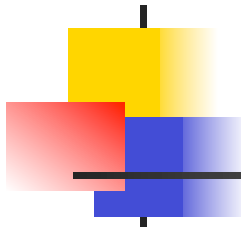


Eigenvalues of $\Theta = (\mathrm{Trace}(\mathcal{L}^+))^{-1}\mathcal{L}^+$



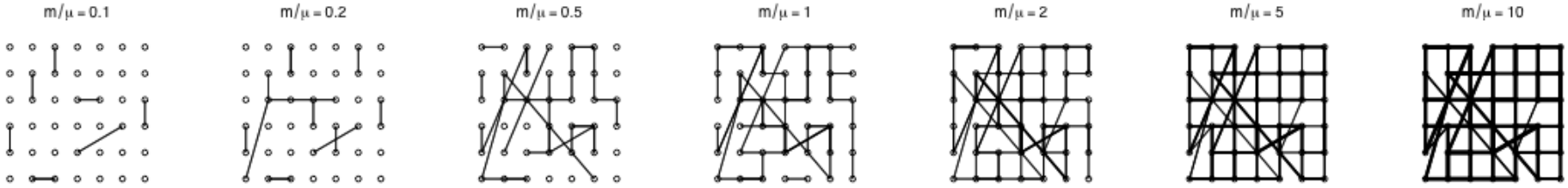
Draws from $\overset{\alpha}{\text{Dirichlet}}(\alpha)$

The similarity *suggests* that the prior qualitatively matches simulation procedure, with α parameter analogous to sqrt(s/μ).

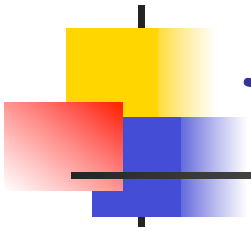


Generating a sample

Given a population graph with Laplacian $\mathcal{L}$, we generate a sample Laplacian L by sampling m edges. In the experiments, we get to observe L but not $\mathcal{L}$.



As m/μ increases, sample Laplacian L approaches the population Laplacian $\mathcal{L}$.



Two estimators for population Laplacian

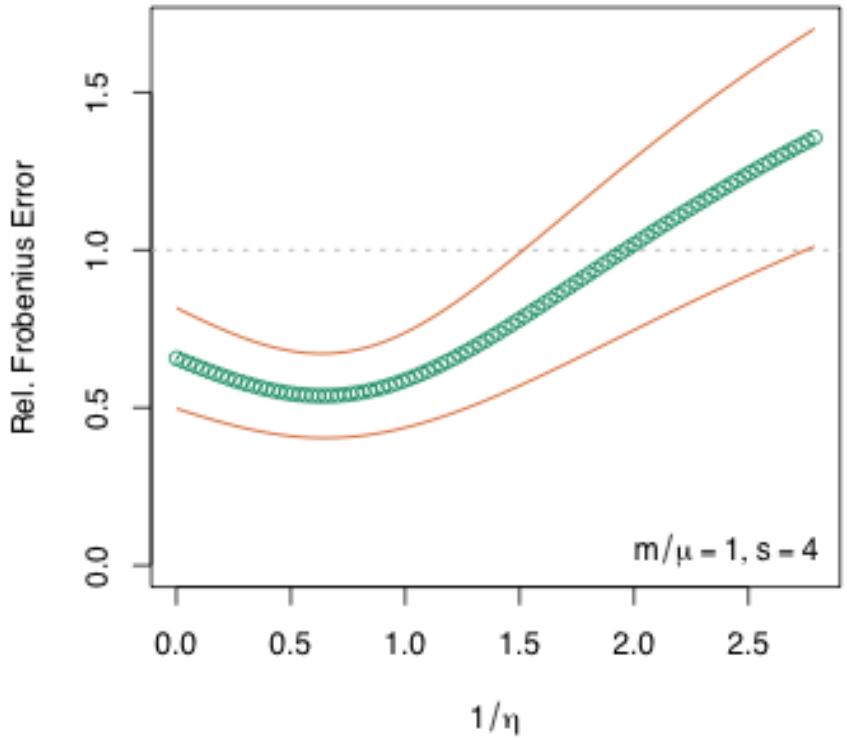
Two estimators for $\mathcal{L}$:

- **Unregularized:** $\hat{\mathcal{L}} = L$
- **Regularized:** $\hat{\mathcal{L}}_\eta$, the solution to the MO regularized SDP with $G(X) = -\log|X|$

Notation: $\tau = \text{Trace}(\mathcal{L}^+)$, $\Theta = \tau^{-1}\mathcal{L}^+$; $\hat{\tau} = \text{Trace}(\hat{\mathcal{L}}^+)$, $\hat{\Theta} = \hat{\tau}^{-1}\hat{\mathcal{L}}^+$; $\hat{\tau}_\eta = \text{Trace}(\hat{\mathcal{L}}_\eta^+)$, $\hat{\Theta}_\eta = \hat{\tau}_\eta^{-1}\hat{\mathcal{L}}_\eta^+$; $\bar{\tau}$ is mean of τ over all replicates.

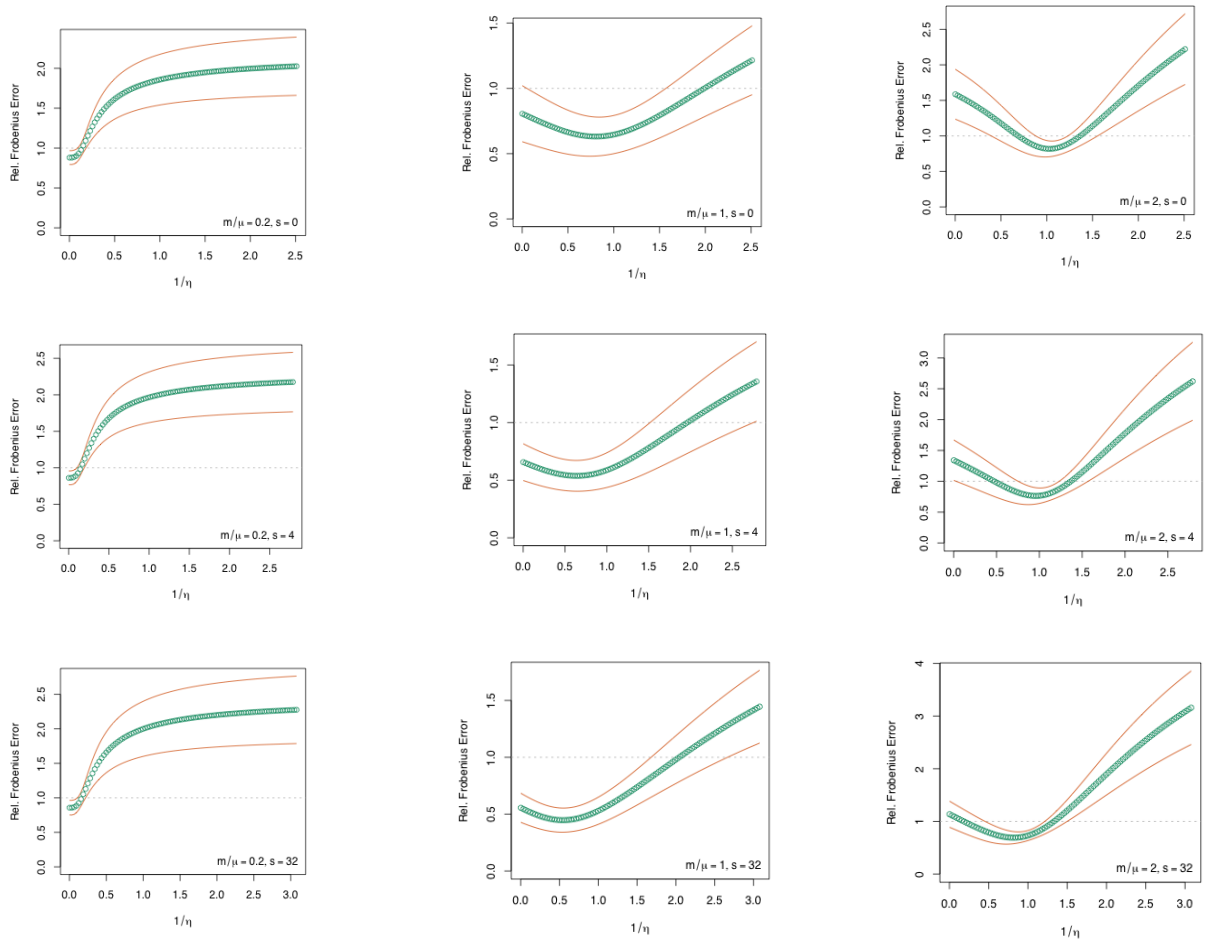
Empirical results (1 of 3)

Perry and Mahoney (2011)



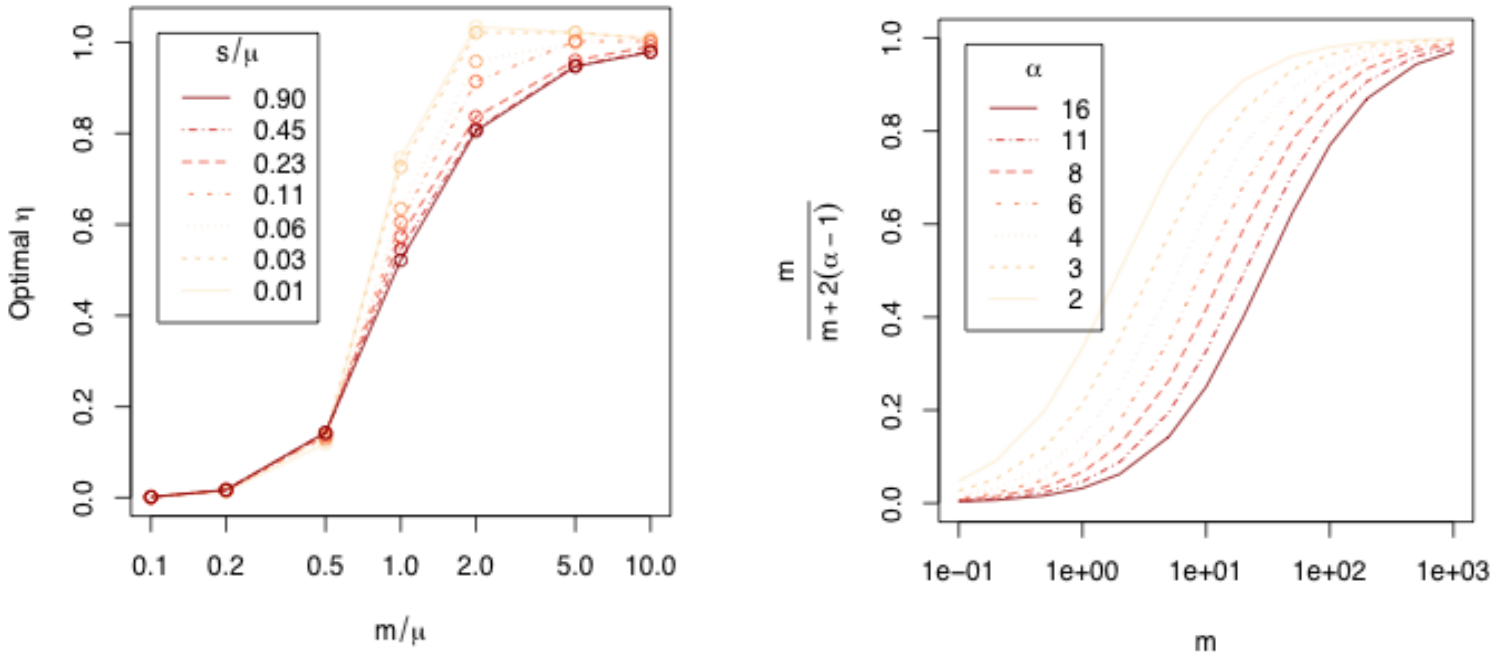
For certain values of η, regularized estimate $\hat{\Delta}_\eta$ outperforms unregularized estimate $\hat{\Delta}$, i.e. $\|\Theta - \hat{\Theta}_\eta\|_F / \|\Theta - \hat{\Theta}\|_F < 1$; and similarly for spectral norm error.

Empirical results (2 of 3)

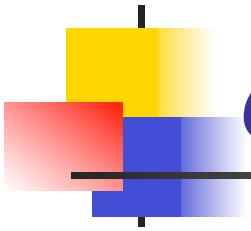


The optimal regularization η depends on m/μ and s.

Empirical results (3 of 3)



The optimal η increases with m and s/μ (left); this agrees qualitatively with the Proposition (right).



Outline and overview

Preamble: algorithmic & statistical perspectives

General thoughts: data & algorithms, and explicit & implicit regularization

Approximate first nontrivial eigenvector of Laplacian

- Three diffusion-based procedures (heat kernel, PageRank, truncated lazy random walk) are *implicitly* solving a regularized optimization *exactly*!

A statistical interpretation of this result

- Analogous to Gaussian/Laplace interpretation of Ridge/Lasso regression

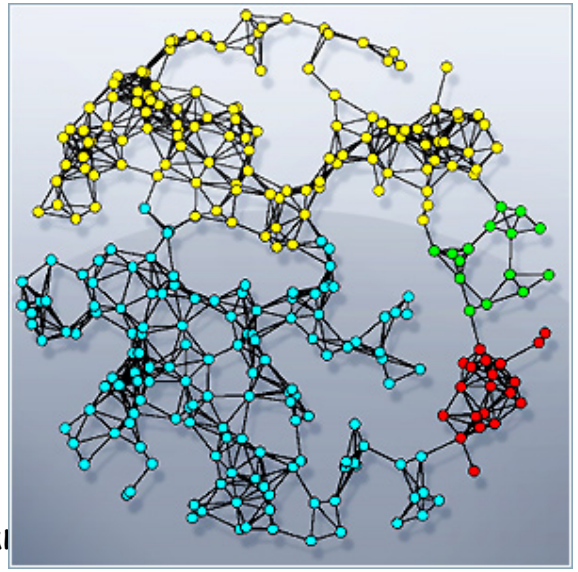
Spectral versus flow-based algs for graph partitioning

- Theory says each regularizes in different ways; empirical results agree!

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)
- β-balanced bisection (minimum cut with 70-30 balance)
- cutsize/min{|A|,|B|}, or cutsize/(|A||B|) (expansion)
- cutsize/min{Vol(A),Vol(B)}, or cutsize/(Vol(A)Vol(B)) (conductan

All of these formalizations of the bi-criterion are NP-hard!

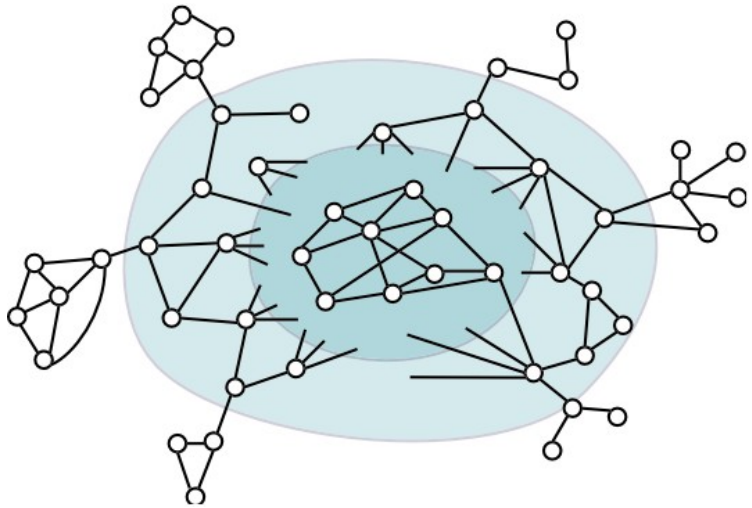
Networks and networked data

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



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]

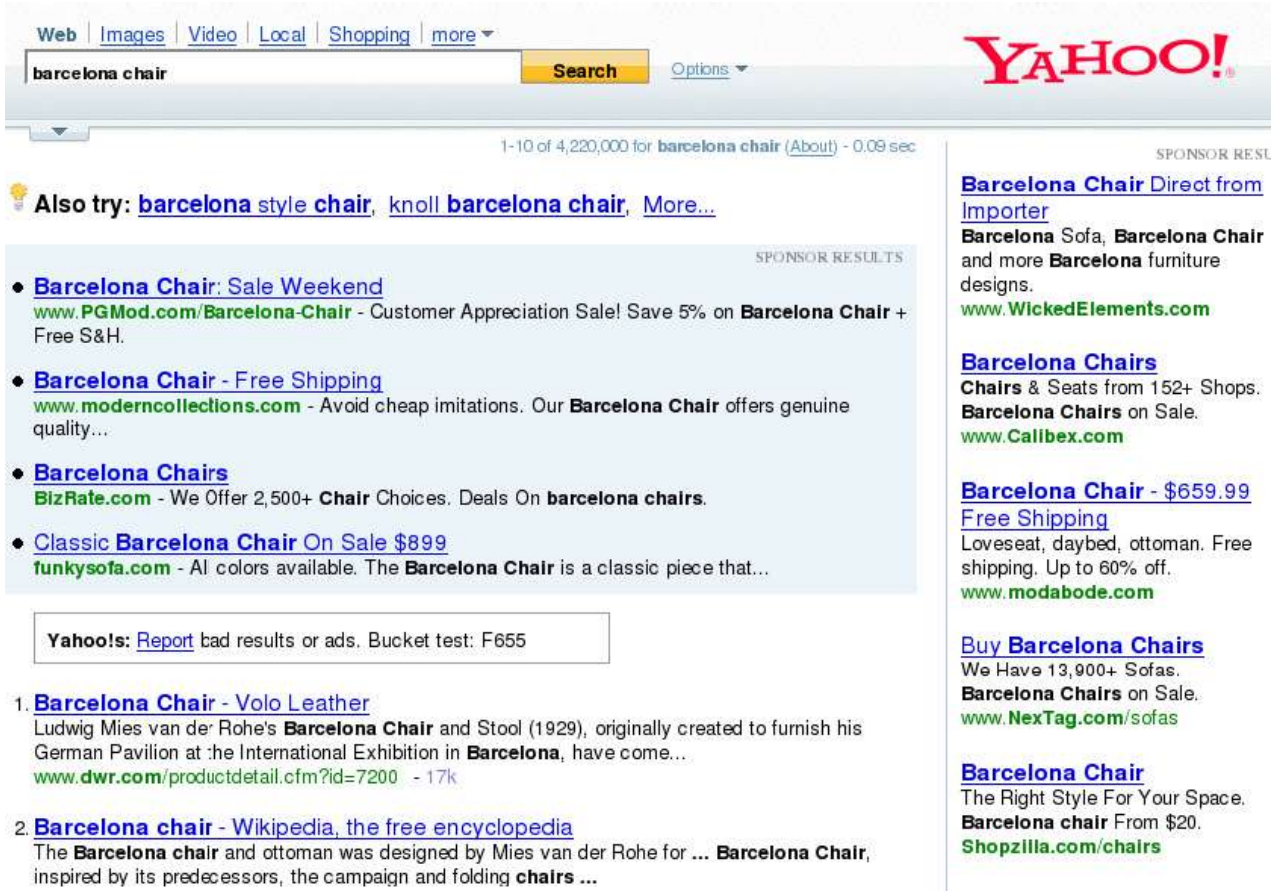
Table 1: Some of the network datasets we studied.

Motivation: Sponsored ("paid") Search

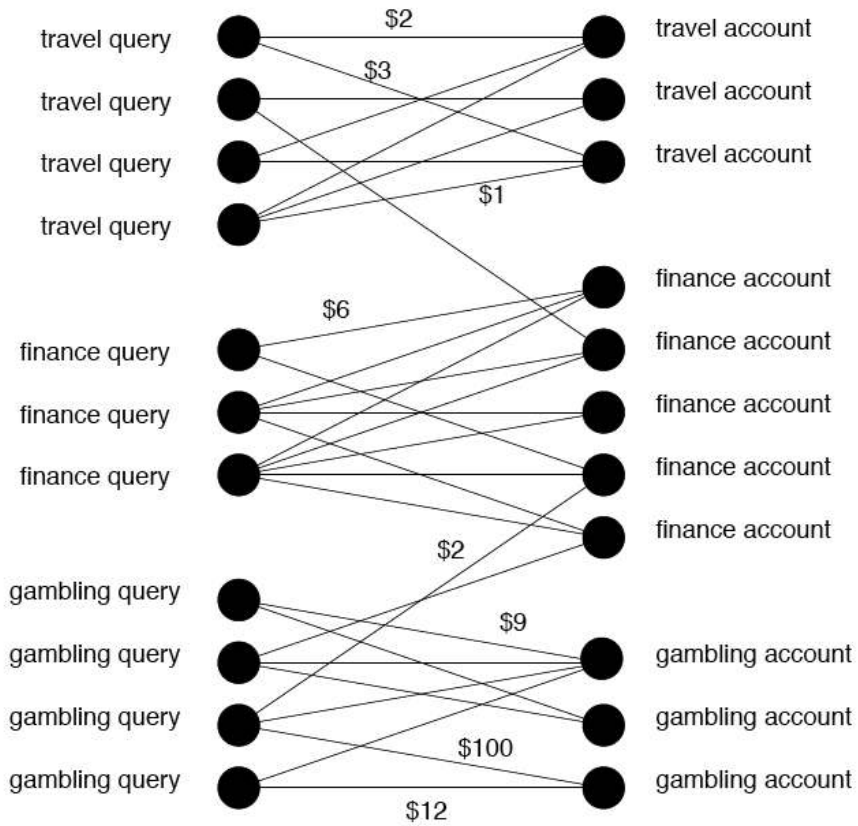
Text based ads driven by user specified query

The process:

- Advertisers bids on query phrases.
- Users enter query phrase.
- Auction occurs.
- Ads selected, ranked, displayed.
- When user clicks, advertiser pays!



Bidding and Spending Graphs



A "social network" with "term-document" aspects.

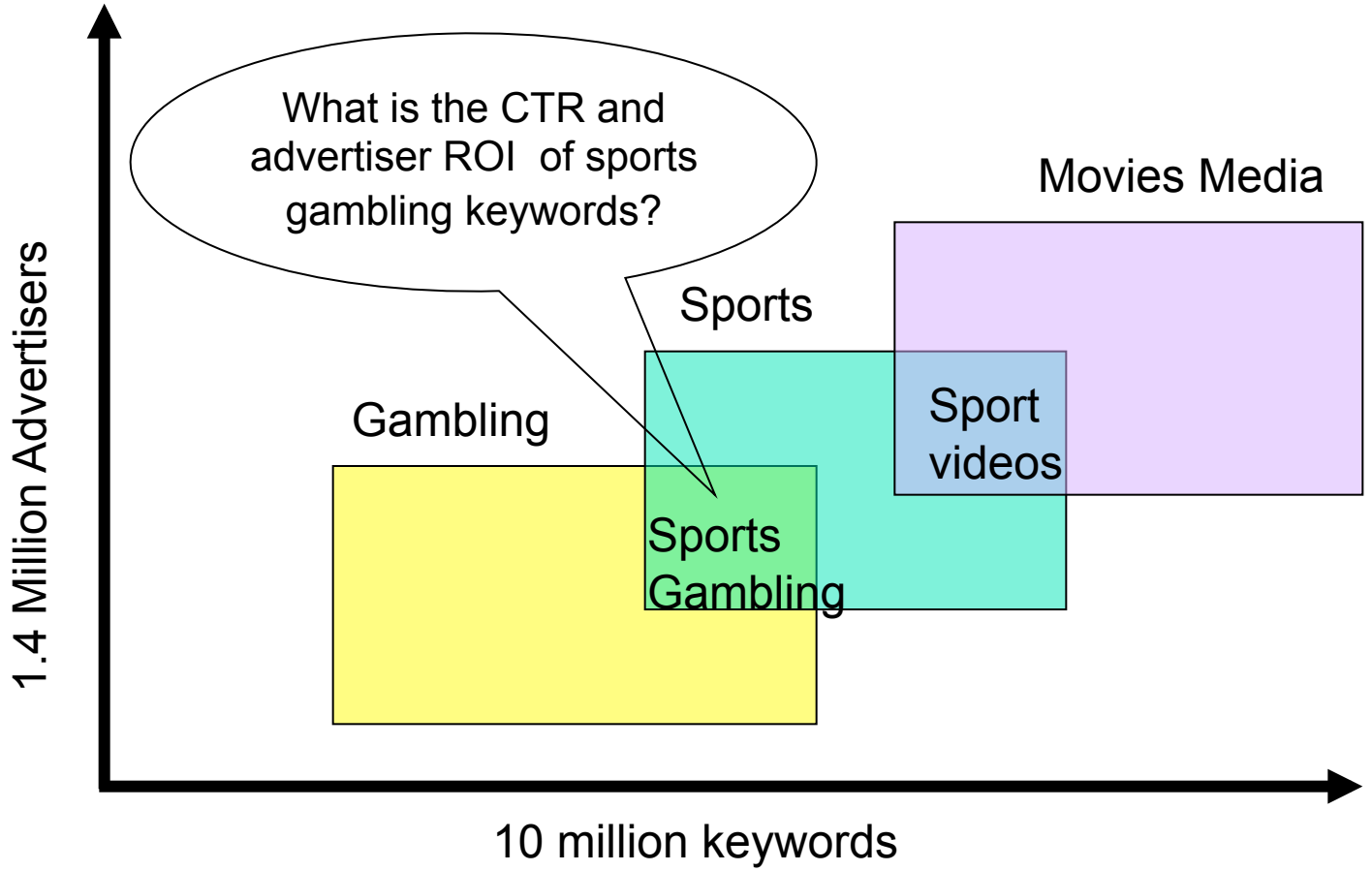
Uses of Bidding and Spending graphs:

- "deep" micro-market identification.
- improved query expansion.

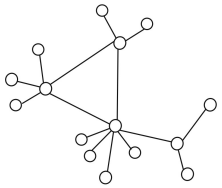
More generally, user segmentation for behavioral targeting.

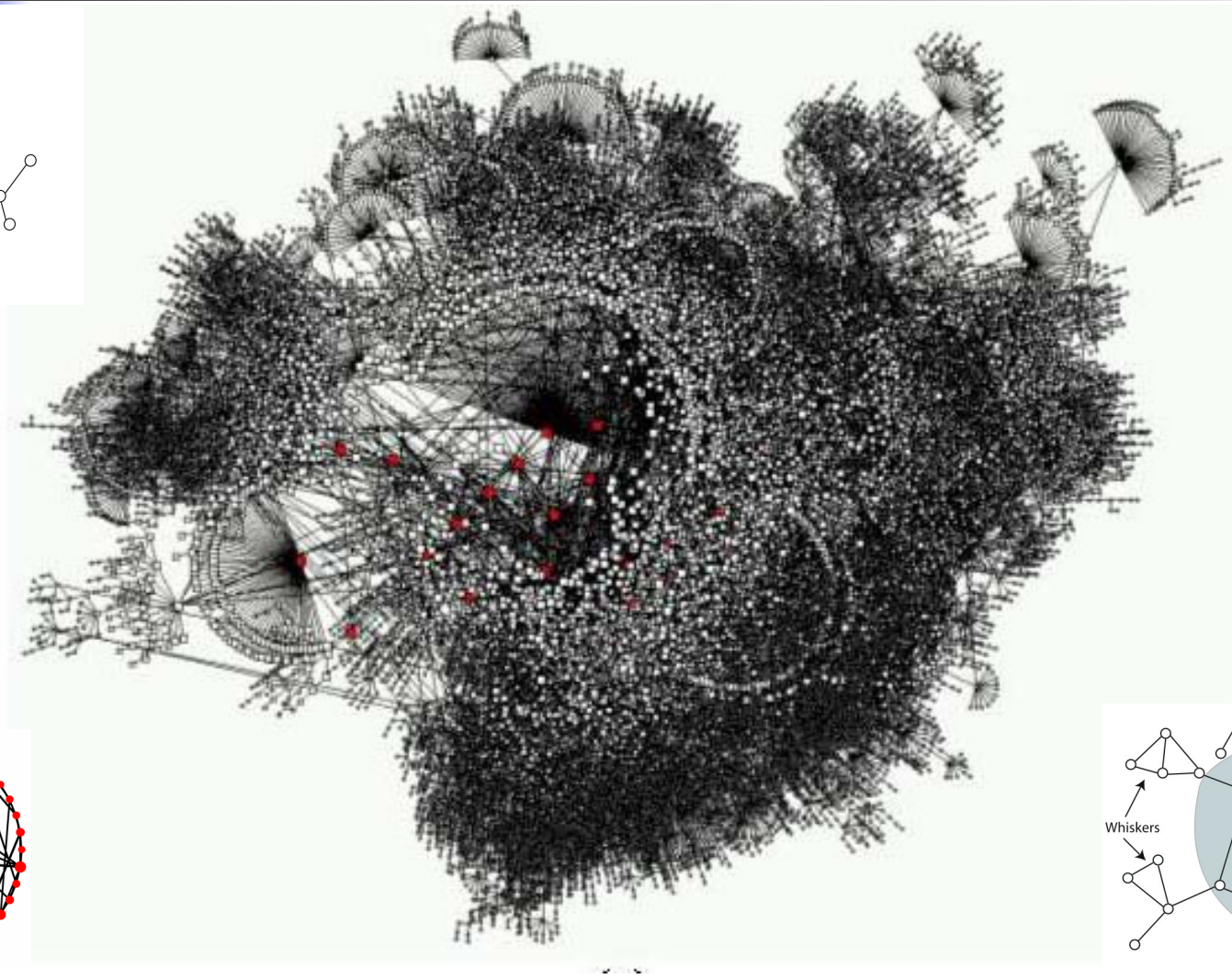
Micro-markets in sponsored search

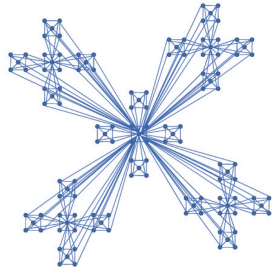
Goal: Find *isolated* markets/clusters with *sufficient money/clicks* with *sufficient coherence*.
Ques: Is this even possible?

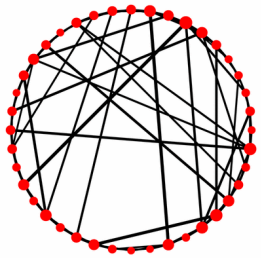


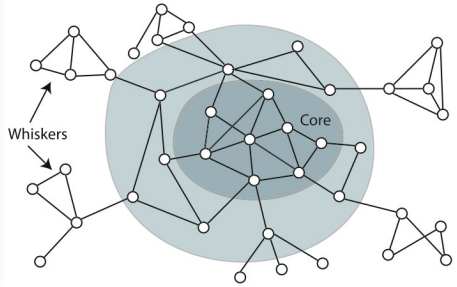
What do these networks "look" like?

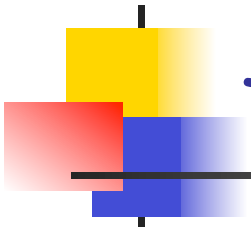












The "lay of the land"

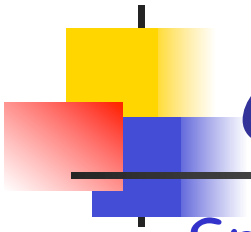
Spectral methods* - compute eigenvectors of associated matrices

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* underlying theory to guide heuristics.



Comparison of "spectral" versus "flow"

Spectral:

- Compute an eigenvector
- "Quadratic" worst-case bounds
- Worst-case achieved -- on "long stringy" graphs
- Worse-case is "local" property
- Embeds you on a line (or K_n)

Flow:

- Compute a LP
- O(log n) worst-case bounds
- Worst-case achieved -- on expanders
- Worst case is "global" property
- Embeds you in L1

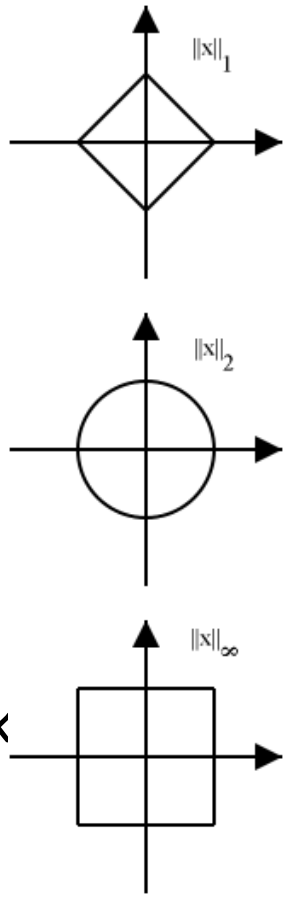
Two methods -- complementary strengths and weaknesses

- What we compute is determined at least as much by as the approximation algorithm as by objective function.

Explicit versus implicit geometry

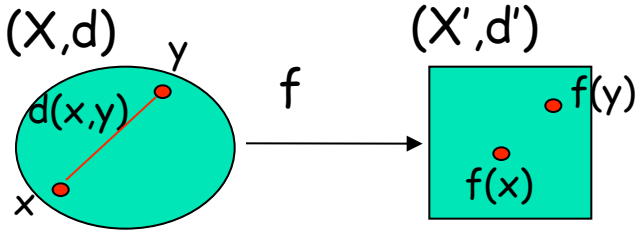
Explicitly-imposed geometry

- Traditional regularization uses *explicit* norm constraint to make sure solution vector is "small" and not-too-complex



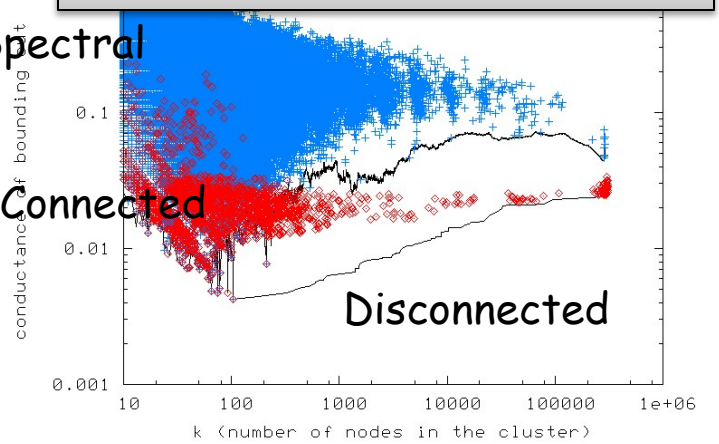
Implicitly-imposed geometry

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

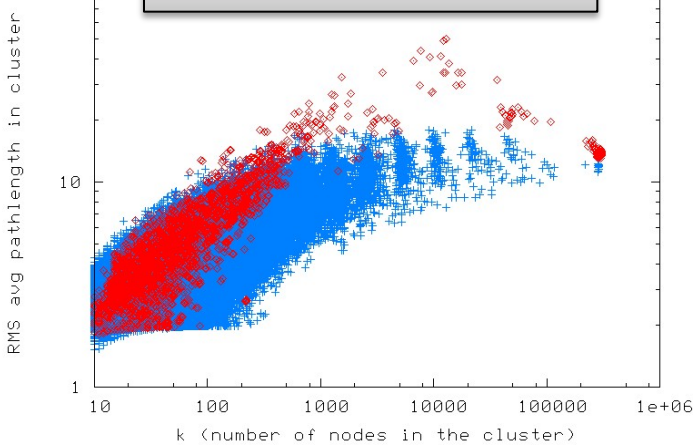


Regularized and non-regularized communities (1 of 2)

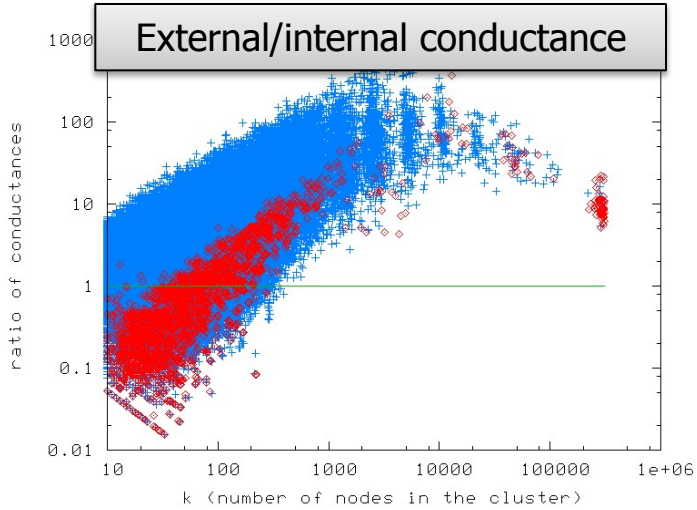
Conductance of bounding cut



Diameter of the cluster



External/internal conductance

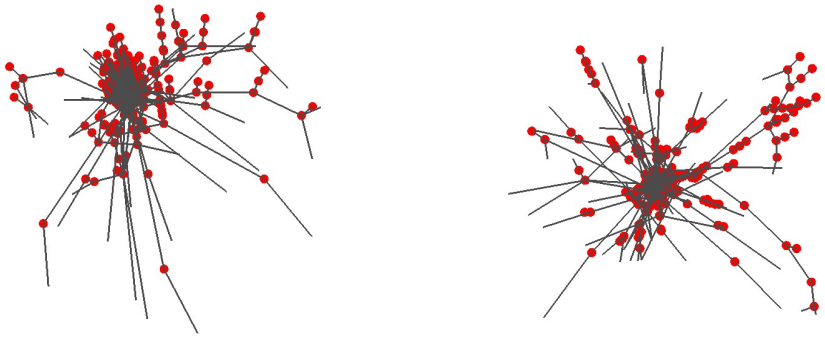


Lower is good

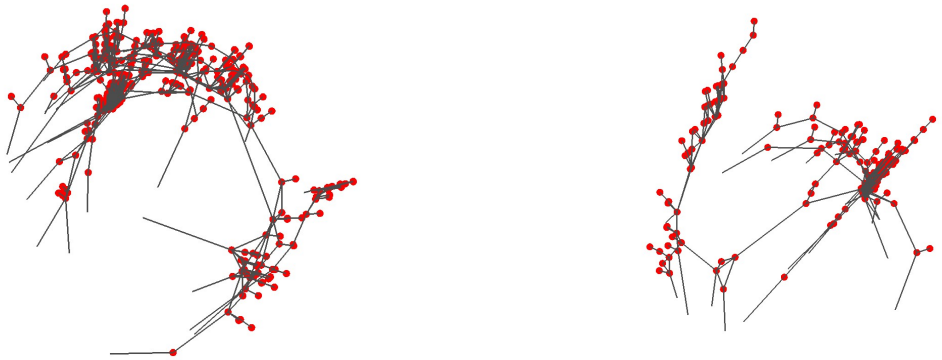
- Metis+MQI - a Flow-based method (red) gives sets with better conductance.
- Local Spectral (blue) gives tighter and more well-rounded sets.

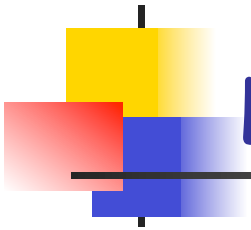
Regularized and non-regularized communities (2 of 2)

Two ca. 500 node communities from Local Spectral Algorithm:



Two ca. 500 node communities from Metis+MQI:





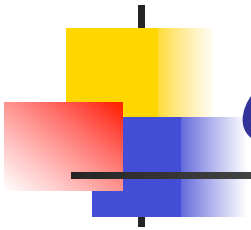
Looking forward …

A common *modus operandi* in many (really*) large-scale applications is:

- Run a procedure that bears some resemblance to the procedure you would run if you were to solve a given problem exactly
- Use the output in a way similar to how you would use the exact solution, or prove some result that is similar to what you could prove about the exact solution.

BIG Question: Can we make this more statistically principled? E.g., can we "engineer" the approximations to solve (exactly but implicitly) some regularized version of the original problem---to do large scale analytics in a statistically more principled way?

*e.g., industrial production, publication venues like WWW, SIGMOD, VLDB, etc.



Conclusions

Regularization is:

- central to Stats & nearly area that applies algorithms to noisy data
- absent from CS, which historically has studied computation per se
- gets at the heart of the algorithmic-statistical "disconnect"

Approximate computation, *in and of itself*, can *implicitly* regularize

- theory & the empirical signatures in matrix *and* graph problems

In very large-scale analytics applications:

- can we "engineer" database operations so "worst-case" approximation algorithms exactly solve regularized versions of original problem?
- I.e., can we get best of both worlds for more statistically-principled very large-scale analytics?