Algorithmic and Statistical Perspectives on BIG Data ... and BIG Information Theory?

Michael W. Mahoney

Stanford University

March 2013

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

BIG data??? MASSIVE data????

The Economist
The data deluge
AND HOW TO HANDLE IT: A 14-PAGE SPECIAL REPORT
Science
data
research
IBM
data management
TAMING BIG DATA
POPULAR SCIENCE
THE CONTROL CENTERS
DATA IS POWER
WIRED
FACEBOOK'S $5 BILLION MISTAKE: HOW IPOs KILL INNOVATION
Deep in the Utah desert, the National Security Agency is building the country's biggest spy center. It's the final piece of a secret surveillance network that will intercept and store your phone calls, emails, Google searches...
(Watch what you say.)
INSIDE THE MATRIX
nature
BIG DATA
SCIENCE IN THE PETABYTE ERA

NYT, Feb 11, 2012: "The Age of Big Data"

- "What is Big Data? A meme and a marketing term, for sure, but also shorthand for advancing trends in technology that open the door to a new approach to understanding the world and making decisions. …"

Why are big data big?

- Generate data at different places/times and different resolutions
- Factor of 10 more data is not just more data, but *different data*

BIG data??? MASSIVE data????

MASSIVE data:

- Internet, Customer Transactions, Astronomy/HEP = "Petascale"
- One Petabyte = watching 20 years of movies (HD) = listening to 20,000 years of MP3 (128 kbits/sec) = way too much to browse or comprehend

massive data:

- 10^5 people typed at 10^6 DNA SNPs; 10^6 or 10^9 node social network; etc.

In either case, main issues:

- Memory management issues, e.g., push computation to the data
- Hard to answer even basic questions about what data "looks like"

Algorithmic vs. Statistical Perspectives

Lambert (2000); Mahoney "Algorithmic and Statistical Perspectives on Large-Scale Data Analysis" (2010)

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.

Perspectives are NOT incompatible

- Statistical/probabilistic ideas are central to recent work on developing improved randomized algorithms for matrix problems.

- Intractable optimization problems on graphs/networks yield to approximation when assumptions are made about network participants.

- In boosting (a statistical technique that fits an additive model by minimizing an objective function with a method such as gradient descent), the computation parameter (i.e., the number of iterations) also serves as a regularization parameter.

But they are VERY different **paradigms**

Statistics, natural sciences, scientific computing, etc:

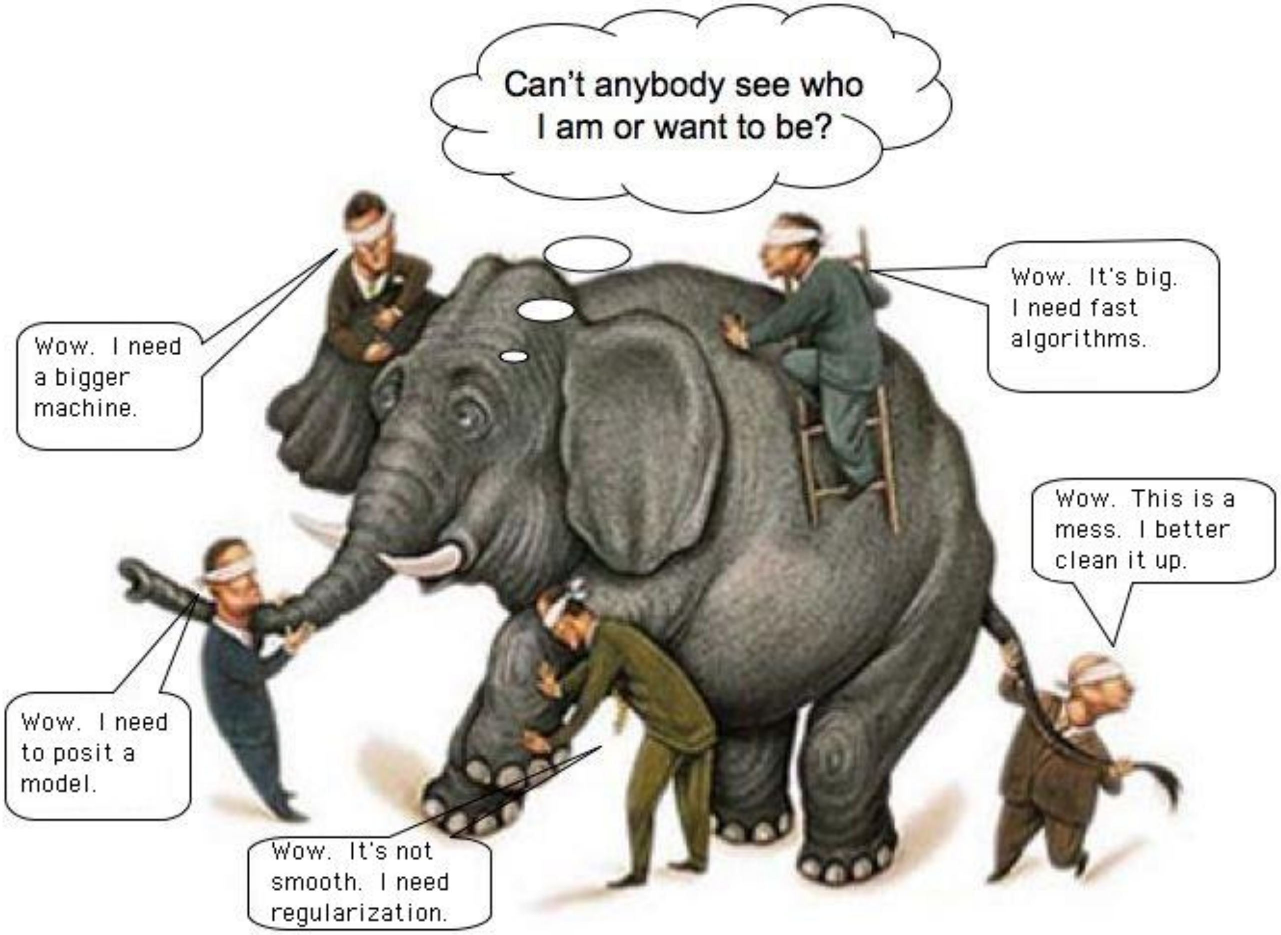
- Problems often involve computation, but the study of computation *per se* is secondary
- Only makes sense to develop algorithms for well-posed* problems
- First, write down a model, and think about computation later

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

How do we view BIG data?


Can't anybody see who I am or want to be?
Wow. I need a bigger machine.
Wow. It's big. I need fast algorithms.
Wow. This is a mess. I better clean it up.
Wow. I need to posit a model.
Wow. It's not smooth. I need regularization.

In Two Parts

Part One: Algorithmic and Statistical Perspectives on Large-scale Data Analysis:

- Describes these two approaches with two "anecdotes" from genetics and internet advertising applications
- Preprint: arXiv:1010.1609 (2010); In: Combinatorial Scientific Computing, pp. 427-469, edited by U. Naumann and O. Schenk, 2012

Part Two: Approximate Computation and Implicit Regularization in Large-scale Data Analysis:

- Describes regularization, the concept at the heart of this difference, in traditional and novel contexts
- Preprint: arXiv:1203.0786 (2012);Proc. of the 2012 ACM Symposium on Principles of Database Systems, 143-154, 2012

In Two Parts

Part One: Algorithmic and Statistical Perspectives on Large-scale Data Analysis:

• Describes these two approaches with two “anecdotes” from genetics and internet advertising applications

• Preprint: arXiv:1010.1609 (2010); In: Combinatorial Scientific Computing, pp. 427-469, edited by U. Naumann and O. Schenk, 2012

Part Two: Approximate Computation and Implicit Regularization in Large-scale Data Analysis:

• Describes regularization, the concept at the heart of this difference, in traditional and novel contexts

• Preprint: arXiv:1203.0786 (2012);Proc. of the 2012 ACM Symposium on Principles of Database Systems, 143-154, 2012

Matrices and graphs in data analysis

Graphs:

- *model* information network with graph G = (V,E) -- vertices represent “entities” and edges represent “interactions” between pairs of entities

Matrices:

- *model* data sets by a matrix -- since an m x n matrix A can encode information about m objects, each of which is described by n features

Matrices and graphs represent a nice tradeoff between:

- descriptive flexibility
- algorithmic tractability

But, the issues that arise are very different than in traditional linear algebra or graph theory AND the data place very different demands on hardware than in traditional database or supercomputer applications.

Outline for Part One

- "Algorithmic" and "statistical" perspectives on data problems
- Genetics application

  DNA SNP analysis --> choose columns from a matrix

  PMJKPGKD, *Genome Research* '07; PZBCRMD, *PLOS Genetics* '07; Mahoney and Drineas, *PNAS* '09; DMM, *SIMAX* '08; BMD, *SODA* '09

- Internet application

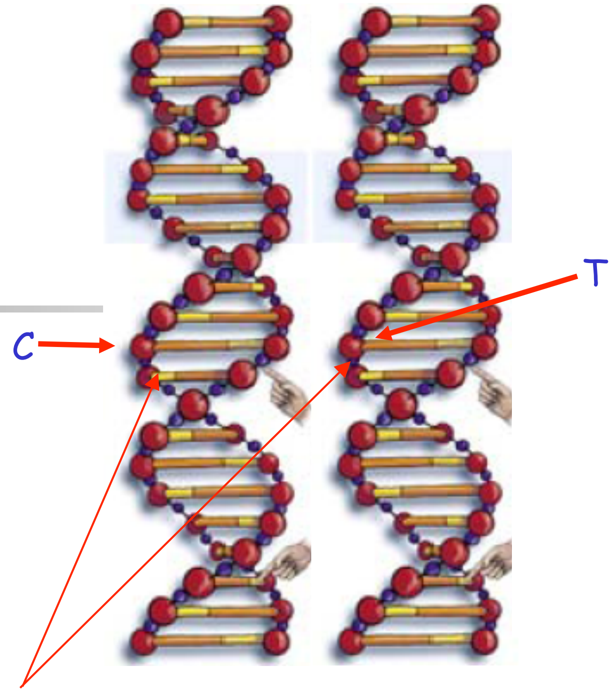
  Community finding --> partitioning a graph

  LLDM (*WWW* '08 & *TR* '08-IM'09 & *WWW* '10)

We will focus on what was going on "under the hood" in these two applications --- use statistical properties implicit in worst-case algorithms to make domain-specific claims!

DNA SNPs and human genetics

- Human genome ≈ 3 billion base pairs
- 25,000 – 30,000 genes
- Functionality of 97% of the genome is unknown.
- Individual "polymorphic" variations at ≈ 1 b.p./thousand.



SNPs are known locations at the human genome where two alternate nucleotide bases (alleles) are observed (out of A, C, G, T).

SNPs

individuals

```
... AG CT GT GG CT CC CC CC CC AG AG AG AG AG AA CT AA GG GG CC GG AG CG AC CC AA CC AA GG TT AG CT CG CG CG AT CT CT AG CT AG GG GT GA AG ...
... GG TT TT GG TT CC CC CC CC GG AA AG AG AG AA CT AA GG GG CC GG AA GG AA CC AA CC AA GG TT AA TT GG GG GG TT TT CC GG TT GG GG TT GG AA ...
... GG TT TT GG TT CC CC CC CC GG AA AG AG AA AG CT AA GG GG CC AG AG CG AC CC AA CC AA GG TT AG CT CG CG CG AT CT CT AG CT AG GG GT GA AG ...
... GG TT TT GG TT CC CC CC CC GG AA AG AG AG AA CC GG AA CC CC AG GG CC AC CC AA CG AA GG TT AG CT CG CG CG AT CT CT AG CT AG GT GT GA AG ...
... GG TT TT GG TT CC CC CC CC GG AA GG GG GG AA CT AA GG GG CT GG AA CC AC CG AA CC AA GG TT GG CC CG CG CG AT CT CT AG CT AG GG TT GG AA ...
... GG TT TT GG TT CC CC CG CC AG AG AG AG AG AA CT AA GG GG CT GG AG CC CC CG AA CC AA GT TT AG CT CG CG CG AT CT CT AG CT AG GG TT GG AA ...
... GG TT TT GG TT CC CC CC CC GG AA AG AG AG AA TT AA GG GG CC AG AG CG AA CC AA CG AA GG TT AA TT GG GG GG TT TT CC GG TT GG GT TT GG AA ...
```

SNPs occur quite frequently within the genome and thus are effective genomic markers for the tracking of disease genes and population histories.

DNA SNPs and data analysis

A common *modus operandi* in applying NLA to data problems:

- Write the gene/SNP data as an m x n matrix A.
- Do SVD/PCA to get a small number of eigenvectors
- Either: interpret the eigenvectors as meaningful i.t.o. underlying genes/SNPs

  use a heuristic to get actual genes/SNPs from those eigenvectors

Unfortunately, eigenvectors themselves are meaningless (recall reification in stats):

- "EigenSNPs" (being linear combinations of SNPs) can not be assayed …
- … nor can "eigengenes" from micro-arrays be isolated and purified …
- … nor do we really care about "eigenpatients" respond to treatment ...

DNA SNPs and low-rank methods

PMJKPGKD, *Genome Research* '07 (data from K. Kidd, Yale University)
PZBCRMD, *PLOS Genetics* '07 (data from E. Ziv and E. Burchard, UCSF)

- Common genetics task: find a small subset of informative *actual SNPs*

  to cluster individuals depending on their ancestry

  to determine predisposition to diseases

- **Algorithmic question:** Can we find the best k actual columns from a matrix?

  Can we find *actual SNPs* that "capture" information in singular vectors?

  Can we find *actual SNPs* that are maximally uncorrelated?

- Common formalization of "best" lead to intractable optimization problems.

Column Subset Selection Problem (CSSP)

Input: an m-by-n matrix A and a rank parameter k.

Goal: choose *exactly k columns* of A s.t. the m-by-k matrix C minimizes the error:

$$\min ||A - P_C A||_\xi = \min ||A - CC^+ A||_\xi \quad (\xi = 2, F)$$

- Widely-studied problem in numerical linear algebra and optimization.
- Related to unsupervised feature selection.
- Choose the "best" k documents from a document corpus.

A hybrid two-stage algorithm

Boutsidis, Mahoney, and Drineas (2007)

** Diagonal elements of the "hat matrix"--- see later.*

Algorithm: Given an m-by-n matrix *A* and rank parameter k:

- (Randomized phase)

  Randomly select *c = O(k logk)* columns according to "leverage score probabilities*".

- (Deterministic phase)

  Run a deterministic algorithm on the above columns to pick exactly *k* columns of *A*.

Theorem: Let C be the m-by-k matrix of the selected columns. Our algorithm runs in "O(mmk)" and satisfies, w.p. ≥ $1\text{-}10^{-20}$,

$$||A - P_C A||_F \leq O\left(k \log^{1/2} k\right) ||A - A_k||_F$$

$$||A - P_C A||_2 \leq O\left(k^{3/4} \log^{1/2}(k) \left(n-k\right)^{1/2}\right) ||A - A_k||_2$$

Comparison with previous results

Running time: comparable with NLA algorithms.

Spectral norm:

- Spectral norm bound is $k^{1/4}\log^{1/2}k$ worse than previous work.

Frobenius norm:

- An efficient *algorithmic* result at most $(k \log k)^{1/2}$ worse than the previous *existential* result.

NLA: Deterministic algorithms.

Spectral norm.

TCS: Randomized algorithms.

Sample more than k columns.

Frobenius norm bounds.

Computation: usually interested in columns for the *bases they span*!

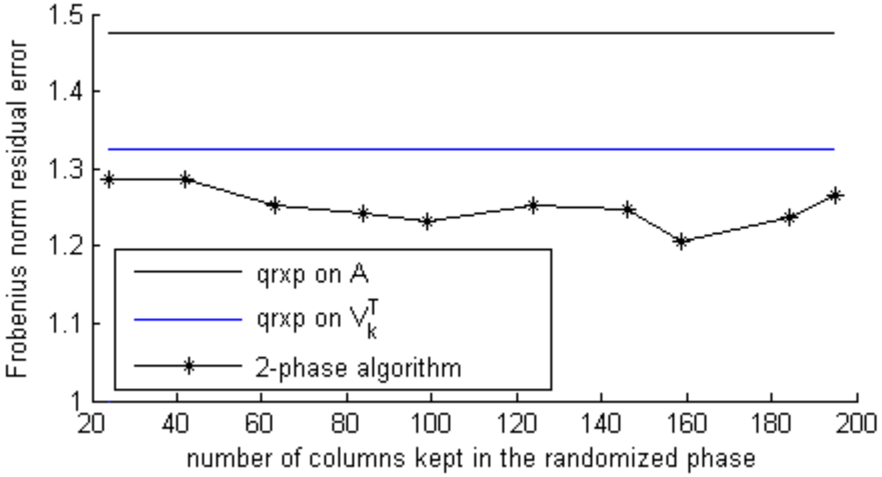
Data analysis: usually interested in the *columns themselves*!

Evaluation on term-document data

TechTC (Technion Repository of Text Categorization Datasets)

- lots of diverse test collections from ODP
- ordered by categorization difficulty
- use hierarchical structure of the directory as background knowledge
- Davidov, Gabrilovich, and Markovitch 2004

Fix k=10 and measure Frobenius norm error:



Things to note …

Different versions of QR (i.e., different pivot rules) perform differently …

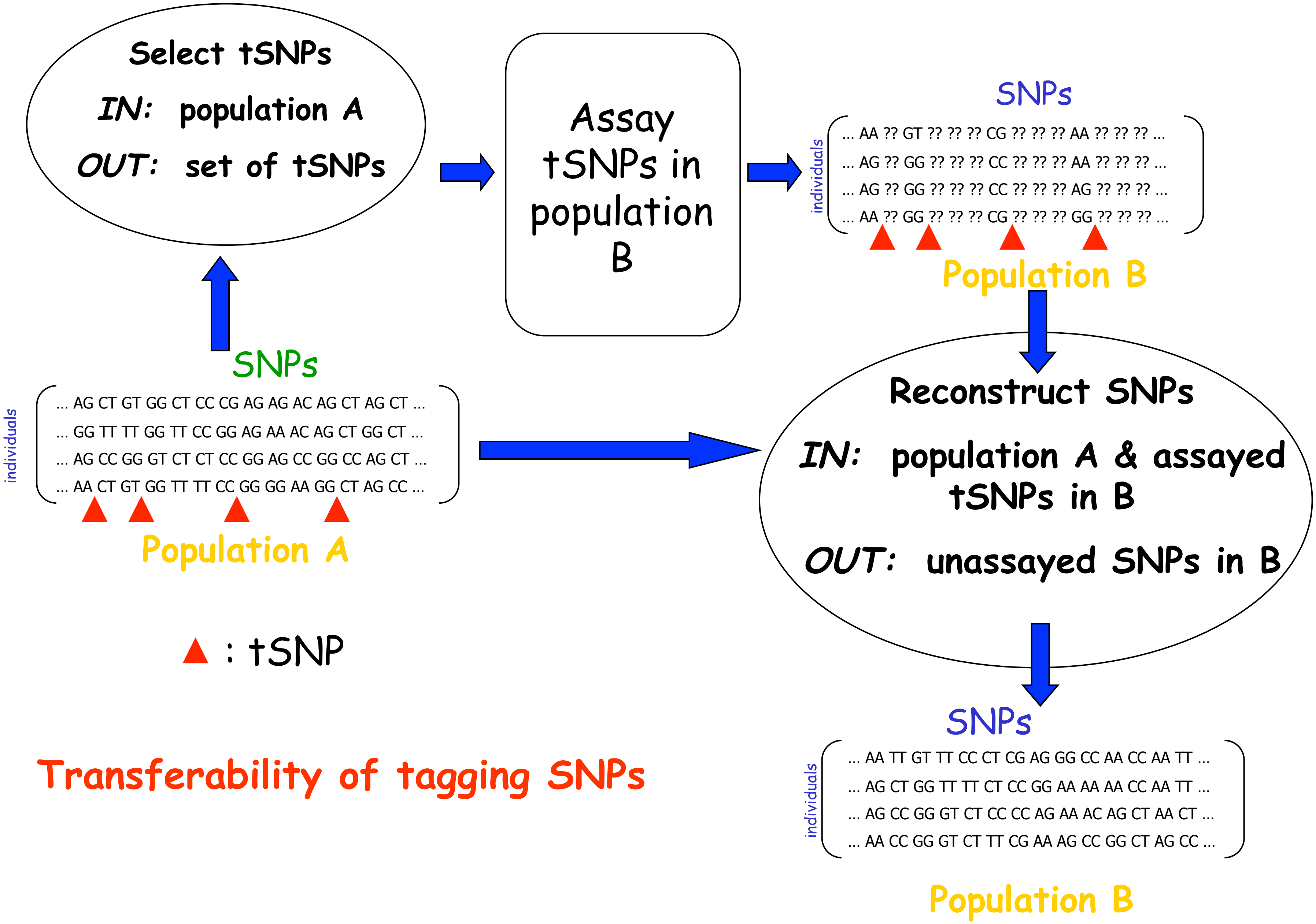
- “obviously,” but be careful with “off the shelf” implementations.

QR applied directly to V_k^T typically does better than QR applied to *A* …

- since V_k^T defined the relevant non-uniformity structure in *A*
- since columns “spread out,” have fewer problems with pivot rules

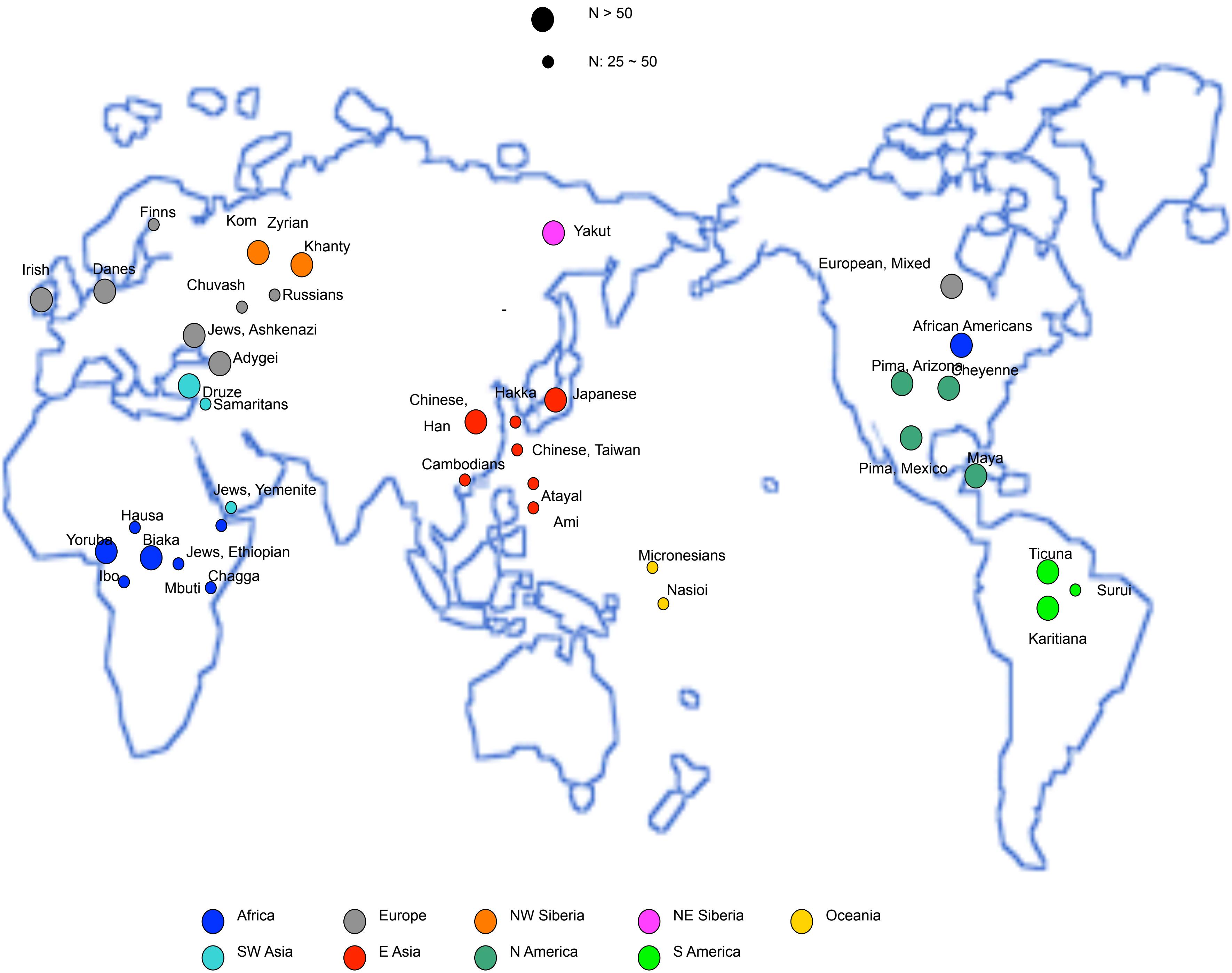
“Randomized preprocessing” improves things even more …

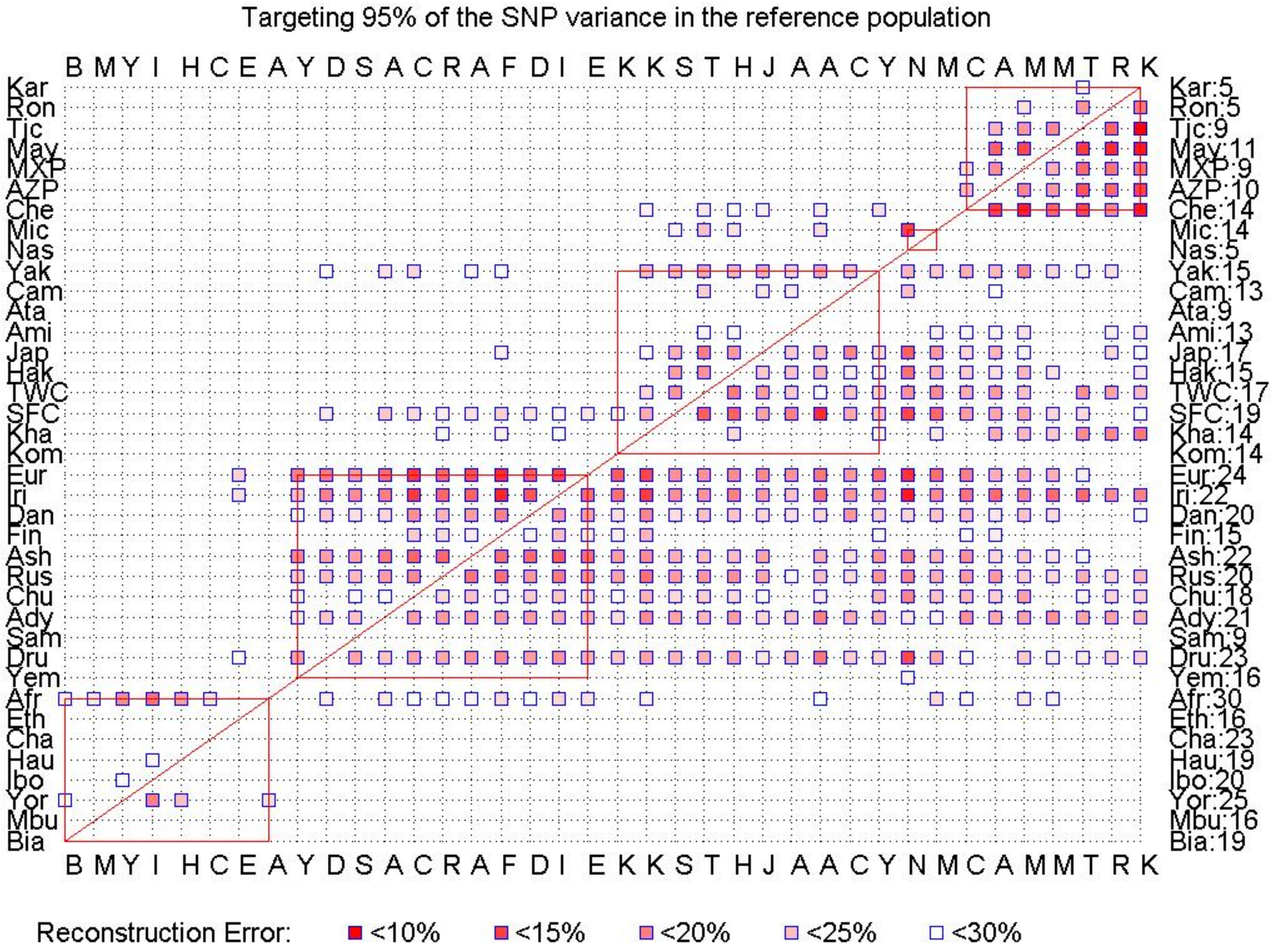
- due to *implicit* regularization
- (if you are careful with various parameter choices)
- and it improves worse QR implementations more than better code


Select tSNPs
IN: population A
OUT: set of tSNPs
Assay tSNPs in population B
SNPs
individuals
... AA ?? GT ?? ?? ?? CG ?? ?? ?? AA ?? ?? ?? ...
... AG ?? GG ?? ?? ?? CC ?? ?? ?? AA ?? ?? ?? ...
... AG ?? GG ?? ?? ?? CC ?? ?? ?? AG ?? ?? ?? ...
... AA ?? GG ?? ?? ?? CG ?? ?? ?? GG ?? ?? ?? ...
Population B
SNPs
individuals
... AG CT GT GG CT CC CG AG AG AC AG CT AG CT ...
... GG TT TT GG TT CC GG AG AA AC AG CT GG CT ...
... AG CC GG GT CT CT CC GG AG CC GG CC AG CT ...
... AA CT GT GG TT TT CC GG GG AA GG CT AG CC ...
Population A
Reconstruct SNPs
IN: population A & assayed tSNPs in B
OUT: unassayed SNPs in B
▲ : tSNP
SNPs
individuals
... AA TT GT TT CC CT CG AG GG CC AA CC AA TT ...
... AG CT GG TT TT CT CC GG AA AA AA CC AA TT ...
... AG CC GG GT CT CC CC AG AA AC AG CT AA CT ...
... AA CC GG GT CT TT CG AA AG CC GG CT AG CC ...
Population B

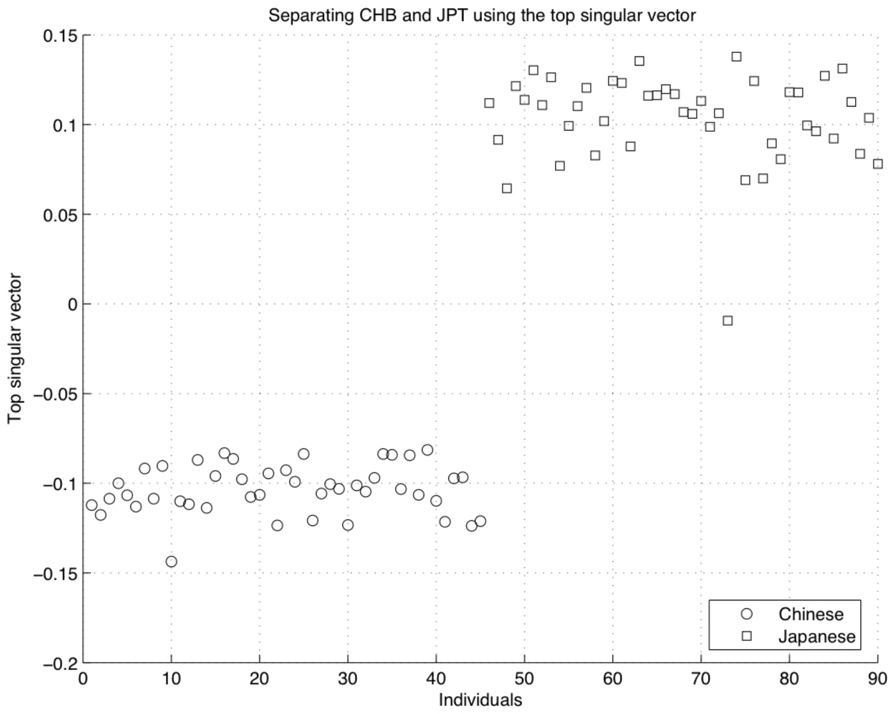
Transferability of tagging SNPs

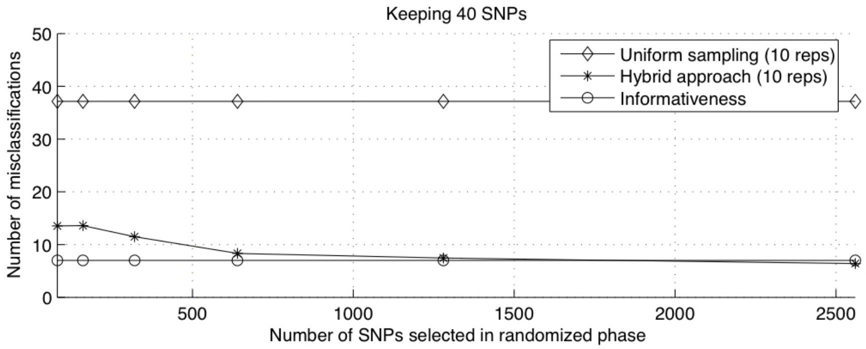
FIG. 6

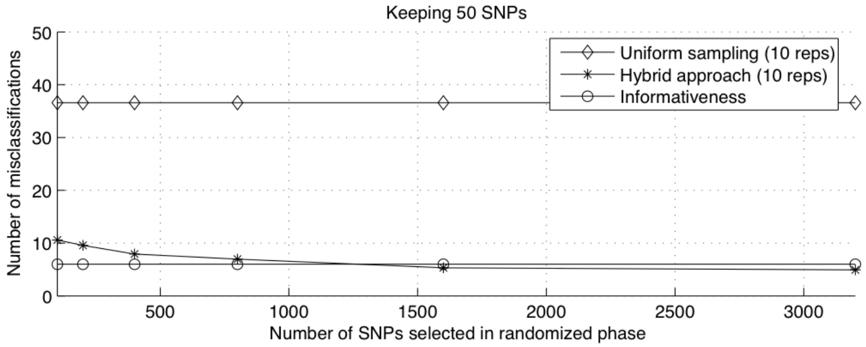

N > 50
N: 25 ~ 50
Finns
Kom
Zyrian
Khanty
Irish
Danes
Chuvash
Russians
Jews, Ashkenazi
Adygei
Druze
Samaritans
Yakut
Chinese,
Han
Hakka
Japanese
Chinese, Taiwan
Cambodians
Atayal
Ami
Jews, Yemenite
Hausa
Yoruba
Biaka
Jews, Ethiopian
Ibo
Chagga
Mbuti
Micronesians
Nasioi
European, Mixed
African Americans
Pima, Arizona
Cheyenne
Pima, Mexico
Maya
Ticuna
Surui
Karitiana
Africa
Europe
NW Siberia
NE Siberia
Oceania
SW Asia
E Asia
N America
S America


Targeting 95% of the SNP variance in the reference population
B M Y I H C E A Y D S A C R A F D I E K K S T H J A A C Y N M C A M M T R K
Kar
Ron
Tic
May
MXP
AZP
Che
Mic
Nas
Yak
Cam
Ata
Ami
Jap
Hak
TWC
SFC
Kha
Kom
Eur
Tri
Dan
Fin
Ash
Rus
Chu
Ady
Sam
Dru
Yem
Afr
Eth
Cha
Hau
Ibo
Yor
Mbu
Bia
Kar:5
Ron:5
Tic:9
May:11
MXP:9
AZP:10
Che:14
Mic:14
Nas:5
Yak:15
Cam:13
Ata:9
Ami:13
Jap:17
Hak:15
TWC:17
SFC:19
Kha:14
Kom:14
Eur:24
Tri:22
Dan:20
Fin:15
Ash:22
Rus:20
Chu:18
Ady:21
Sam:9
Dru:23
Yem:16
Afr:30
Eth:16
Cha:23
Hau:19
Ibo:20
Yor:25
Mbu:16
Bia:19
B M Y I H C E A Y D S A C R A F D I E K K S T H J A A C Y N M C A M M T R K
Reconstruction Error:
<10%
<15%
<20%
<25%
<30%

DNA HapMap SNP data

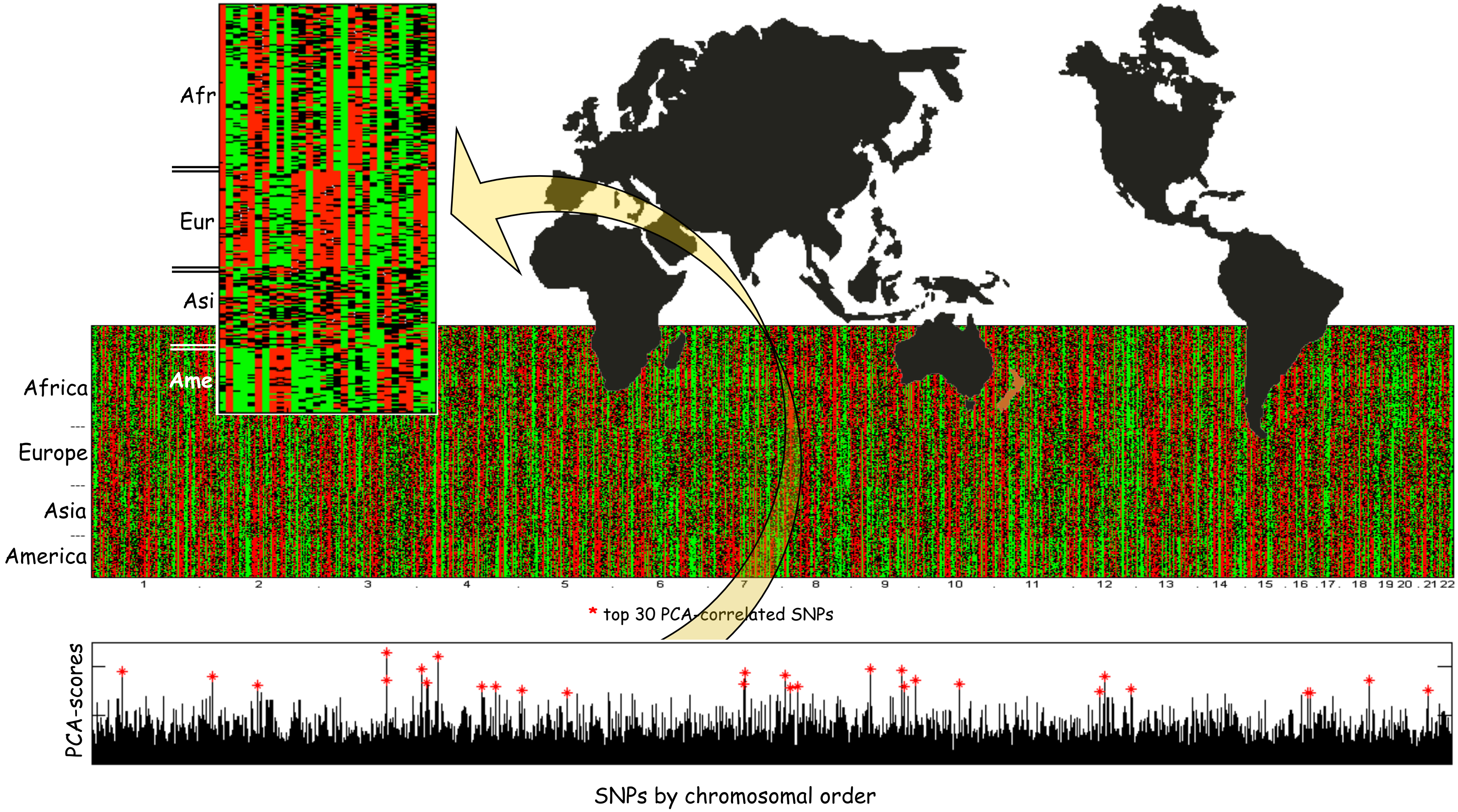






- Most NLA codes don't even run on this 90 x 2M matrix.
- Informativeness is a state of the art supervised technique in genetics.

Selecting PCA-correlated SNPs for individual assignment to four continents (Africa, Europe, Asia, America)



Paschou et al (2007) PLoS Genetics

An Aside on: Least Squares (LS) Approximation

$$\begin{pmatrix} \\ \\ A \\ \\ \\ \end{pmatrix} \begin{pmatrix} \widehat{x} \end{pmatrix} \approx \begin{pmatrix} \\ \\ b \\ \\ \\ \end{pmatrix}$$

$n \times d \quad , \quad n \gg d$

⇒

$$\begin{aligned} \mathcal{Z}_2 &= \min_{x \in \mathbb{R}^d} ||b - Ax||_2 \\ &= ||b - A\hat{x}||_2 \end{aligned}$$

Ubiquitous in applications & central to theory:

Statistical interpretation: best linear unbiased estimator.

Geometric interpretation: orthogonally project b onto span(A).

Algorithmic and Statistical Perspectives

$$\begin{aligned} \mathcal{Z}_2 &= \min_{x \in R^d} ||b - Ax||_2 \\ &= ||b - A\hat{x}||_2 \end{aligned}$$

Algorithmic Question: How long does it take to solve this LS problem?

Answer: O(nd^2) time, with Cholesky, QR, or SVD*

Statistical Question: When is solving this LS problem the right thing to do?

Answer: When the data are "nice," as quantified by the leverage scores.

*BTW, we used statistical leverage score ideas to get the first (1+ε)-approximation worst-case-analysis algorithm for the general LS problem that runs in o(nd^2) time for *any* input matrix.

Theory: DM06,DMM06,S06,DMMS07

Numerical implementation: Tygert, Rokhlin, etc. (2008), Avron, Maymounkov, and Toledo (2009)

Statistical Issues and Regression Diagnostics

Statistical Model: $b = Ax+\varepsilon$

ε = "nice" error process

$b' = A\ x_{opt} = A(A^TA)^{-1}A^Tb$ = prediction

$H = A(A^TA)^{-1}A^T$ is the "hat" matrix, i.e. projection onto span(A)

Statistical Interpretation:

H_{ij} -- measures the leverage or influence exerted on b'_i by b_j,

Note: $H_{ii} = |U^{(i)}|_2^2$ = row "lengths" of spanning orthogonal matrix

Note 1: these are the sampling probabilities we used for our worst-case algorithms!

Note 2: high leverage scores traditionally used to flag outliers!

Note 3: can compute all of them to $(1\pm\varepsilon)$ in $o(nd^2)$ time!

An Aside on the Aside on LS: Traditional algorithms

For L2 regression:

- *direct methods*: QR, SVD, and normal equation ($O(mn^2 + n^2)$ time)
  - Pros: high precision & implemented in LAPACK
  - Cons: hard to take advantage of sparsity & hard to implement in parallel environments

- *iterative methods*: CGLS, LSQR, etc.
  - Pros: low cost per iteration, easy to implement in some parallel environments, & capable of computing approximate solutions
  - Cons: hard to predict the number of iterations needed

For L1 regression:

- linear programming
- interior-point methods (or simplex, ellipsoid? methods)
- re-weighted least squares
- first-order methods

Two important notions: leverage and condition

Statistical leverage. (Think: eigenvectors & low-precision solutions.)

- The *statistical leverage scores* of A (assume m>>n) are the diagonal elements of the projection matrix onto the column span of A.
- They equal the L2-norm-squared of any orthogonal basis spanning A.
- They measure:
    - how well-correlated the singular vectors are with the canonical basis
    - which constraints have largest "influence" on the LS fit
    - a notion of "coherence" or "outlierness"
- *Computing them exactly is as hard as solving the LS problem*.

Condition number. (Think: eigenvalues & high-precision solutions.)

- The *L2-norm condition number* of A is (A) = $\sigma_{max}(A)/\sigma_{min}(A)$.
- $\kappa(A)$ bounds the number of iterations
    - for ill-conditioned problems (e.g., $\kappa(A) \cong 10^6 >> 1$), convergence speed is slow.
- *Computing $\kappa(A)$ is generally as hard as solving the LS problem*.

These are for the L2-norm. Generalizations exist for the L1-norm.

Condition number, well-conditioned bases and leverage scores for L1 norm

(Dasgupta, Drineas, Harb, Kumar, Mahoney (2008); Clarkson, Drineas, Magdon-Ismail, Mahoney, Meng, Woodruff (2012))

Convenient to formulate L1 regression in what follows as:

$$\min_{x \varepsilon R^n} ||Ax||_1 \text{ s.t. } c^T x=1$$

- **Def:** A matrix $U \varepsilon R^{m \times n}$ is *(α, β, p = 1)-conditioned* if $||U||_1 \leq \alpha$ and $||x||_\infty \leq \beta\, ||Ux||_1$, forall x; and *L1-well-conditioned* if α, β = poly(n).

- **Def:** The *L1 leverage scores* of an m x n matrix A, with m > n, are the L1-norms-squared of the rows of any L1-well-conditioned basis of A. (Only well-defined up to poly(n) factors.)

- **Def:** The *L1-norm condition number* of A, denoted by $\kappa_1(A)$, is:

$$\kappa_1(A) = \sigma_{1,max}(A) / \sigma_{1,min}(A)$$
$$= (Max_{||x||2=1} ||Ax||_1) / (Min_{||x||2=1} ||Ax||_1)$$

Note that this implies:

$$\sigma_{1,min}(A)||x||_2 \leq ||Ax||_1 \leq \sigma_{1,max}(A)||x||_2 \quad \text{, forall } x \varepsilon R^n.$$

Meta-algorithm for L2 regression

(Drineas, Mahoney, etc., 2006, 2008, etc., starting with SODA 2006; Mahoney FnTML, 2011.)

1: Using the L2 statistical leverage scores of A, construct an importance sampling distribution $\{p_i\}_{i=1,\ldots,m}$

2: Randomly sample a small number of constraints according to $\{p_i\}_{i,\ldots,m}$ to construct a subproblem.

3: Solve the L2-regression problem on the subproblem.

Naïve implementation: $1 + \varepsilon$ approximation in $O(mn^2/\varepsilon)$ time. (Ugh.)

"Fast" $O(mn \log(n)/\varepsilon)$ in RAM if

- Hadamard-based projection and sample uniformly
- Quickly compute approximate leverage scores

"High precision" $O(mn \log(n)\log(1/\varepsilon))$ in RAM if:

- use the random projection/sampling basis to construct a preconditioner

Question: can we extend these ideas to parallel-distributed environments?

Meta-algorithm for L1 (& Lp) regression

(Clakson 2005, DDHKM 2008, Sohler and Woodruff 2011, CDMMMW 2012, Meng and Mahoney 2012.)

1: Using the L1 statistical leverage scores of A, construct an importance sampling distribution $\{p_i\}_{i=1,\ldots,m}$

2: Randomly sample a small number of constraints according to $\{p_i\}_{i,\ldots,m}$ to construct a subproblem.

3: Solve the L1-regression problem on the subproblem.

Naïve implementation: $1 + \varepsilon$ approximation in $O(mn^5/\varepsilon)$ time. (Ugh.)

"Fast" in RAM if

- we perform a fast "L1 projection" to uniformize them approximately
- we approximate the L1 leverage scores quickly

"High precision" in RAM if:

- we use the random projection/sampling basis to construct an L1 preconditioner

Question: can we extend these ideas to parallel-distributed environments?

Parallel and distributed algorithms

Meng, Saunders, and Mahoney (2011, arXiv); Meng and Mahoney (2013)

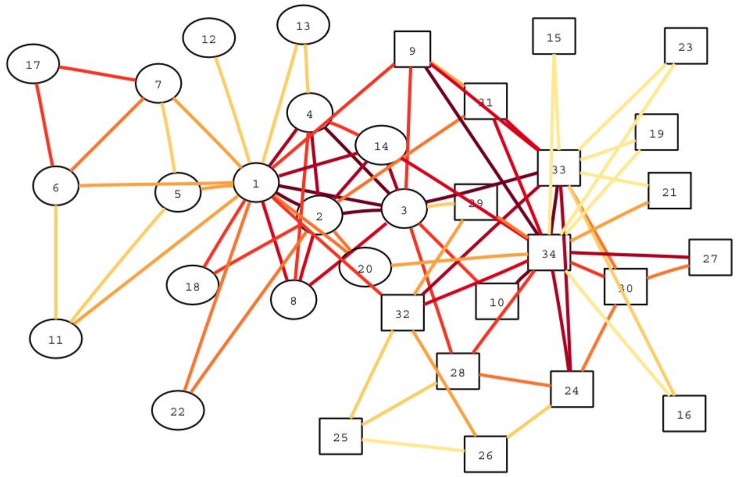
For L2 regression (LSRN):

- computes unique min-length solution to $\min_x ||Ax-b||_2$
- very over/under-constrained, full-rank or rank-deficient A
- A can be dense, sparse, or a linear operator
- easy to implement using threads or with MPI, and scales well in parallel environments
- Minimize communication with the Chebyshev semi-iterative method
- Do L2 regression on communication-constrained Amazon EC2

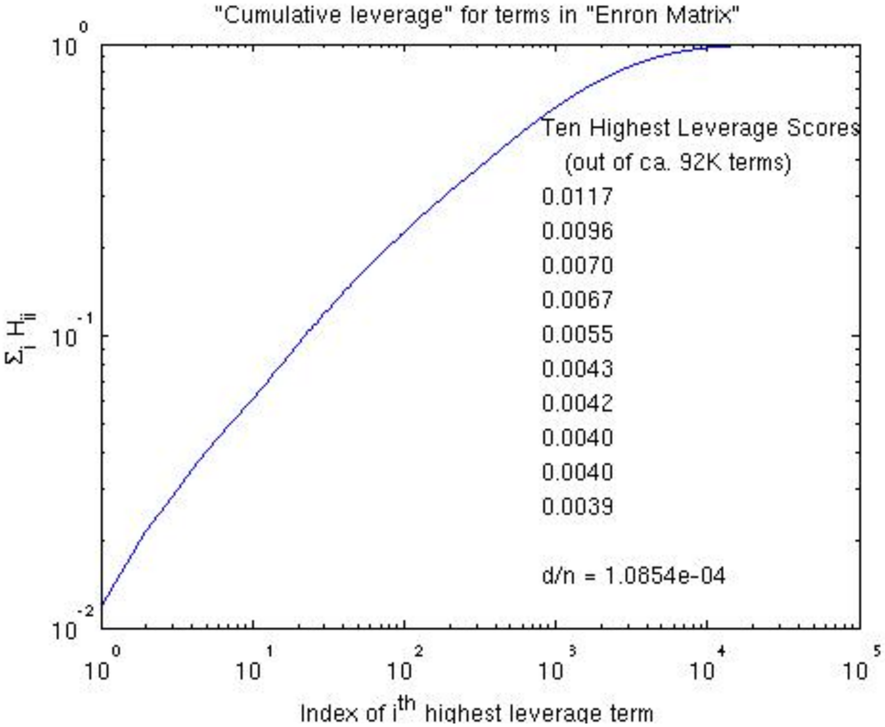
For L1 regression (beyond the FCT):

- Single-pass deterministic conditioning algorithm;
- Single-pass random sampling with map and reduce functions;
- Effective initialization by using multiple subsampled solutions;
- Effective iterative solving with a randomized IPCPM method by perfroming in parallel multiple queries at each iteration.
- Do L1 regression on a tera-byte of data in MapReduce

Leverage Scores of “Real” Data Matrices

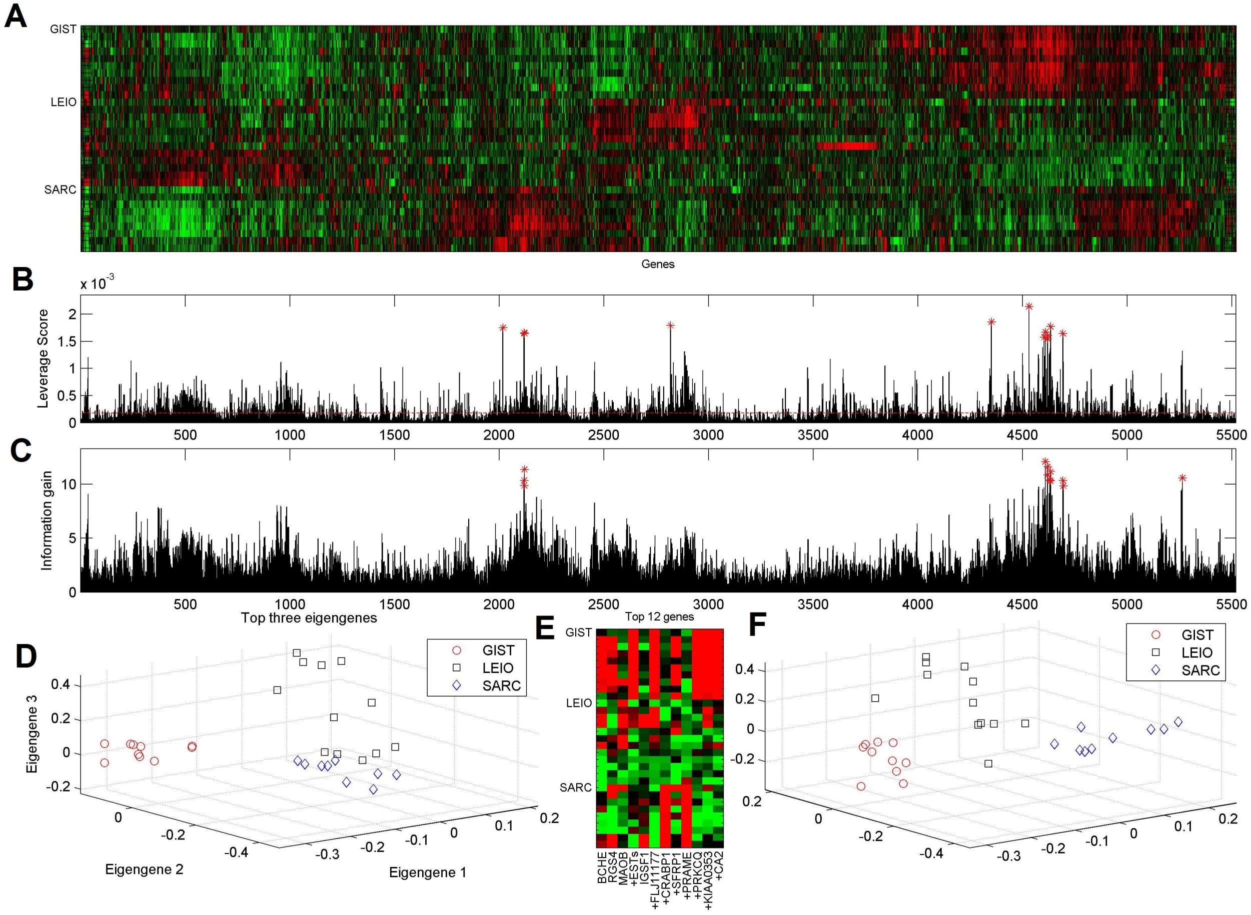


Leverage scores of Zachary karate network edge-incidence matrix.



Cumulative leverage score for the Enron email data matrix.

Leverage Scores and Information Gain



Similar strong correlation between (unsupervised) Leverage Scores and (supervised) Informativeness elsewhere!

A few general thoughts

Q1: Why does a statistical concept like leverage help with worst-case analysis of traditional NLA problems?

- ***A1***: If a data point has high leverage and is *not* an error, as worst-case analysis *implicitly* assumes, it is very important!

Q2: Why are statistical leverage scores so non-uniform in many modern large-scale data analysis applications?

- ***A2***: Statistical models are often *implicitly* assumed for computational and not statistical reasons---many data points “stick out” relative to obviously inappropriate models!

Outline

- “Algorithmic” and “statistical” perspectives on data problems
- Genetics application

  DNA SNP analysis --> choose columns from a matrix

- Internet application

  Community finding --> partitioning a graph

In many large-scale data applications, “algorithmic” and “statistical” perspectives interact in fruitful ways --- *we use statistical properties implicit in worst-case algorithms to make domain-specific claims!*

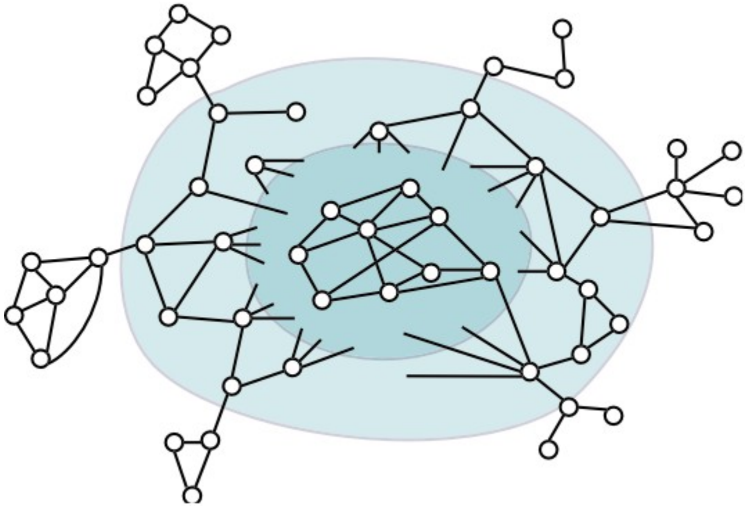
Networks and networked data

Lots of "networked" data!!

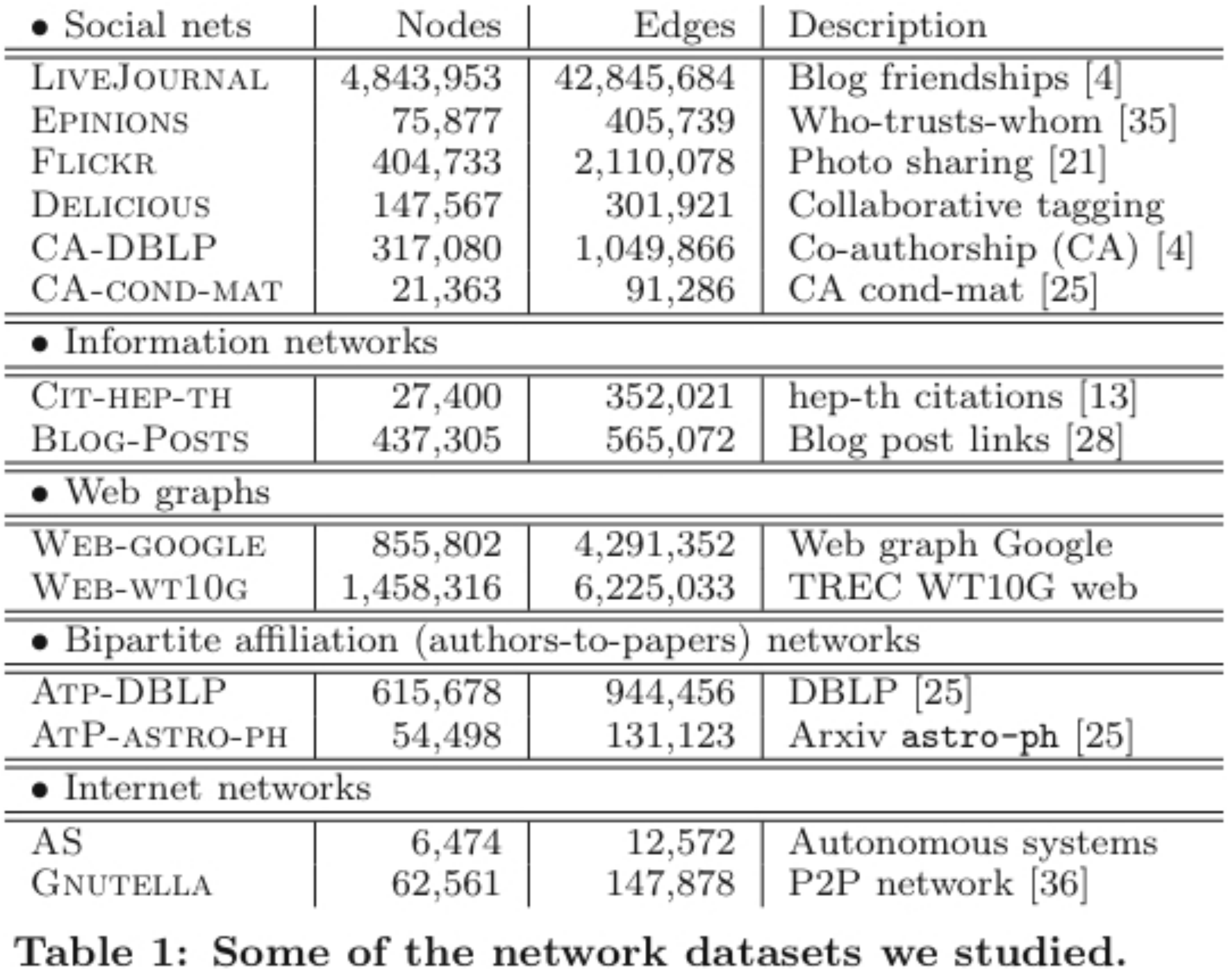
- technological networks
  - AS, power-grid, road networks
- biological networks
  - food-web, protein networks
- social networks
  - collaboration networks, friendships
- information networks
  - co-citation, blog cross-postings, advertiser-bidded phrase graphs...
- language networks
  - semantic networks...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

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]

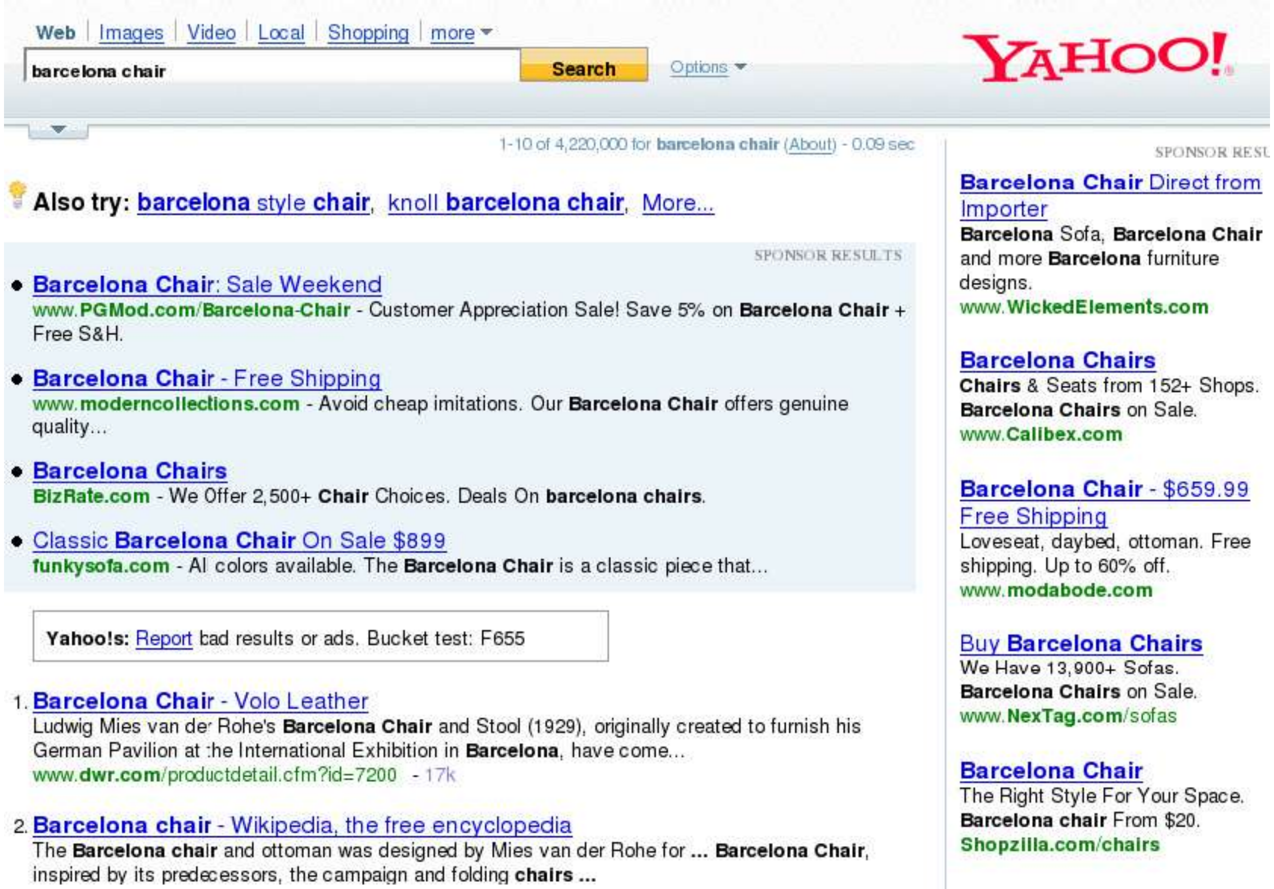
Table 1: Some of the network datasets we studied.

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!



barcelona chair

1-10 of 4,220,000 for **barcelona chair** (About) - 0.09 sec

Also try: **barcelona** style **chair**, knoll **barcelona chair**, More...

SPONSOR RESULTS

- **Barcelona Chair**: Sale Weekend
  www.**PGMod.com/Barcelona-Chair** - Customer Appreciation Sale! Save 5% on **Barcelona Chair** + Free S&H.
- **Barcelona Chair** - Free Shipping
  www.**moderncollections.com** - Avoid cheap imitations. Our **Barcelona Chair** offers genuine quality...
- **Barcelona Chairs**
  BizRate.com - We Offer 2,500+ **Chair** Choices. Deals On **barcelona chairs**.
- Classic **Barcelona Chair** On Sale $899
  funkysofa.com - Al colors available. The **Barcelona Chair** is a classic piece that...

Yahoo!s: Report bad results or ads. Bucket test: F655

1. **Barcelona Chair** - Volo Leather
   Ludwig Mies van der Rohe's **Barcelona Chair** and Stool (1929), originally created to furnish his German Pavilion at the International Exhibition in **Barcelona**, have come...
   www.**dwr.com**/productdetail.cfm?id=7200 - 17k
2. **Barcelona chair** - Wikipedia, the free encyclopedia
   The **Barcelona chair** and ottoman was designed by Mies van der Rohe for **...** **Barcelona Chair**, inspired by its predecessors, the campaign and folding **chairs ...**

SPONSOR RESU

Barcelona Chair Direct from Importer
Barcelona Sofa, **Barcelona Chair** and more **Barcelona** furniture designs.
www.**WickedElements.com**

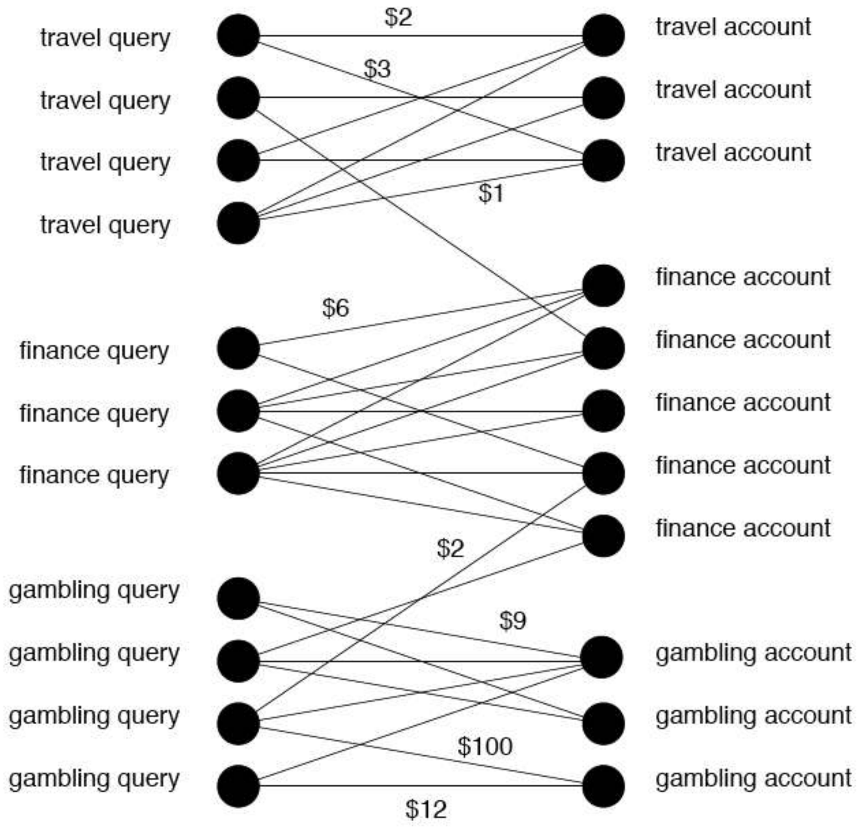
Barcelona Chairs
Chairs & Seats from 152+ Shops. **Barcelona Chairs** on Sale.
www.**Calibex.com**

Barcelona Chair - $659.99 Free Shipping
Loveseat, daybed, ottoman. Free shipping. Up to 60% off.
www.**modabode.com**

Buy **Barcelona Chairs**
We Have 13,900+ Sofas. **Barcelona Chairs** on Sale.
www.**NexTag.com**/sofas

Barcelona Chair
The Right Style For Your Space. **Barcelona chair** From $20.
Shopzilla.com/chairs

Bidding and Spending Graphs



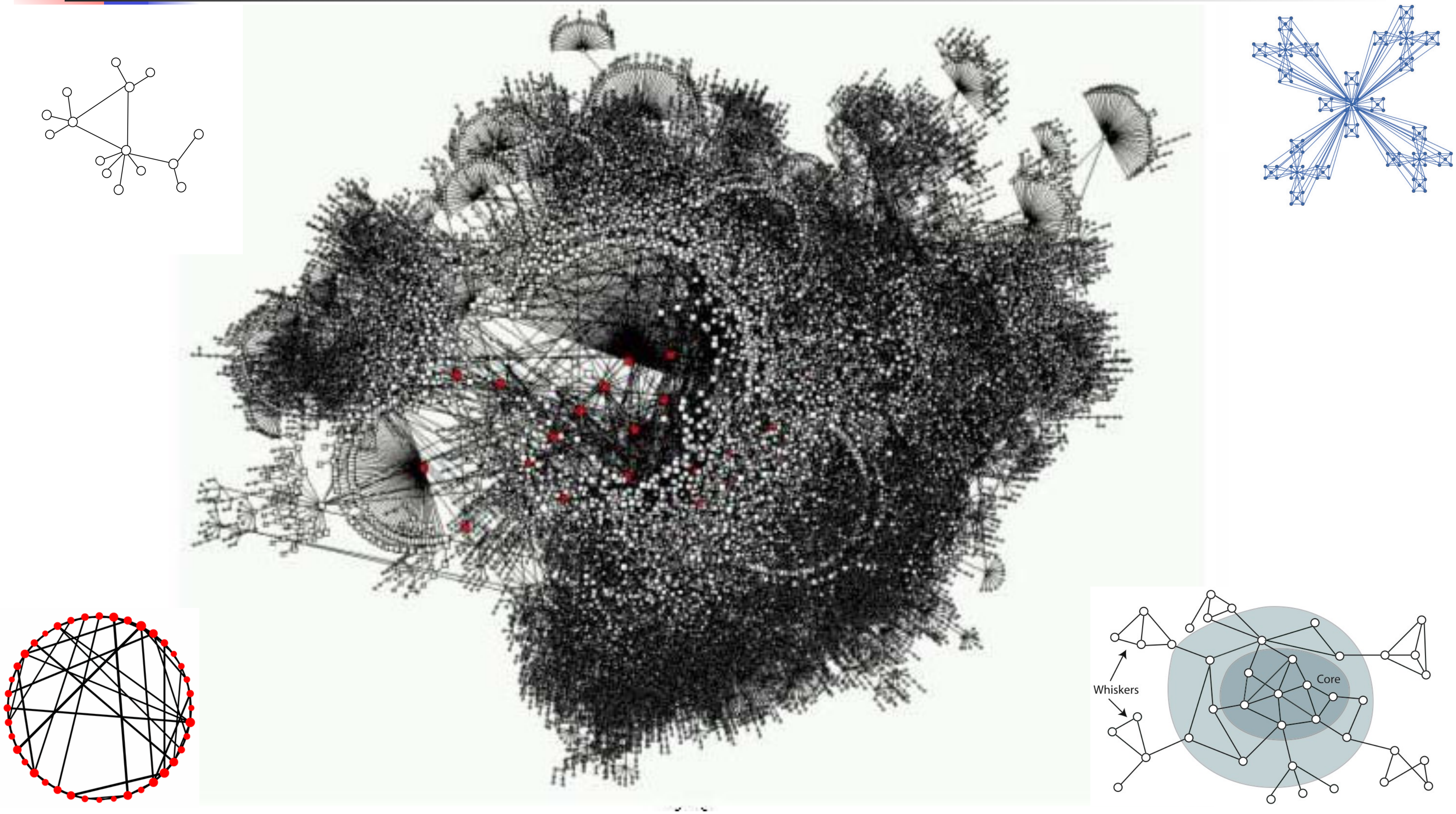
A "social network" with "term-document" aspects.

Uses of Bidding and Spending graphs:

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

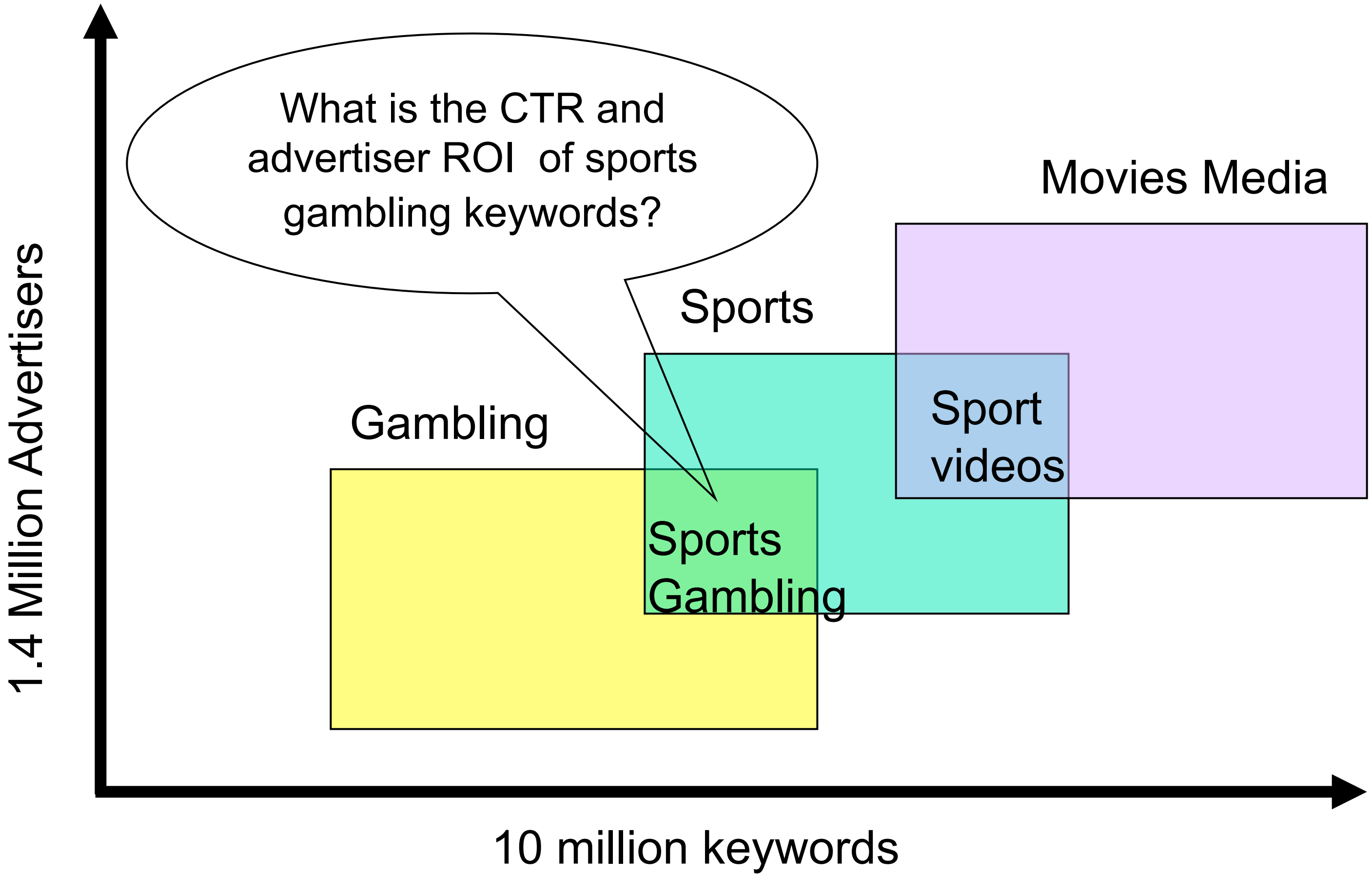
More generally, user segmentation for behavioral targeting.

What do these networks “look” like?

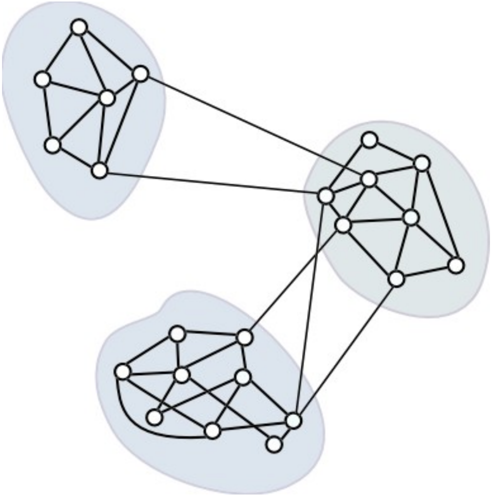


Micro-markets in sponsored search

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



Clustering and Community Finding



- Linear (Low-rank) methods

  If Gaussian, then low-rank space is good.

- Kernel (non-linear) methods

  If low-dimensional manifold, then kernels are good

- Hierarchical methods

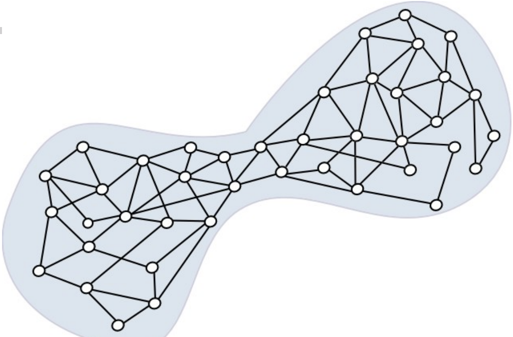
  Top-down and botton-up -- common in the social sciences

- Graph partitioning methods

  Define "edge counting" metric 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."

Communities, Conductance, and NCPPs



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

The conductance ϕ of a set S of nodes is:

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

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

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

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

← A “size-resolved” community-quality measure!

- *Just as conductance captures the “gestalt” notion of cluster/ community quality, the NCP plot measures cluster/community quality as a function of size*.
- NCP plot is intractable to compute exactly
- Use approximation algorithms to approximate it (*even better than exactly*)

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)

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

Approximation algorithms as experimental probes?

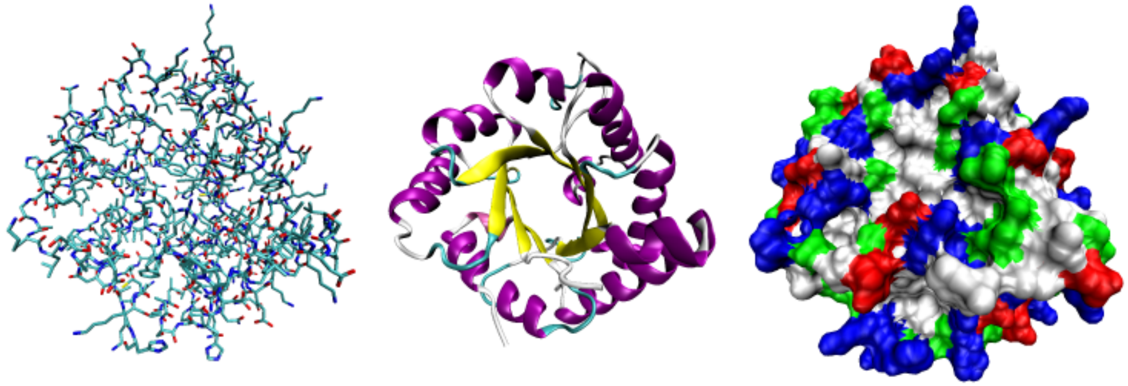
The usual *modus operandi* for approximation algorithms:

- define an objective, the numerical value of which is intractable to compute
- develop approximation algorithm that returns approximation to that number
- graph achieving the approximation may be unrelated to the graph achieving the exact optimum.

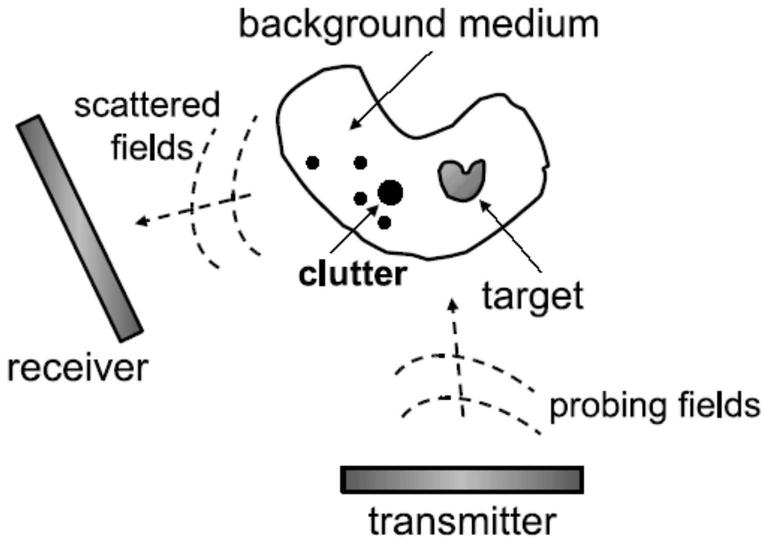
But, for randomized approximation algorithms with a geometric flavor (e.g. matrix algorithms, regression algorithms, eigenvector algorithms; duality algorithms, etc):

- often can approximate the vector achieving the exact solution
- randomized algorithms compute an ensemble of answers -- the details of which depend on choices made by the algorithm
- maybe compare different approximation algorithms for the same problem.

Analogy: What does a protein look like?



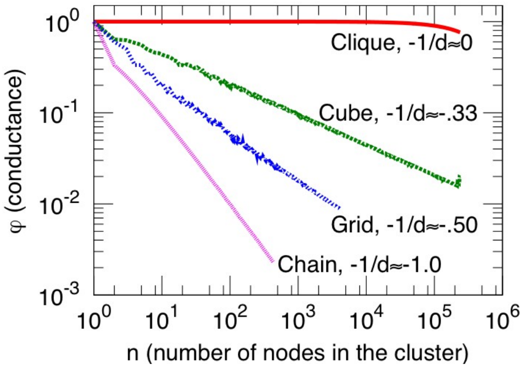
Three possible representations (all-atom; backbone; and solvent-accessible surface) of the three-dimensional structure of the protein triose phosphate isomerase.



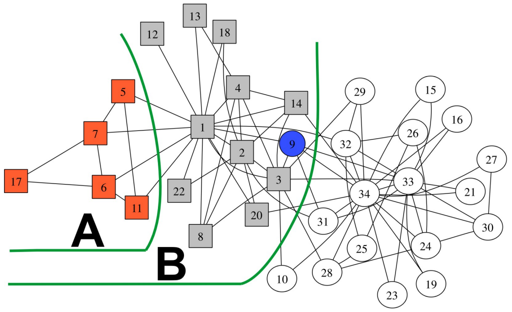
Experimental Procedure:

- 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.

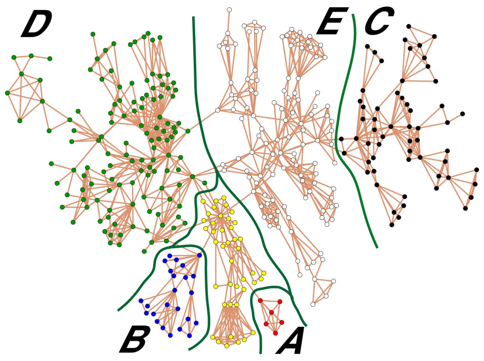
Low-dimensional and small social networks



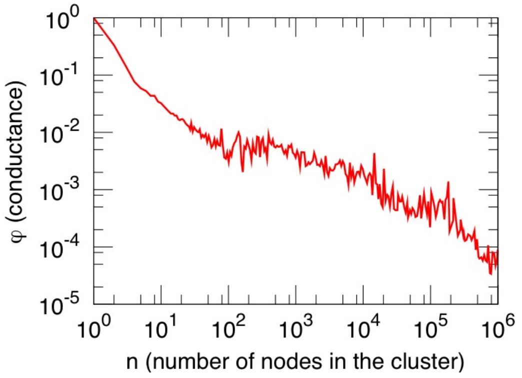
d-dimensional meshes



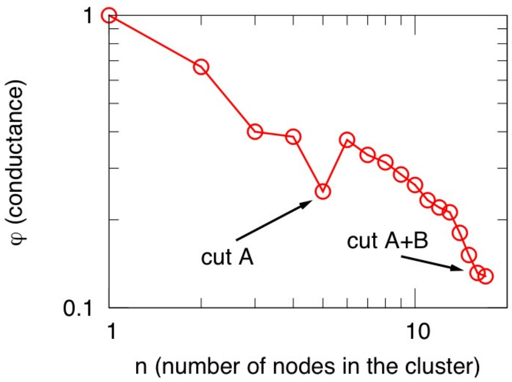
Zachary's karate club

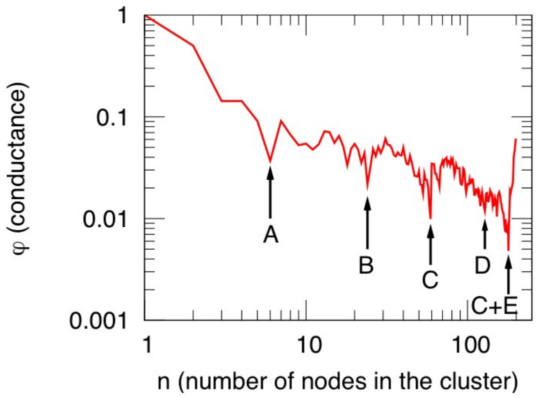


Newman's Network Science

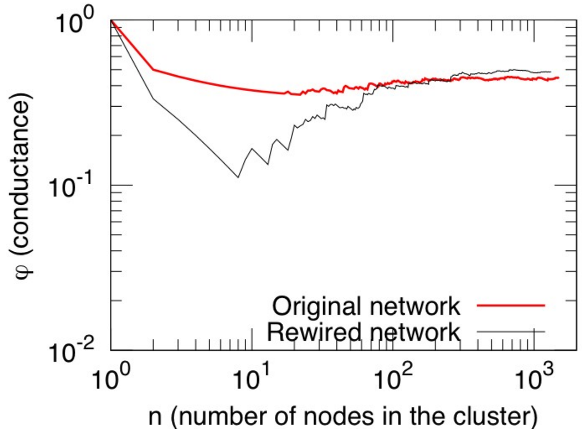


RoadNet-CA

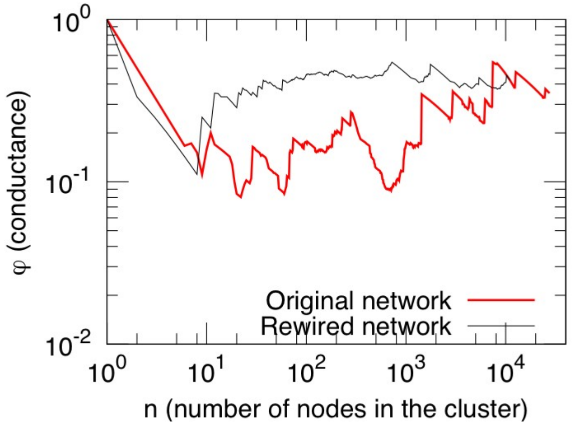




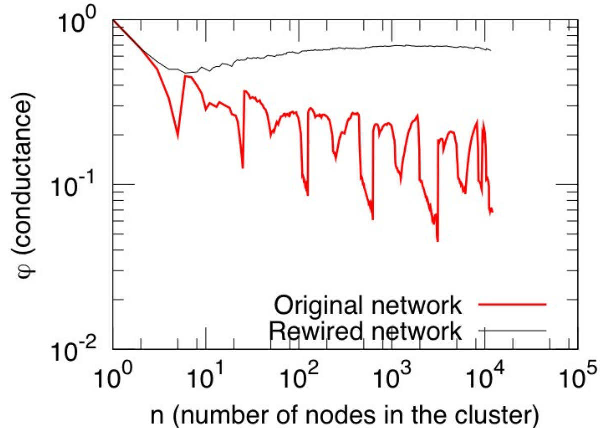
NCP for common generative models



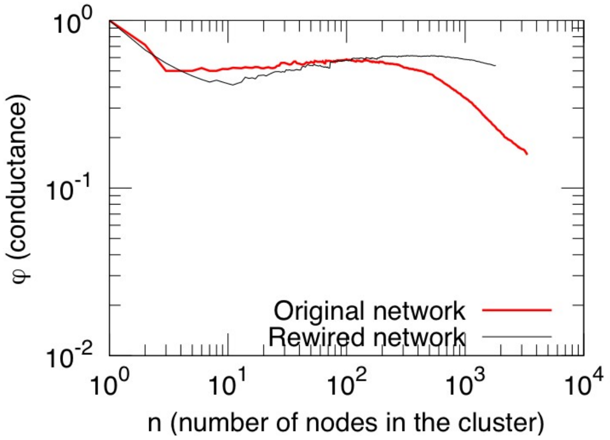
Preferential Attachment



Copying Model

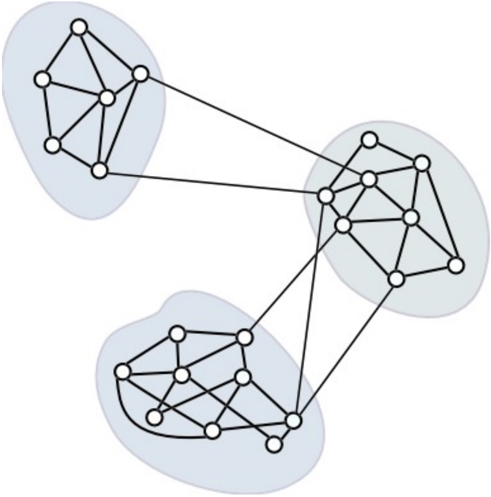


RB Hierarchical



Geometric PA

What do large networks look like?



Downward sloping NCPP

small social networks (validation)

"low-dimensional" networks (intuition)

hierarchical networks (model building)

existing generative models (incl. community models)

Natural interpretation in terms of isoperimetry

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

Large social/information networks are very *very* different

We examined more than 70 large social and information networks

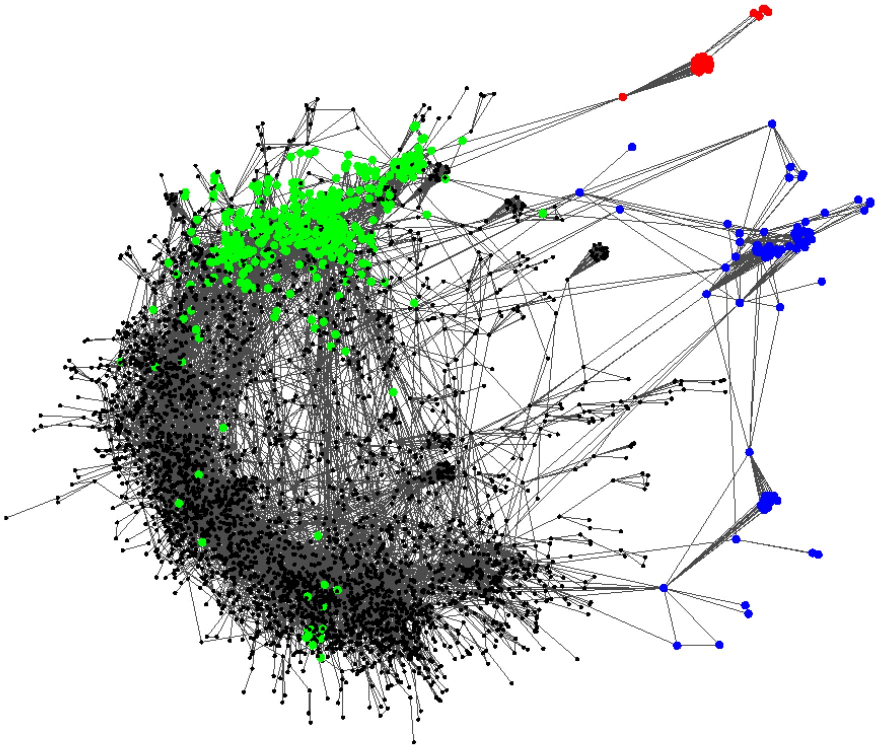
We developed principled methods to interrogate large networks

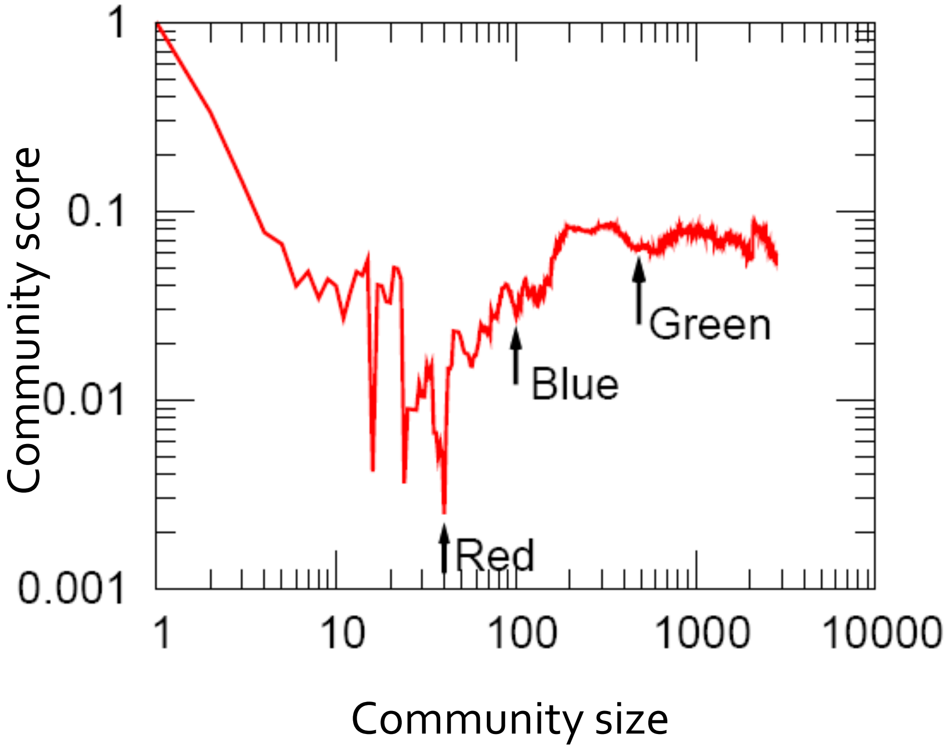
Previous community work: on small social networks (hundreds, thousands)

Typical example of our findings

Leskovec, Lang, Dasgupta, and Mahoney (WWW 2008 & arXiv 2008 & WWW 2010)

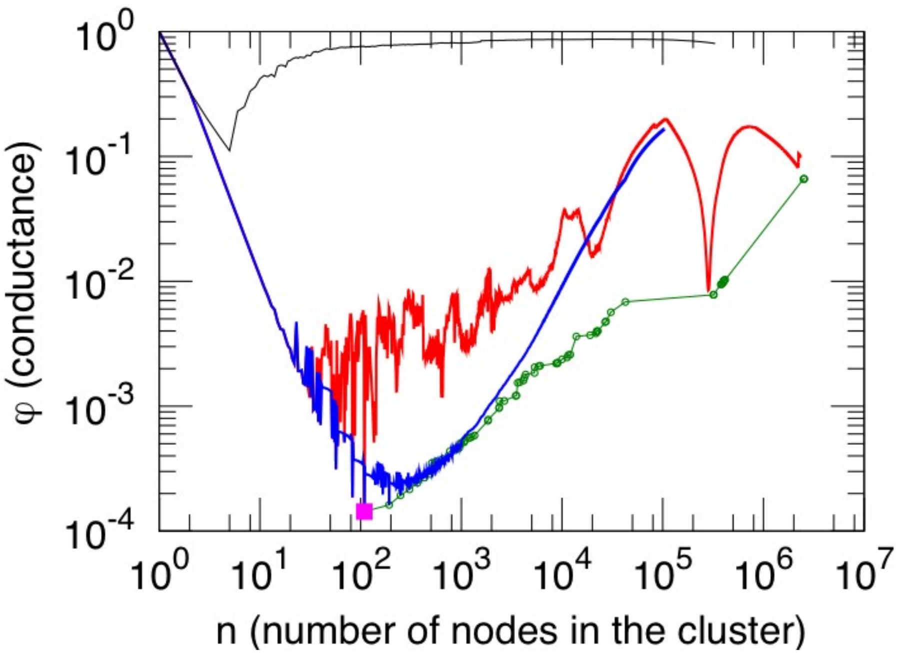
General relativity collaboration network
(4,158 nodes, 13,422 edges)



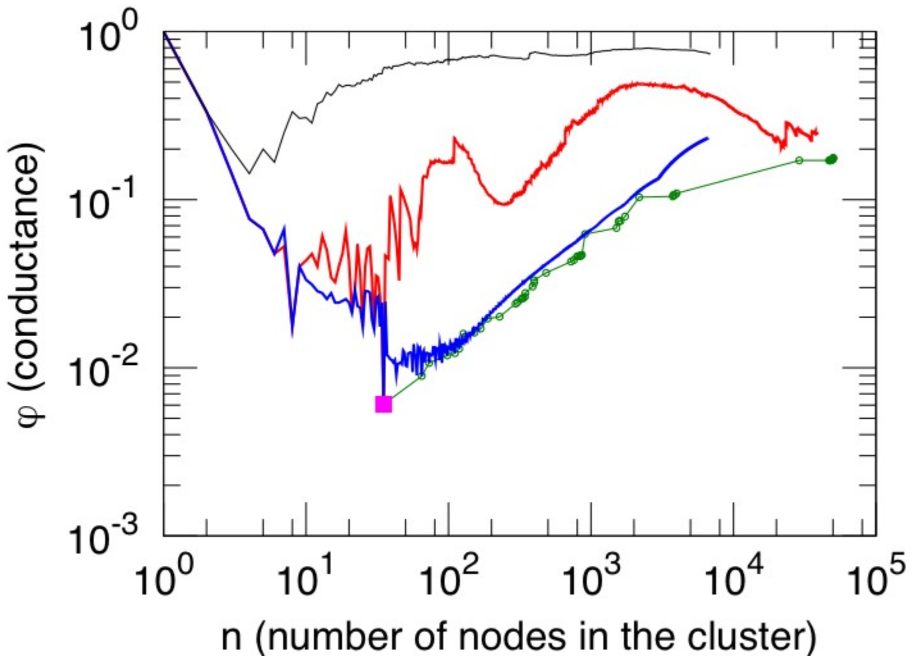


Large Social and Information Networks

Leskovec, Lang, Dasgupta, and Mahoney (WWW 2008 & arXiv 2008 & WWW 2010)



LiveJournal

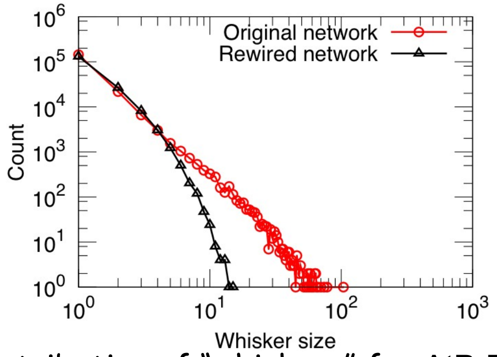


Epinions

Focus on the red curves (local spectral algorithm) - blue (Metis+Flow), green (Bag of whiskers), and black (randomly rewired network) for consistency and cross-validation.

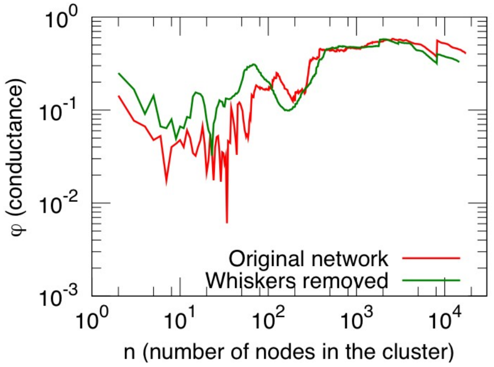
“Whiskers” and the “core”

- Whiskers
  - maximal sub-graph detached from network by removing a single edge
  - Contain (on average) 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

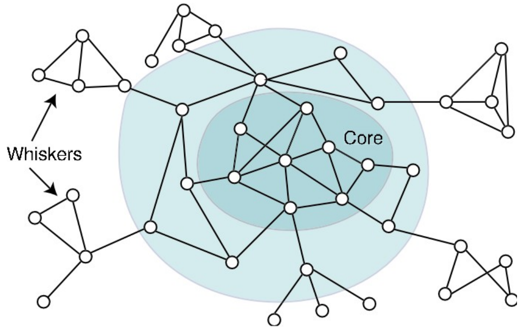


Distribution of “whiskers” for AtP-DBLP.

If remove whiskers, then the lowest conductance sets (the “best” communities) are “2-whiskers”:



Epinions



How do we know this plot it "correct"?

- Lower Bound Result

  Spectral and SDP lower bounds for large partitions

- Modeling Result

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

- Structural Result

  Small barely-connected "whiskers" responsible for minimum

- Algorithmic Result

  Ensemble of sets returned by different algorithms are very different

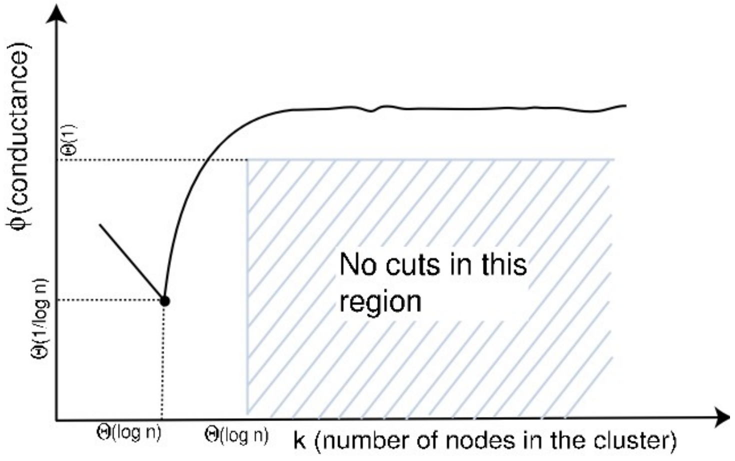
  Spectral vs. flow vs. bag-of-whiskers heuristic

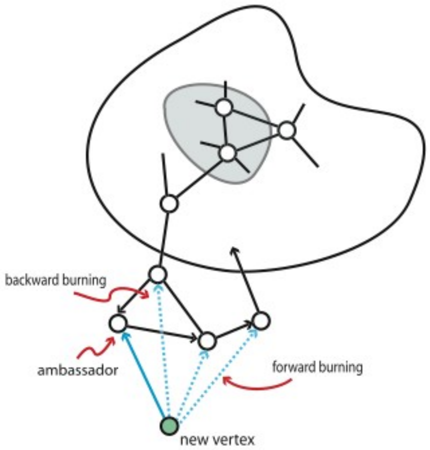
  Spectral method implicitly regularizes, gets more meaningful communities

Random graphs and forest fires

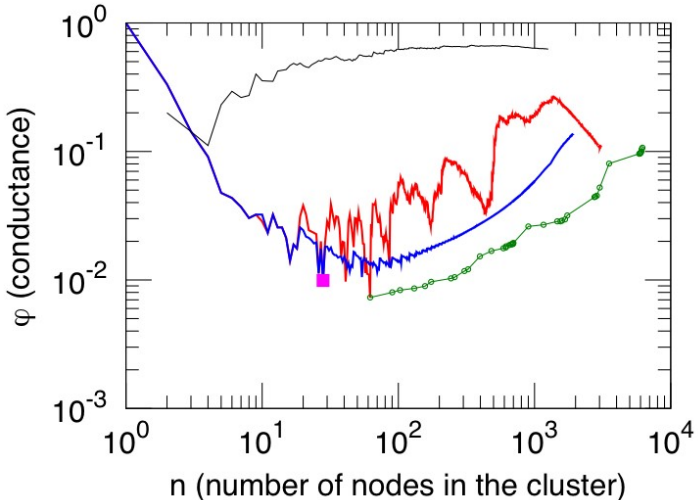
Let $\mathbf{w} = (w_1, \ldots, w_n)$, where
$w_i = c i^{-1/(\beta-1)}, \quad \beta \in (2,3)$.
Connect nodes i and j w.p.
$p_{ij} = w_i w_j / \sum_k w_k$.

A "power law random graph" model (Chung-Lu)

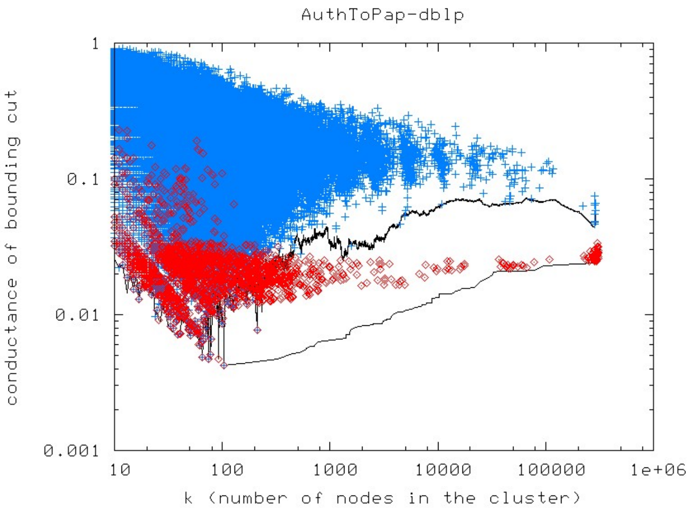


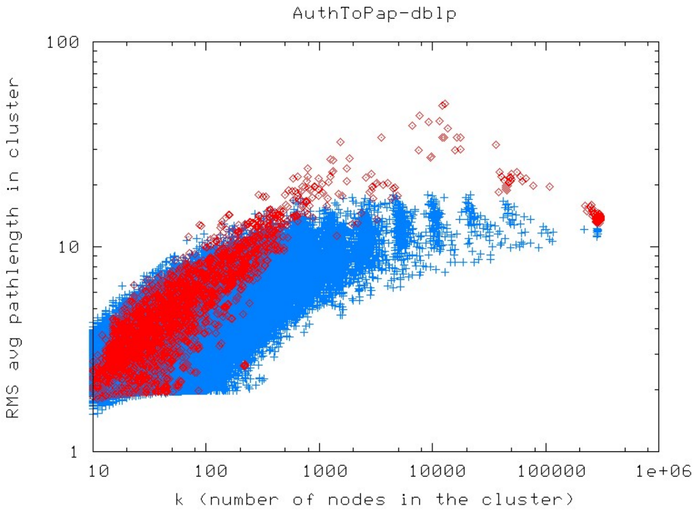


A "forest fire" model (LKF05)

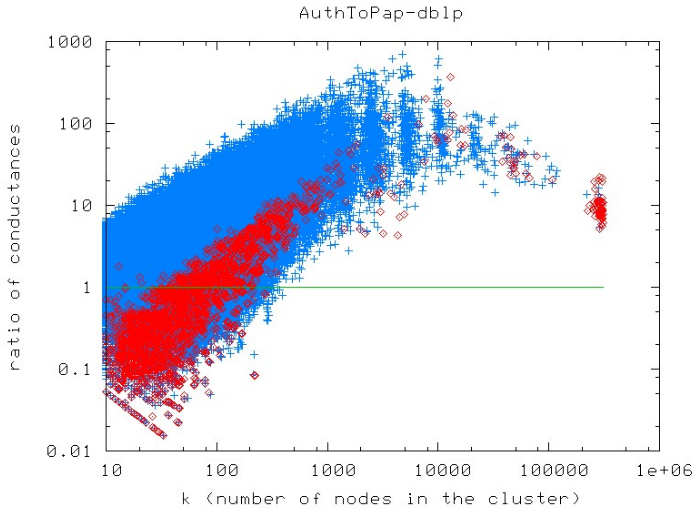


Regularized and non-regularized communities (1 of 2)



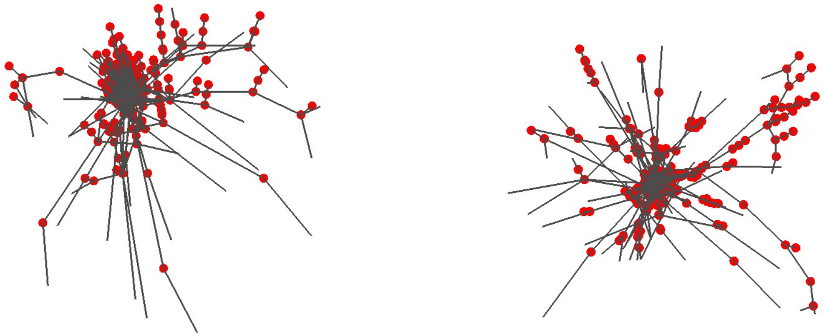


- Metis+MQI (red) gives sets with better conductance.
- Local Spectral (blue) gives tighter and more well-rounded sets.

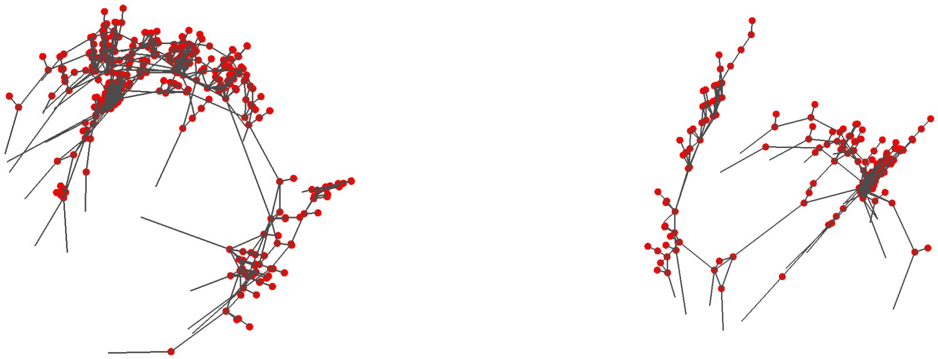


Regularized and non-regularized communities (2 of 2)

Two ca. 500 node communities from Local Spectral Algorithm:



Two ca. 500 node communities from Metis+MQI:



A few general thoughts

Regularization is typically *implemented* by adding a norm constraint

- makes the problem harder (think L1-regularized L2-regression).

Approximation algorithms for intractable graph problems *implicitly* regularize

- relative to combinatorial optimum
- incorporate empirical signatures of bias-variance tradeoff.

Use statistical properties *implicit* in worst-case algorithms to provide insights into informatics graphs

- good since networks are large, sparse, and noisy.

A “claimer” and a “disclaimer”:

- Today, mostly took a “10,000 foot” view:
  - But, “drilled down” on two specific examples that illustrate “algorithmic-statistical” interplay in a novel way

- Mostly avoided* “rubber-hits-the-road” issues:
  - Multi-core and multi-processor issues
  - Map-Reduce and distributed computing
  - Other large-scale implementation issues

*But, these issues are *very* much a motivation and “behind-the-scenes” and important looking forward!

Conclusions to Part One

- "Algorithmic" and "statistical" perspectives on data problems
- Genetics application

  DNA SNP analysis --> choose columns from a matrix
- Internet application

  Community finding --> partitioning a graph

In many large-scale data applications, "algorithmic" and "statistical" perspectives interact in fruitful ways.

In Two Parts

Part One: Algorithmic and Statistical Perspectives on Large-scale Data Analysis:

• Describes these two approaches with two “anecdotes” from genetics and internet advertising applications

• Preprint: arXiv:1010.1609 (2010); In: Combinatorial Scientific Computing, pp. 427-469, edited by U. Naumann and O. Schenk, 2012

Part Two: Approximate Computation and Implicit Regularization in Large-scale Data Analysis:

• Describes regularization, the concept at the heart of this difference, in traditional and novel contexts

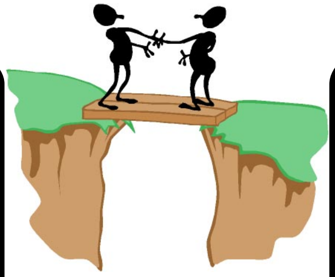
• Preprint: arXiv:1203.0786 (2012);Proc. of the 2012 ACM Symposium on Principles of Database Systems, 143-154, 2012

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



Practical applications

- NLA, ML, statistics, data analysis, genetics, etc
- Fast JL transform
- Relative-error algs
- Numerically-stable algs
- Good statistical properties

How to "bridge the gap"?

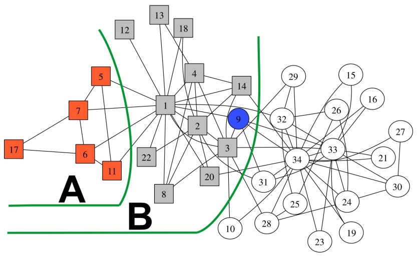
- decouple randomization from linear algebra
- importance of statistical leverage scores!

Anecdote 2:
Communities in large informatics graphs

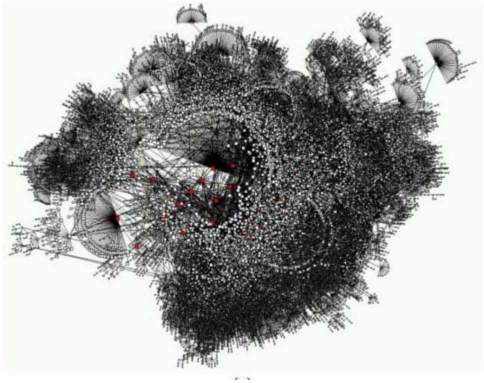
Mahoney "Algorithmic and Statistical Perspectives on Large-Scale Data Analysis" (2010)
Leskovec, Lang, Dasgupta, & Mahoney "Community Structure in Large Networks ..." (2009)

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

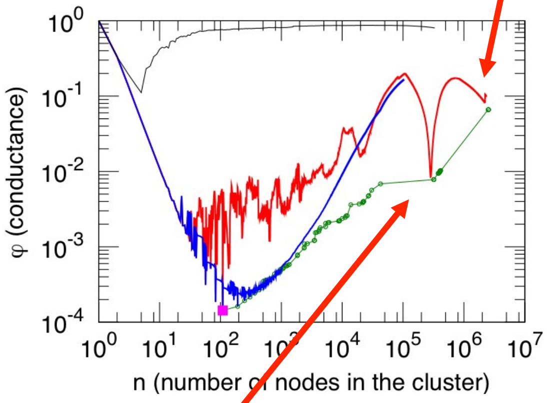
People imagine social networks to look like:



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

Lessons from the anecdotes

Mahoney "Algorithmic and Statistical Perspectives on Large-Scale Data Analysis" (2010)

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 for Part Two

Preamble: algorithmic & statistical perspectives

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

Approximate first nontrivial eigenvector of Laplacian

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

Spectral versus flow-based algs for graph partitioning

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

Weakly-local and strongly-local graph partitioning methods

- Operationally like L1-regularization and already used in practice!

Outline and overview

Preamble: algorithmic & statistical perspectives

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

Approximate first nontrivial eigenvector of Laplacian

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

Spectral versus flow-based algs for graph partitioning

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

Weakly-local and strongly-local graph partitioning methods

- Operationally like L1-regularization and already used in practice!

Thoughts on models of data (1 of 2)

Data are whatever data are

- records of banking/financial transactions, hyperspectral medical/astronomical images, electromagnetic signals in remote sensing applications, DNA microarray/SNP measurements, term-document data, search engine query/click logs, user interactions on social networks, corpora of images, sounds, videos, etc.

To do something useful, you must *model the data*

Two criteria when choosing a data model

- (data acquisition/generation side): want a structure that is "close enough" to the data that you don't do too much "damage" to the data
- (downstream/analysis side): want a structure that is at a "sweet spot" between descriptive flexibility and algorithmic tractability

Thoughts on models of data (2 of 2)

Examples of data models:

- *Flat tables and the relational model:* one or more two-dimensional arrays of data elements, where different arrays can be related by predicate logic and set theory.

- *Graphs, including trees and expanders:* G=(V,E), with a set of nodes V that represent "entities" and edges E that represent "interactions" between pairs of entities.

- *Matrices, including SPSD matrices:* m "objects," each of which is described by n "features," i.e., an n-dimensional Euclidean vector, gives an m x n matrix A.

Much modern data are relatively-unstructured; matrices and graphs are often useful, especially when traditional databases have problems.

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

Before the digital computer:

- Natural sciences rich source of problems, statistical methods developed 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” 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: Δy=y*-y
- Backward error: smallest Δx s.t. f(x+Δx) = 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 procedures 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:

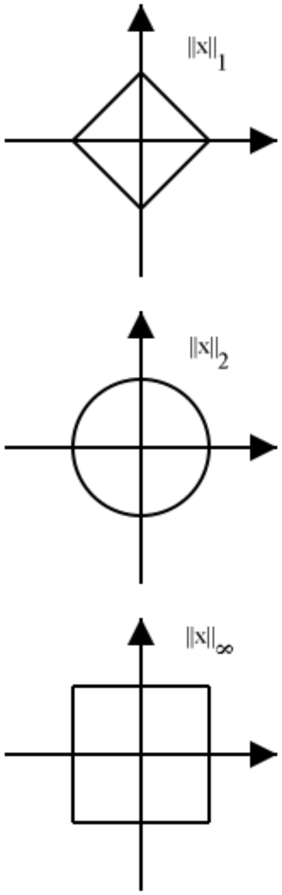
- add a norm constraint (or “geometric capacity control function”) $g(x)$ to objective function $f(x)$

- solve the modified optimization problem

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

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

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



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

Spectral versus flow-based algs for graph partitioning

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

Weakly-local and strongly-local graph partitioning methods

- Operationally like L1-regularization and already used in practice!

Notation for weighted undirected graph

- vertex set $V = \{1, \ldots, n\}$
- edge set $E \subset V \times V$
- edge weight function $w : E \rightarrow R_+$
- degree function $d : V \rightarrow R_+$, $d(u) = \sum_v w(u, v)$
- 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 an SPSD (e.g., Laplacian) matrix A,

- Power method starts with v_0, and iteratively computes

$$v_{t+1} = Av_t / ||Av_t||_2 \quad .$$

- Then, $v_t = \Sigma_i \gamma_i^t v_i \to v_1 \quad .$

- If we truncate after (say) 3 or 10 iterations, still have some mixing from other eigen-directions

What objective does the exact eigenvector optimize?

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

- But can also express this as an SDP, for a SPSD *matrix* X.

- (We will put regularization on this SDP!)

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:

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

Question: 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: ↔ **SDP:**

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

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

↓ ↓

R-VP:

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

R-SDP:

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

A simple theorem

Mahoney and Orecchia (2010)

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

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

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

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

Three simple corollaries

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

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

$F_D(X) = -\mathrm{logdet}(X)$ (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 $\lambda \sim \eta$

(F(•) specifies the algorithm; "number of steps" specifies the η)

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

A Statistical interpretation of this result (& framework for regularized graph estimation)

Perry and Mahoney (2011)

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

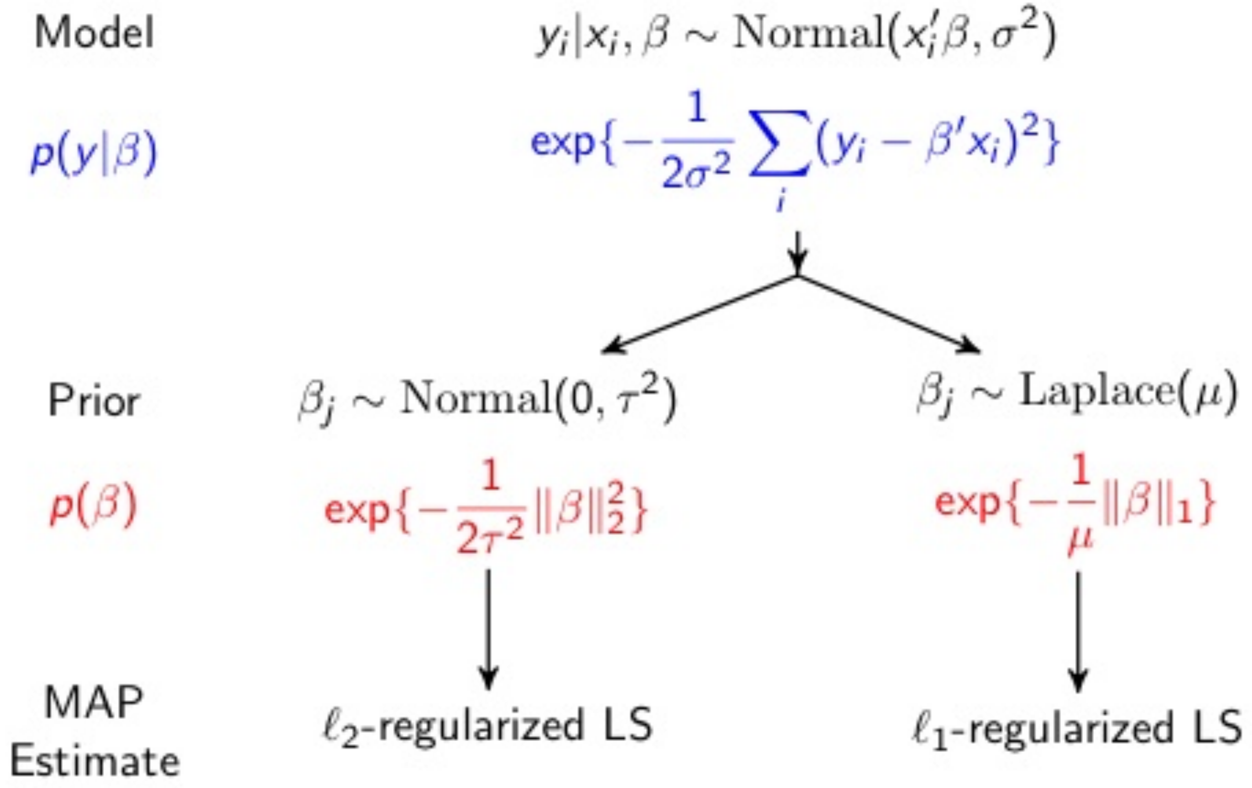
- Issue 1: Best to think of the graph (e.g., Web graph) as a single data point, so what is the "ensemble"?

- 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



Model $y_i|x_i,\beta \sim \mathrm{Normal}(x_i'\beta,\sigma^2)$

$p(y|\beta)$ $\exp\{-\frac{1}{2\sigma^2}\sum_i (y_i-\beta' x_i)^2\}$

Prior $\beta_j \sim \mathrm{Normal}(0,\tau^2)$ $\beta_j \sim \mathrm{Laplace}(\mu)$

$p(\beta)$ $\exp\{-\frac{1}{2\tau^2}\|\beta\|_2^2\}$ $\exp\{-\frac{1}{\mu}\|\beta\|_1\}$

MAP Estimate ℓ_2-regularized LS ℓ_1-regularized LS

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)

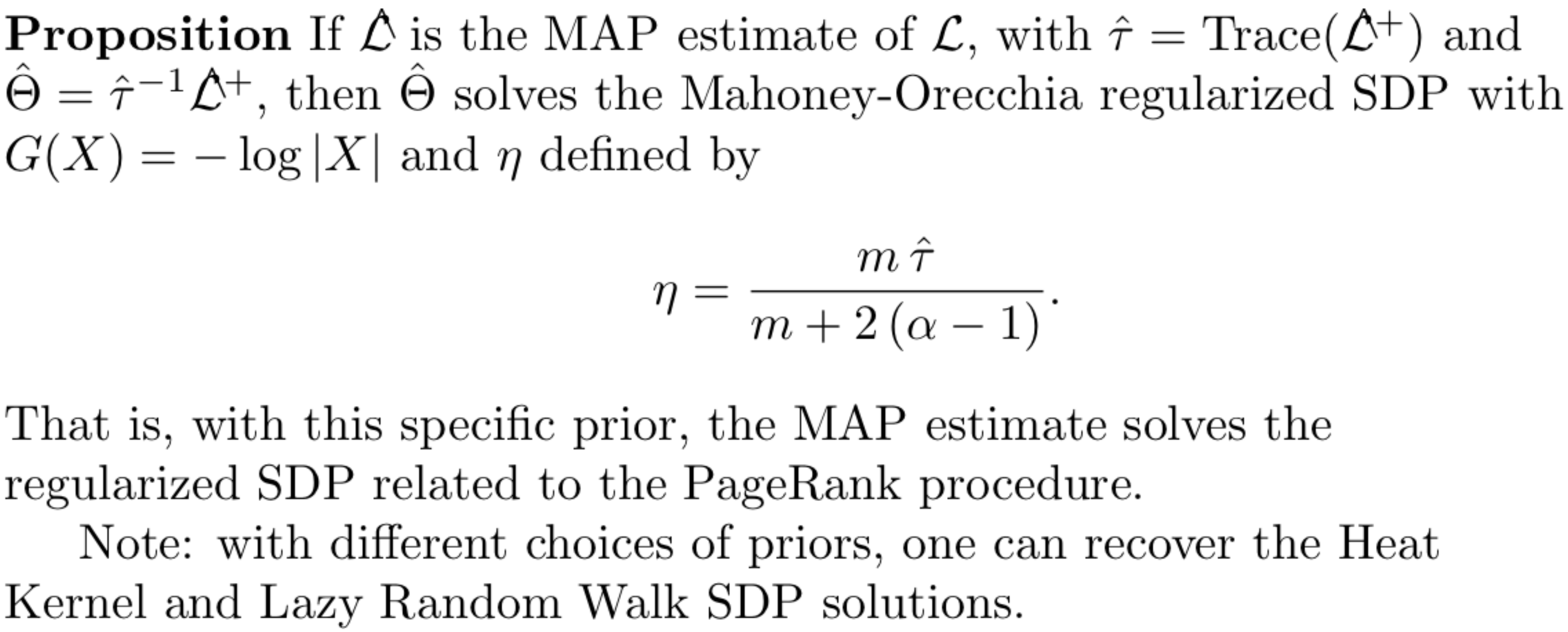
Let $\mathcal{L}^+ = \tau O \Lambda O'$ be the spectral decomposition of $\mathcal{L}^+$, where $\tau = \mathrm{Trace}(\mathcal{L}^+) \geq 0$ is a scale factor, $O \in R^{n \times n-1}$ is an orthogonal matrix, and $\Lambda = \mathrm{diag}\big(\lambda(1), \ldots, \lambda(n-1)\big)$, where $\sum_v \lambda(v)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

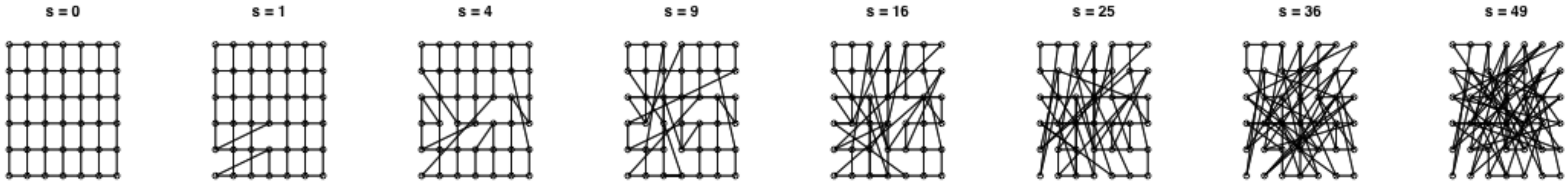
$$\eta = \frac{m\,\hat{\tau}}{m + 2\,(\alpha - 1)}.$$

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

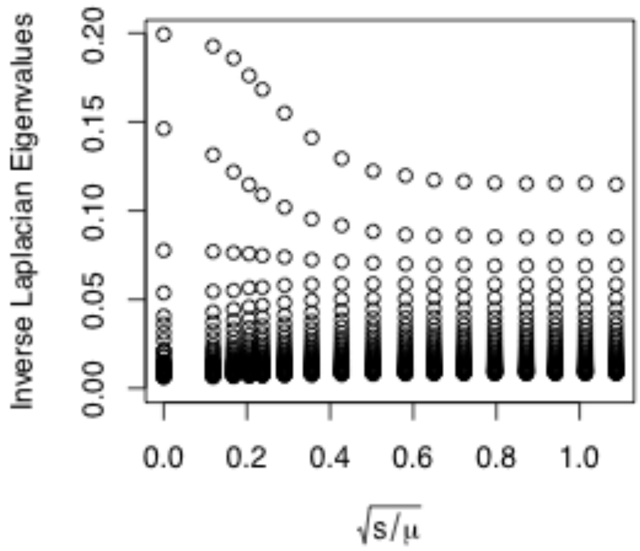
Generate a population Laplacian $\mathcal{L}$ by performing s edge swaps starting from a 2-dimensional grid with n nodes and μ edges.



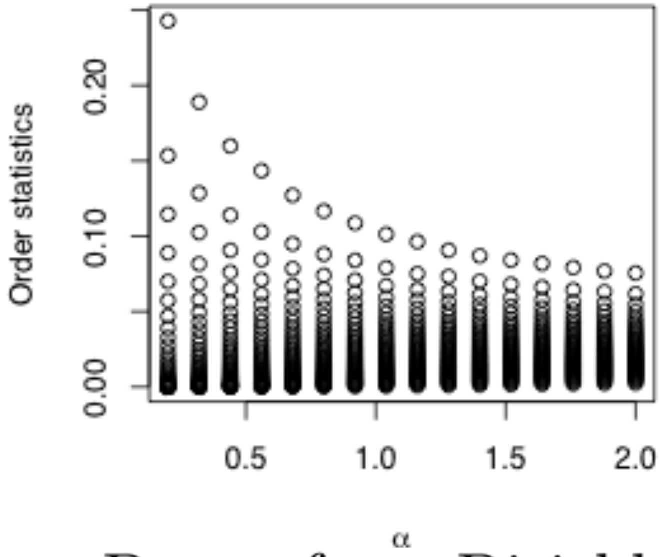
When $s = 0$ the population graph with Laplacian $\mathcal{L}$ is a low-dimensional grid; as $s \to \infty$, it becomes an expander-like random graph.

The prior vs. the simulation procedure

Perry and Mahoney (2011)



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

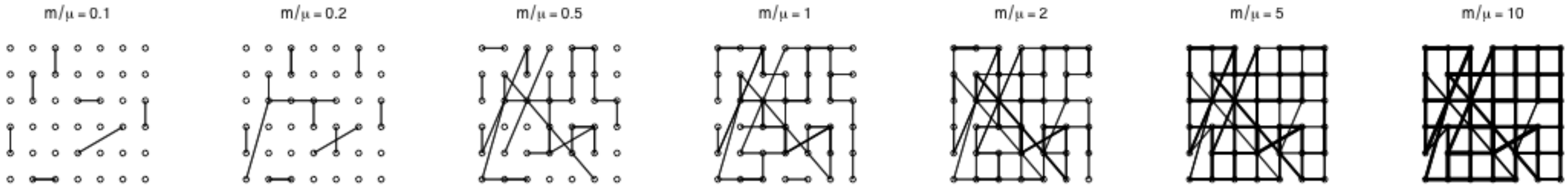


Draws from Dirichlet(α)

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

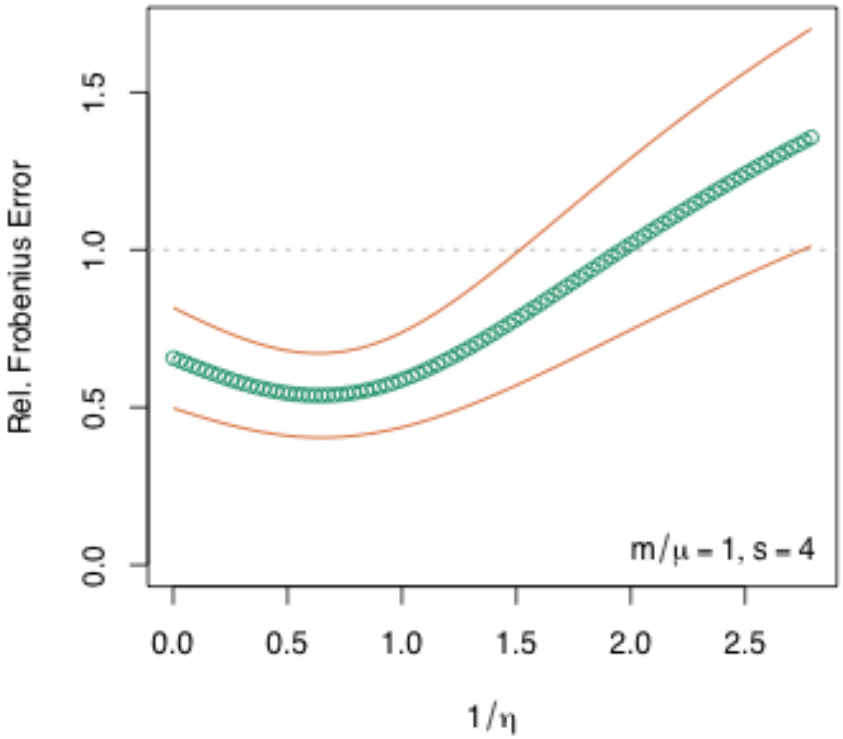
Two estimators for $\mathcal{L}$:

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

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

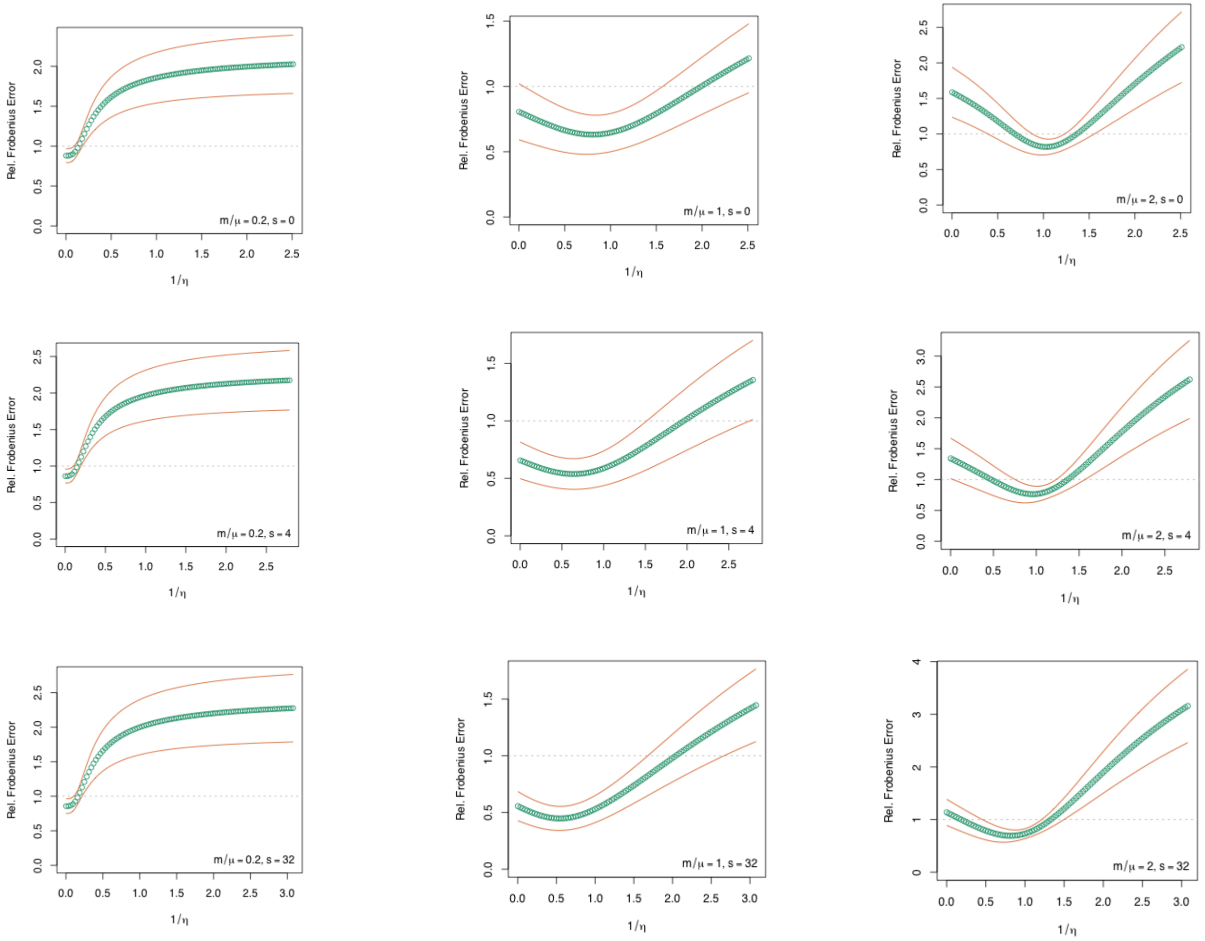
Empirical results (1 of 3)

Perry and Mahoney (2011)



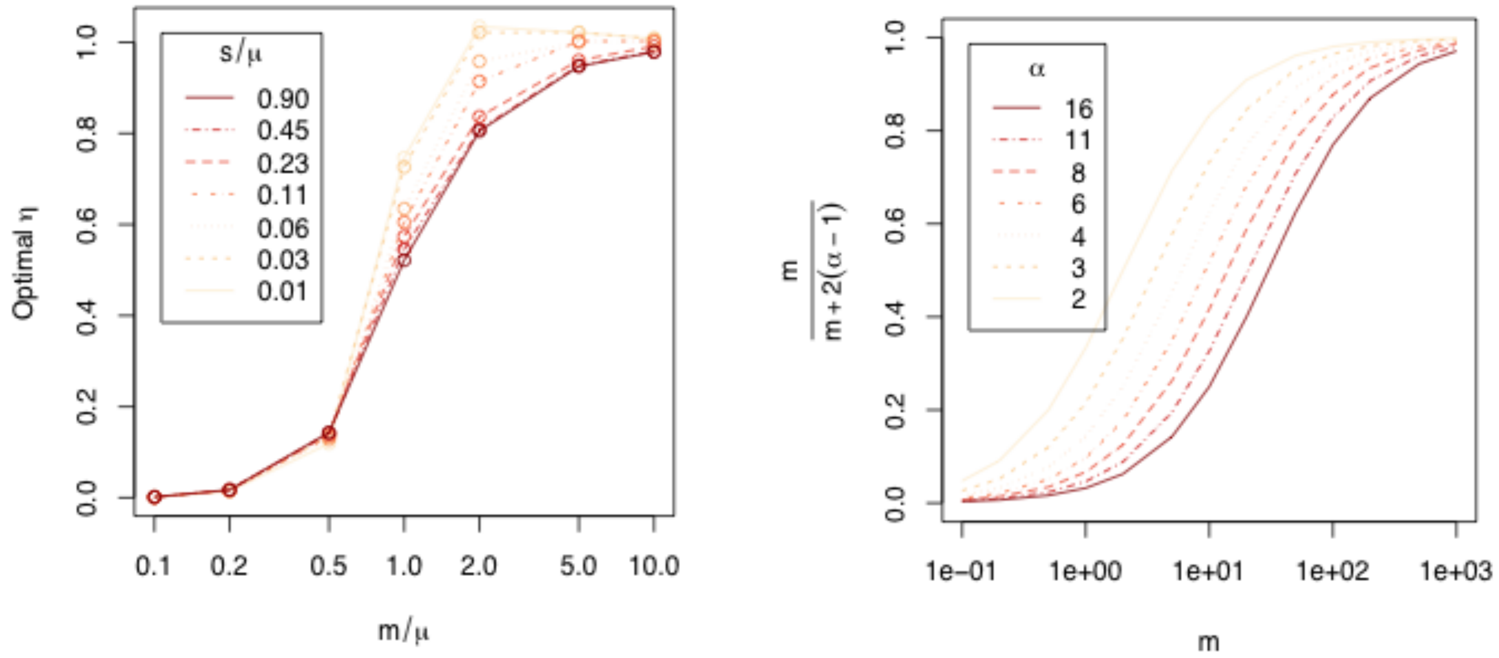
For certain values of η, regularized estimate $\hat{\mathcal{L}}_\eta$ outperforms unregularized estimate $\hat{\mathcal{L}}$, i.e. $\|\Theta - \hat{\Theta}_\eta\|_{\mathrm{F}} / \|\Theta - \hat{\Theta}\|_{\mathrm{F}} < 1$; and similarly for spectral norm error.

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

Spectral versus flow-based algs for graph partitioning

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

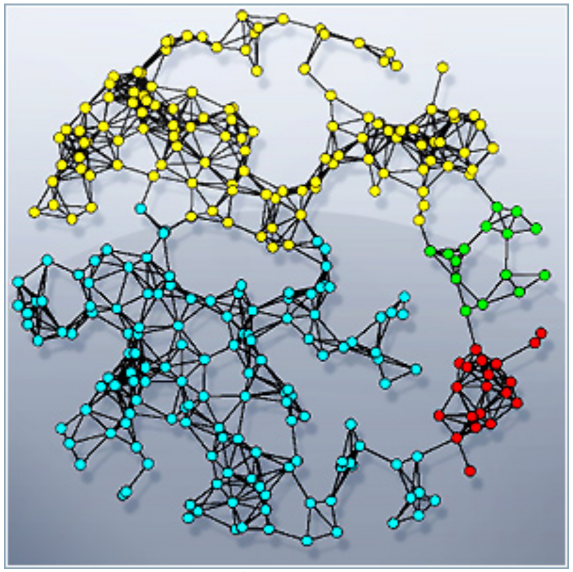
Weakly-local and strongly-local graph partitioning methods

- Operationally like L1-regularization and already used in practice!

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)) (conductance or N-Cuts)

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

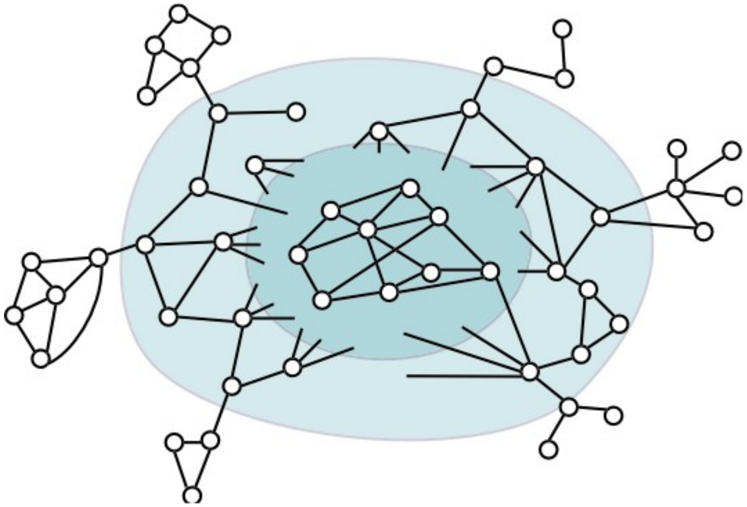
Networks and networked data

Lots of "networked" data!!

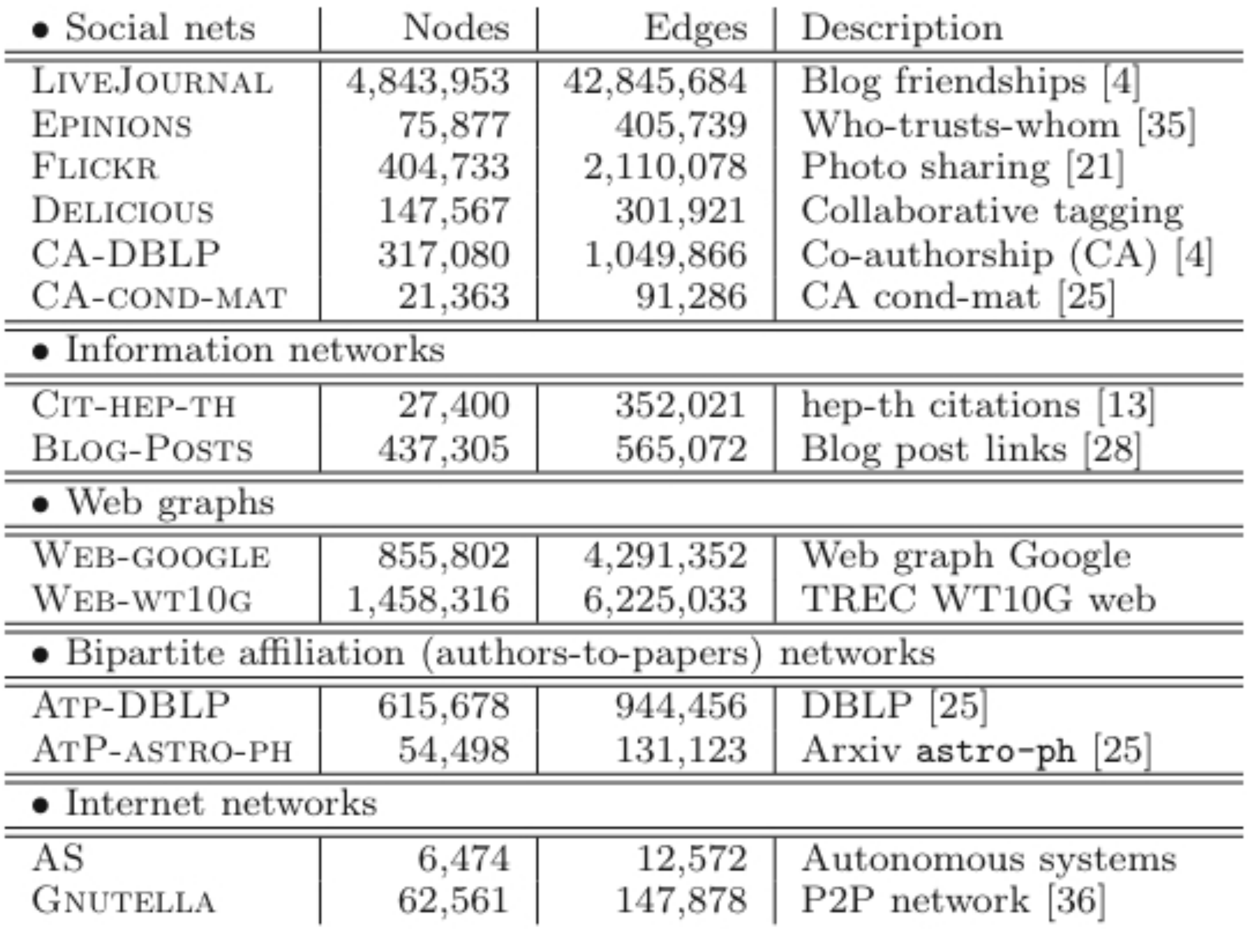
- technological networks
  - AS, power-grid, road networks
- biological networks
  - food-web, protein networks
- social networks
  - collaboration networks, friendships
- information networks
  - co-citation, blog cross-postings, advertiser-bidded phrase graphs...
- language networks
  - semantic networks...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

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]

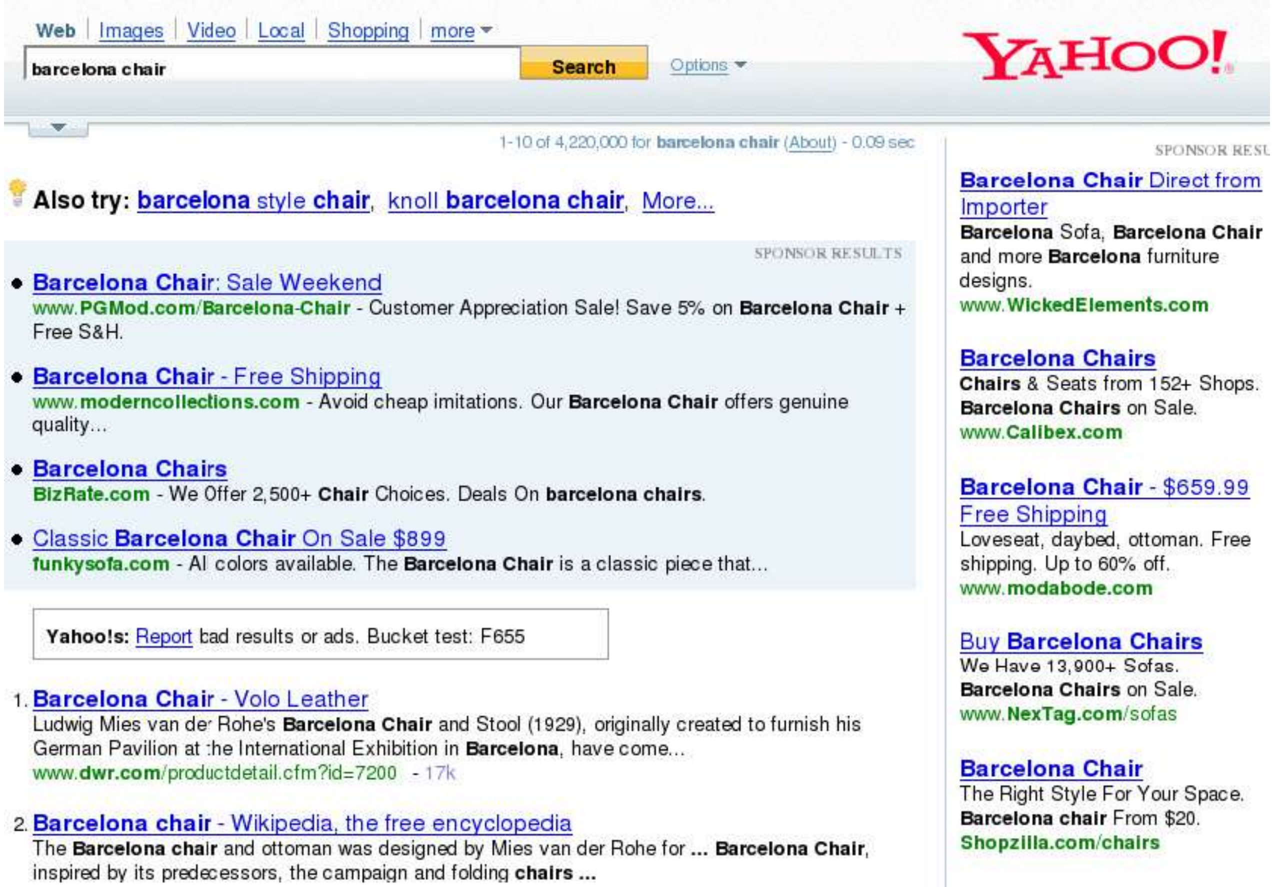
Table 1: Some of the network datasets we studied.

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!



barcelona chair

1-10 of 4,220,000 for **barcelona chair** (About) - 0.09 sec

Also try: **barcelona** style **chair**, knoll **barcelona chair**, More...

SPONSOR RESULTS

- **Barcelona Chair**: Sale Weekend
  www.**PGMod.com/Barcelona-Chair** - Customer Appreciation Sale! Save 5% on **Barcelona Chair** + Free S&H.
- **Barcelona Chair** - Free Shipping
  www.**moderncollections.com** - Avoid cheap imitations. Our **Barcelona Chair** offers genuine quality...
- **Barcelona Chairs**
  BizRate.com - We Offer 2,500+ **Chair** Choices. Deals On **barcelona chairs**.
- Classic **Barcelona Chair** On Sale $899
  funkysofa.com - Al colors available. The **Barcelona Chair** is a classic piece that...

Yahoo!s: Report bad results or ads. Bucket test: F655

1. **Barcelona Chair** - Volo Leather
   Ludwig Mies van der Rohe's **Barcelona Chair** and Stool (1929), originally created to furnish his German Pavilion at the International Exhibition in **Barcelona**, have come...
   www.**dwr.com**/productdetail.cfm?id=7200 - 17k
2. **Barcelona chair** - Wikipedia, the free encyclopedia
   The **Barcelona chair** and ottoman was designed by Mies van der Rohe for ... **Barcelona Chair**, inspired by its predecessors, the campaign and folding **chairs** ...

SPONSOR RESU

Barcelona Chair Direct from Importer
Barcelona Sofa, **Barcelona Chair** and more **Barcelona** furniture designs.
www.**WickedElements.com**

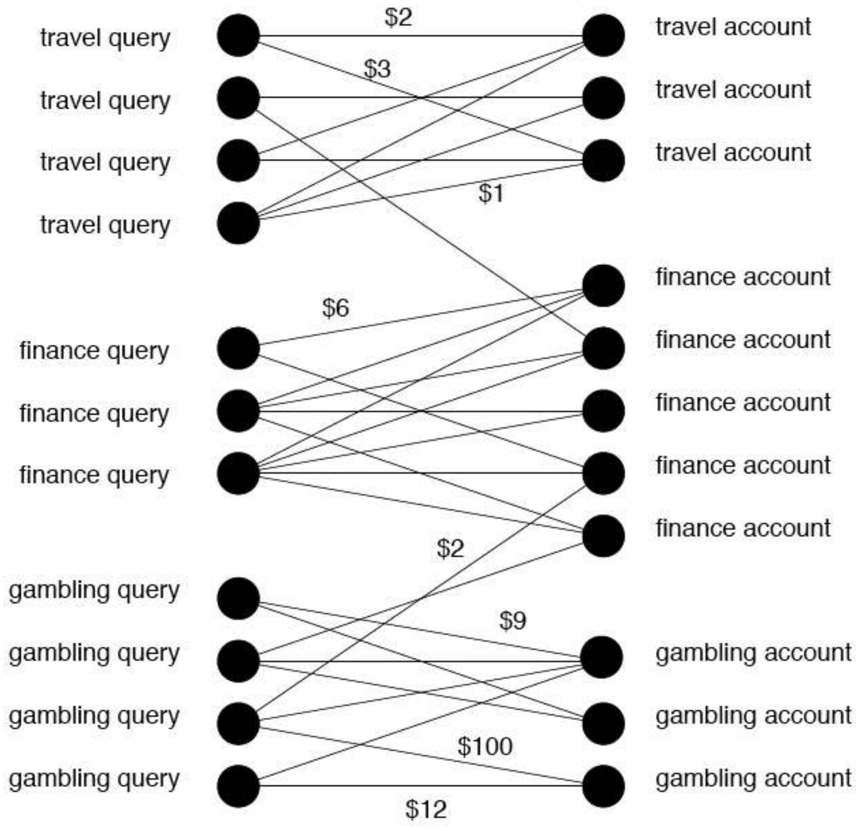
Barcelona Chairs
Chairs & Seats from 152+ Shops. **Barcelona Chairs** on Sale.
www.**Calibex.com**

Barcelona Chair - $659.99 Free Shipping
Loveseat, daybed, ottoman. Free shipping. Up to 60% off.
www.**modabode.com**

Buy **Barcelona Chairs**
We Have 13,900+ Sofas. **Barcelona Chairs** on Sale.
www.**NexTag.com**/sofas

Barcelona Chair
The Right Style For Your Space. **Barcelona chair** From $20.
Shopzilla.com/chairs

Bidding and Spending Graphs



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

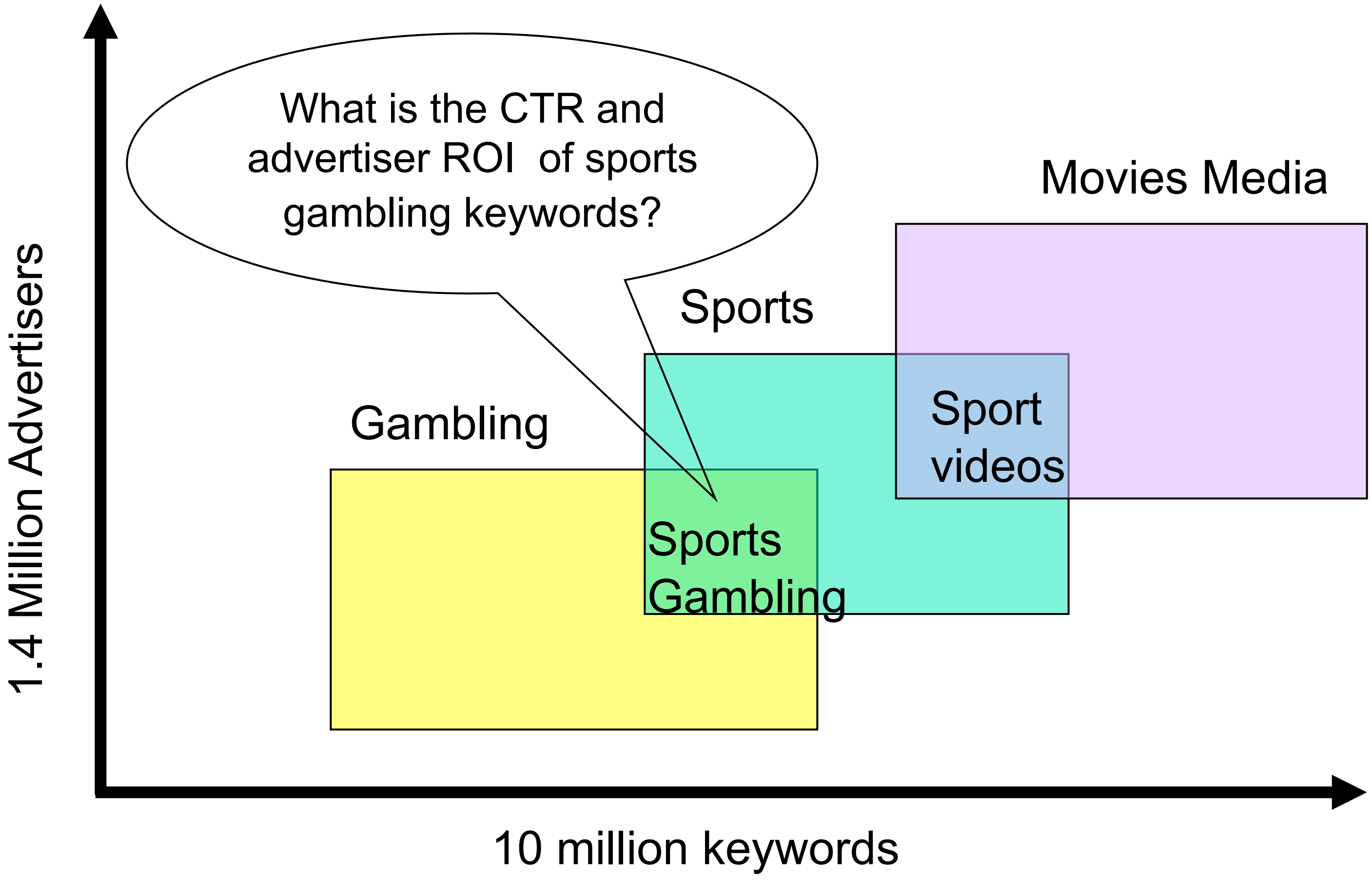
Uses of Bidding and Spending graphs:

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

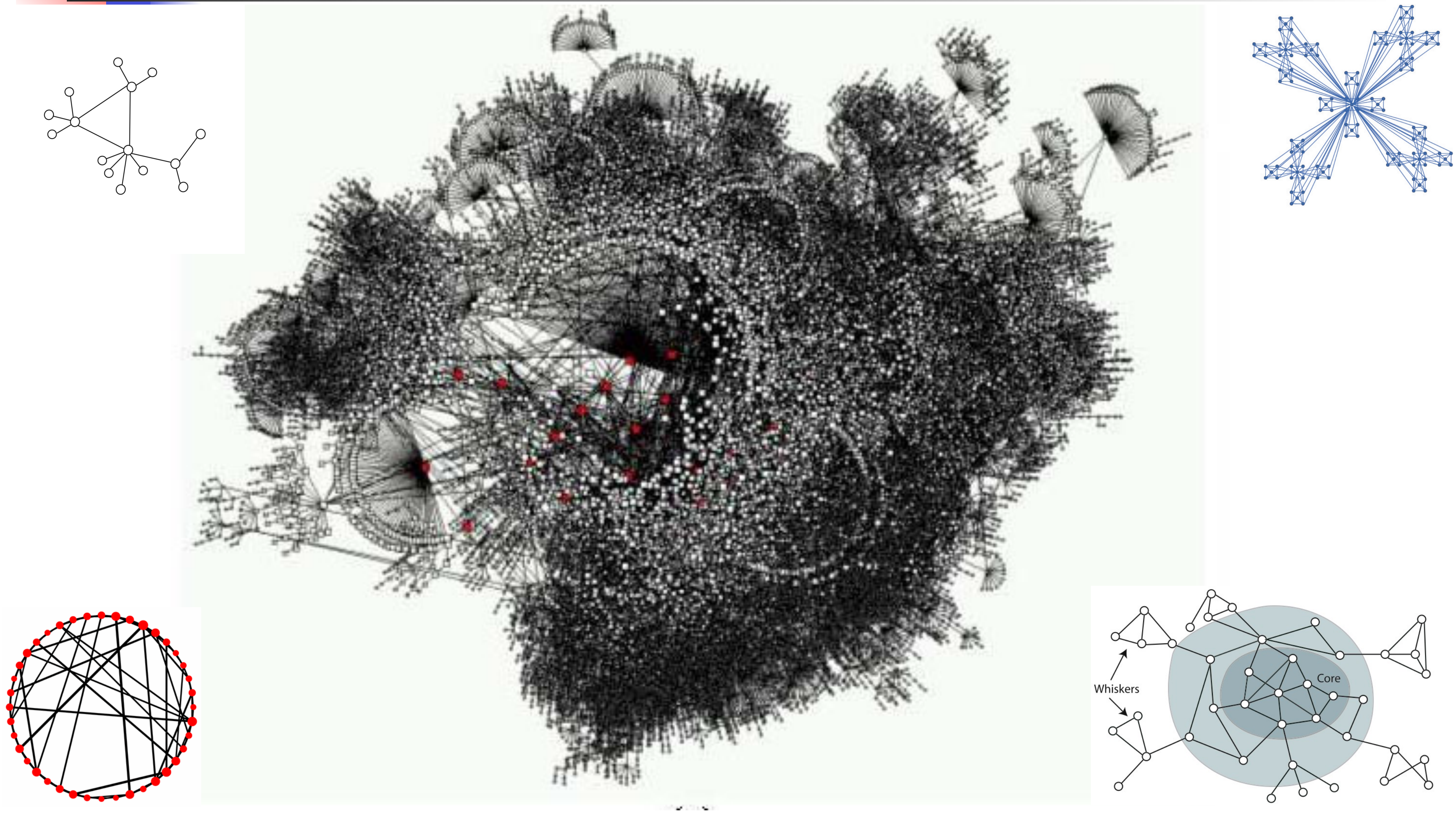
More generally, user segmentation for behavioral targeting.

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

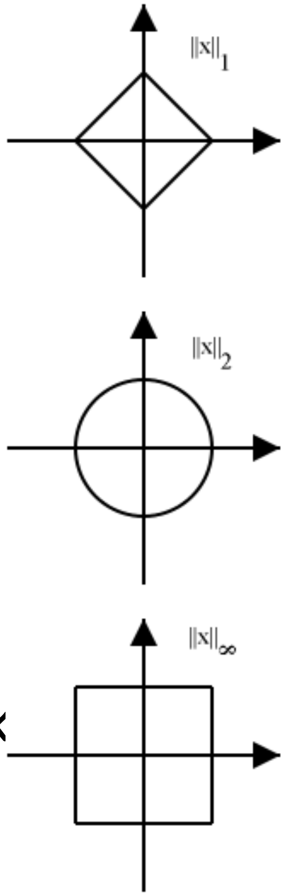
Two methods -- complementary strengths and weaknesses

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

Explicit versus implicit geometry

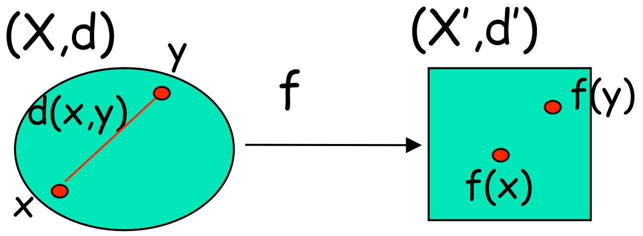
Explicitly-imposed geometry

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

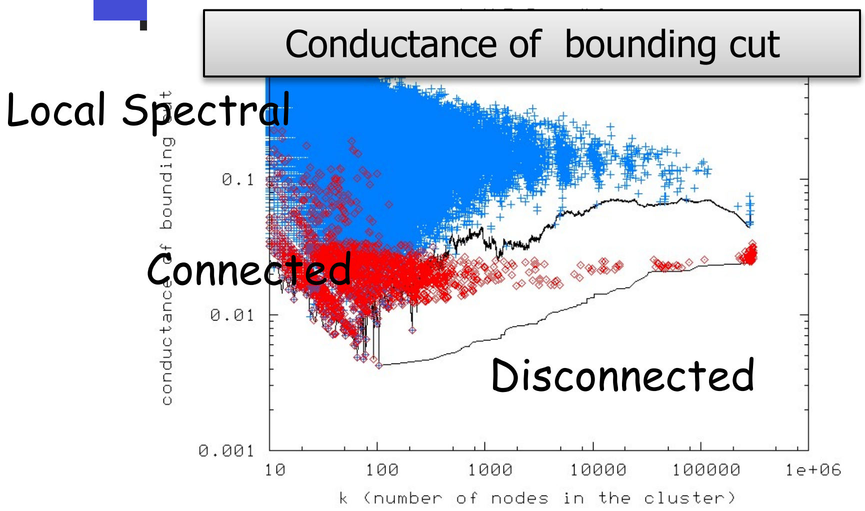


Implicitly-imposed geometry

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



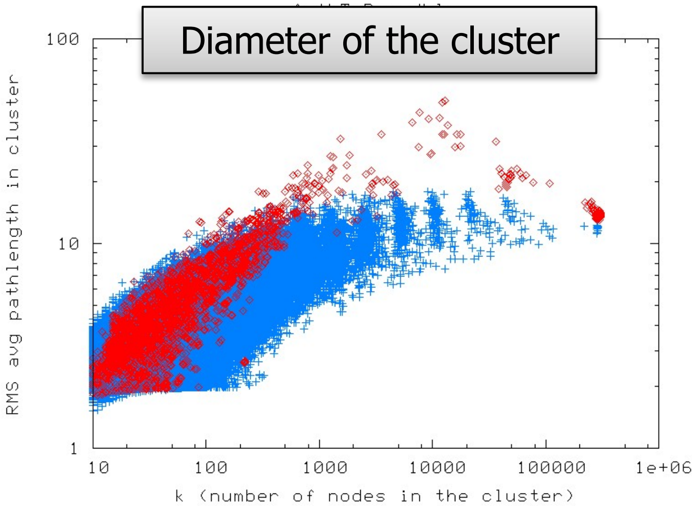
Regularized and non-regularized communities (1 of 2)


Conductance of bounding cut

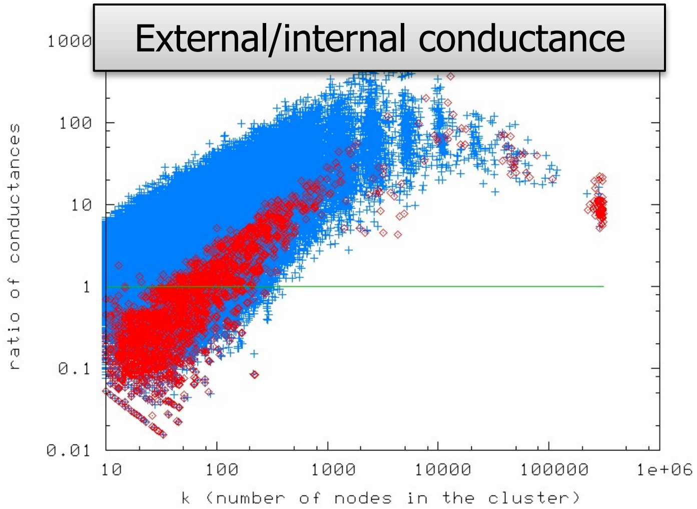
Local Spectral

Connected

Disconnected


Diameter of the cluster

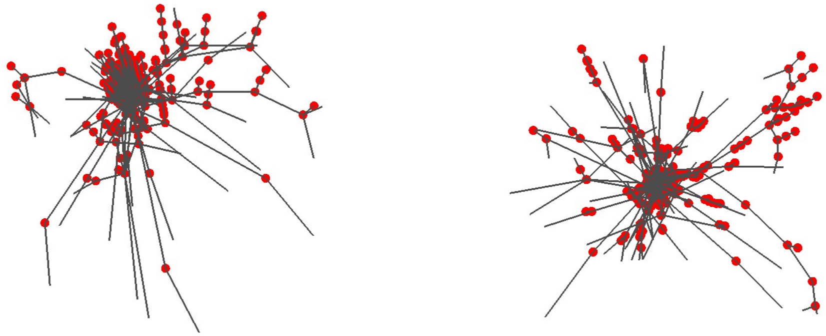
- Metis+MQI - a Flow-based method (red) gives sets with better conductance.
- Local Spectral (blue) gives tighter and more well-rounded sets.


External/internal conductance

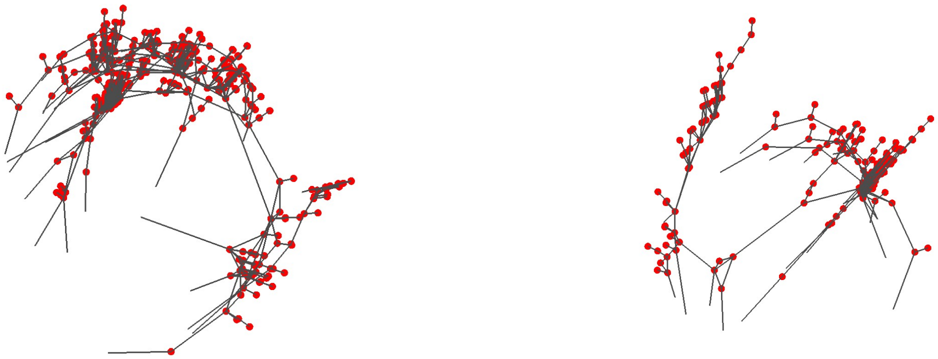
Lower is good

Regularized and non-regularized communities (2 of 2)

Two ca. 500 node communities from Local Spectral Algorithm:



Two ca. 500 node communities from Metis+MQI:



Outline and overview

Preamble: algorithmic & statistical perspectives

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

Approximate first nontrivial eigenvector of Laplacian

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

Spectral versus flow-based algs for graph partitioning

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

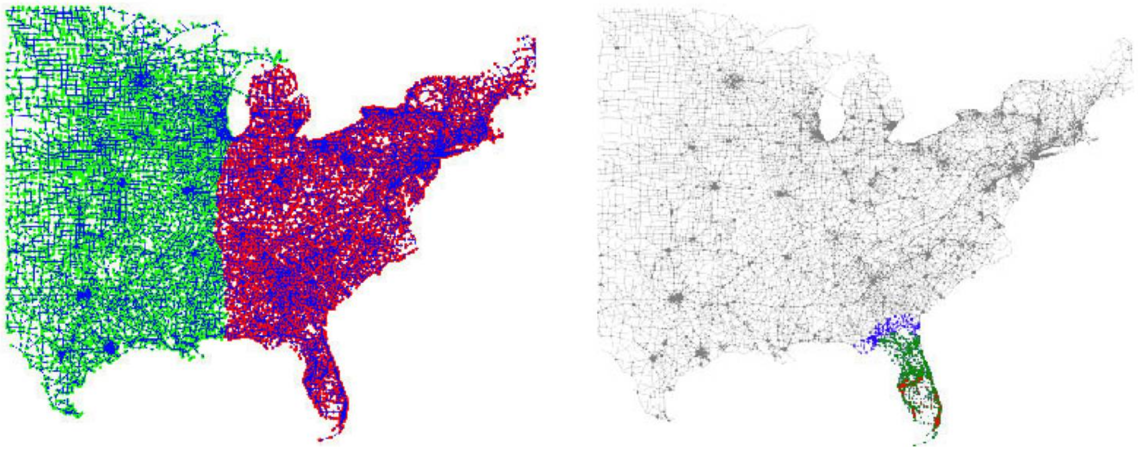
Weakly-local and strongly-local graph partitioning methods

- Operationally like L1-regularization, and already used in practice!

Computing locally-biased partitions

Often want clusters "near" a pre-specified set of nodes:

- Large social graphs have good small clusters, don't have good large clusters
- Might have domain knowledge, so find "semi-supervised" clusters
- As algorithmic primitives, e.g., to solve linear equations fast.



Recall global spectral graph partitioning

The basic optimization problem:

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

- Relaxation of:

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

- Solvable via the eigenvalue problem:

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

- Sweep cut of second eigenvector yields:

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

Idea to compute locally-biased partitions:

- Modify this objective with a locality constraint
- Show that some/all of these nice properties still hold locally

Local spectral partitioning *ansatz*

Mahoney, Orecchia, and Vishnoi (2010)

Primal program:

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

Interpretation:

- Find a cut well-correlated with the seed vector s.

- If s is a single node, this relaxes:

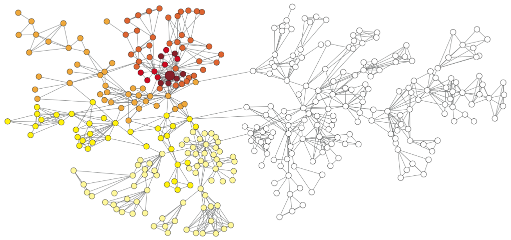
$$\min_{S \subset V, s \in S, |S| \leq 1/k} \frac{E(S, \bar{S})}{Vol(S) Vol(\bar{S})}$$

Dual program:

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

Interpretation:

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



Main theoretical results

Mahoney, Orecchia, and Vishnoi (2010)

Theorem: If x^* is an optimal solution to LocalSpectral,

(*) it is a Generalized Personalized PageRank vector, and can be computed as solution to a set of linear equations;

Fast running time guarantee.

(*) one can find a cut of conductance $\leq 8\lambda(G,s,\kappa)$ in time $O(n \lg n)$ with sweep cut of x^*;

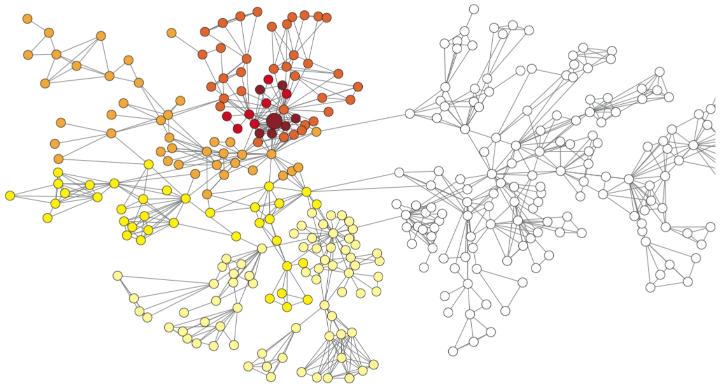
Upper bound, as usual from sweep cut & Cheeger.

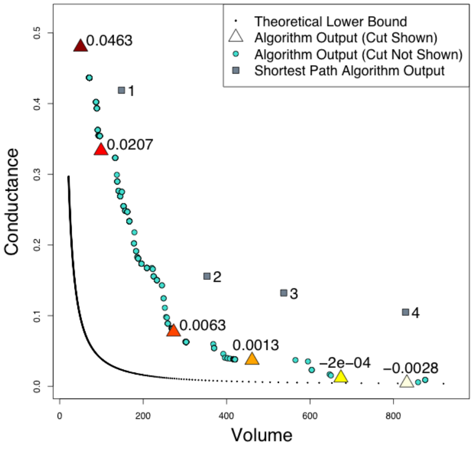
(*) For all sets of nodes T s.t. $\kappa' := \langle s, s_T \rangle_D^2$, we have: $\phi(T) \geq \lambda(G,s,\kappa)$ if $\kappa \leq \kappa'$, and $\phi(T) \geq (\kappa'/\kappa)\lambda(G,s,\kappa)$ if $\kappa' \leq \kappa$.

Lower bound: Spectral version of flow-improvement algs.

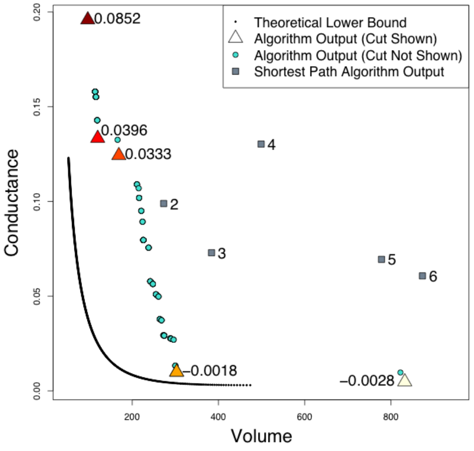
Illustration on small graphs

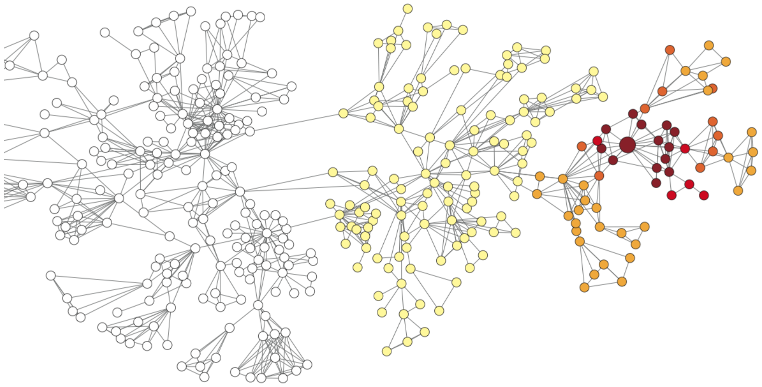
Mahoney, Orecchia, and Vishnoi (2010)





- 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.)





A somewhat different approach

Strongly-local spectral methods

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

These are the diffusion-based procedures

that we saw before

except truncate/round/clip/push small things to zero

starting with localized initial condition

Also get provably-good local version of global spectral

What's the connection?

"Optimization" approach:

- Well-defined objective f
- Weakly local (touch all nodes), so good for medium-scale problems
- Easy to use

"Operational" approach*:

- *Very* fast algorithm
- Strongly local (clip/truncate small entries to zero), good for large-scale
- Very difficult to use

* *Informally, optimize $f+\lambda g$ (... almost formally!):* steps are structurally-similar to the steps of how, e.g., L1-regularized L2 regression algorithms, implement regularization

More importantly,

- This "operational" approach is *already* being adopted in PODS/VLDB/SIGMOD/KDD/WWW environments!
- Let's make the regularization explicit—and know what we compute!

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 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 to Part Two

Regularization is:

- absent from CS, which historically has studied computation per se
- central to nearly area that applies algorithms to noisy data
- 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
- Solutions of approximation algorithms don't need to be something we "settle for," they can be "better" than the "exact" solution

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 very large-scale analytics?

Conclusions ... And Looking Forward

In many BIG data applications, "algorithmic" and "statistical" perspectives interact in fruitful ways

- Genetics: DNA SNP analysis --> choose columns from a matrix
- Internet: Community finding --> partitioning a graph

Regularization lies at the heart of the algorithmic-statistical disconnect

- Absent from CS, but central to every area that computes on noisy data
- Approximate computation, in and of itself, regularizes

Connections with BIG Information Theory?

- What is information? What is data? What is signal? What is noise? How to use these ideas in information theory?
- You tell me ...