Eigenvector localization, implicit regularization, and algorithmic anti-differentiation for large-scale graphs and network data

Michael W. Mahoney

ICSI and Dept of Statistics, UC Berkeley

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

First, parse the title ...

Eigenvector localization:

- Eigenvectors are "usually" global entities
- But they can be localized in extremely sparse/noisy graphs/matrices

Implicit regularization:

- Usually "exactly" optimize $f+\lambda g$, for some λ and g
- Regularization often a side effect of approximations to f

Algorithmic anti-differentiation:

- What is the objective that approximate computation exactly optimizes

Large-scale graphs and network data:

- Small versus medium versus large versus big
- Social/information networks versus "constructed" graphs

Outline

Motivation: large informatics graphs

- Downward-sloping, flat, and upward-sloping NCPs (i.e., not “nice” *at large size scales*, but instead expander-like/tree-like)
- Implicit regularization in graph approximation algorithms

Eigenvector localization & semi-supervised eigenvectors

- Strongly and weakly local diffusions
- Extension to semi-supervised eigenvectors

Implicit regularization & algorithmic anti-differentiation

- Early stopping in iterative diffusion algorithms
- Truncation in diffusion algorithms

Outline

Motivation: large informatics graphs

- Downward-sloping, flat, and upward-sloping NCPs (i.e., not "nice" *at large size scales*, but instead expander-like/tree-like)
- Implicit regularization in graph approximation algorithms

Eigenvector localization & semi-supervised eigenvectors

- Strongly and weakly local diffusions
- Extension to semi-supervised eigenvectors

Implicit regularization & algorithmic anti-differentiation

- Early stopping in iterative diffusion algorithms
- Truncation in diffusion algorithms

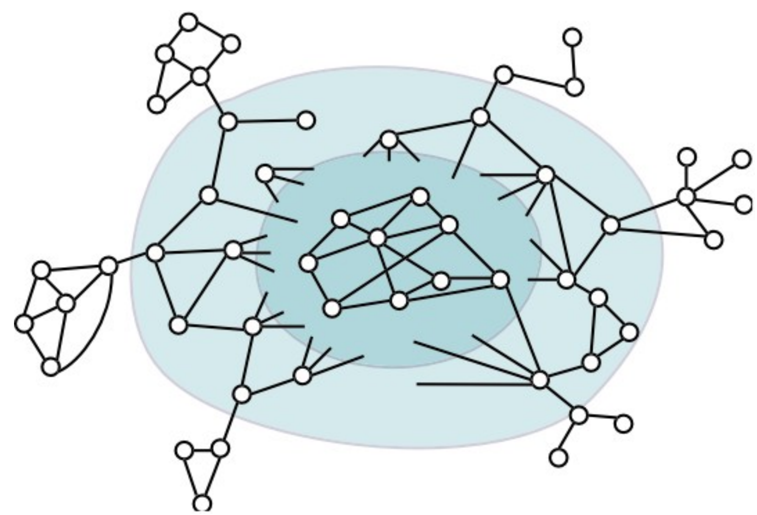
Networks and networked data

Lots of "networked" data!!

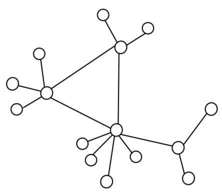
- technological networks
  - AS, power-grid, road networks
- biological networks
  - food-web, protein networks
- social networks
  - collaboration networks, friendships
- information networks
  - co-citation, blog cross-postings, advertiser-bidded phrase graphs...
- language networks
  - semantic networks...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

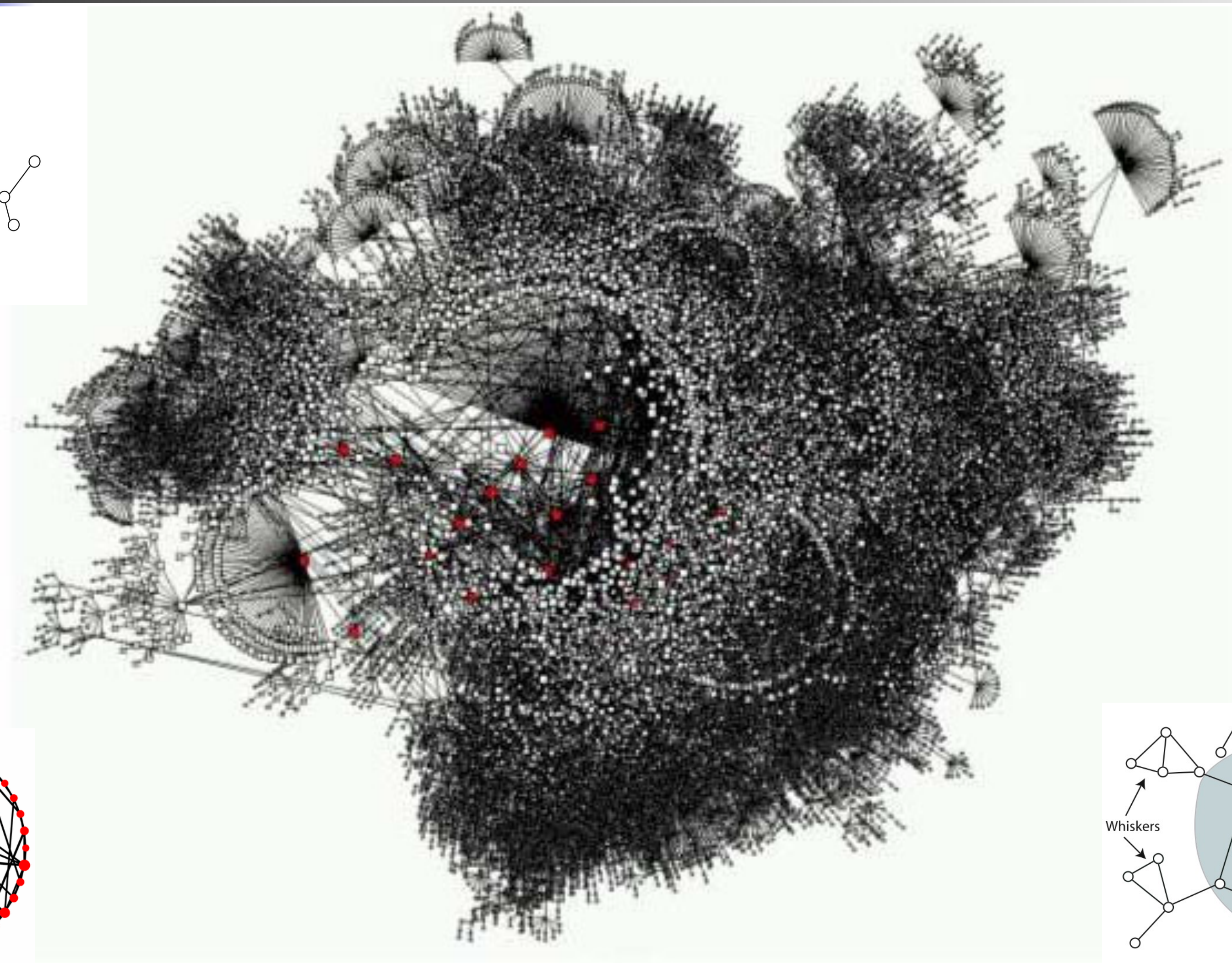
Interaction graph model of networks:

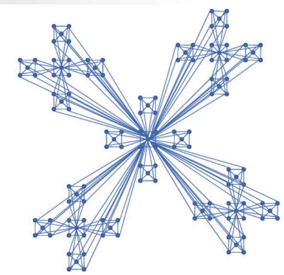
- Nodes represent "entities"
- Edges represent "interaction" between pairs of entities

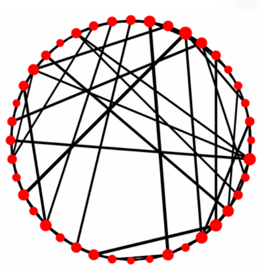


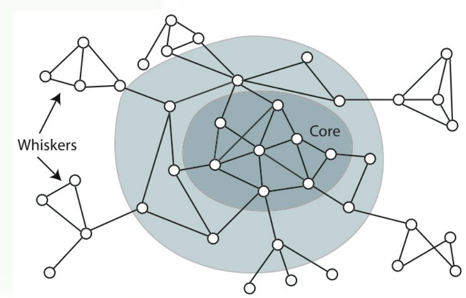
What do these networks "look" like?



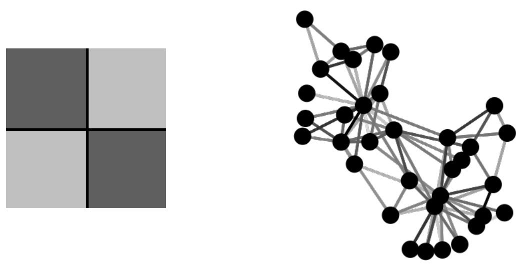




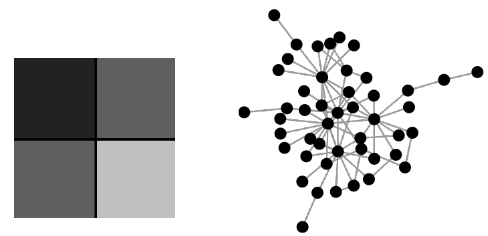




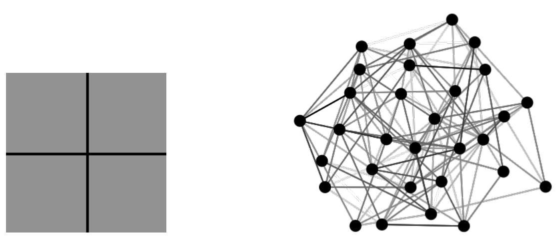
Possible ways a graph might look



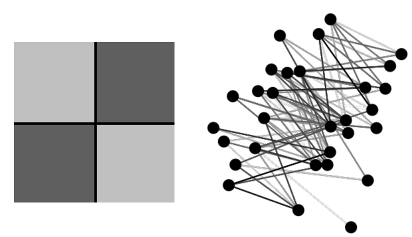
Low-dimensional structure



Core-periphery structure

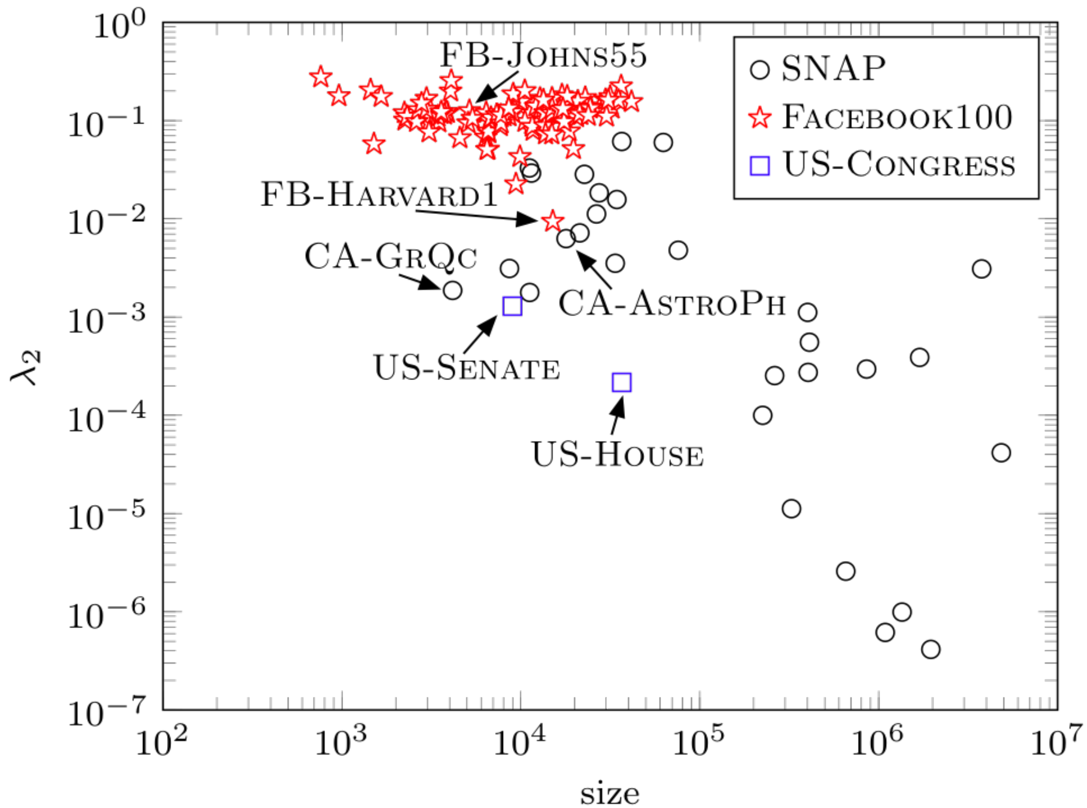


Expander or complete graph



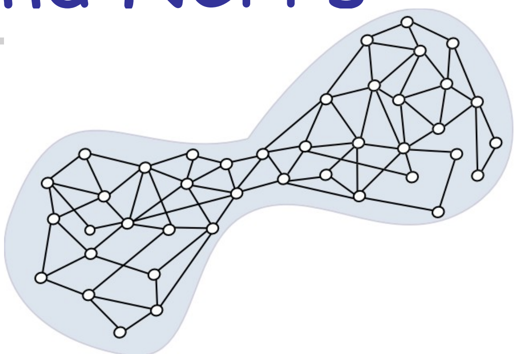
Bipartite structure

Scatter plot of λ2 for real networks



Question: does this plot really tell us much about these networks?

Communities, Conductance, and NCPPs



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

The conductance ϕ of a set S of nodes is:

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

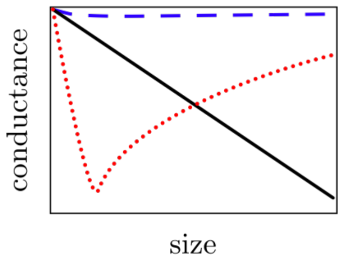
$$A(S) = \sum_{i \in S} \sum_{j \in V} A_{ij}$$

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

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

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

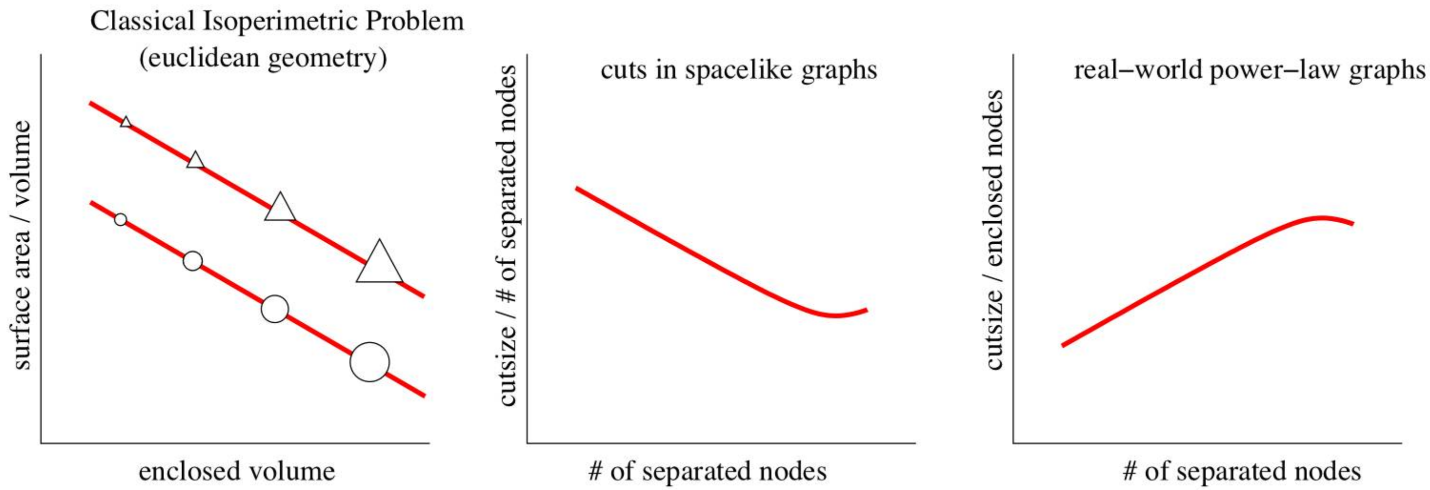
- *the NCP captures a Size-Resolved Surface-Area-To-Volume notion*
- *captures the idea of size-resolved bottlenecks to diffusion*



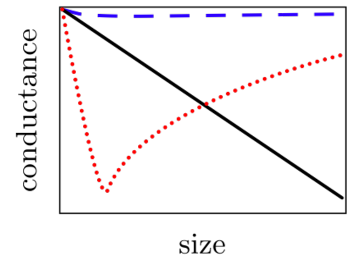
Why worry about both criteria?

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

Tradeoff between cut quality and balance



• For other classes of graphs (e.g., informatics graphs, as we will see) there is a "tradeoff," i.e., better cuts lead to worse balance
• For still other graphs (e.g., expanders) there are no good cuts of any size



Probing Large Networks with Approximation Algorithms

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

Spectral - (quadratic approx) - confuses "long paths" with "deep cuts"

Multi-commodity flow - (log(n) approx) - difficulty with expanders

SDP - (sqrt(log(n)) approx) - best in theory

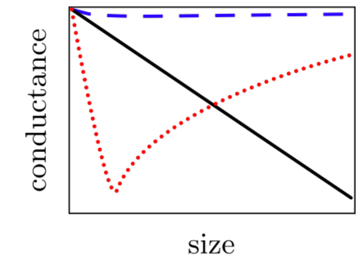
Metis - (multi-resolution for mesh-like graphs) - common in practice

X+MQI - post-processing step on, e.g., Spectral of Metis

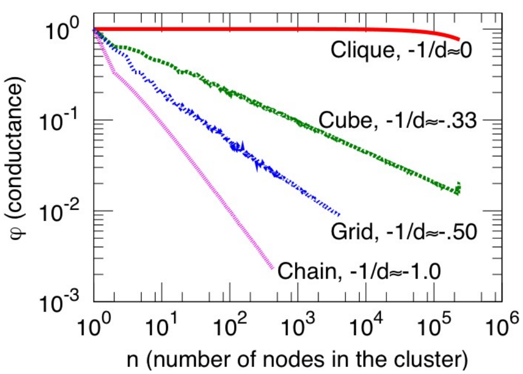
Metis+MQI - best conductance (empirically)

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

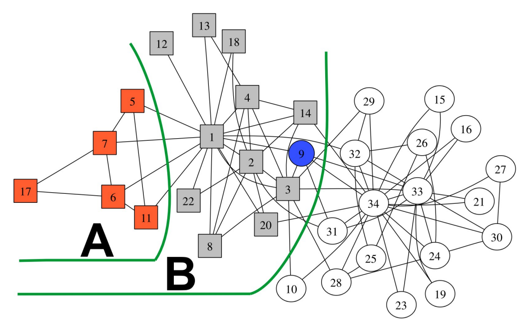
- *We exploit the "statistical" properties implicit in "worst case" algorithms.*



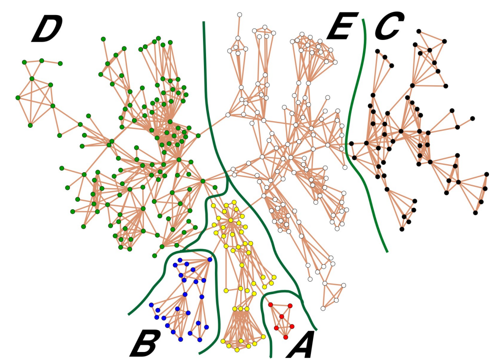
Typical intuitive networks



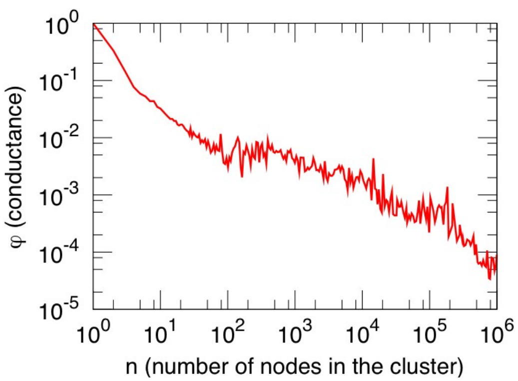
d-dimensional meshes



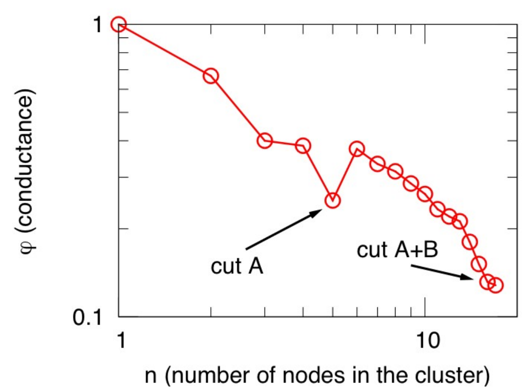
Zachary's karate club

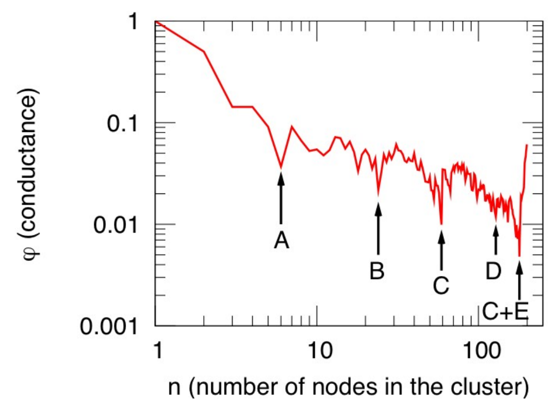


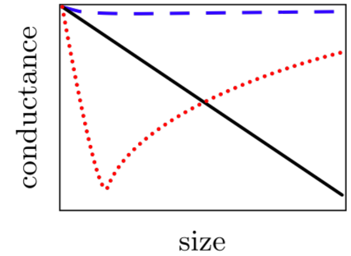
Newman's Network Science



RoadNet-CA



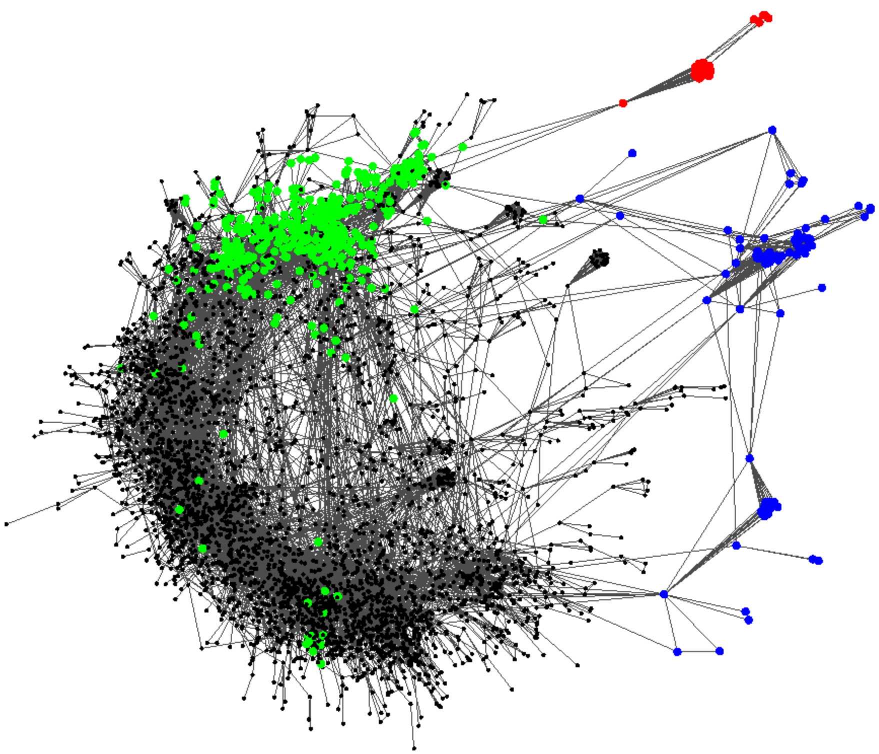


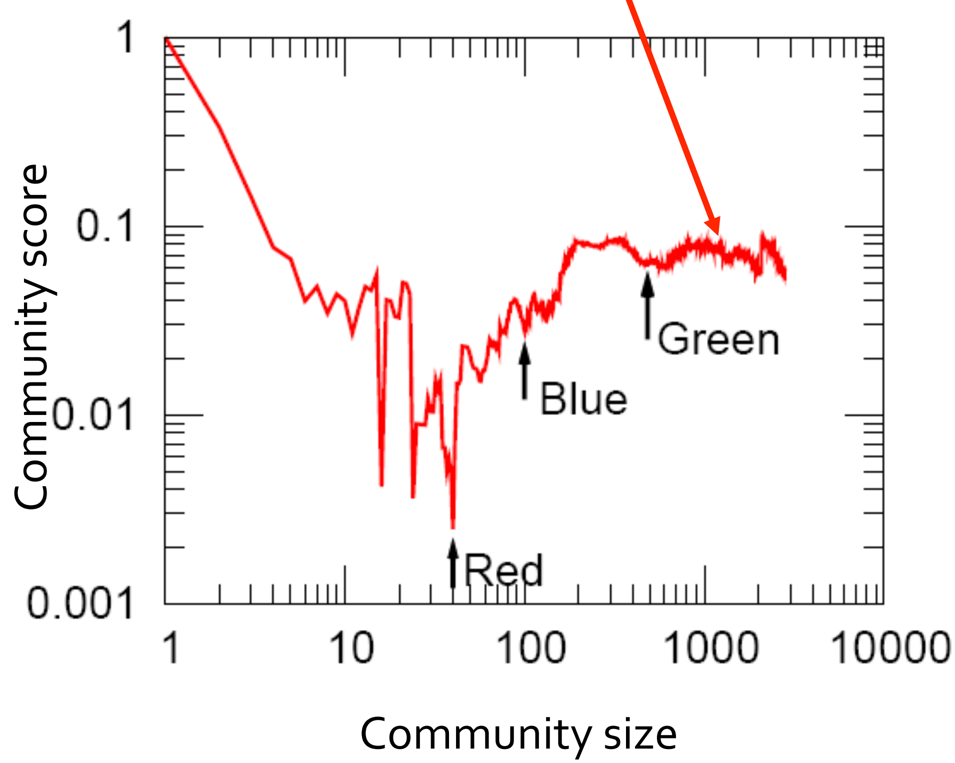


Typical real network

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

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

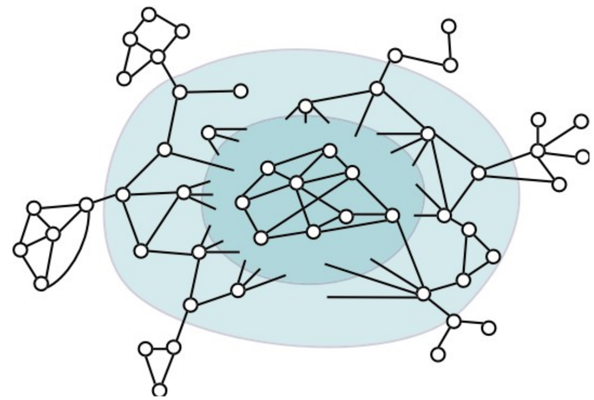


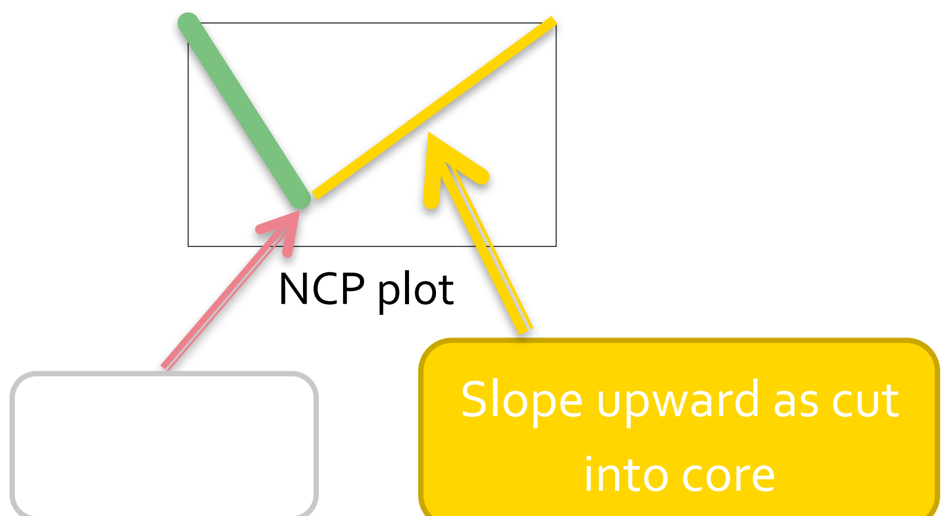


"Whiskers" and the "core"

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

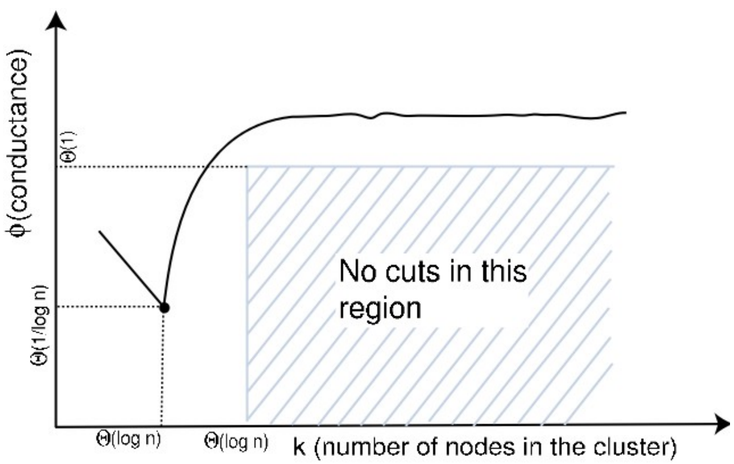
- *And, the core has a core-peripehery structure, recursively ...*



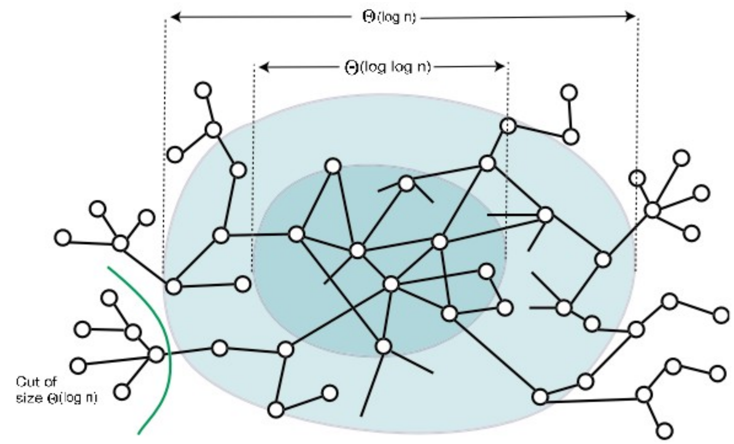


A simple theorem on random graphs

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



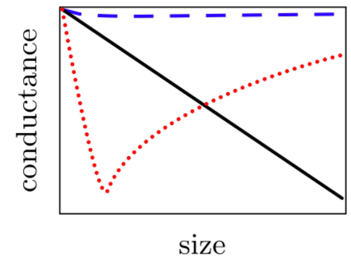
Power-law random graph with β ε (2,3).



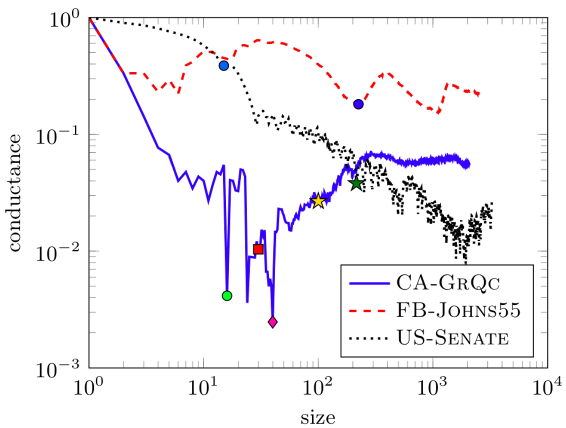
Structure of the G(w) model, with β ε (2,3).

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

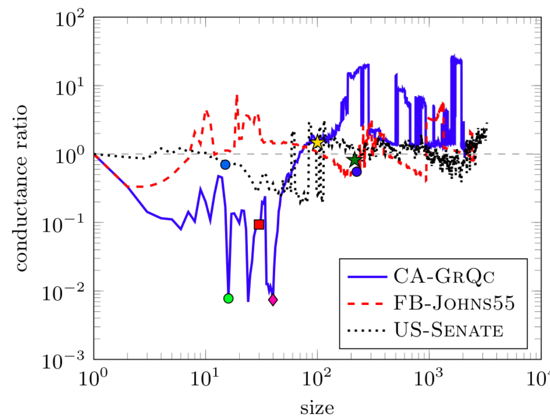
Think of the data as: local-structure on global-noise; not small noise on global structure!



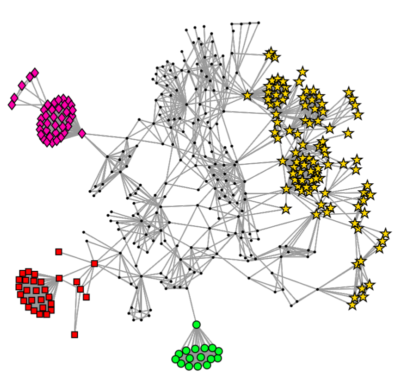
Three different types of real networks



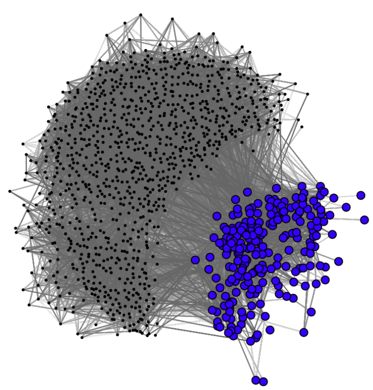
NCP: conductance value of best conductance set in graph, as a function of size



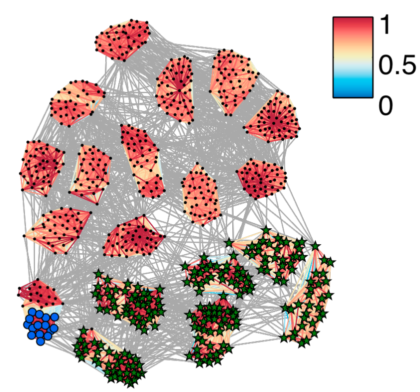
CRP: ratio of internal to external conductance, as a function of size



CA-GrQc



FB-Johns55



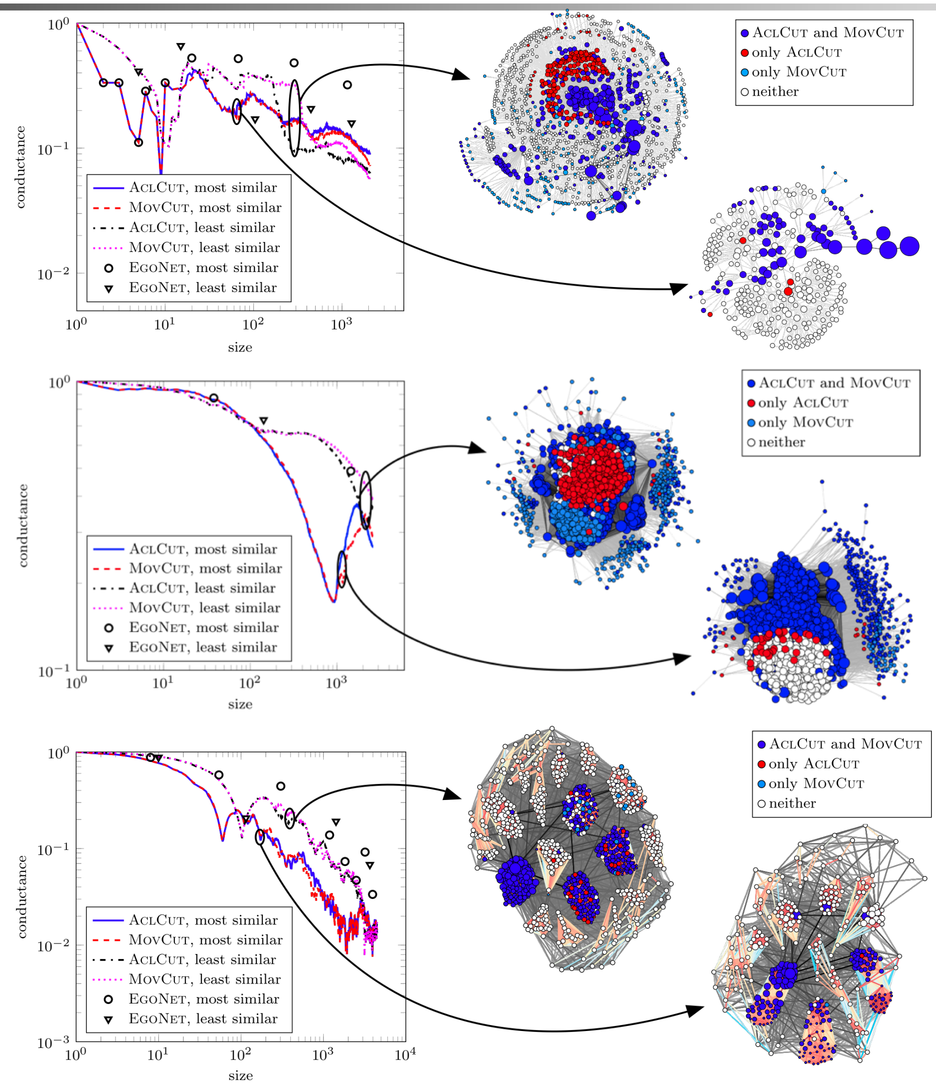
US-Senate

Local structure for graphs with upward versus downward sloping NCPs

CA-GrQc: upward-sloping global NCP

FB-Johns55: flat global NCP

US-Senate: downward-sloping global NCP



AclCut (strongly local spectral method)

versus

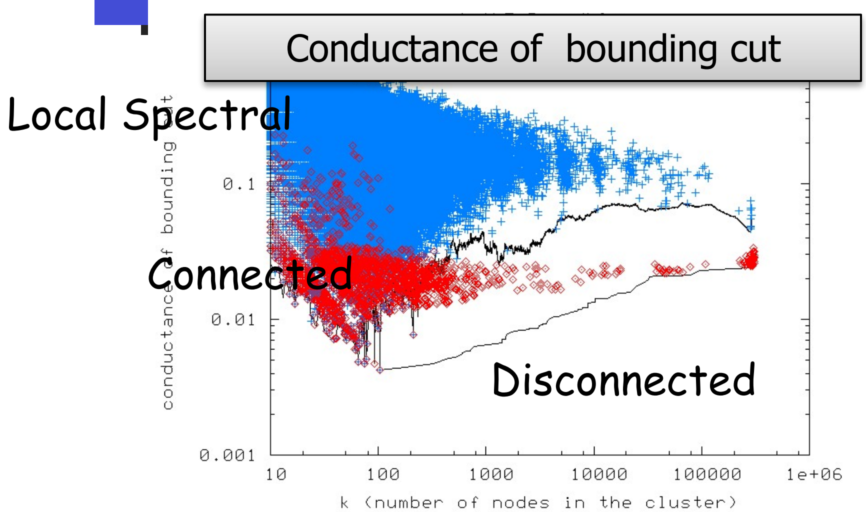
MovCut (weakly local spectral method)

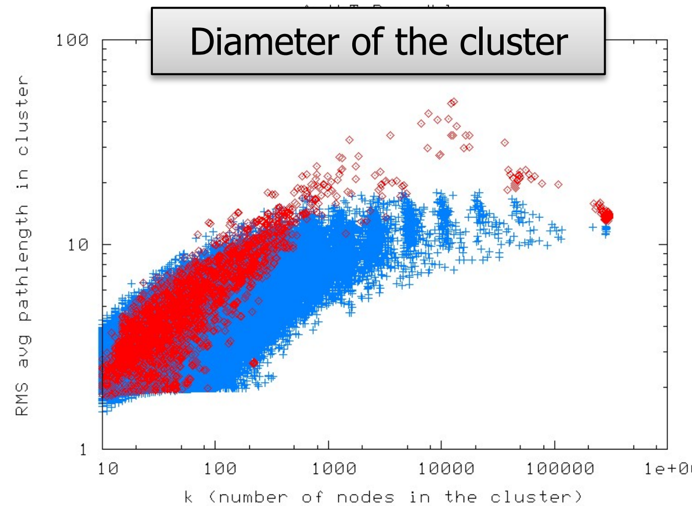
Two very similar methods often give very different results.

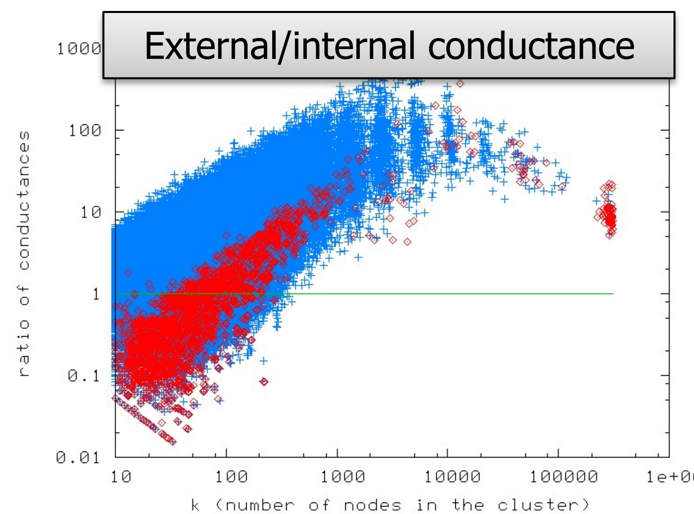
Former is often preferable---for both algorithmic *and* statistical reasons.

Why? And what does problem does it solve?

Regularized and non-regularized communities







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.

Summary of lessons learned

Local-global properties of real data are very different ...

- ... than practical/theoretical people implicitly/explicitly assume

Local spectral methods were a big winner

- For both algorithmic and statistical reasons

Little design decisions made a big difference

- Details of how deal with truncation and boundary conditions are not second-order issues when graphs are expander-like

Approximation algorithm usefulness uncoupled from theory

- Often useful when they implicitly regularize

Outline

Motivation: large informatics graphs

- Downward-sloping, flat, and upward-sloping NCPs (i.e., not "nice" *at large size scales*, but instead expander-like/tree-like)
- Implicit regularization in graph approximation algorithms

Eigenvector localization & semi-supervised eigenvectors

- Strongly and weakly local diffusions
- Extension to semi-supervised eigenvectors

Implicit regularization & algorithmic anti-differentiation

- Early stopping in iterative diffusion algorithms
- Truncation in diffusion algorithms

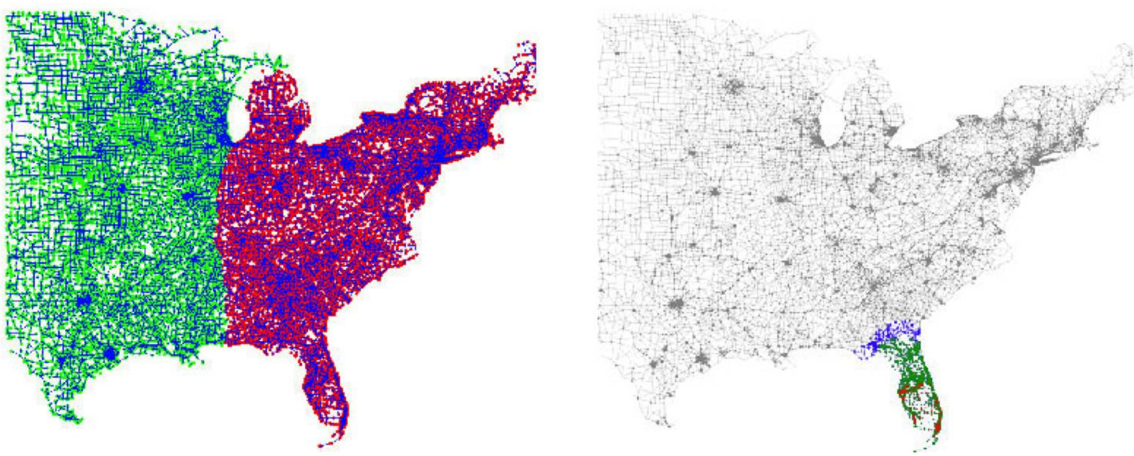
Local spectral optimization methods

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

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

ACL06: approximate locally-biased PageRank vector computations

Chung08: approximate heat-kernel computation to get a vector



Q1: What do these procedures optimize approximately/exactly?

Q2: Can we write these procedures as optimization programs?

Recall spectral graph partitioning

The basic optimization problem:

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

- Relaxation of:

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

- Solvable via the eigenvalue problem:

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

- Sweep cut of second eigenvector yields:

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

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

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

Geometric correlation and generalized PageRank vectors

Given a cut T, define the vector:

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

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

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

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

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

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

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

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

Local spectral partitioning *ansatz*

Mahoney, Orecchia, and Vishnoi (2010)

Primal program:

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

Dual program:

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

Interpretation:

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

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

Interpretation:

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

Main results (1 of 2)

Mahoney, Orecchia, and Vishnoi (2010)

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

Proof:

(1) Relax non-convex problem to convex SDP

(2) Strong duality holds for this SDP

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

(4) Rank one solution is GPPR vector.

Main results (2 of 2)

Mahoney, Orecchia, and Vishnoi (2010)

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

Upper bound, as usual from sweep cut & Cheeger.

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

Lower bound: Spectral version of flow-improvement algs.

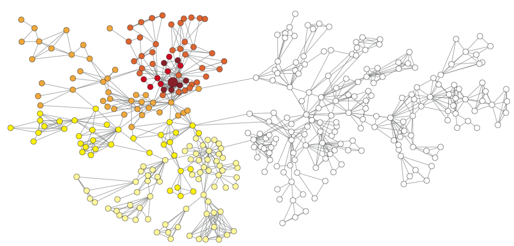
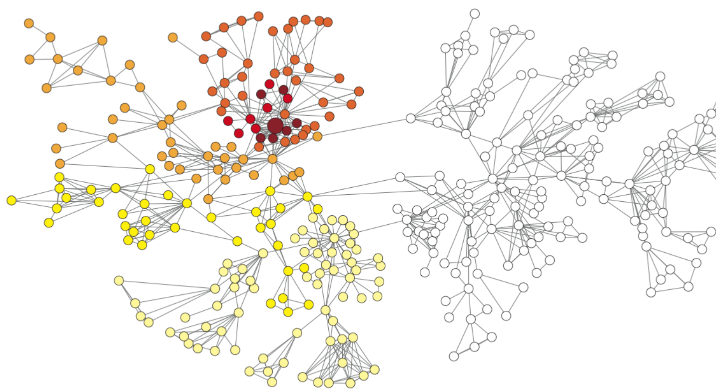
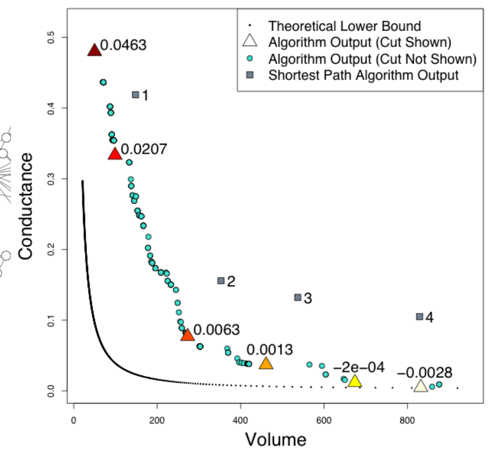


Illustration on small graphs





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

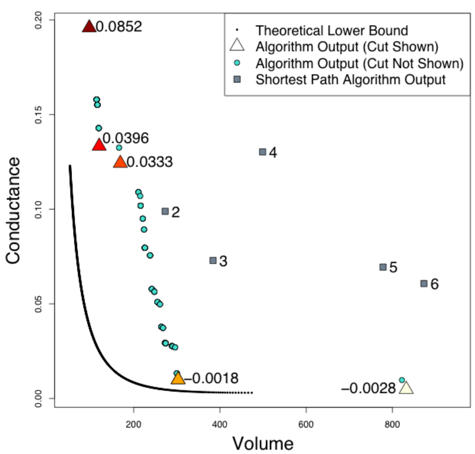
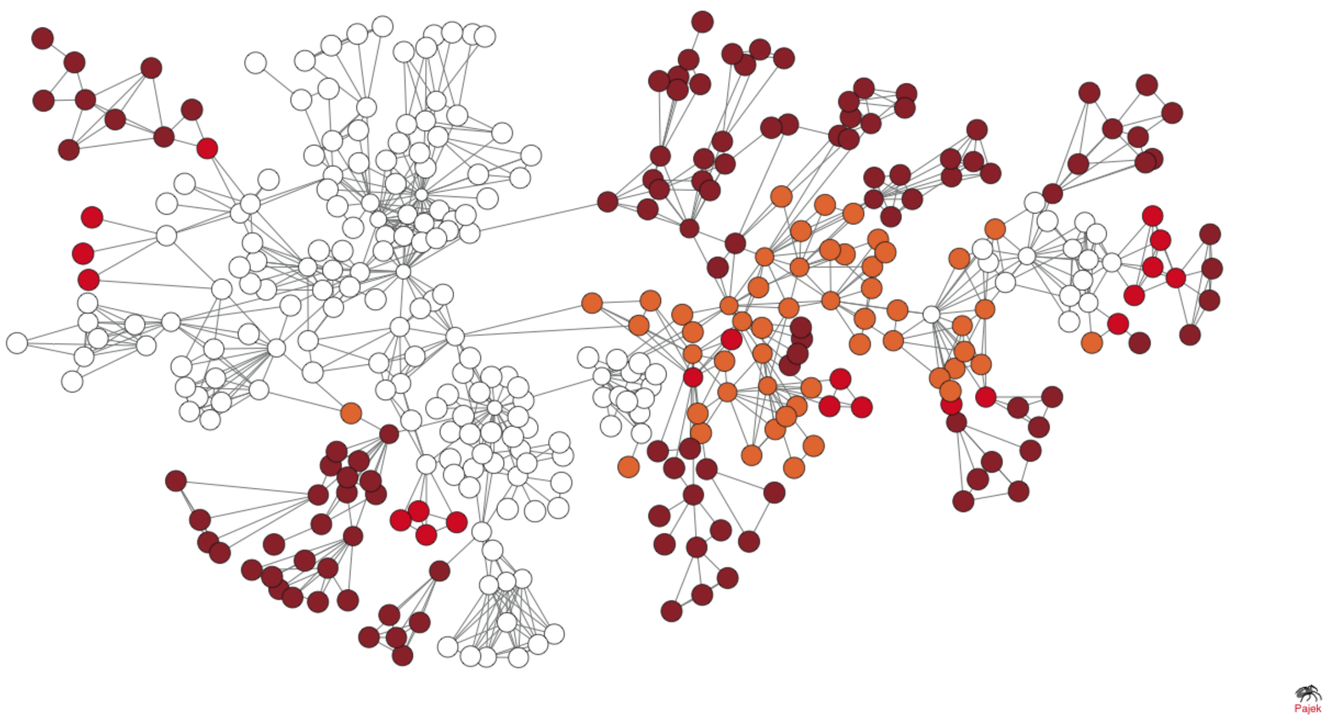


Illustration with general seeds

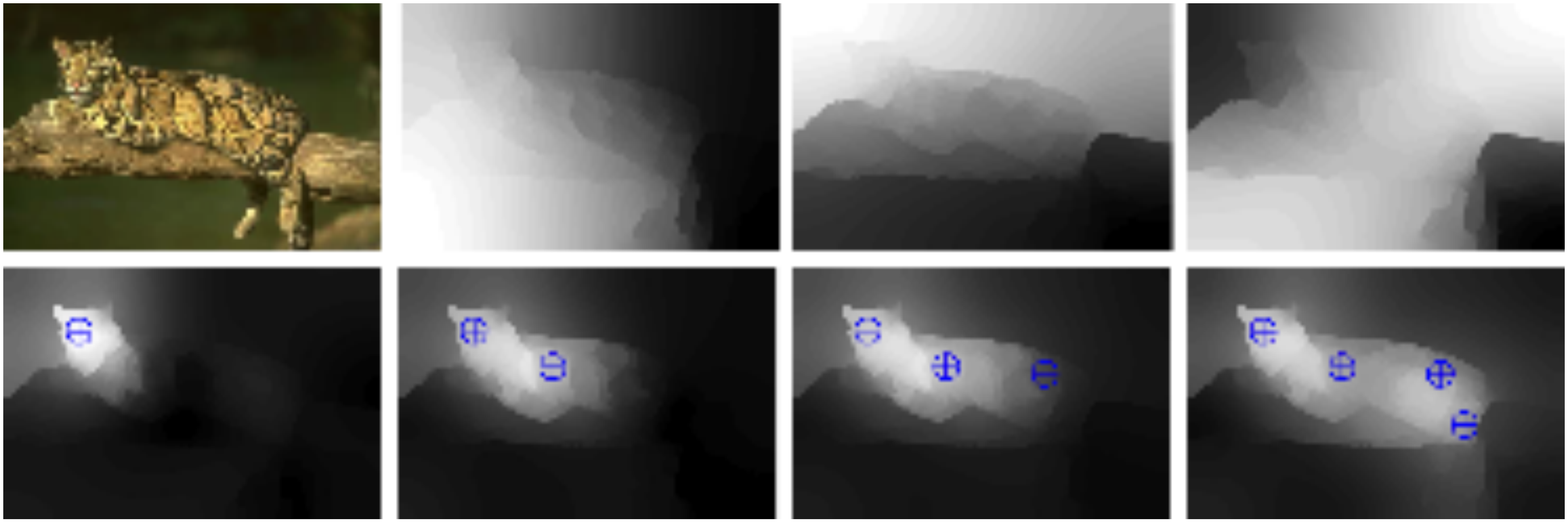
- Seed vector doesn't need to correspond to cuts.

- It could be any vector on the nodes, e.g., can find a cut "near" low-degree vertices with $s_i = -(d_i - d_{av})$, $i \varepsilon [n]$.



New methods are useful more generally

Maji, Vishnoi,and Malik (2011) applied Mahoney, Orecchia, and Vishnoi (2010)



- Cannot find the tiger with global eigenvectors.
- Can find the tiger with our LocalSpectral method!

Semi-supervised eigenvectors

Hansen and Mahoney (NIPS 2013, JMLR 2014)

Eigenvectors are inherently global quantities, and the leading ones may therefore fail at modeling relevant local structures.

GlobalSpectral

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

Generalized eigenvalue problem. Solution is given by the second smallest eigenvector, and yields a "Normalized Cut".

LocalSpectral

$$\begin{aligned} \text{minimize} \quad & x^T L_G x \\ \text{s.t} \quad & x^T D_G x = 1 \\ & x^T D_G 1 = 0 \\ & x^T D_G s \geq \sqrt{\kappa} \end{aligned}$$

Locally-biased analogue of the second smallest eigenvector. Optimal solution is a generalization of Personalized PageRank and can be computed in nearly-linear time [MOV2012].

Generalized LocalSpectral

$$\begin{aligned} \text{minimize} \quad & x^T L_G x \\ \text{s.t} \quad & x^T D_G x = 1 \\ & x^T D_G X = 0 \\ & x^T D_G s \geq \sqrt{\kappa} \end{aligned}$$

Semi-supervised eigenvector generalization of [MOV2012]. This objective incorporates a general orthogonality constraint, allowing us to compute a sequence of "localized eigenvectors".

Semi-supervised eigenvectors are efficient to compute and inherit many of the nice properties that characterizes global eigenvectors of a graph.

Semi-supervised eigenvectors

Hansen and Mahoney (NIPS 2013, JMLR 2014)

Provides a natural way to interpolate between very localized solutions and the global eigenvectors of the graph Laplacian.

For $\kappa = 0$ this becomes the usual generalized eigenvalue problem.

GENERALIZED LOCALSPECTRAL

$$\begin{aligned} \text{minimize} \quad & x^T L_G x \\ \text{s.t} \quad & x^T D_G x = 1 \quad \leftarrow \text{Norm constraint} \\ & x^T D_G X = 0 \quad \leftarrow \text{Orthogonality constraint} \\ & x^T D_G s \geq \sqrt{\kappa} \quad \leftarrow \text{Locality constraint} \end{aligned}$$

The solution can be viewed as the first step of the Rayleigh quotient iteration, where γ is the current estimate of the eigenvalue, and $D_G s$ the current estimate of the eigenvector.

Leading solution

Seed vector

$$x_1^* = c\,(L_G - \gamma_1 D_G)^+\, D_G s$$

Projection operator

$$x^* \propto (FF^T(L_G - \gamma D_G)FF^T)^+ FF^T D_G s$$

General solution

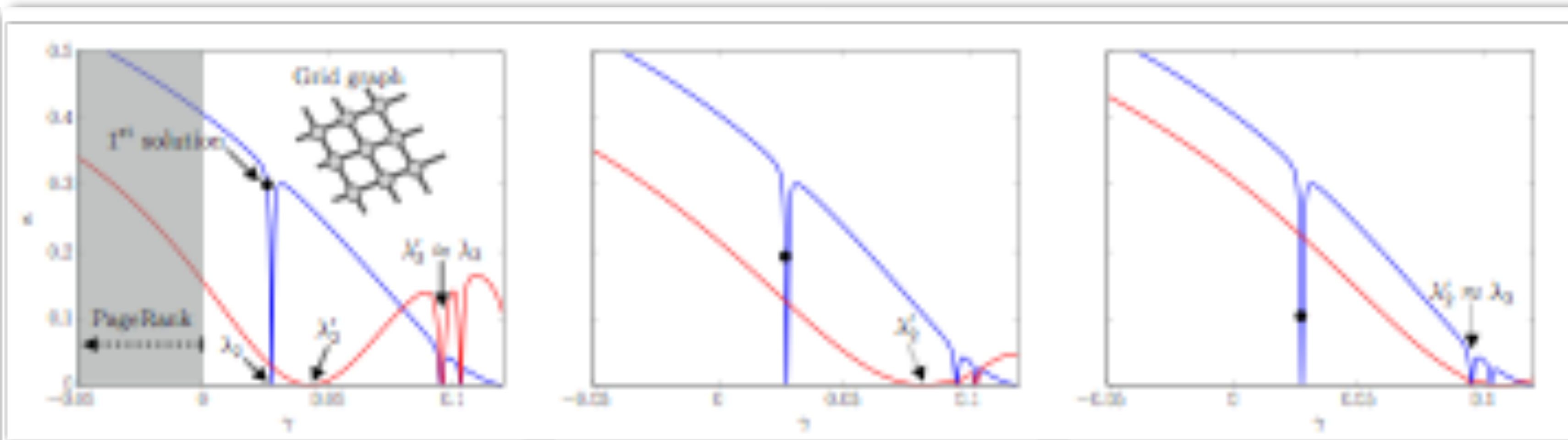
Determines the locality of the solution.

Convex for $\gamma \in (-\infty, \lambda_2(G))$

Semi-supervised eigenvectors

Hansen and Mahoney (NIPS 2013, JMLR 2014)

Convexity - The interplay between γ and κ .



(a) $(x_1^T D_G s)^2 = 0.3$ (b) $(x_1^T D_G s)^2 = 0.2$ (c) $(x_1^T D_G s)^2 = 0.1$

For $\gamma < 0$, one we can compute semi-supervised eigenvectors using local graph diffusions, *i.e.*, personalized PageRank.

$$x^* = (L_G - \gamma D_G)^+ D_G s$$

↓

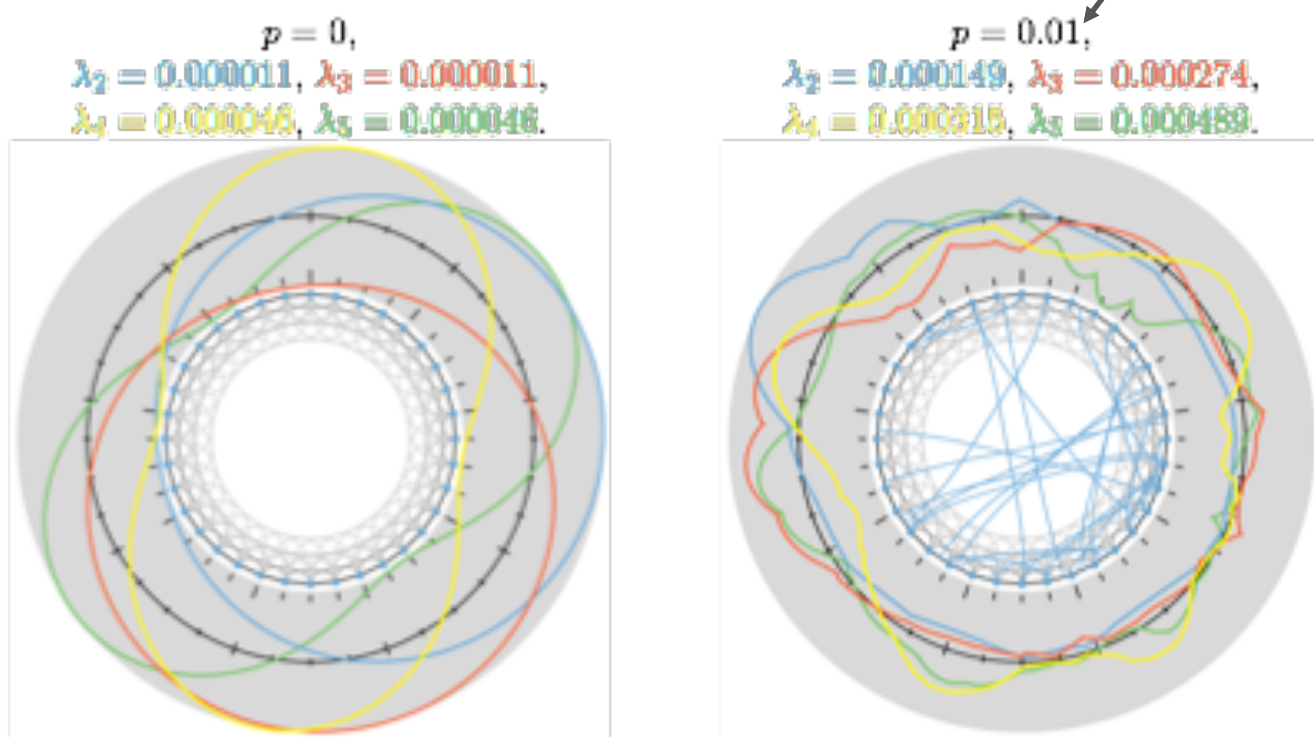
$$x^* = \frac{c}{1-\gamma} D_G^{-1} \left(I + \sum_{i=1}^{\infty} \left(\frac{1}{1-\gamma} D_G^{-1} A_G \right)^i \right) D_G s$$

Approximate the solution using the Push algorithm [Andersen2006].

Semi-supervised eigenvectors

Small-world example - The eigenvectors having smallest eigenvalues capture the slowest modes of variation.

Probability of random edges

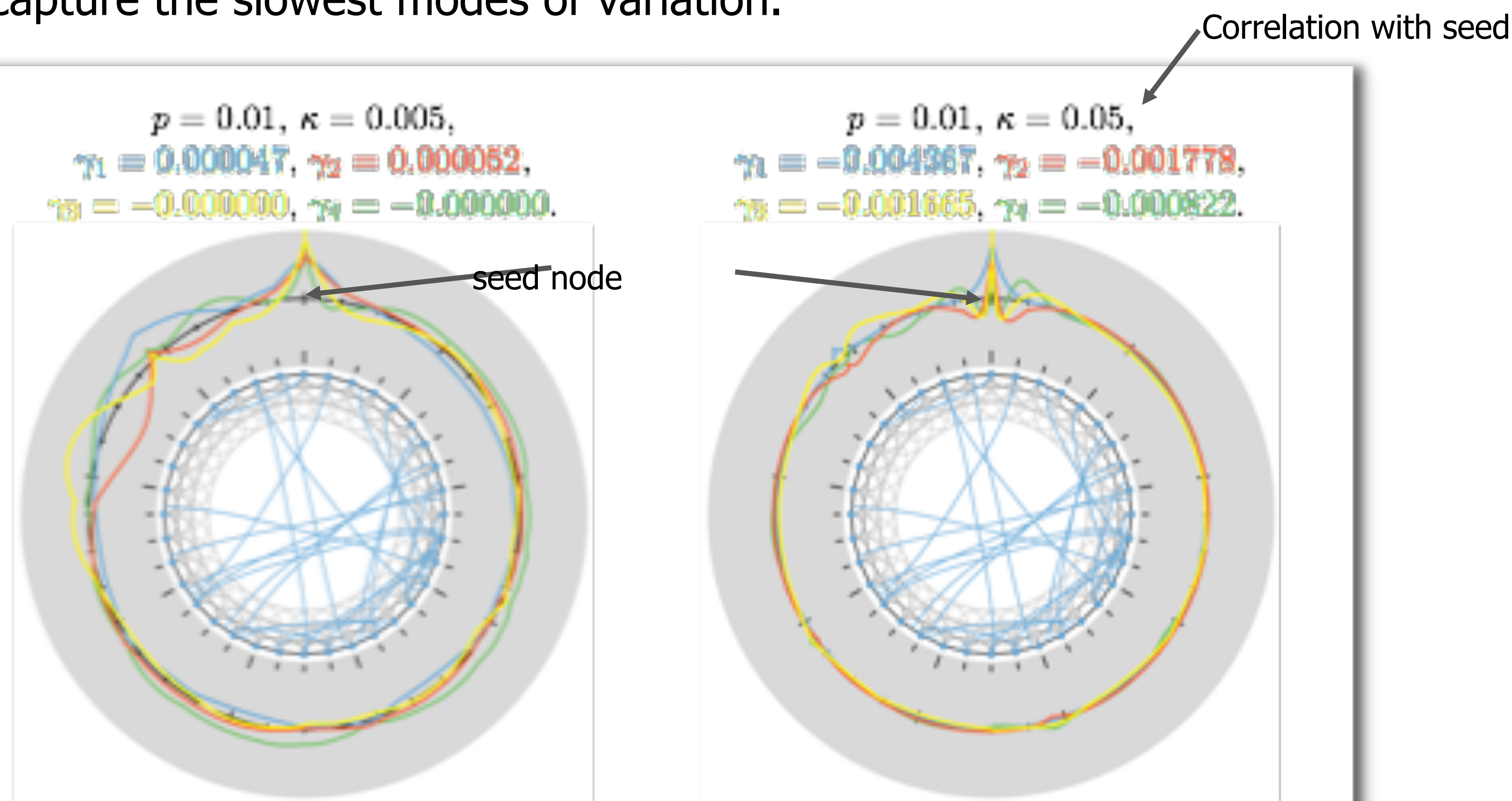


Global eigenvectors

Global eigenvectors

Semi-supervised eigenvectors

Small-world example - The eigenvectors having smallest eigenvalues capture the slowest modes of variation.



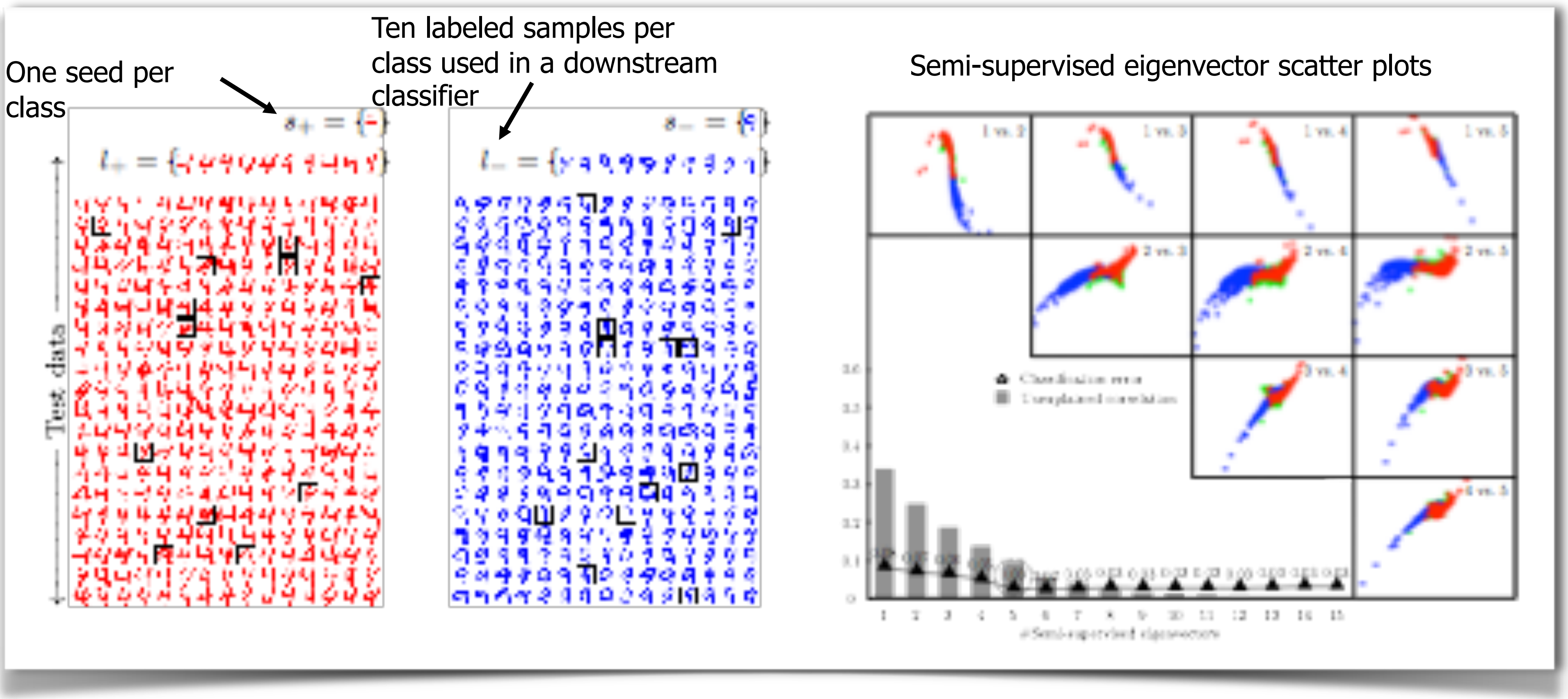
Semi-supervised eigenvectors
Low correlation

Semi-supervised eigenvectors
High correlation

Semi-supervised eigenvectors

Hansen and Mahoney (NIPS 2013, JMLR 2014)

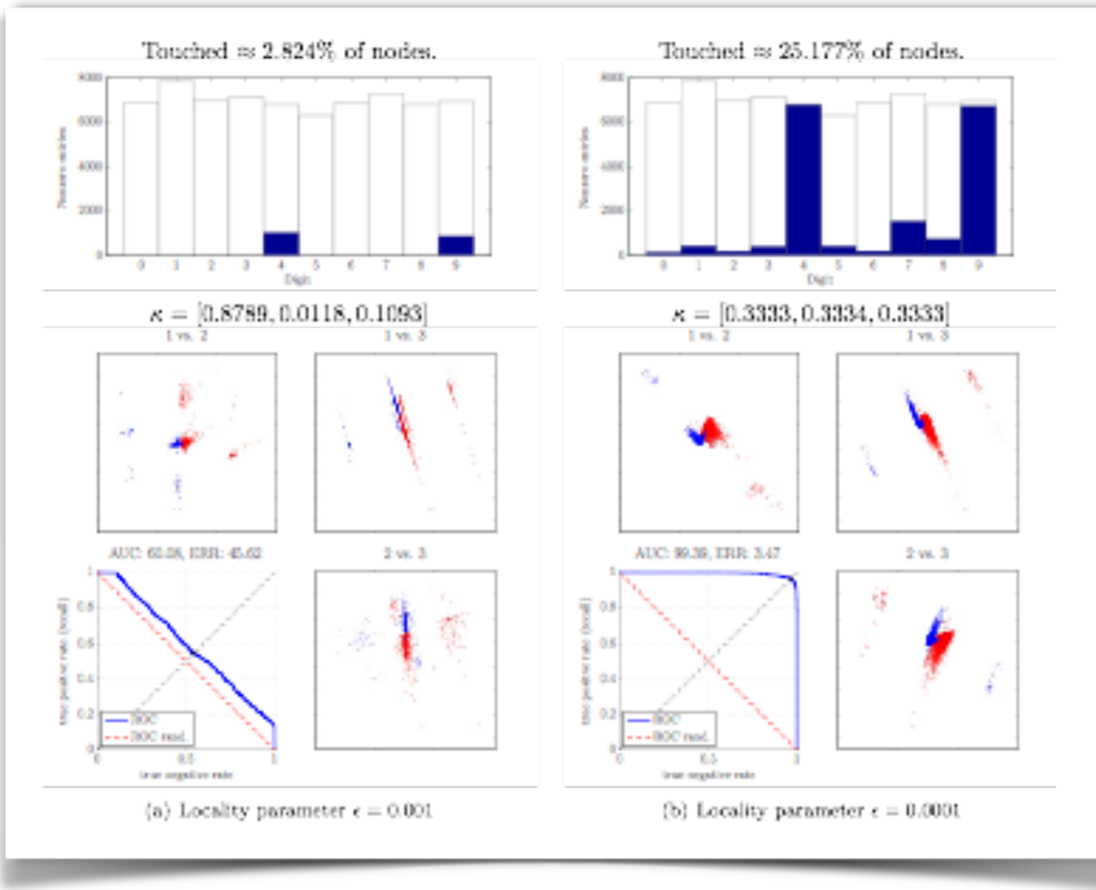
Semi-supervised learning example - Discard the majority of the labels from MNIST dataset. We seek a basis in which we can discriminate between *fours* and *nines.*



Semi-supervised eigenvectors

Hansen and Mahoney (NIPS 2013, JMLR 2014)

Localization/approximation of the Push algorithm is controlled by the ϵ parameter that defines a threshold for propagating mass away from the seed set.



Semi-supervised eigenvectors

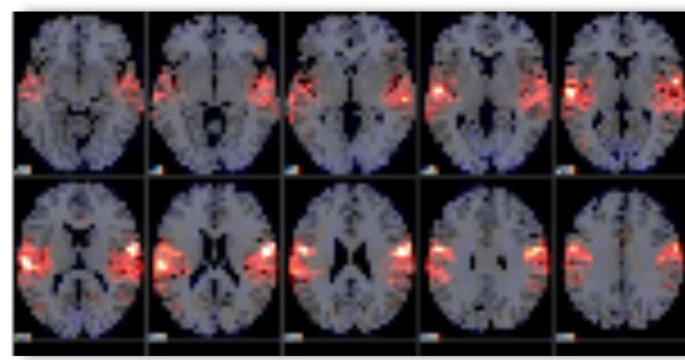
Hansen and Mahoney (NIPS 2013, JMLR 2014)

Methodology to construct semi-supervised eigenvectors of a graph, *i.e.*, local analogues of the global eigenvectors.

- Efficient to compute
- Inherit many nice properties that characterizes global eigenvectors of a graph
- Larger-scale: couples cleanly with Nystrom-based low-rank approximations
- Larger-scale: couples with local graph diffusions
- Code is available at: https://sites.google.com/site/tokejansenhansen/

Many applications:

- A spatially guided "searchlight" technique that compared to [Kriegeskorte2006] account for spatially distributed signal representations.
- Local structure in astronomical data
- Large-scale and small-scale structure in DNA SNP data in population genetics



Outline

Motivation: large informatics graphs

- Downward-sloping, flat, and upward-sloping NCPs (i.e., not "nice" *at large size scales*, but instead expander-like/tree-like)
- Implicit regularization in graph approximation algorithms

Eigenvector localization & semi-supervised eigenvectors

- Strongly and weakly local diffusions
- Extension to semi-supervised eigenvectors

Implicit regularization & algorithmic anti-differentiation

- Early stopping in iterative diffusion algorithms
- Truncation in diffusion algorithms

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)

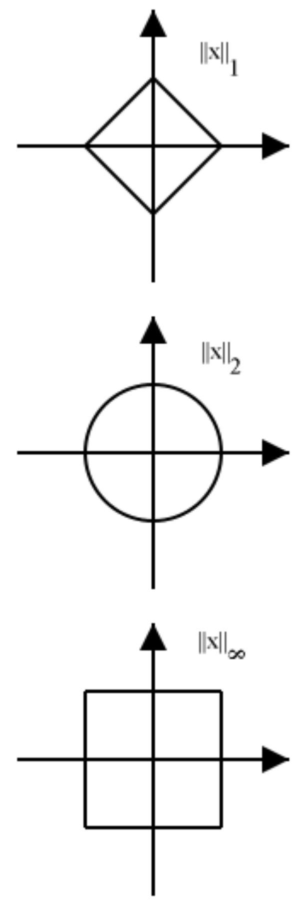
Usually *implemented* in 2 steps:

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

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

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?

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, $v_t = \Sigma_i \gamma_i^t v_i \rightarrow v_1$.

- 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 $R(A,x) = x^T A x / x^T x$, 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:**

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

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

↓ ↓

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.

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

Three simple corollaries

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

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

$F_D(X) = -\text{logdet}(X)$ (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*!

Implicit Regularization and Algorithmic Anti-differentiation

Gleich and Mahoney (2014)

The Ideal World

Given: Problem P
Derive: solution characterization C

Show: algorithm A finds a solution where C holds

Publish, Profit?

Given: "min-cut"
Derive: "max-flow is equivalent to min-cut"

Show: push-relabel solves max-flow

Publish, Profit!

Implicit Regularization and Algorithmic Anti-differentiation

Gleich and Mahoney (2014)

(The Ideal World)′

Given: Problem P
Derive: *approximate* solution characterization *C′*

Show: algorithm *A′ quickly* finds a solution where *C′* holds

Publish, Profit?

Given: "sparsest-cut"
Derive: Rayleigh-quotient approximation

Show: power-method finds a good Rayleigh-quotient

Publish, Profit!

Implicit Regularization and Algorithmic Anti-differentiation

Gleich and Mahoney (2014)

The Real World

Given: *Ill-defined task P*
Hack around until you find something useful

Write paper presenting "novel heuristic" H for P and …

Publish, Profit …

Given: "find communities"
Hack around with details buried in code & never described
Write paper describing novel community detection method that finds hidden communities

Publish, Profit …

Implicit Regularization and Algorithmic Anti-differentiation

Gleich and Mahoney (2014)

Given heuristic H, is there a problem P′ such that H is an algorithm for P′ ?

Understand why H works

Show *heuristic H solves problem P′*

Guess and check until you find something H solves

Given: "find communities"

Hack around until you find some useful heuristic H

Derive characterization of heuristic H

E.g., Mahoney and Orecchia implicit regularization results.

Implicit Regularization and Algorithmic Anti-differentiation

Gleich and Mahoney (2014)

Given heuristic H, is there a problem P' such that H is an algorithm for P' ?

If your algorithm is related to optimization, this is:

Given a procedure H, what objective does it optimize?

In an unconstrained case, this is:

Just "anti-differentiation"!!

- *Just as anti-differentiation is harder than differentiation, expect that algorithmic anti-differentiation to he harder than algorithm design.*
- *These details matter in many empirical studies, and can dramatically impact performance (speed or quality)*
- *Can we get a suite of scalable primitives to "cut and paste" to obtain goos algorithmic and good statistical properties?*

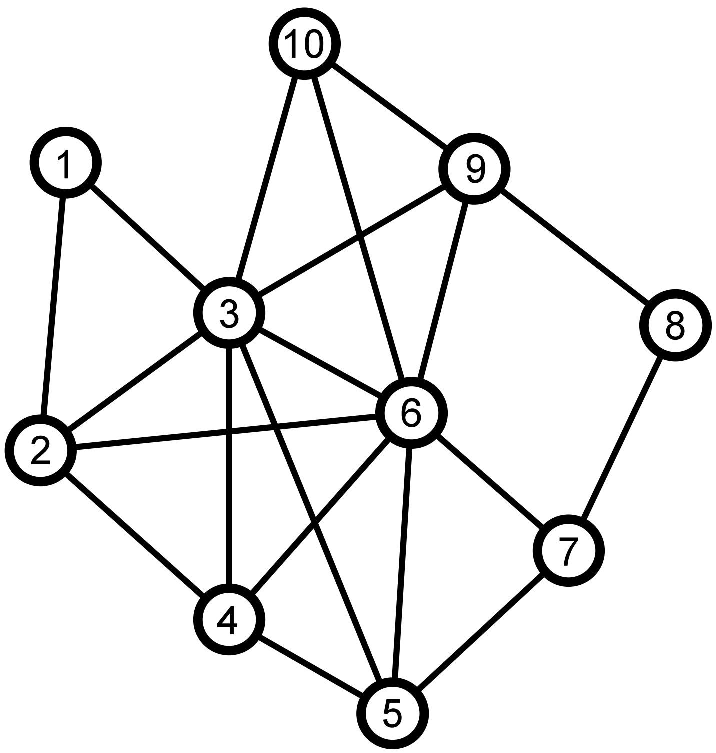
Application: new connections between PageRank, spectral, and localized flow

Gleich and Mahoney (2014)

- A new derivation of the PageRank vector for an undirected graph based on Laplacians, cuts, or flows
- A new understanding of the "push" methods to compute Personalized PageRank
- An empirical improvement to methods for semi-supervised learning

- Explains remarkable empirical success of "push" methods
- An example of algorithmic anti-differentiation

The PageRank problem/solution



- The PageRank random surfer

1. With probability beta, follow a random-walk step
2. With probability (1-beta), jump randomly ~ dist. .

- **Goal:** find the stationary dist. $\mathbf{x}$

$$\mathbf{x} = \beta \mathbf{A}\mathbf{D}^{-1}\mathbf{x} + (1-\beta)\mathbf{v}$$

- **Alg:** Solve the linear system

$$(\mathbf{I} - \beta \mathbf{A}\mathbf{D}^{-1})\mathbf{x} = (1-\beta)\mathbf{v}$$

Symmetric adjacency matrix

Diagonal degree matrix

Solution

Jump vector

PageRank and the Laplacian

1. $(\mathbf{I} - \beta \mathbf{A}\mathbf{D}^{-1})\mathbf{x} = (1 - \beta)\mathbf{v}$;

2. $(\mathbf{I} - \beta \mathcal{A})\mathbf{y} = (1 - \beta)\mathbf{D}^{-1/2}\mathbf{v}$,
   where $\mathcal{A} = \mathbf{D}^{-1/2}\mathbf{A}\mathbf{D}^{-1/2}$ and $\mathbf{x} = \mathbf{D}^{1/2}\mathbf{y}$; and

3. $[\alpha \mathbf{D} + \mathbf{L}]\mathbf{z} = \alpha \mathbf{v}$ where $\beta = 1/(1+\alpha)$ and $\mathbf{x} = \mathbf{D}\mathbf{z}$.

Combinatorial Laplacian

Push Algorithm for PageRank

- Proposed (in closest form) in Andersen, Chung, Lang (also by McSherry, Jeh & Widom) for *personalized PageRank*
  - Strongly related to Gauss-Seidel (see Gleich's talk at Simons for this)
- Derived to show improved runtime for balanced solvers

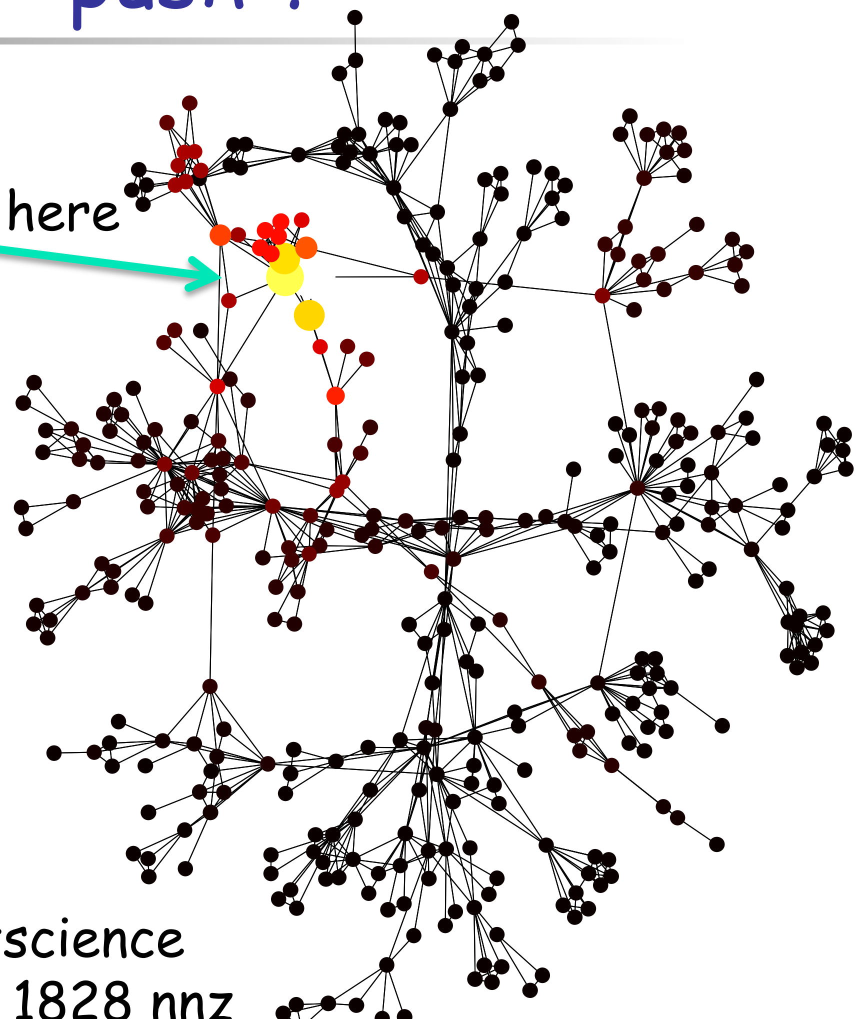
The Push Method τ, ρ

1. $\mathbf{x}^{(1)} = 0, \mathbf{r}^{(1)} = (1-\beta)\mathbf{e}_i, k = 1$
2. *while* any $r_j > \tau d_j$ (*d_j is the degree of node j*)
3. $\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + (r_j - \tau d_j \rho)\mathbf{e}_j$
4. $$r_i^{(k+1)} = \begin{cases} \tau d_j \rho & i = j \\ r_i^{(k)} + \beta(r_j - \tau d_j \rho)/d_j & i \sim j \\ r_i^{(k)} & \text{otherwise} \end{cases}$$
5. $k \leftarrow k + 1$

Why do we care about "push"?

1. Used for empirical studies of "communities"
2. Used for "fast PageRank" approximation

- Produces *sparse* approximations to PageRank!
- Why does the "push method" have such empirical utility?



Newman's netscience
379 vertices, 1828 nnz
"zero" on most of the nodes

Recall the s-t min-cut problem

Unweighted incidence matrix

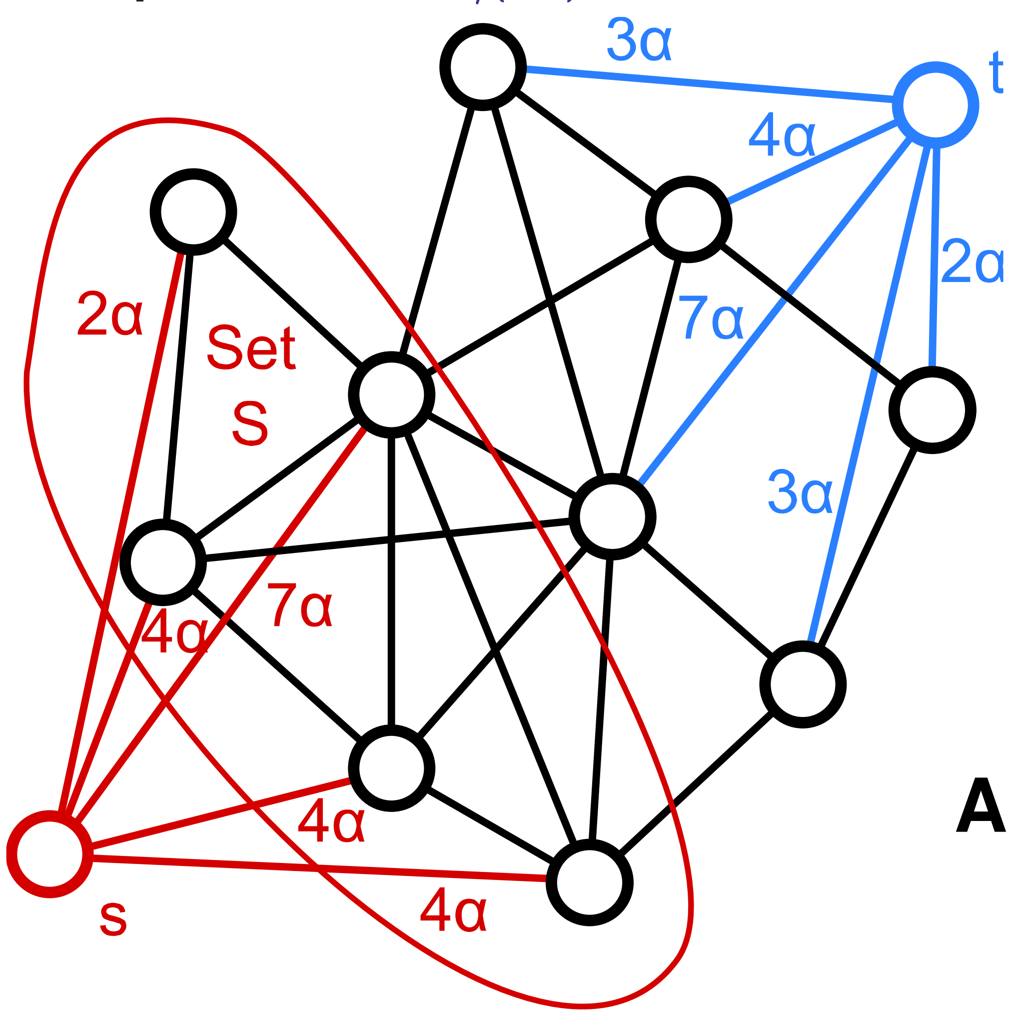
Diagonal capacity matrix

minimize $\quad \|\mathbf{Bx}\|_{C,1} = \sum_{ij \in E} C_{i,j} |x_i - x_j|$

subject to $\quad x_s = 1, x_t = 0, \mathbf{x} \geq 0.$

The localized cut graph

Gleich and Mahoney (2014)



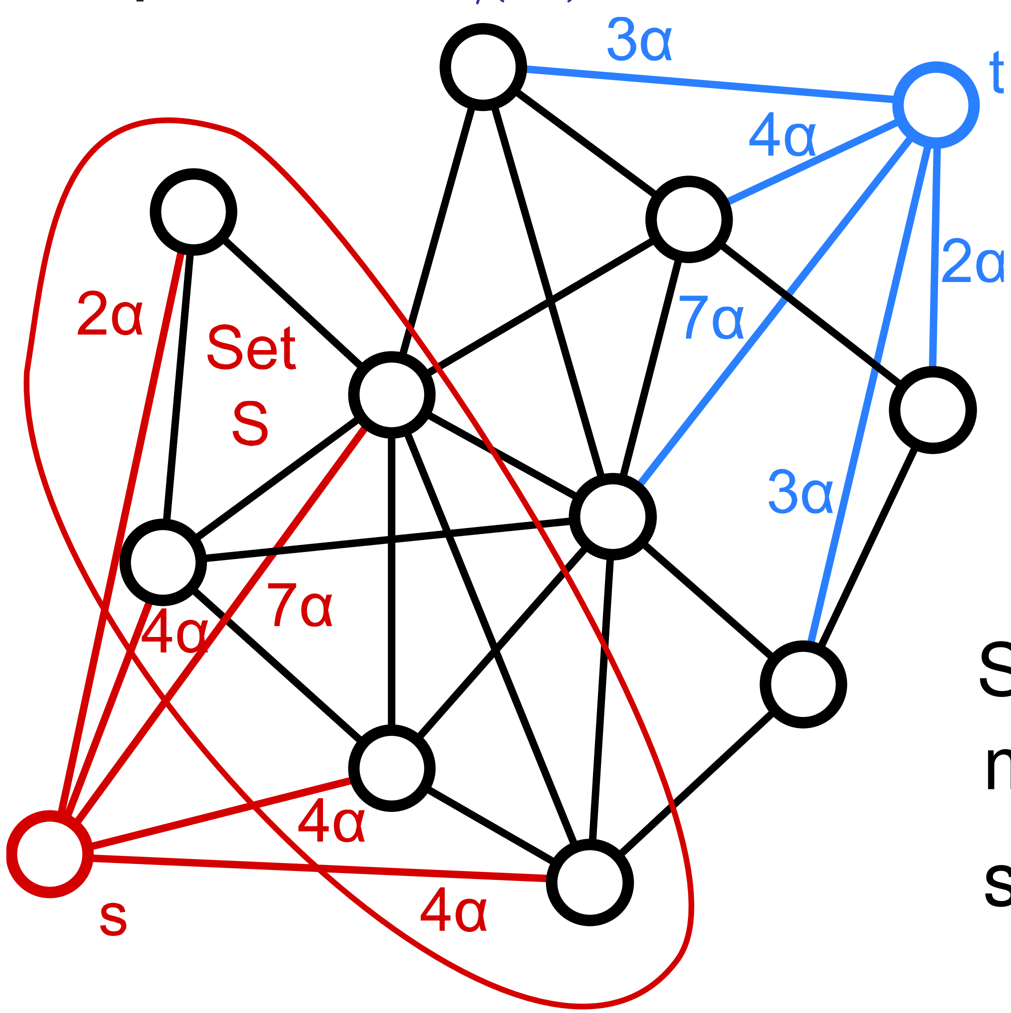
Connect s to vertices in S with weight $\alpha \cdot$ degree
Connect t to vertices in $\bar{S}$ with weight $\alpha \cdot$ degree

- Related to a construction used in "FlowImprove" Andersen & Lang (2007); and Orecchia & Zhu (2014)

$$\mathbf{A}_S = \begin{bmatrix} 0 & \alpha \mathbf{d}_S^T & 0 \\ \alpha \mathbf{d}_S & \mathbf{A} & \alpha \mathbf{d}_{\bar{S}} \\ 0 & \alpha \mathbf{d}_{\bar{S}}^T & 0 \end{bmatrix}$$

The localized cut graph

Gleich and Mahoney (2014)



Connect *s* to vertices in *S* with weight $\alpha \cdot$ degree
Connect *t* to vertices in $\bar{S}$ with weight $\alpha \cdot$ degree

$$\mathbf{B}_S = \begin{bmatrix} \mathbf{e} & -\mathbf{I}_S & 0 \\ 0 & \mathbf{B} & 0 \\ 0 & -\mathbf{I}_{\bar{S}} & \mathbf{e} \end{bmatrix}$$

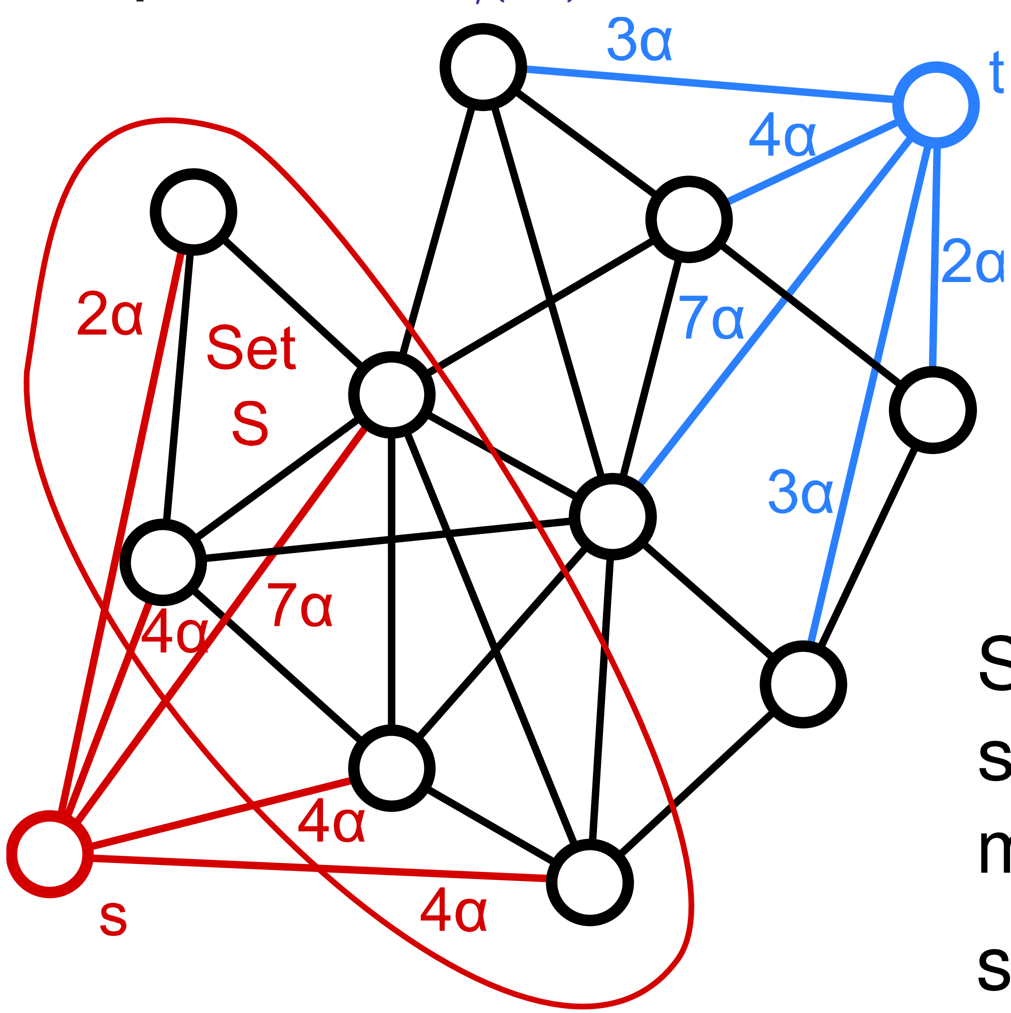
Solve the s-t min-cut

minimize $\|\mathbf{B}_S \mathbf{x}\|_{C(\alpha),1}$

subject to $x_s = 1, x_t = 0$
$\mathbf{x} \geq 0.$

The localized cut graph

Gleich and Mahoney (2014)



Connect *s* to vertices
in *S* with weight $\alpha \cdot$ degree
Connect *t* to vertices
in $\bar{S}$ with weight $\alpha \cdot$ degree

$$\mathbf{B}_S = \begin{bmatrix} \mathbf{e} & -\mathbf{I}_S & 0 \\ 0 & \mathbf{B} & 0 \\ 0 & -\mathbf{I}_{\bar{S}} & \mathbf{e} \end{bmatrix}$$

Solve the "electrical flow"
s-t min-cut

$$\text{minimize} \quad \|\mathbf{B}_S \mathbf{x}\|_{C(\alpha),2}$$

$$\text{subject to} \quad x_s = 1, x_t = 0$$

s-t min-cut -> PageRank

Gleich and Mahoney (2014)

The PageRank vector **z** that solves

$$(\alpha \mathbf{D} + \mathbf{L})\mathbf{z} = \alpha \mathbf{v}$$

with $\mathbf{v} = \mathbf{d}_S / \mathrm{vol}(S)$ is a renormalized solution of the electrical cut computation:

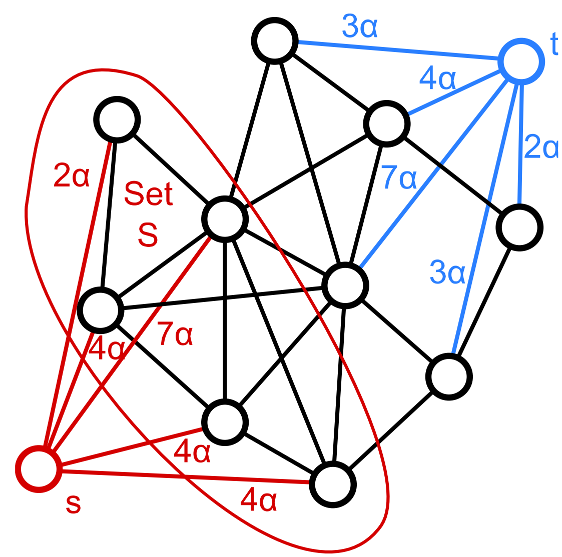
$$\begin{array}{ll} \text{minimize} & \|\mathbf{B}_S \mathbf{x}\|_{C(\alpha),2} \\ \text{subject to} & x_s = 1, x_t = 0. \end{array}$$

Specifically, if **x** is the solution, then

$$\mathbf{x} = \begin{bmatrix} 1 \\ \mathrm{vol}(S)\mathbf{z} \\ 0 \end{bmatrix}$$

Proof

Square and expand the objective into a Laplacian, then apply constraints.

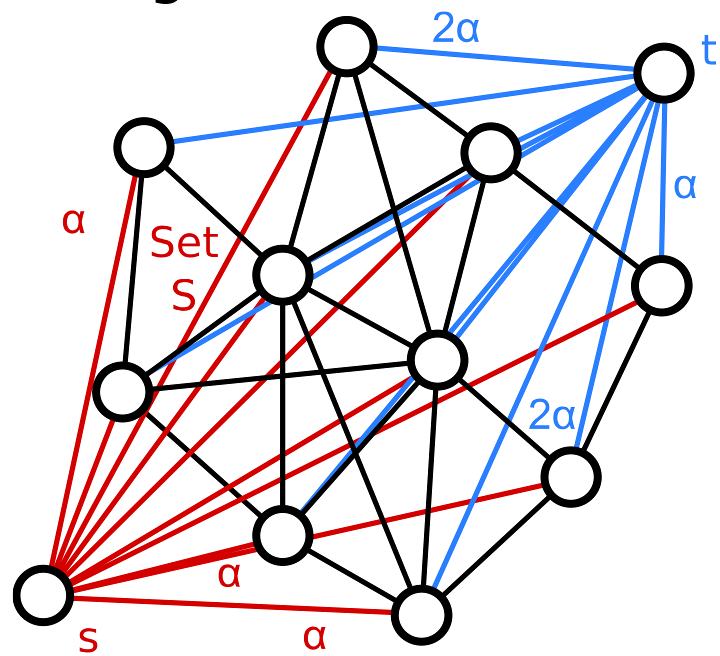


PageRank -> s-t min-cut

Gleich and Mahoney (2014)

- That equivalence works if **v** is degree-weighted.
- What if **v** is the uniform vector?

$$\mathbf{A}(\mathbf{s}) = \begin{bmatrix} 0 & \alpha\mathbf{s}^T & 0 \\ \alpha\mathbf{s} & \mathbf{A} & \alpha(\mathbf{d}-\mathbf{s}) \\ 0 & \alpha(\mathbf{d}-\mathbf{s})^T & 0 \end{bmatrix}.$$



- Easy to cook up popular diffusion-like problems and adapt them to this framework. E.g., semi-supervised learning (Zhou et al. (2004).

Back to the push method

Gleich and Mahoney (2014)

Let $\mathbf{x}$ be the output from the push method with $0 < \beta < 1$, $\mathbf{v} = \mathbf{d}_S/\text{vol}(S)$, $\rho = 1$, and $\tau > 0$.

Set $\alpha = \frac{1-\beta}{\beta}$, $\kappa = \tau\text{vol}(S)/\beta$, and let $\mathbf{z}_G$ solve:

$$\begin{array}{ll} \text{minimize} & \frac{1}{2}\|\mathbf{B}_S\mathbf{z}\|^2_{C(\alpha),2} + \kappa\|\mathbf{D}\mathbf{z}\|_1 \\ \text{subject to} & z_s = 1, z_t = 0, \mathbf{z} \geq 0 \end{array},$$

Need for normalization

Regularization for sparsity

where $\mathbf{z} = \begin{bmatrix} 1 \\ \mathbf{z}_G \\ 0 \end{bmatrix}$.

Then $\mathbf{x} = \mathbf{D}\mathbf{z}_G/\text{vol}(S)$.

Proof Write out KKT conditions
Show that the push method solves them. Slackness was “tricky”

Large-scale applications

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

• Fuxman, Tsaparas, Achan, and Agrawal (2008): random walks on query-click for automatic keyword generation

• Najork, Gallapudi, and Panigraphy (2009): carefully "whittling down" neighborhood graph makes SALSA faster and better

• Lu, Tsaparas, Ntoulas, and Polanyi (2010): test which page-rank-like implicit regularization models are most consistent with data

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

Conclusions

Motivation: large informatics graphs

- Downward-sloping, flat, and upward-sloping NCPs (i.e., not “nice” *at large size scales*, but instead expander-like/tree-like)
- Implicit regularization in graph approximation algorithms

Eigenvector localization & semi-supervised eigenvectors

- Strongly and weakly local diffusions
- Extension to semi-supervised eigenvectors

Implicit regularization & algorithmic anti-differentiation

- Early stopping in iterative diffusion algorithms
- Truncation in diffusion algorithms

MMDS Workshop on "Algorithms for Modern Massive Data Sets"

(http://mmds-data.org)

at UC Berkeley, June 17-20, 2014

Objectives:

- Address algorithmic, statistical, and mathematical challenges in modern statistical data analysis.

- Explore novel techniques for modeling and analyzing massive, high-dimensional, and nonlinearly-structured data.

- Bring together computer scientists, statisticians, mathematicians, and data analysis practitioners to promote cross-fertilization of ideas.

Organizers: M. W. Mahoney, A. Shkolnik, P. Drineas, R. Zadeh, and F. Perez

Registration is available now!