Implicit regularization in sublinear approximation algorithms

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")

Motivation (1 of 2)

- Data are medium-sized, but things we want to compute are "intractable," e.g., NP-hard or n^3 time, so develop an approximation algorithm.
- Data are large/Massive/BIG, so we can't even touch them all, so develop a sublinear approximation algorithm.

Goal: Develop an algorithm s.t.:

Typical Theorem: My algorithm is faster than the exact algorithm, and it is only a little worse.

Motivation (2 of 2)

Mahoney, "Approximate computation and implicit regularization ..." (PODS, 2012)

- Fact 1: I have not seen many examples (yet!?) where sublinear algorithms are a useful guide *for LARGE-scale "vector space" or "machine learning" analytics*

- Fact 2: I have seen real examples where sublinear algorithms are very useful, *even for rather small problems*, but their usefulness is *not* primarily due to the bounds of the Typical Theorem.

- Fact 3: I have seen examples where (both linear and sublinear) approximation algorithms yield "better" solutions than the output of the more expensive exact algorithm.

Overview for today

Consider two approximation algorithms from spectral graph theory to approximate the Rayleigh quotient $f(x)$

Roughly (more precise versions later):

- Diffuse a small number of steps from starting condition
- Diffuse a few steps and zero out small entries (a local spectral method that is sublinear in the graph size)

These approximation algorithms implicitly regularize

- They exactly solve regularized versions of the Rayleigh quotient, $f(x) + \lambda g(x)$, for familiar $g(x)$

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)

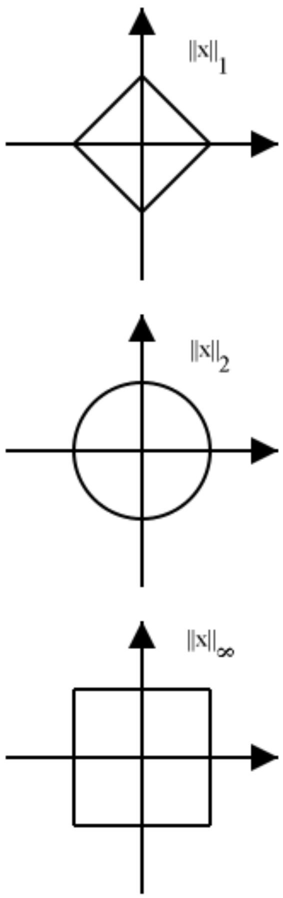
Usually *implemented* in 2 steps:

- 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?
- In general and/or for sublinear approximation algorithms?

Notation for weighted undirected graph

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

Approximating the top eigenvector

Basic idea: Given 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

Mahoney and Orecchia (2010)

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

Mahoney and Orecchia (2010)

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

Mahoney and Orecchia (2010)

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

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

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

$\downarrow$ $\downarrow$

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)

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

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.

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

Three simple corollaries

Mahoney and Orecchia (2010)

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

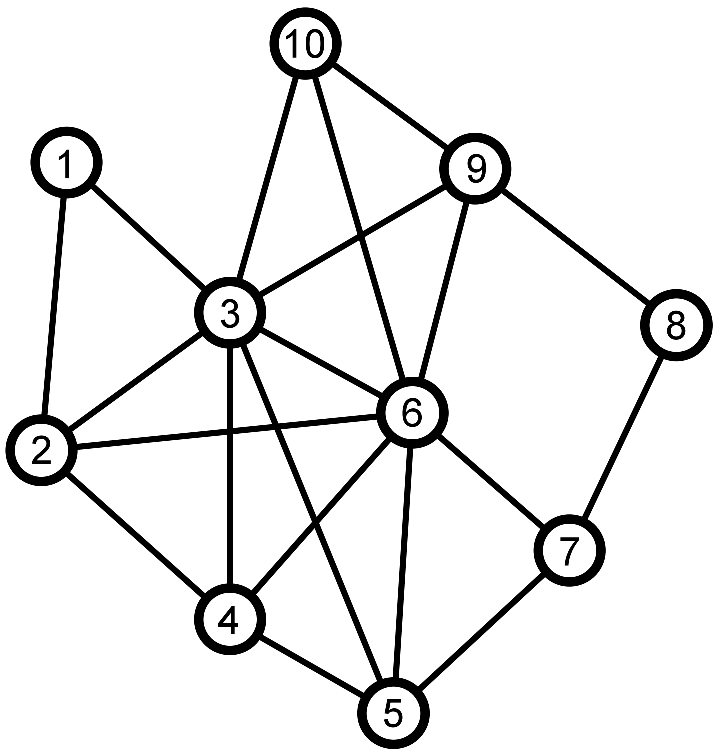
$F_p(X) = (1/p)||X||_p^p$ (i.e., matrix p-norm, for $p>1$)

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*!

Spectral algorithms and the PageRank problem/solution



- The PageRank random surfer
  1. With probability β, follow a random-walk step
  2. With probability (1-β), jump randomly ~ dist. **v**
- **Goal:** find the stationary dist. **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

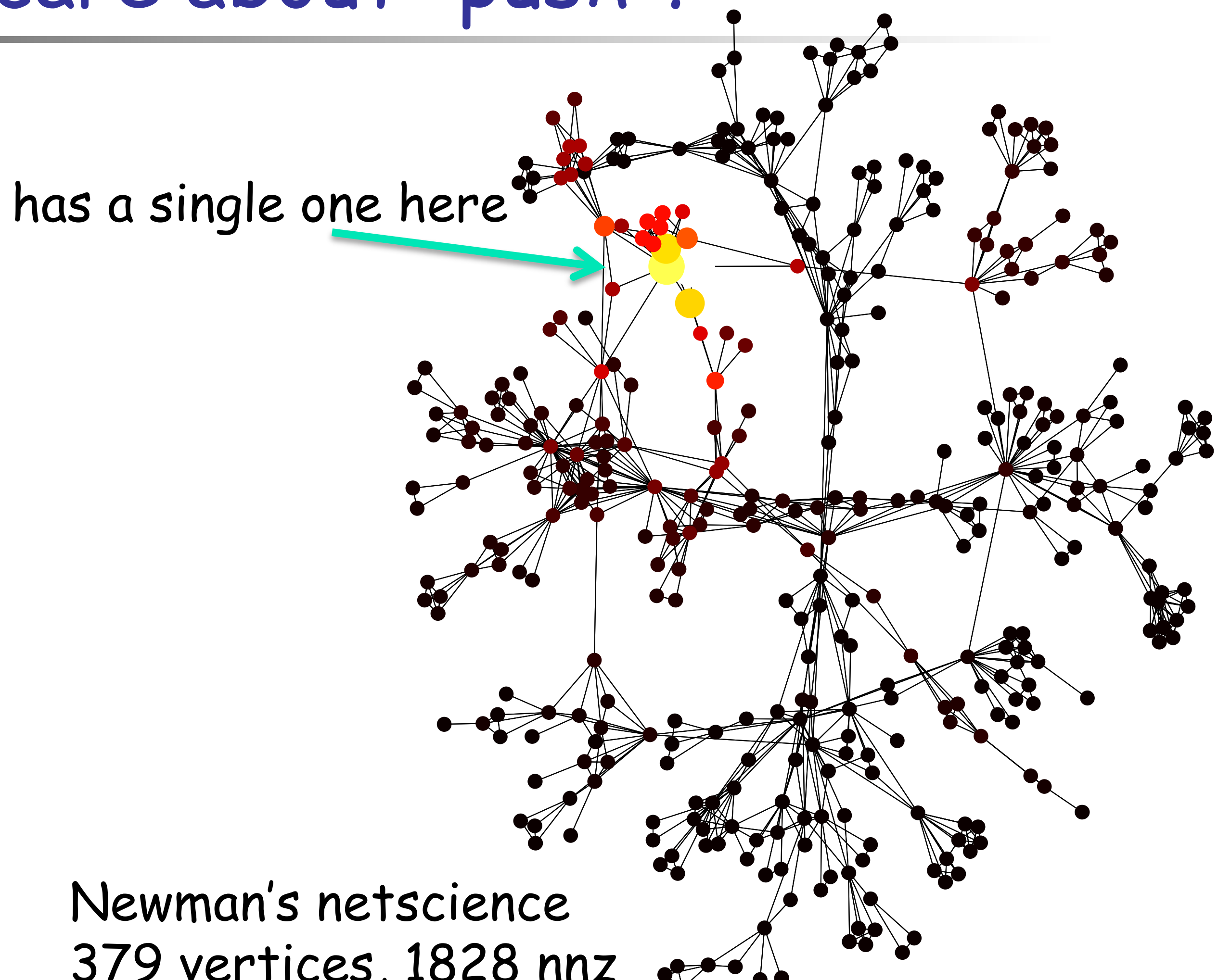
The Push Method τ, ρ

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

New connections between PageRank, spectral methods, localized flow, and sparsity inducing regularization terms

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

- The "push" method is a sublinear algorithm with an implicit regularization characterization ...
- ...that "explains" it remarkable empirical success.

The s-t min-cut problem

Unweighted incidence matrix

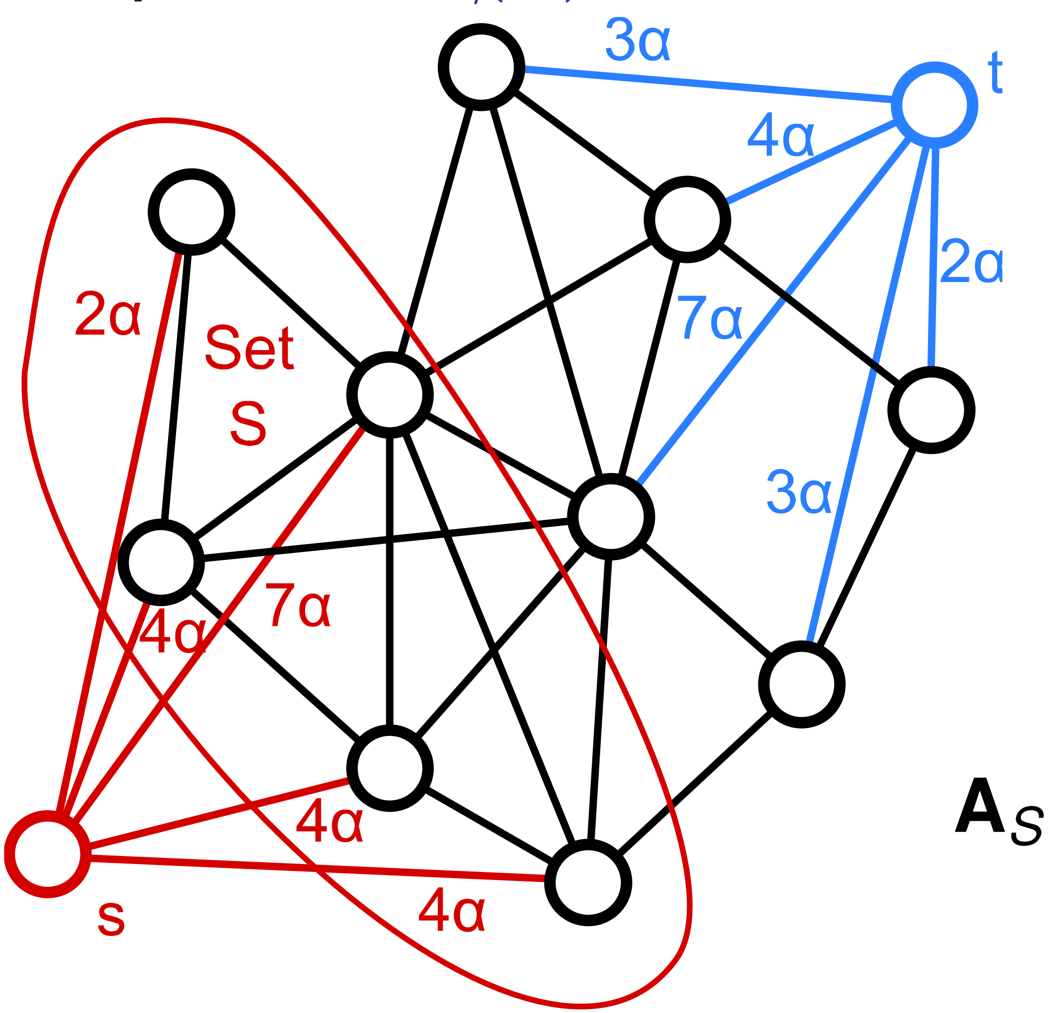
Diagonal capacity matrix

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

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

The localized cut graph

Gleich and Mahoney (2014)



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

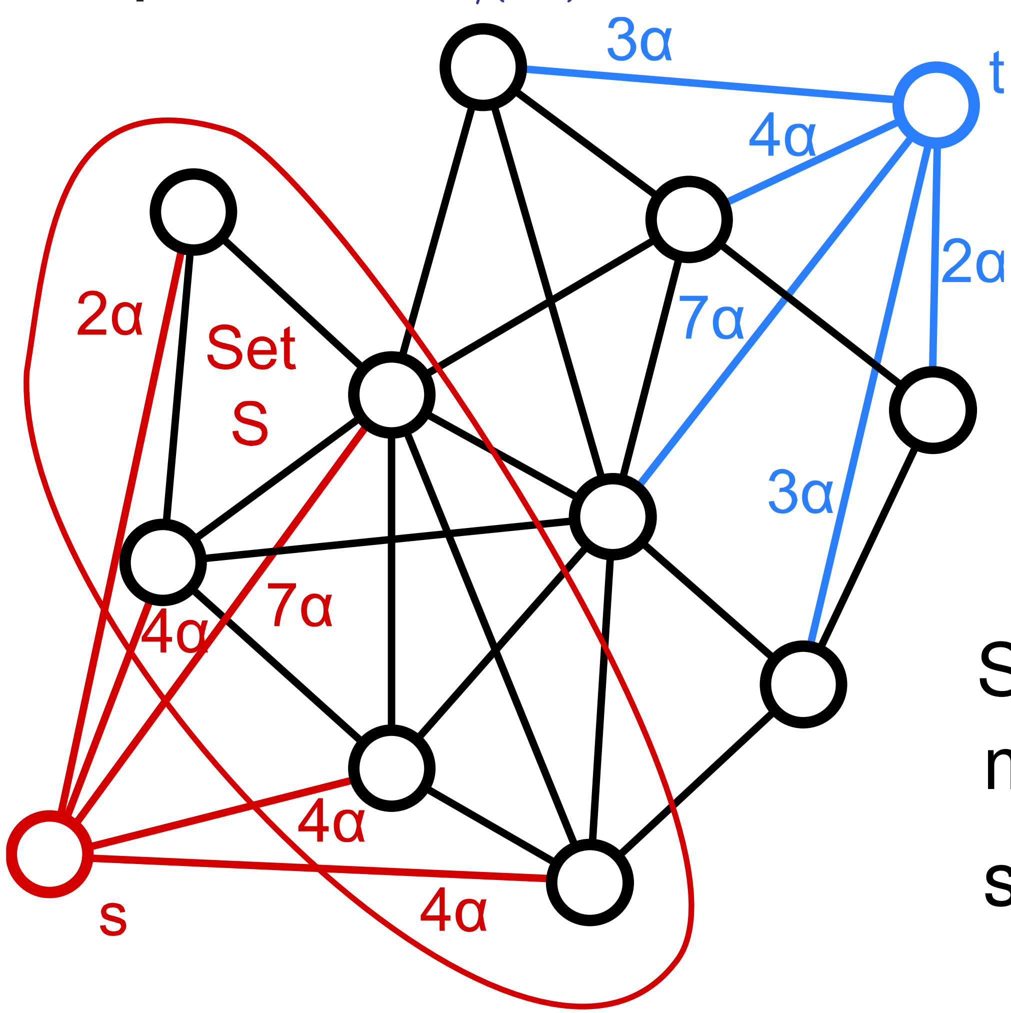
Connect *t* to vertices in $\bar{S}$ with weight $\alpha \cdot \text{degree}$

- 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

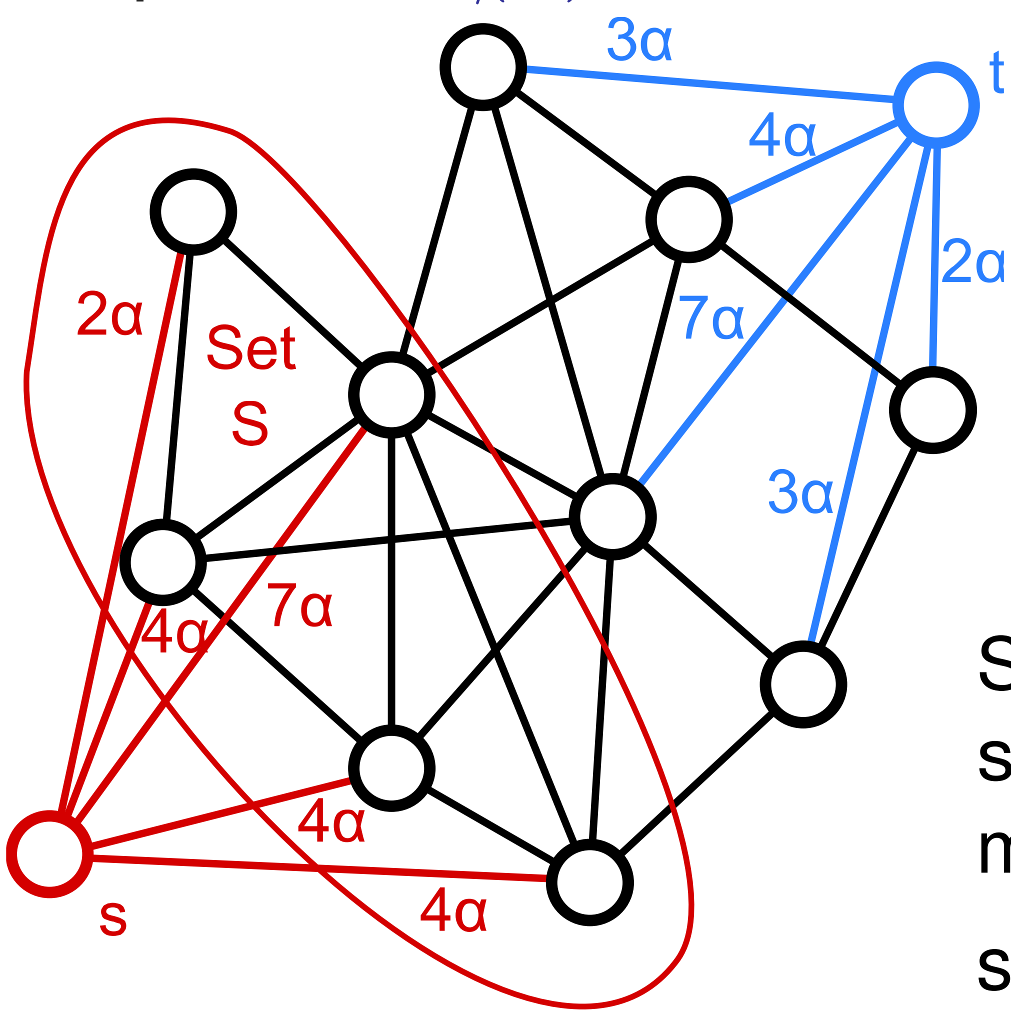
$$\mathbf{B}_S = \begin{bmatrix} \mathbf{e} & -\mathbf{I}_S & 0 \\ 0 & \mathbf{B} & 0 \\ 0 & -\mathbf{I}_{\bar{S}} & \mathbf{e} \end{bmatrix}$$

Solve the s-t min-cut

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

The localized cut graph

Gleich and Mahoney (2014)



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

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

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

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

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:

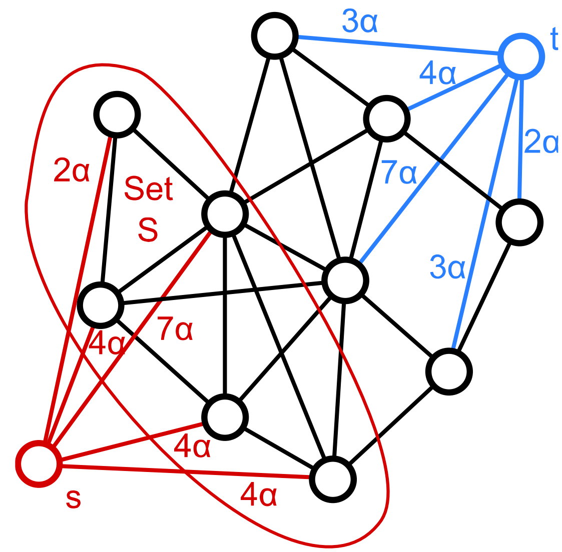
$$\begin{array}{ll} \text{minimize} & \|\mathbf{B}_S \mathbf{x}\|_{C(\alpha),2} \\ \text{subject to} & x_s = 1, x_t = 0. \end{array}$$

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.

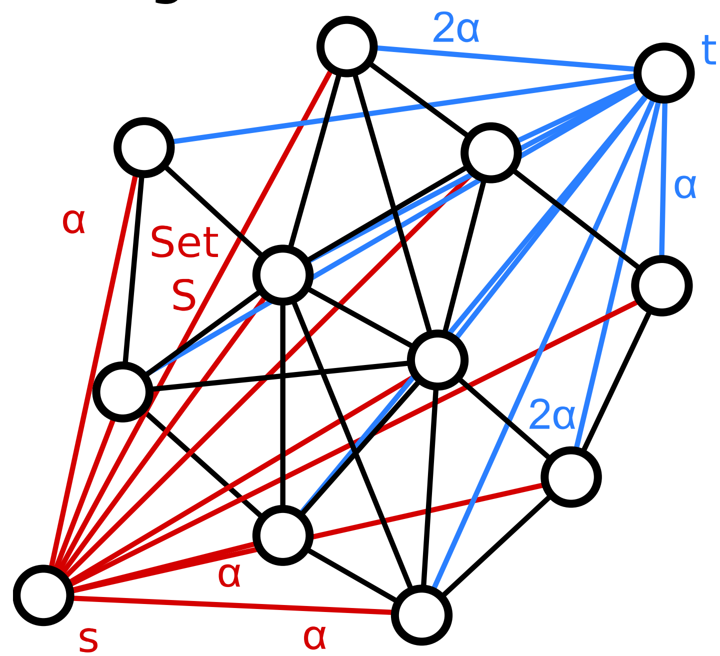


PageRank -> s-t min-cut

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: sparsity-inducing regularization

Gleich and Mahoney (2014)

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

Set $\alpha = \frac{1-\beta}{\beta}$, $\kappa = \tau \mathrm{vol}(S)/\beta$, 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 $\mathbf{x} = \mathbf{D}\mathbf{z}_G/\mathrm{vol}(S)$.

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

Conclusions

Characterize of the solution of a sublinear graph approximation algorithm in terms of an implicit sparsity-inducing regularization term.

How much more general is this in sublinear algorithms?

Characterize the implicit regularization properties of a (non-sublinear) approximation algorithm, in and of iteslf, in terms of regularized SDPs.

How much more general is this in approximation 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!