Local diffusion algorithms for fast, personalized graph applications

David F. Gleich
CS Department

Kyle Kloster
Math Department

PURDUE UNIVERSITY

supported by
NSF CAREER CCF-1149756,
IIS-IS-1422918, DARPA SIMPLEX,

Brief outline

Introduction & Application 20-30 min
Coordinate Relaxation for Strong Convergence 30 min
Break 15 min

Katz Diffusion 10 min
Weak Convergence for PageRank 20 min
Monte Carlo methods 15 min
Break 15 min

Implicit Regularization 25 min
Discussion 15 min

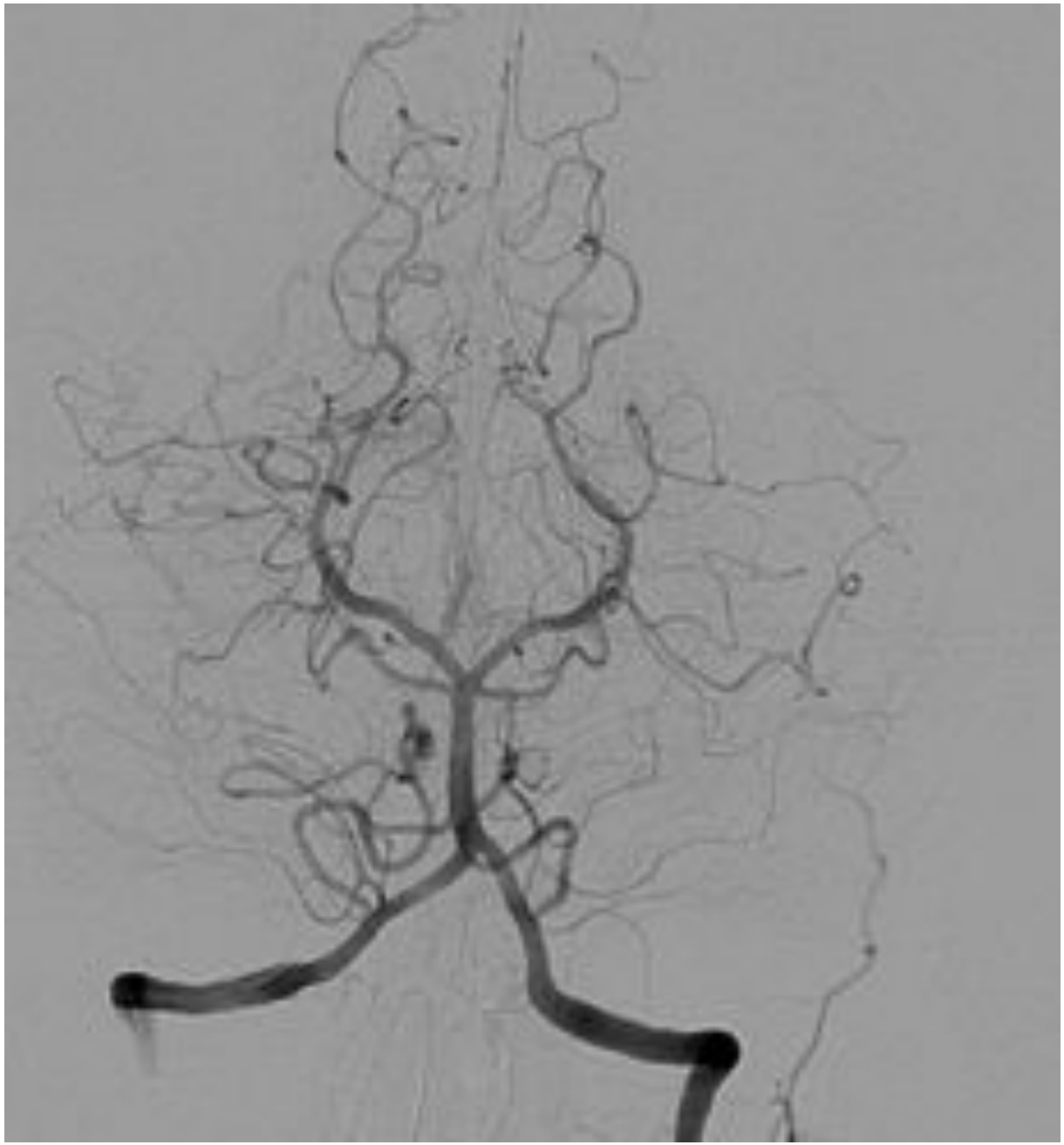
Brief outline

Introduction & Application 20-30 min
Coordinate Relaxation for Strong Convergence 30 min
Break 15 min

Katz Diffusion 10 min
Weak Convergence for PageRank 20 min
Monte Carlo methods 15 min
Break 15 min

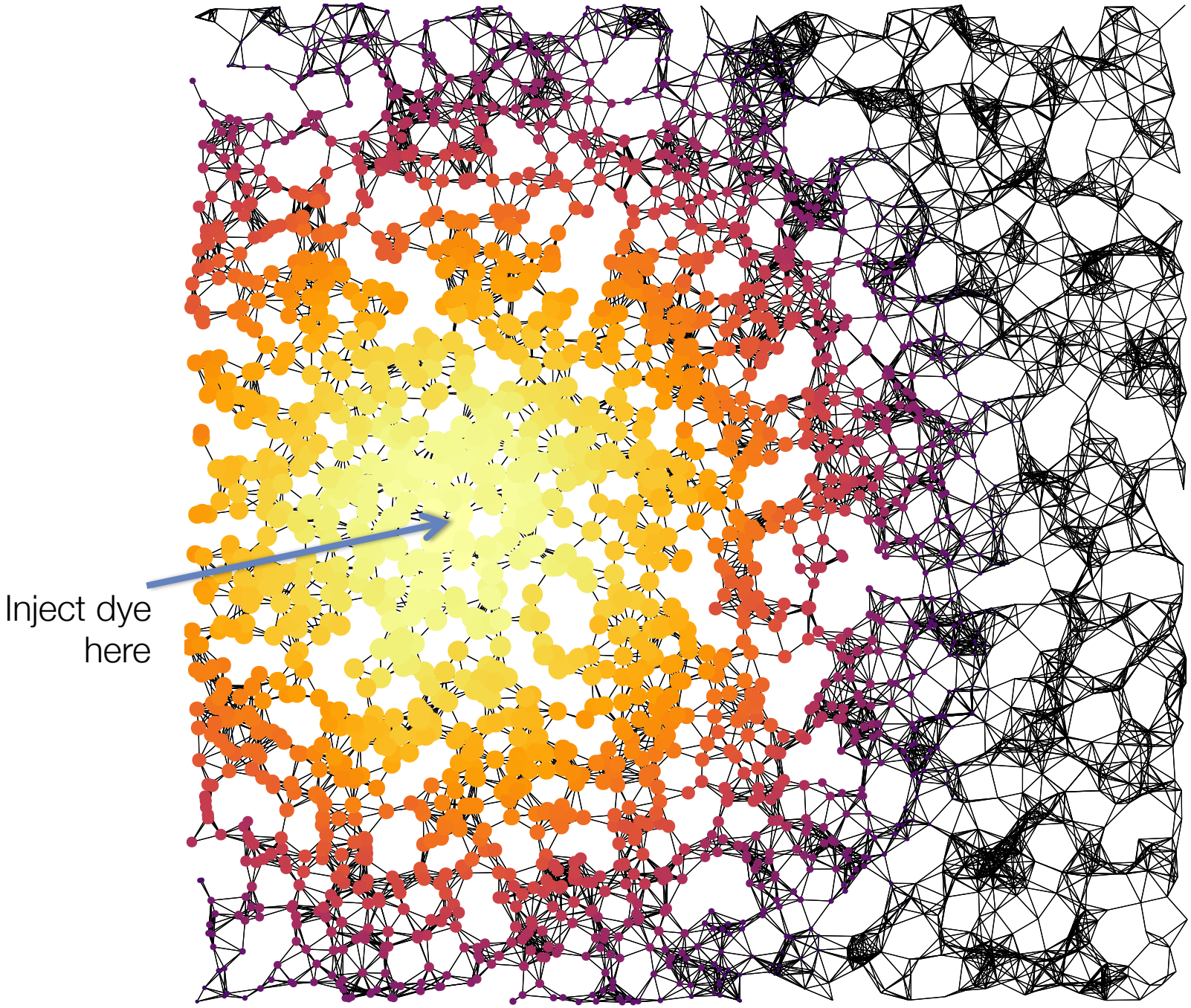
Implicit Regularization 25 min
Discussion 15 min

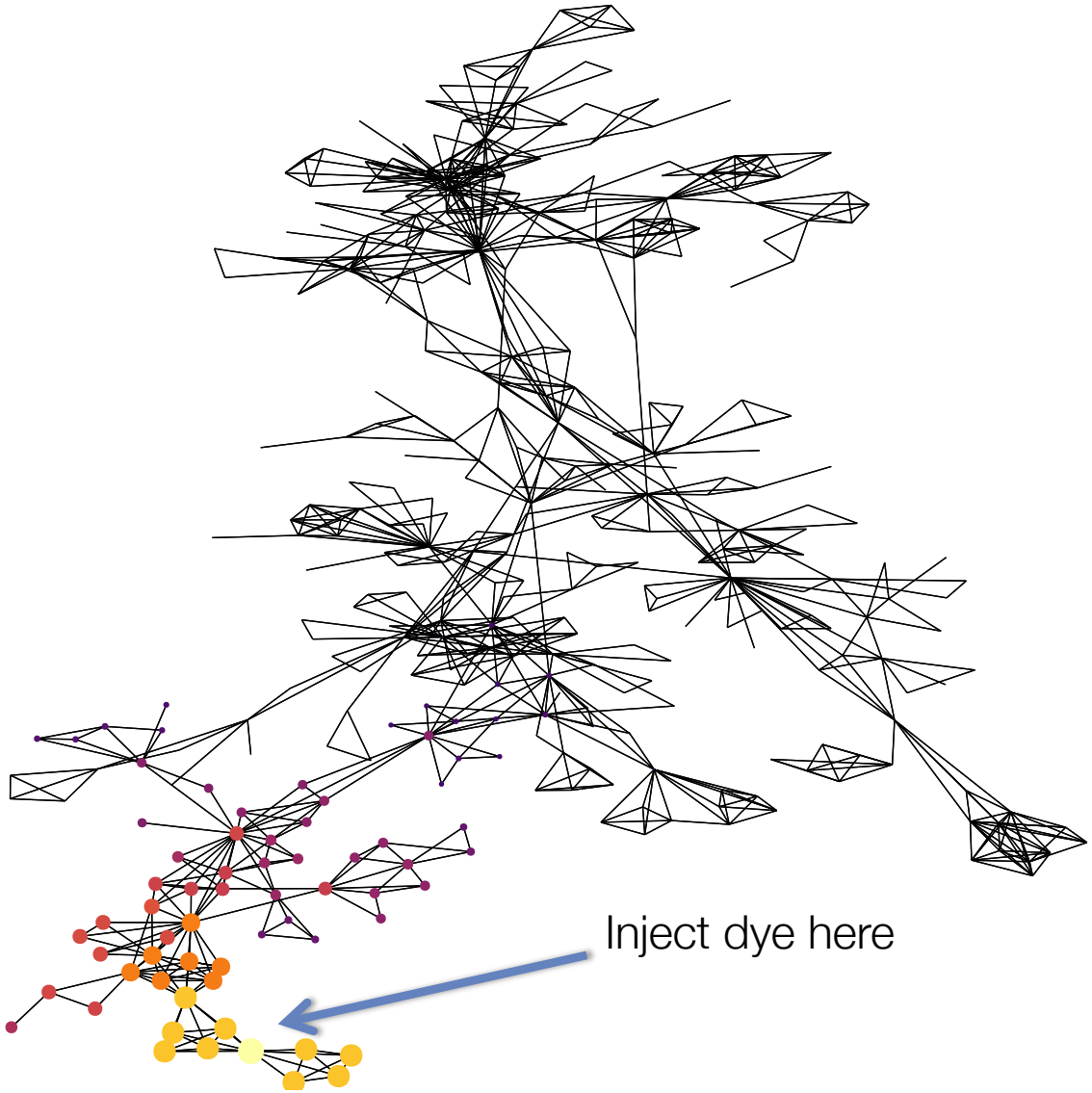
http://education.seattlepi.com/effect-kinetic-energy-relates-diffusion-rate-4870.html



Wikipedia: Angiography

Let $G = (V, E)$ ...


Inject dye
here


Inject dye here

… work out sample diffusion on the board ...

Examples

Everything in the world can be explained by a matrix, and we see how deep the rabbit hole goes

The talk ends, you believe -- whatever you want to.

Image from rockysprings, deviantart, CC share-alike

MATRIX COMPUTATIONS

Matrices derived from networks

adjacency	$\boldsymbol{A}$		
diag degree matrix	$\boldsymbol{D}$	$d_{ii} = \deg$	Directed
random walk	$\boldsymbol{P}$	$= \boldsymbol{A}^T \boldsymbol{D}^{-1}$	
Laplacian	$\boldsymbol{L}$	$= \boldsymbol{D} - \boldsymbol{A}$	
normalized Laplacian	$\mathcal{L}$	$= \boldsymbol{D}^{-1/2}(\boldsymbol{D} - \boldsymbol{A})\boldsymbol{D}^{-1/2}$	Undirected
		$= \boldsymbol{I} - \boldsymbol{D}^{-1/2}\boldsymbol{A}\boldsymbol{D}^{-1/2}$	
normalized adjacency	$\mathcal{A}$	$= \boldsymbol{D}^{-1/2}\boldsymbol{A}\boldsymbol{D}^{-1/2}$	

A general diffusion definition

You'll see this again, Kyle has more.

$$\mathbf{f} = \sum_{k=0}^{\infty} c_k \boldsymbol{M}^k \mathbf{s}$$ Source

PageRank	Katz scores	Heat kernel
$\mathbf{f} = \sum_{k=0}^{\infty} (1-\alpha)\alpha^k \boldsymbol{P}^k \mathbf{s}$	$\mathbf{f} = \sum_{k=0}^{\infty} (1-\alpha)\alpha^k \boldsymbol{A}^k \mathbf{s}$	$\mathbf{f} = \sum_{k=0}^{\infty} e^{-t} \frac{t^k}{t!} \boldsymbol{A}^k \mathbf{s}$
$(\boldsymbol{I} - \alpha \boldsymbol{P})\mathbf{f} = (1-\alpha)\mathbf{s}$	$(\boldsymbol{I} - \alpha \boldsymbol{A})\mathbf{f} = (1-\alpha)\mathbf{s}$	$\mathbf{f} = \exp\{-t(\boldsymbol{I} - \boldsymbol{P})\}\mathbf{s}$

Katz, 1953; Page et al. 1999; ??? For Heat Kernel

… not just me ...

Adjacency Kernels

$$
\begin{aligned}
F_{\mathrm{EXP}}(\mathbf{A}) &= \exp(\alpha \mathbf{A}) \\
F_{\mathrm{EXP}}(\mathcal{A}) &= \exp(\alpha \mathcal{A}) \\
F_{\mathrm{NEU}}(\mathbf{A}) &= (I - \alpha \mathbf{A})^{-1} \\
F_{\mathrm{NEU}}(\mathcal{A}) &= (I - \alpha \mathcal{A})^{-1}
\end{aligned}
$$

Laplacian Kernels

$$
\begin{aligned}
F_{\mathrm{HEAT}}(\mathbf{L}) &= exp(-\alpha \mathbf{L}) \\
F_{\mathrm{HEAT}}(\mathcal{L}) &= exp(-\alpha \mathcal{L})
\end{aligned}
$$

$$
\begin{aligned}
F_{\mathrm{COMR}}(\mathbf{L}) &= (\mathbf{I} + \alpha \mathbf{L})^{-1} \\
F_{\mathrm{COMR}}(\mathcal{L}) &= (\mathbf{I} + \alpha \mathcal{L})^{-1}
\end{aligned}
$$

$$
\begin{aligned}
F_{\mathrm{COM}}(\mathbf{L}) &= \mathbf{L}^{+} \\
F_{\mathrm{COM}}(\mathcal{L}) &= \mathcal{L}^{+}
\end{aligned}
$$

Kunegis & Lommatzsch, ICML 2009

… demo …

github.com/dgleich/diffusion-tutorial

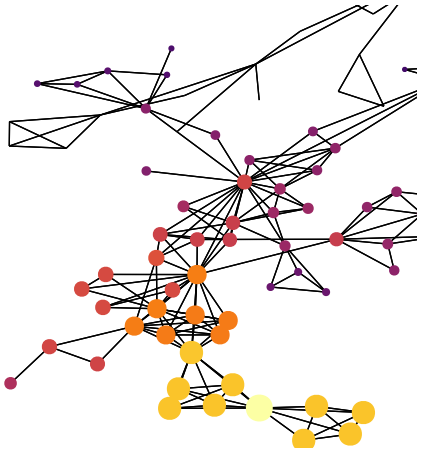
Applications

Applications of localized and personalized diffusion

Seeds → Scores → “Learning”

Graph Kernels

- Kondor & Lafferty, ICML 2002

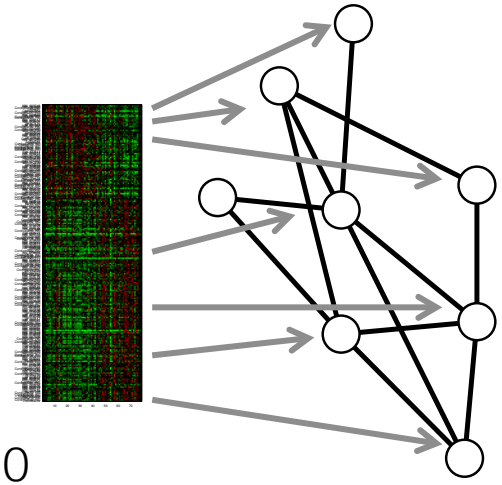


Link prediction

- Liben-Nowell & Kleinberg, 2005, 2006
- Kunegis, Lommatzsch, 2009
- … more ...

“Attribute prediction” “semi-dense vectors”

- GeneRank, Morrison et al. 2005
- ProteinRank, Freschi 2007
- Genes for cancer, Winter et al. 2012
- Cross-modal discovery, Pan et al. 2004
- Global information diffusion, Venner et al. 2010



PageRank beyond the Web

SIAM Review, Sept. 2015

$$(\mathbf{I} - \alpha \mathbf{P})\mathbf{x}$$

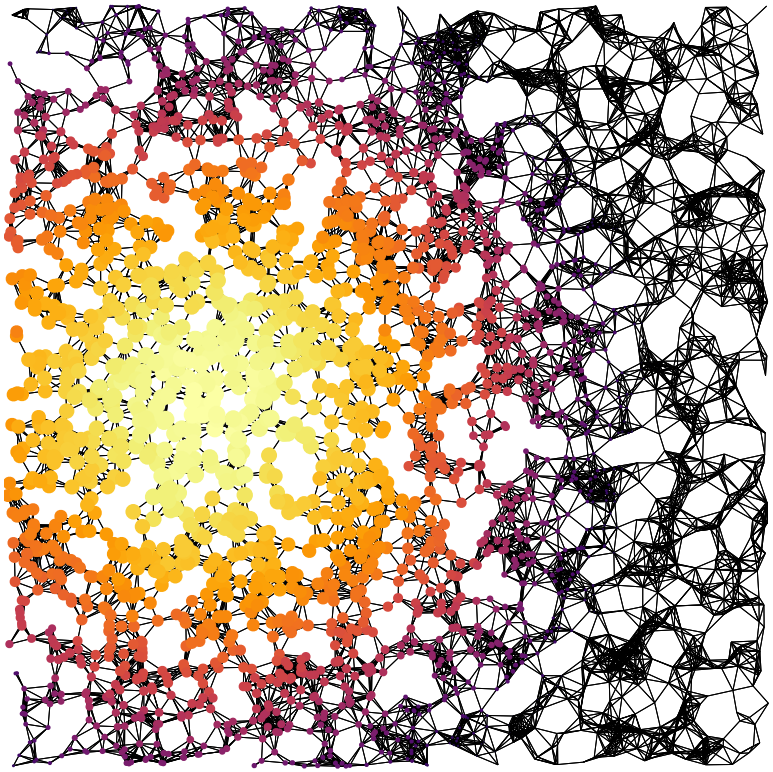
Vertex similarity

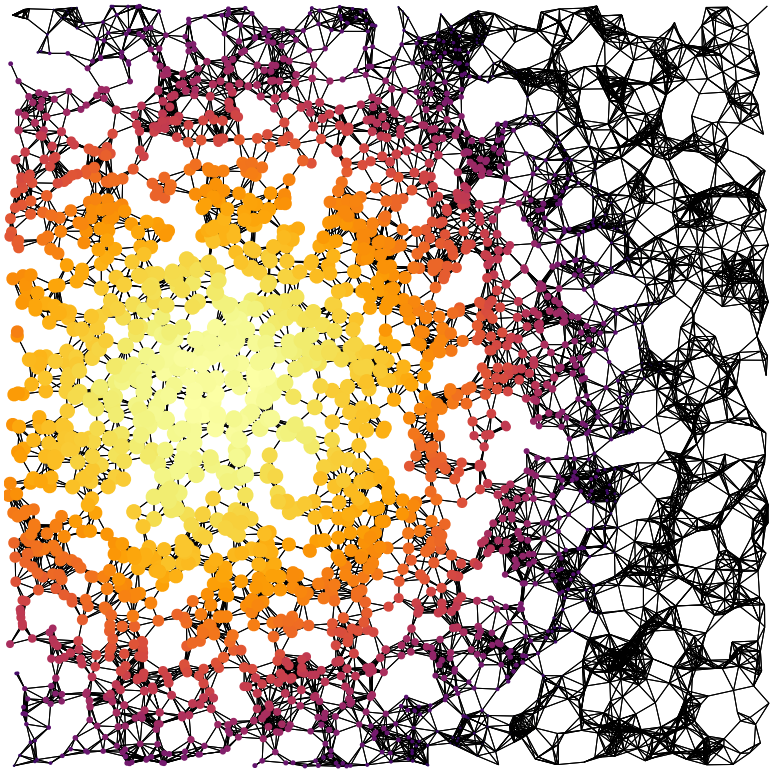
Diffusions in Open-Directory graph used for *semantic relatedness*

1. Reverse the direction of all edges
2. Compute the seeded PageRank matrix
   e.g. “diffuse” from all individual seeds
3. Compute cosine distances between columns.

Bar-Yossef, Ziv and Mashiach, Li-Tal, 2008

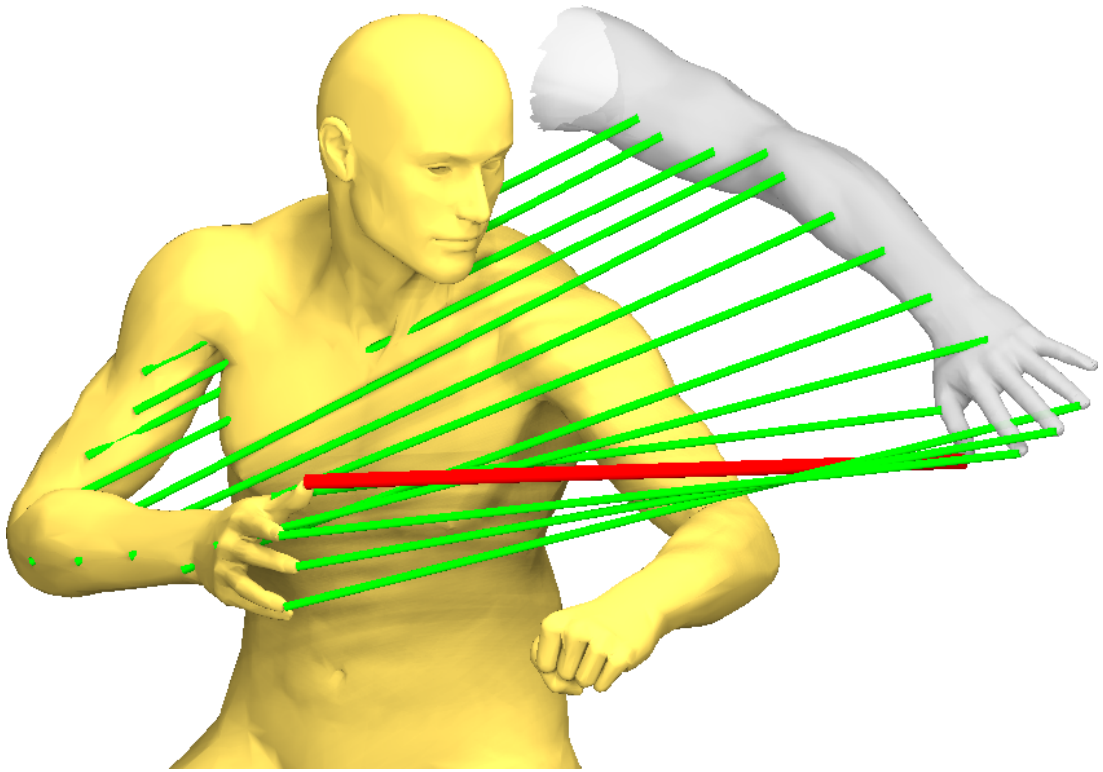
Vertex similarity





$$\|\mathbf{x} - \mathbf{y}\|_{\text{👁}} \approx 0$$

Vertex similarity

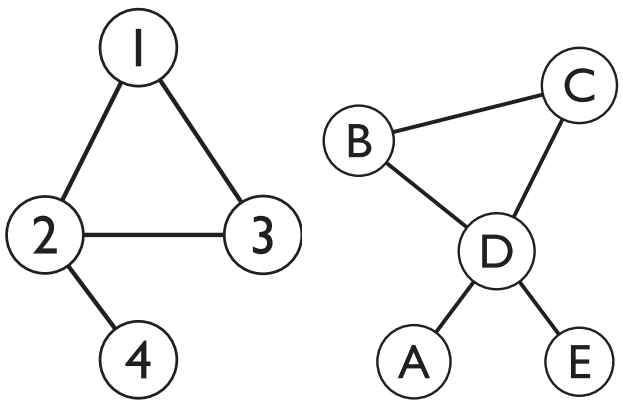


M. Ovsjanikov, Q. Mérigot, F. Mémoli, and L. Guibas, One Point Isometric Matching with the Heat Kernel, Proc. Eurographics Symposium on Geometry Processing (SGP) 2010.

Network alignment & near isomorphisms

IsoRank

Diffuse from all potential matches through the Kronecker graph



$$\mathbf{P} = \begin{bmatrix} 0 & 1/3 & 1/2 & 0 \\ 1/2 & 0 & 1/2 & 1 \\ 1/2 & 1/3 & 0 & 0 \\ 0 & 1/3 & 0 & 0 \end{bmatrix}$$

$$\mathbf{Q} = \begin{bmatrix} 0 & 0 & 0 & 1/4 & 0 \\ 0 & 0 & 1/2 & 1/4 & 0 \\ 0 & 1/2 & 0 & 1/4 & 0 \\ 1 & 1/2 & 1/2 & 0 & 1 \\ 0 & 0 & 0 & 1/4 & 0 \end{bmatrix}$$

$$\begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \end{array} \begin{array}{c} \begin{array}{ccccc} A & B & C & D & E \end{array} \\ \begin{bmatrix} 0.03 & 0.05 & 0.05 & 0.09 & 0.03 \\ 0.04 & 0.07 & 0.07 & 0.15 & 0.04 \\ 0.03 & 0.05 & 0.05 & 0.09 & 0.03 \\ 0.02 & 0.03 & 0.03 & 0.05 & 0.02 \end{bmatrix} \end{array}$$

BIG PROBLEMS!

But structured

Dataset	Size	Nonzeros
LCSH-2	59,849	227,464
WC-3	70,509	403,960
Product graph	4,219,893,141	91,886,357,440

Singh et al. PNAS 2008

Gleich et al. SIAM J. Sci. Comp. 2011

Opinion dynamics

$\boldsymbol{Y}_1$ initial opinions

$$(\boldsymbol{I} - \alpha \boldsymbol{P}^T)\boldsymbol{Y}_\infty = \beta \boldsymbol{Y}_1$$

$\boldsymbol{P}$ gives influences

Friedkin & Johnson, 1990, 1999

Voting in social networks

Vicious democracy

Consider a social network

- Some small fraction of users express a vote ☺
- Other users delegate their vote with decay 😐
- The others don't vote at all ☹
- Determine a final vote by taking the expected diffusion where each non-voter picks a neighbor at random and then diffuse the vote

Boldi et al. CIKM '09, CACM '11

Communities & Clusters

Theory

- *Early* Fiedler, Anderson & Morley 1985, Mihail, Chung, Pothen et al., Simon et al., Lovász & Simonovits, FOCS 1990, Random. Struct. Alg. 1993
- Spielman & Teng, 2004, 2013
- Andersen, Chung, Lang, FOCS 2006
- Chung, PNAS 2007
- Ghosh et al. KDD 2014

Practice

- Andersen & Lang, WWW 2006
- Leskovec et al. Internet Math. 2009
- Gargi et al. 2011 (Google, YouTube communities)
- Epasto et al. 2014 (Google, Competing advertisers)
- … so many ...

Andersen-Chung-Lang personalized PageRank community theorem

[Andersen et al. 2006]

Informally

Suppose the seeds are in a set of good conductance, then the personalized PageRank method *will find* a set with conductance that's nearly as good.

… also, it's really fast.

Examples

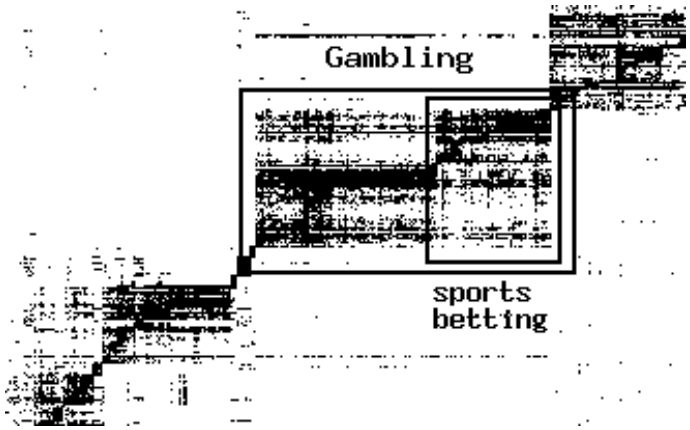


Figure 4: A low-resolution view of part of a very small version of the Yahoo sponsored search bipartite incidence matrix. The gambling co-cluster contains a sports betting subcluster.

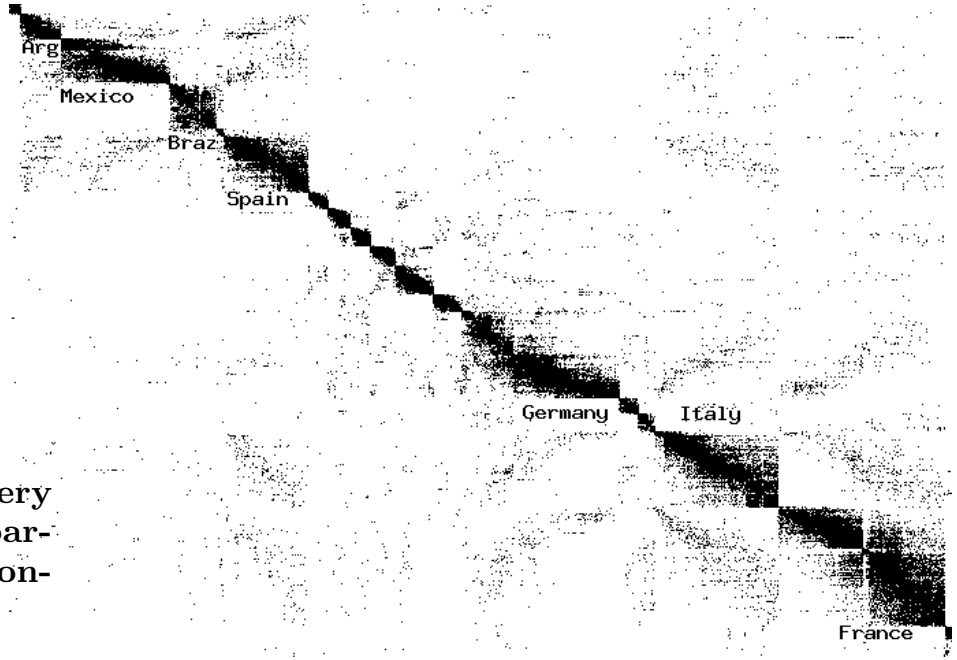
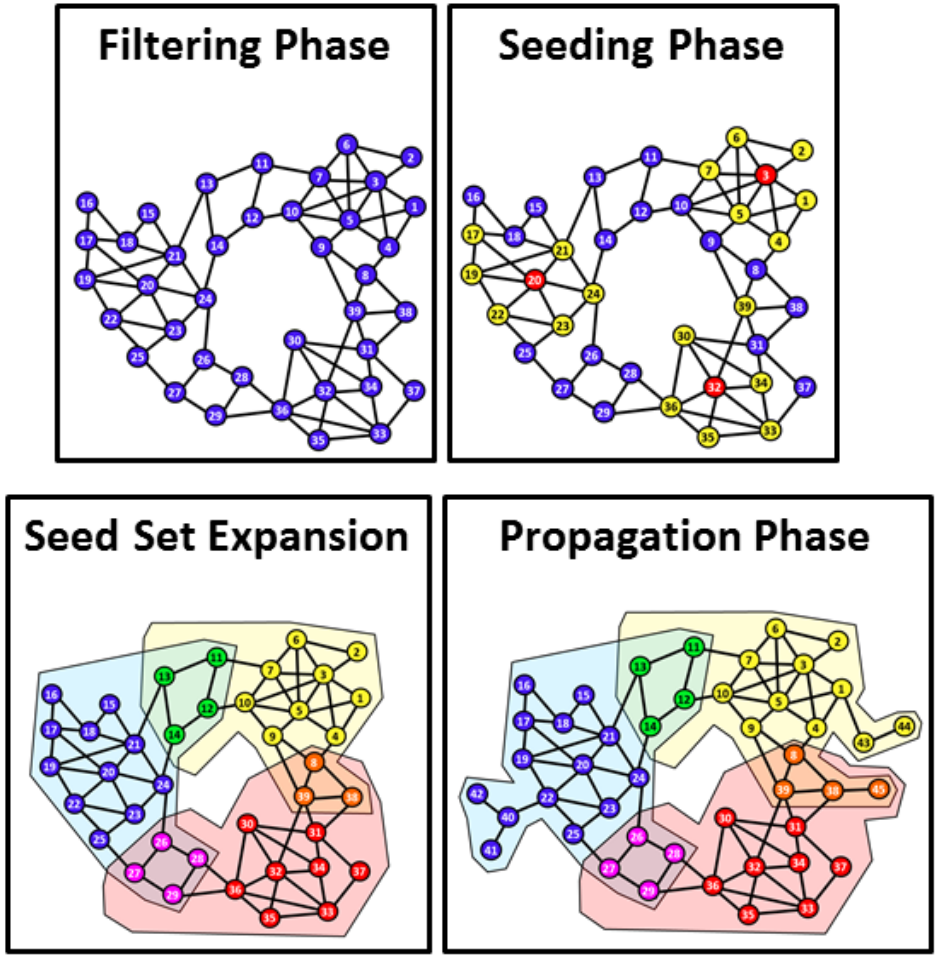


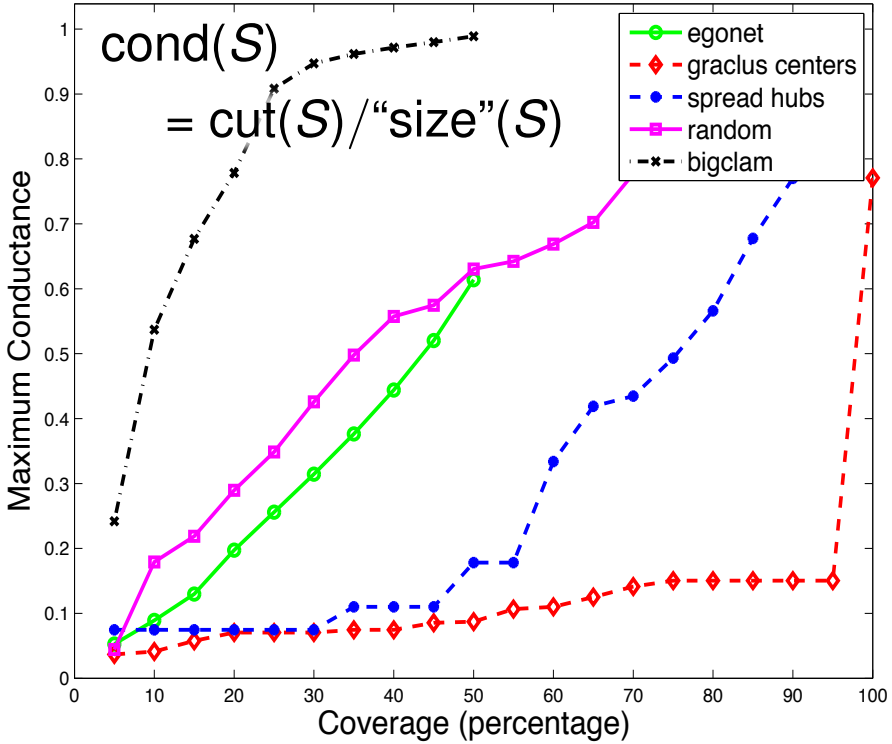
Figure 10: A low-resolution view of part of the movies vs actresses incidence matrix. The Spain co-cluster lies within a supercluster of Spanish- and Portuguese-language countries. Also there are many edges leading from Spain to other Romance-language countries. See section 3.5.

Andersen and Lang, 2006

Overlapping communities via seed set expansion works nicely.



Flickr social network
2M vertices, 22M edges



We can cover 95% of network with communities of cond. ~0.15.

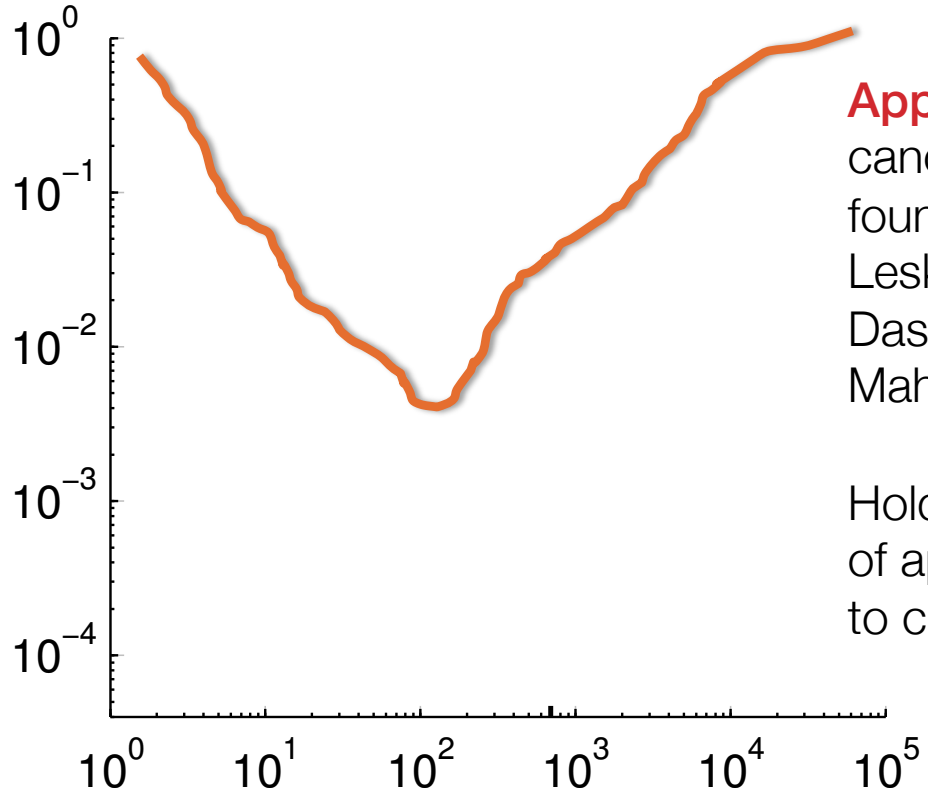
Whang, Gleich, Dhillon 2013, 2015

… demo ...

github.com/dgleich/diffusion-tutorial

Empirical Evaluation using Network Community Profiles

Minimum conductance for any community of the given size



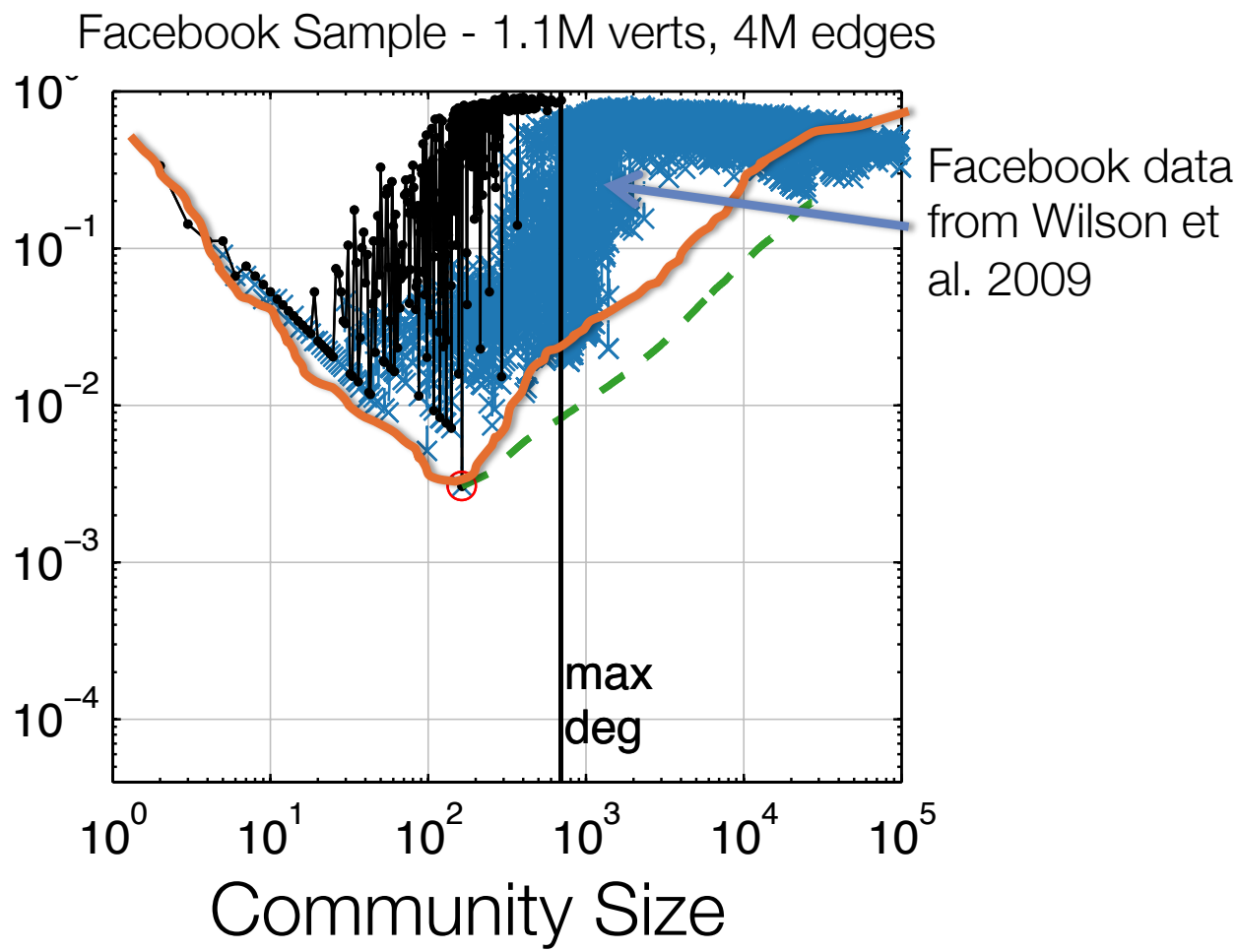
Community Size

Approximate canonical shape found by Leskovec, Lang, Dasgupta, and Mahoney

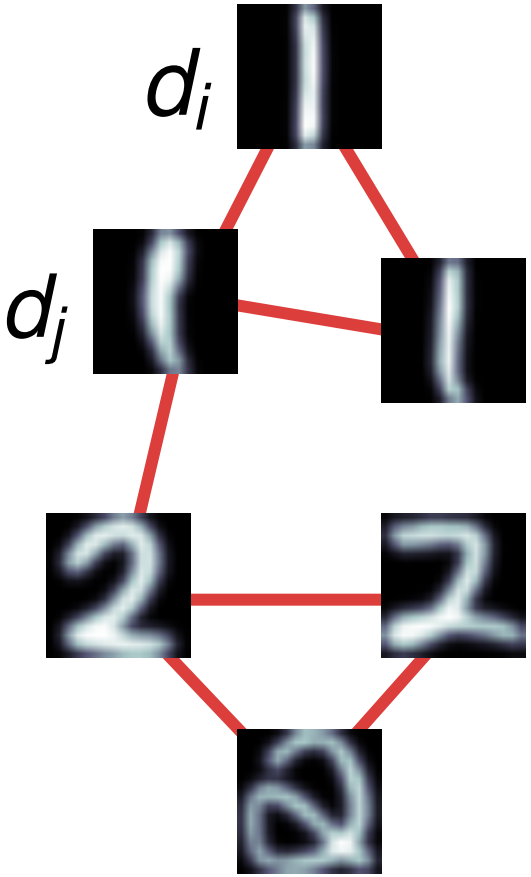
Holds for a variety of approximations to conductance.

Network Community Profile

Minimum conductance for any community of the given size

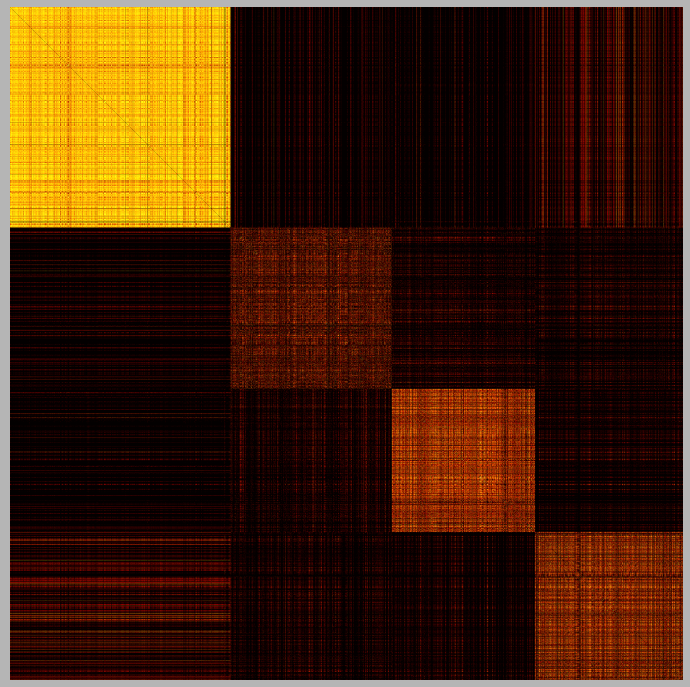


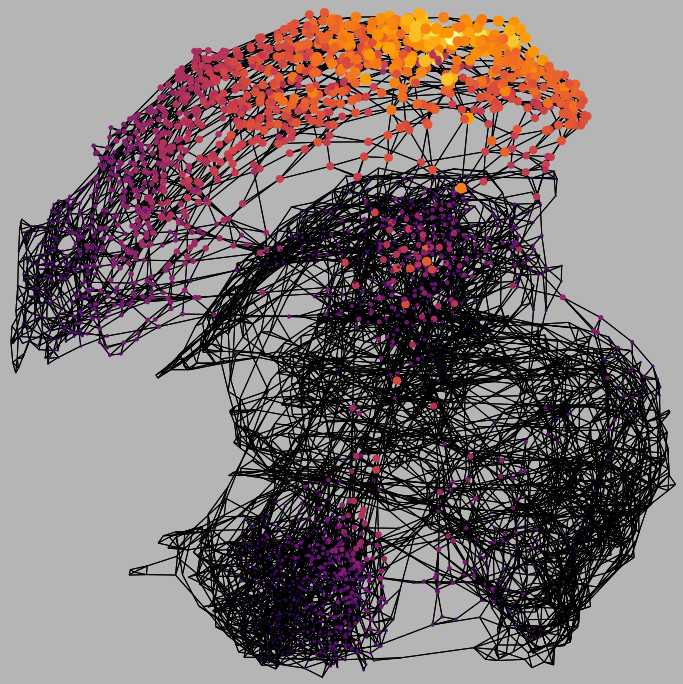
Semi-supervised Learning on Graphs



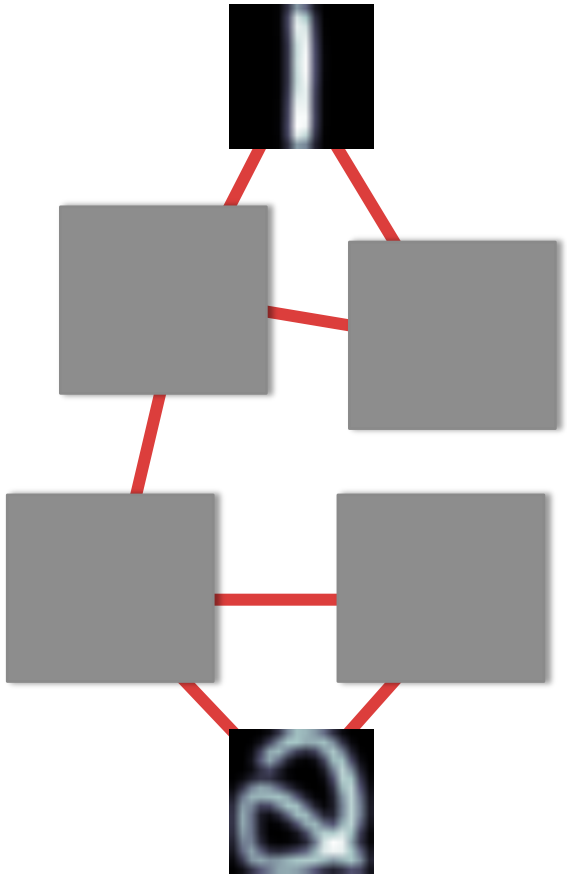
$$A_{i,j} = \exp\left(-\frac{\|d_i - d_j\|_2^2}{2\sigma^2}\right)$$

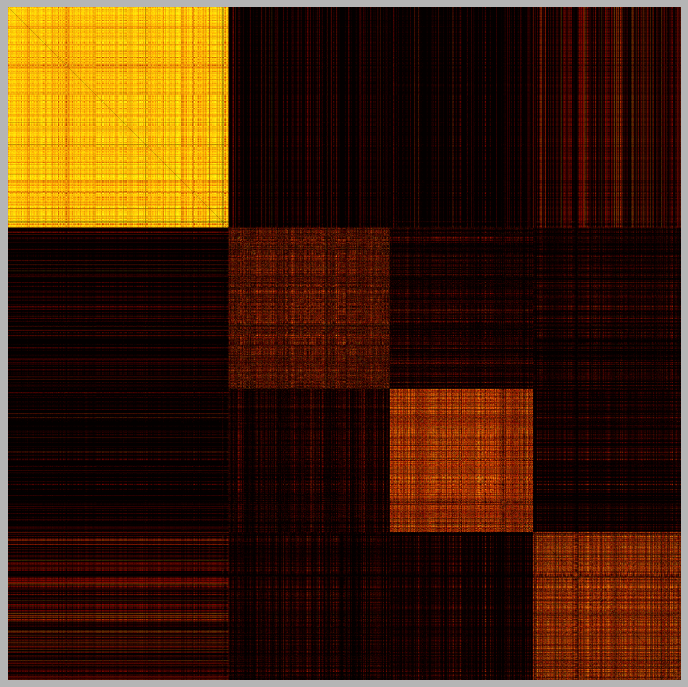
Zhou et al. NIPS (2003)

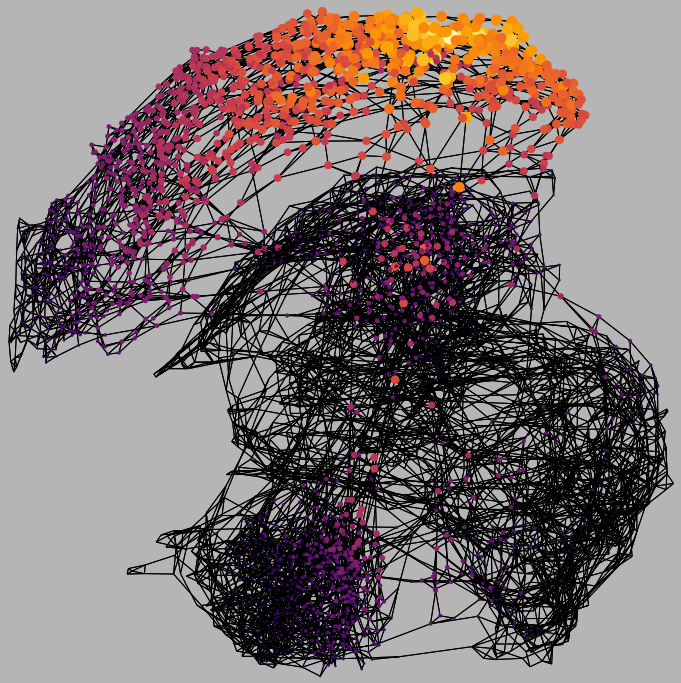




Semi-supervised Learning on Graphs







Experiment predict unlabeled images from the labeled ones

Semi-supervised Learning on Graphs

$$\mathbf{K}_1 = (\mathbf{I} - \beta \mathcal{A})^{-1}$$

$$\mathbf{K}_2 = (\mathbf{D} - \beta \mathbf{A})^{-1}$$

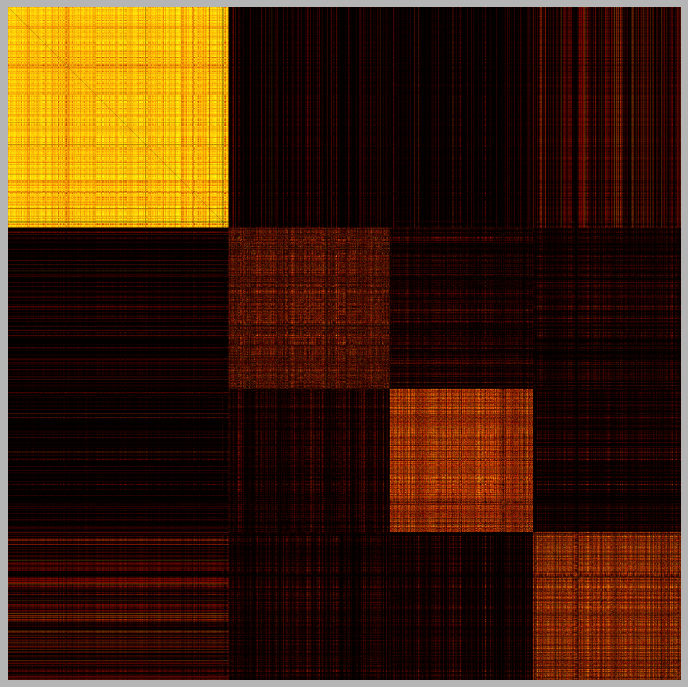
Predictions

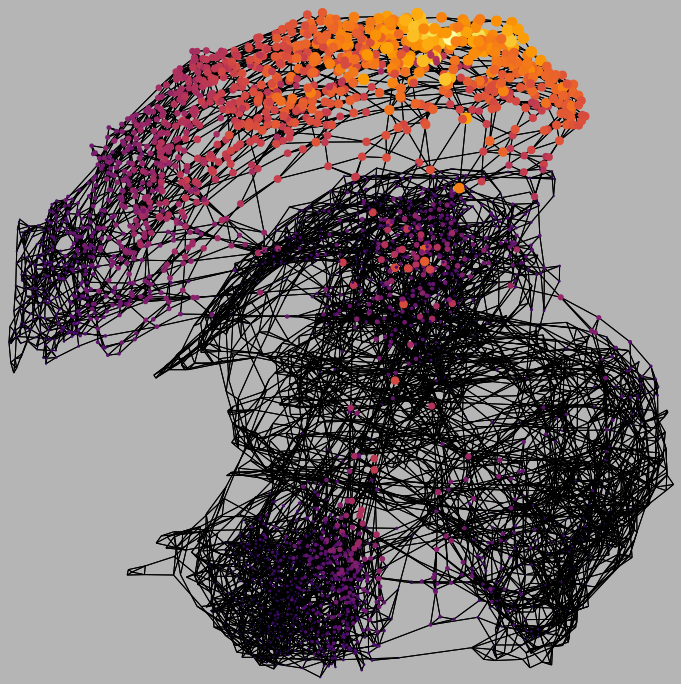
Indicators on the revealed labels

$$\mathbf{Y} = \mathbf{K}_i \mathbf{L}$$

$$\mathbf{y} = \text{argmax}_j \, \mathbf{Y}$$

Experiment vary number of labeled images and track perf.





Even more diffusions

SIR and viral thresholds

- Wang et al. 2003
- Berger et al. 2005

Rumor spreading

- Chierichetti et al. 2010

Information cascades

- Farajtaba et al. 2015

Laplacian variations

- Bridle & Zhu, MLG 2013

Diffusions over semi-rings

- Kepner & Gilbert, 2011 (e.g. Peer-pressure clustering)

Generalized coefficients

- Boldi et al. 2005 – TotalRank
- Baeza-Yates et al. 2006 - Generalized
- Constantine & Gleich, 2007, 2010 – Random alpha PageRank

Brief outline

Introduction & Application 20-30 min
Coordinate Relaxation for Strong Convergence 30 min
Break 15 min

Katz Diffusion 10 min
Weak Convergence for PageRank 20 min
Monte Carlo methods 15 min
Break 15 min

Implicit Regularization 25 min
Discussion 15 min

Diffusion vectors

Given a graph G and a set of seed node(s)…

A **diffusion vector** assigns nodes values that quantify somehow the relationship of S to the rest of G, by propagating information from S to the rest of G.

$$\mathbf{f} = \sum_{k=0}^{\infty} c_k \boldsymbol{M}^k \mathbf{s}$$

Diffusion coefficients

Graph related matrix

Seed vector

Diffusion vector

Another application, another diffusion

Different applications and sources of data call for different diffusions – change c_k, $\mathbf{M}^k$, $\mathbf{s}$, pre- and post-processing.

Algorithmic approaches

- deterministic
- randomized

$$\mathbf{f} = \sum_{k=0}^{\infty} c_k \boldsymbol{M}^k \mathbf{s}$$

Diffusion coefficients

- (many)

Graph related matrix

- (A,P,L, more)

Seed vector

- constructions
- normalizations

Diffusion vector

- normalizations
- accuracy

Another application, another diffusion

Different applications and sources of data call for different diffusions – change c_k, $\mathbf{M}^k$, $\mathbf{s}$, pre- and post-processing.

Algorithms – coordinate relaxation / push, monte carlo

Coefficients – probability, Katz, adaptive, many more

Graph matrix – adjacency, prob trans, Laplacians, more

Seed vector – seed set indicator vector, normalizations

Diffusion post-processing

- Degree scaling
- weak vs strong accuracy
- Sweep-procedure

Coordinate relaxation methods for graph diffusions

Local node rankings, similarity

Rank nodes with respect to S by taking the largest values from a diffusion **f**. More accurate rankings require the solution to be precise enough that large diffusion values that are close to each other aren't mis-ranked:

Ranks **Diffusion values** **Approx Ranks**

$$\begin{bmatrix} 4 \\ 2 \\ 3 \\ 1 \end{bmatrix} \mathbf{f} = \begin{bmatrix} 0.139 \\ 0.251 \\ 0.25 \\ 0.36 \end{bmatrix}, \hat{\mathbf{f}} = \begin{bmatrix} 0.137 \\ 0.24 \\ 0.245 \\ 0.33 \end{bmatrix} \quad \begin{bmatrix} 4 \\ 3 \\ 2 \\ 1 \end{bmatrix}$$

Local node rankings, similarity

Rank nodes with respect to S by taking the largest values from a diffusion **f**. More accurate rankings require the solution to be precise enough …

GOAL: compute **f** with accuracy $\|\mathbf{f} - \hat{\mathbf{f}}\|_1 < \varepsilon$

We begin with the popular personalized PageRank diffusion vector, but will later apply this accuracy setting to a number of diffusions.

PageRank diffusion

The PageRank diffusion can be defined as the solution to

$$(\boldsymbol{I} - \alpha \boldsymbol{P})\mathbf{x} = (1 - \alpha)\mathbf{s}$$

for some alpha in (0,1). The seed vector **s** should be normalized to sum to 1. This linear system is equivalent to our definition for a diffusion:

$$\mathbf{f} = \sum_{k=0}^{\infty} \alpha^k \boldsymbol{P}^k \mathbf{s}$$

PageRank diffusion

The PageRank diffusion can be defined as the solution to

$$(\boldsymbol{I} - \alpha \boldsymbol{P})\mathbf{x} = (1 - \alpha)\mathbf{s}$$

for some alpha in (0,1). The seed vector **s** should be normalized to sum to 1. This linear system is equivalent to our definition for a diffusion:

$$\mathbf{f} = \sum_{k=0}^{\infty} \alpha^k \boldsymbol{P}^k \mathbf{s}$$

This holds when $\|\alpha \boldsymbol{P}\| < 1$ because of the Neumann series:

$$(\boldsymbol{I} - \alpha \boldsymbol{P})^{-1} = \sum_{k=0}^{\infty} \alpha^k \boldsymbol{P}^k$$

(proof: geometric series)

Coordinate relaxation for PageRank

For a fast approximation, **f**, to the following

$$(\boldsymbol{I} - \alpha \boldsymbol{P})\mathbf{x} = (1 - \alpha)\mathbf{s} \quad = \tilde{\mathbf{s}}$$

we introduce a coordinate relaxation scheme:

Initial solution and residual: $\mathbf{x}^{(0)} = 0, \mathbf{r}^{(0)} = (1 - \alpha)\mathbf{s}$

Iterative updates: first pick entry of residual, j

(There are a number of ways of picking an entry

-- for now, assume just that the entry is non-zero)

Coordinate relaxation for PageRank

For a fast approximation, **f**, to the following

$$(\boldsymbol{I} - \alpha \boldsymbol{P})\mathbf{x} = \tilde{\mathbf{s}}$$

we introduce a coordinate relaxation scheme:

Initial solution and residual: $\mathbf{x}^{(0)} = 0, \mathbf{r}^{(0)} = (1 - \alpha)\mathbf{s}$

Iterative updates: first pick entry of residual, j

- update solution: $\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + r_j \cdot \mathbf{e}_j$

Coordinate relaxation for PageRank

For a fast approximation, **f**, to the following

$$(\boldsymbol{I} - \alpha \boldsymbol{P})\mathbf{x} = \tilde{\mathbf{s}}$$

we introduce a coordinate relaxation scheme:

Initial solution and residual: $\mathbf{x}^{(0)} = 0, \mathbf{r}^{(0)} = (1 - \alpha)\mathbf{s}$

Iterative updates: first pick entry of residual, j

- update solution: $\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + r_j \cdot \mathbf{e}_j$
- update residual: $\mathbf{r}^{(k+1)} = \tilde{\mathbf{s}} - (\boldsymbol{I} - \alpha \boldsymbol{P})\mathbf{x}^{(k+1)}$

$$= \mathbf{r}^{(k)} - r_j(\boldsymbol{I} - \alpha \boldsymbol{P})\mathbf{e}_j$$

$$= \mathbf{r}^{(k)} - r_j\mathbf{e}_j + r_j\alpha \boldsymbol{P}\mathbf{e}_j$$

Coordinate relaxation for PageRank

Approximating

$$(\boldsymbol{I} - \alpha \boldsymbol{P})\mathbf{x} = \tilde{\mathbf{s}}$$

Initial solution and residual: $\mathbf{x}^{(0)} = 0, \mathbf{r}^{(0)} = (1 - \alpha)\mathbf{s}$

Iterative updates: first pick entry of residual, j

- update solution: $\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + r_j \cdot \mathbf{e}_j$
- update residual: $\mathbf{r}^{(k+1)} = \tilde{\mathbf{s}} - (\boldsymbol{I} - \alpha \boldsymbol{P})\mathbf{x}^{(k+1)}$

$$= \mathbf{r}^{(k)} - r_j \mathbf{e}_j + r_j \alpha \boldsymbol{P} \mathbf{e}_j$$

Coordinate relaxation for PageRank

Approximating

$$(\boldsymbol{I} - \alpha \boldsymbol{P})\mathbf{x} = \tilde{\mathbf{s}}$$

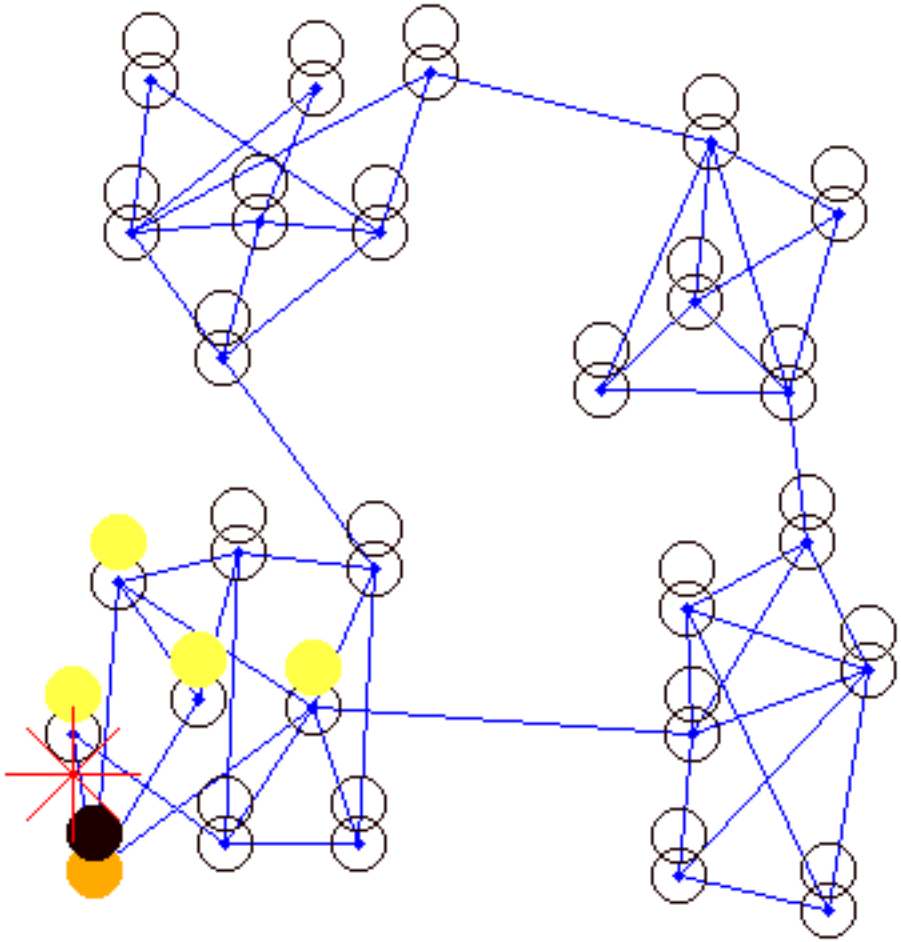
Initial solution and residual: $\mathbf{x}^{(0)} = 0, \mathbf{r}^{(0)} = (1 - \alpha)\mathbf{s}$

Iterative updates: first pick entry of residual, j

- update solution: $x_j^{(k+1)} = x_j^{(k)} + r_j$
- update residual:

$$r_i^{(k+1)} = \begin{cases} 0, & \text{if } i = j \\ r_i^{(k)} + r_j \alpha / d_j, & \text{if } i \sim j \\ r_i^{(k)}, & \text{else} \end{cases}$$

The push algorithm



Coordinate relaxation remarks, part 1

- This is the fundamental operation underlying many of the deterministic methods for diffusions
- Consists of single update to solution **x** and a single column access of matrix **P** to update **r**
- Because of this, fast for sparse matrices (avoids O(|E|) work required by full mat-vec!)
- Nothing special about PageRank, can be applied in other circumstances

Coordinate relaxation remarks, part 1

- *This is the fundamental operation underlying many of the deterministic methods for diffusions*
- Consists of single update to solution **x** and a single column access of matrix **P** to update **r**
- Because of this, fast for sparse matrices (avoids O(|E|) work required by full mat-vec!)
- Nothing special about PageRank, can be applied in other circumstances

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + r_j \mathbf{e}_j$$
$$\mathbf{r}^{(k+1)} = \mathbf{r}^{(k)} - r_j \mathbf{e}_j + r_j \alpha \boldsymbol{P} \mathbf{e}_j$$

Coordinate relaxation remarks, part 2

- Convergence depends on method of choosing entry, as well as the underlying matrix
  - Gauss-Southwell: choose $\mathbf{r}_j$ to be largest entry in $\mathbf{r}$
  - Gauss-Seidel: after j, choose j+1, then j+2, ...
  - Choose any entry >= average magnitude of $\mathbf{r}$
  - Choose any entry above some threshold
  - Even random selection can work

Coordinate relaxation remarks, part 2

- Convergence depends on method of choosing entry, as well as the underlying matrix
  - Gauss-Southwell: choose $\mathbf{r}_j$ to be largest entry in $\mathbf{r}$
  - Gauss-Seidel: after j, choose j+1, then j+2, ...
  - Choose any entry >= average magnitude of $\mathbf{r}$
  - Choose any entry above some threshold
  - Even random selection can work
- Many methods of entry selection converge for diagonally dominant and positive definite matrices (in particular, Gauss-Southwell does).
- Implementation requires intelligent choice of data structure for $\mathbf{r}$ for fast entry selection/updates

PageRank Convergence: error & residual

Approximating a solution to

$$(\boldsymbol{I} - \alpha \boldsymbol{P})\mathbf{x} = \tilde{\mathbf{s}}$$

residual and error satisfy

$$\begin{aligned}\mathbf{r}^{(k)} &= \tilde{\mathbf{s}} - (\boldsymbol{I} - \alpha \boldsymbol{P})\mathbf{x}^{(k)} \\ &= (\boldsymbol{I} - \alpha \boldsymbol{P})\mathbf{x} - (\boldsymbol{I} - \alpha \boldsymbol{P})\mathbf{x}^{(k)}\end{aligned}$$

PageRank Convergence: error & residual

Approximating a solution to

$$(\boldsymbol{I} - \alpha \boldsymbol{P})\mathbf{x} = \tilde{\mathbf{s}}$$

residual and error satisfy

$$\begin{aligned}
\mathbf{r}^{(k)} &= \tilde{\mathbf{s}} - (\boldsymbol{I} - \alpha \boldsymbol{P})\mathbf{x}^{(k)} \\
&= (\boldsymbol{I} - \alpha \boldsymbol{P})\mathbf{x} - (\boldsymbol{I} - \alpha \boldsymbol{P})\mathbf{x}^{(k)} \\
(\boldsymbol{I} - \alpha \boldsymbol{P})^{-1}\mathbf{r}^{(k)} &= (\mathbf{x} - \mathbf{x}^{(k)})
\end{aligned}$$

$$\|\mathbf{x} - \mathbf{x}^{(k)}\| \leq \|(\boldsymbol{I} - \alpha \boldsymbol{P})^{-1}\| \|\mathbf{r}^{(k)}\|$$

for any sub-multiplicative matrix norm $\| \ \|$.

PageRank Convergence: residual bound

Approximating a solution to $(\boldsymbol{I} - \alpha \boldsymbol{P})\mathbf{x} = \tilde{\mathbf{s}}$

Error satisfies $\|\mathbf{x} - \mathbf{x}^{(k)}\| \leq \|(\boldsymbol{I} - \alpha \boldsymbol{P})^{-1}\| \|\mathbf{r}^{(k)}\|$

Initial solution and residual: $\mathbf{x}^{(0)} = 0, \mathbf{r}^{(0)} = (1 - \alpha)\mathbf{s}$

Update residual: $\mathbf{r}^{(k+1)} = \mathbf{r}^{(k)} - r_j \mathbf{e}_j + r_j \alpha \boldsymbol{P} \mathbf{e}_j$

PageRank Convergence: residual bound

Approximating a solution to $(\boldsymbol{I} - \alpha \boldsymbol{P})\mathbf{x} = \tilde{\mathbf{s}}$

Error satisfies $\|\mathbf{x} - \mathbf{x}^{(k)}\| \leq \|(\boldsymbol{I} - \alpha \boldsymbol{P})^{-1}\| \|\mathbf{r}^{(k)}\|$

Initial solution and residual: $\mathbf{x}^{(0)} = 0, \mathbf{r}^{(0)} = (1 - \alpha)\mathbf{s}$

Update residual: $\mathbf{r}^{(k+1)} = \mathbf{r}^{(k)} - r_j \mathbf{e}_j + r_j \alpha \boldsymbol{P} \mathbf{e}_j$

$$\begin{aligned}
\|\mathbf{r}^{(k+1)}\|_1 &\leq \|\mathbf{r}^{(k)} - r_j \mathbf{e}_j\|_1 + \|r_j \alpha \boldsymbol{P} \mathbf{e}_j\|_1 && \text{Triangle inequality} \\
&\leq \|\mathbf{r}^{(k)}\|_1 - r_j + |r_j \alpha| \|\boldsymbol{P} \mathbf{e}_j\|_1 && \text{Residual nonnegative} \\
&\leq \|\mathbf{r}^{(k)}\|_1 - r_j + |r_j \alpha| && \mathbf{P} \text{ is column-stochastic} \\
&\leq \|\mathbf{r}^{(k)}\|_1 - r_j (1 - \alpha) && \text{Residual nonnegative}
\end{aligned}$$

PageRank Convergence: residual bound

Error satisfies $\|\mathbf{x} - \mathbf{x}^{(k)}\| \leq \|(\boldsymbol{I} - \alpha \boldsymbol{P})^{-1}\| \|\mathbf{r}^{(k)}\|$

Initial solution and residual: $\mathbf{x}^{(0)} = 0, \mathbf{r}^{(0)} = (1 - \alpha)\mathbf{s}$

Residual norm: $\|\mathbf{r}^{(k+1)}\|_1 \leq \|\mathbf{r}^{(k)}\|_1 - r_j(1 - \alpha)$

Assume we chose r_j to be at least as big as the average magnitude of the residual entries. Then

$$r_j \geq \|\mathbf{r}^{(k)}\|_1 / \mathrm{nnz}(\mathbf{r}^{(k)}) \qquad \text{(definition of average)}$$

$$\geq \|\mathbf{r}^{(k)}\|_1 / n \qquad \text{(loose bound!)}$$

PageRank Convergence: residual bound

Error satisfies $\|\mathbf{x} - \mathbf{x}^{(k)}\| \leq \|(\boldsymbol{I} - \alpha \boldsymbol{P})^{-1}\| \|\mathbf{r}^{(k)}\|$

Initial solution and residual: $\mathbf{x}^{(0)} = 0, \mathbf{r}^{(0)} = (1 - \alpha)\mathbf{s}$

Residual norm: $\|\mathbf{r}^{(k+1)}\|_1 \leq \|\mathbf{r}^{(k)}\|_1 - r_j(1 - \alpha)$

Assume we chose r_j to be at least as big as the average magnitude of the residual entries. Then

$$r_j \geq \|\mathbf{r}^{(k)}\|_1 / \mathrm{nnz}(\mathbf{r}^{(k)}) \qquad \text{(definition of average)}$$

$$\geq \|\mathbf{r}^{(k)}\|_1 / n \qquad \text{(loose bound!)}$$

$$\begin{aligned}
\|\mathbf{r}^{(k+1)}\|_1 &\leq \|\mathbf{r}^{(k)}\|_1 - \|\mathbf{r}^{(k)}\|_1 (1 - \alpha)/n \\
&\leq \|\mathbf{r}^{(k)}\|_1 \left(1 - \frac{(1-\alpha)}{n}\right) \\
&\leq \|\mathbf{r}^{(0)}\|_1 \left(1 - \frac{(1-\alpha)}{n}\right)^{k+1} = (1 - \alpha)\left(1 - \frac{(1-\alpha)}{n}\right)^{k+1}
\end{aligned}$$

PageRank Convergence: back to error

Error satisfies $\|\mathbf{x} - \mathbf{x}^{(k)}\| \leq \|(\boldsymbol{I} - \alpha \boldsymbol{P})^{-1}\| \|\mathbf{r}^{(k)}\|$

Substituting in for residual...

Residual norm: $\|\mathbf{r}^{(k)}\|_1 \leq (1 - \alpha) \left(1 - \frac{(1-\alpha)}{n}\right)^k$

$\|(\boldsymbol{I} - \alpha \boldsymbol{P})^{-1}\|_1 = \frac{1}{1-\alpha}$ (using Neumann series for inverse)

PageRank Convergence: back to error

Error satisfies $\|\mathbf{x} - \mathbf{x}^{(k)}\| \leq \|(\boldsymbol{I} - \alpha \boldsymbol{P})^{-1}\| \|\mathbf{r}^{(k)}\|$

Substituting in for residual...

Residual norm: $\|\mathbf{r}^{(k)}\|_1 \leq (1 - \alpha) \left(1 - \frac{(1-\alpha)}{n}\right)^k$

$\|(\boldsymbol{I} - \alpha \boldsymbol{P})^{-1}\|_1 = \frac{1}{1-\alpha}$ (using Neumann series for inverse)

Substitution gives $\|\mathbf{x} - \mathbf{x}^{(k)}\|_1 \leq \frac{1}{1-\alpha} \|\mathbf{r}^{(k)}\|$

$$\leq \left(1 - \frac{(1-\alpha)}{n}\right)^k$$

Bounds number of iterations with $O(\log(1/\varepsilon)n)$
(but the bound can be refined to give sublinear work, depending on the underlying graph.)

Related work: strong coordinate relaxation

- Deterministic coordinate relaxation for diffusion
  - [Jeh & Widom ’03] Scaling Personalized PageRank
  - [McSherry ’05] Accelerated PageRank Computation
  - [Berkhin ’07] Bookmark Coloring Algorithm for PPR
  - [Bonchi et al. ’12] Fast Katz and Commute Times
  - [Kloster, Gleich WAW13] Coordinate relaxation for $\exp(\mathbf{P})e_j$
- Selecting entry to relax:
  - [Dhillon, Ravikumar, Tewari ’11] Near neighbor-based greedy coordinate descent
  - [Nutini, et al. 2015] Gauss-Southwell better than random

Brief outline

Introduction & Application 20-30 min
Coordinate Relaxation for Strong Convergence 30 min
Break 15 min

Katz Diffusion 10 min
Weak Convergence for PageRank 20 min
Monte Carlo methods 15 min
Break 15 min

Implicit Regularization 25 min
Discussion 15 min

From PageRank

to the Katz diffusion

Coordinate relaxation for PageRank

Approximating a solution to

$$(\boldsymbol{I} - \alpha \boldsymbol{P})\mathbf{x} = \tilde{\mathbf{s}}$$

Initial solution and residual: $\mathbf{x}^{(0)} = 0, \mathbf{r}^{(0)} = (1 - \alpha)\mathbf{s}$

Iterative updates: first pick entry of residual, j

- update solution: $\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + r_j \cdot \mathbf{e}_j$
- udpate residual: $\mathbf{r}^{(k+1)} = \tilde{\mathbf{s}} - (\boldsymbol{I} - \alpha \boldsymbol{P})\mathbf{x}^{(k+1)}$

$$= \mathbf{r}^{(k)} - r_j \mathbf{e}_j + r_j \alpha \boldsymbol{P} \mathbf{e}_j$$

Coordinate relaxation for Katz diffusion

Approximating a solution to

$$(\boldsymbol{I} - \alpha \boldsymbol{A})\mathbf{x} = \mathbf{b}$$

Initial solution and residual: $\mathbf{x}^{(0)} = 0, \mathbf{r}^{(0)} = \mathbf{b}$

Iterative updates: first pick entry of residual, j

- update solution: $\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + r_j \cdot \mathbf{e}_j$
- udpate residual: $\mathbf{r}^{(k+1)} = \mathbf{b} - (\boldsymbol{I} - \alpha \boldsymbol{A})\mathbf{x}^{(k+1)}$

$$= \mathbf{r}^{(k)} - r_j \mathbf{e}_j + r_j \alpha \boldsymbol{A} \mathbf{e}_j$$

(Not much difference)

Katz diffusion

The Katz diffusion can be defined as the solution to

$$(\boldsymbol{I} - \alpha \boldsymbol{A})\mathbf{x} = \mathbf{s} - (\boldsymbol{I} - \alpha \boldsymbol{A})\mathbf{s}$$

The restrictions on alpha differ from PageRank:

PageRank: $0 < \alpha < 1$

Katz: $0 < \alpha < 1/d_{\text{max}} \leq 1/\lambda_1(\boldsymbol{A})$

where lambda_1 is the dominant eigenvalue, and d_{max} is the largest degree in the graph.

Katz diffusion

The Katz diffusion can be defined as the solution to

$$(\boldsymbol{I} - \alpha \boldsymbol{A})\mathbf{x} = \mathbf{s} - (\boldsymbol{I} - \alpha \boldsymbol{A})\mathbf{s}$$

The restrictions on alpha differ from PageRank:

PageRank: $0 < \alpha < 1$

Katz: $0 < \alpha < 1/d_{\max} \leq 1/\lambda_1(\boldsymbol{A})$

where lambda_1 is the dominant eigenvalue, and $d_{\max}$ is the largest degree in the graph.

Note: $0 < \alpha < 1/d_{\max}$ guarantees $\|\alpha \boldsymbol{A}\|_1 < 1$ so

$$\mathbf{f} = \sum_{k=1}^{\infty} \alpha^k \boldsymbol{A}^k \mathbf{s} = \sum_{k=1}^{\infty} (\alpha d_{\max})^k \left(\frac{1}{d_{\max}} \boldsymbol{A}\right)^k \mathbf{s}$$

Coordinate relaxation for Katz diffusion

Approximating a solution to

$$(\boldsymbol{I} - \alpha \boldsymbol{A})\mathbf{x} = \mathbf{b}$$

Initial solution and residual: $\mathbf{x}^{(0)} = 0, \mathbf{r}^{(0)} = \mathbf{b}$

Iterative updates: first pick entry of residual, j

- update solution: $x_j^{(k+1)} = x_j^{(k)} + r_j$
- udpate residual:

$$r_i^{(k+1)} = \begin{cases} 0, & \text{if } i = j \\ r_i^{(k)} + r_j \alpha, & \text{if } i \sim j \\ r_i^{(k)}, & \text{else} \end{cases}$$

Convergence for PageRank (repeated)

Error satisfies $\|\mathbf{x} - \mathbf{x}^{(k)}\| \leq \|(\boldsymbol{I} - \alpha \boldsymbol{P})^{-1}\| \|\mathbf{r}^{(k)}\|$

Substituting in for residual...

Residual norm: $\|\mathbf{r}^{(k)}\|_1 \leq (1 - \alpha) \left(1 - \frac{(1-\alpha)}{n}\right)^k$

$\|(\boldsymbol{I} - \alpha \boldsymbol{P})^{-1}\|_1 = \frac{1}{1-\alpha}$ (using Neumann series for inverse)

Substitution gives $\|\mathbf{x} - \mathbf{x}^{(k)}\|_1 \leq \frac{1}{1-\alpha} \|\mathbf{r}^{(k)}\|$

$$\leq \left(1 - \frac{(1-\alpha)}{n}\right)^k$$

Bounds number of iterations with $O(\log(1/\varepsilon)n)$
(but the bound can be refined to give sublinear work, depending on the underlying graph.)

Convergence for Katz

Error satisfies $\|\mathbf{x} - \mathbf{x}^{(k)}\| \leq \|(\boldsymbol{I} - \alpha \boldsymbol{A})^{-1}\| \|\mathbf{r}^{(k)}\|$

Differences:

Depending on which norm is desired, scale A by $1/d_{max}$ or $1/\text{lambda}_1$. Then, rest of the convergence analysis for PageRank applies to the scaled adjacency matrix:

$$\|(\boldsymbol{I} - \alpha \boldsymbol{A})^{-1}\|_1 \leq \frac{1}{1 - \alpha d_{\text{max}}} \quad \text{If} \quad \alpha < 1/d_{\text{max}}$$

(If $\alpha < 1/\lambda_1$ the method still converges, but analysis is trickier.)

Brief outline

Introduction & Application 20-30 min
Coordinate Relaxation for Strong Convergence 30 min
Break 15 min

Katz Diffusion 10 min
Weak Convergence for PageRank 20 min
Monte Carlo methods 15 min
Break 15 min

Implicit Regularization 25 min
Discussion 15 min

Weak convergence coordinate relaxation for communities and good conductance sets.

Weak accuracy for community detection

Local community detection/ finding good conductance sets from a diffusions vector **f**:

Seek the largest values in the diffusion vector;
weak accuracy, because identifying the largest values is the goal, not the precise values themselves
(or even the precise ranking of values).

Weak accuracy for community detection

Local community detection/ finding good conductance sets from a diffusions vector **f**:

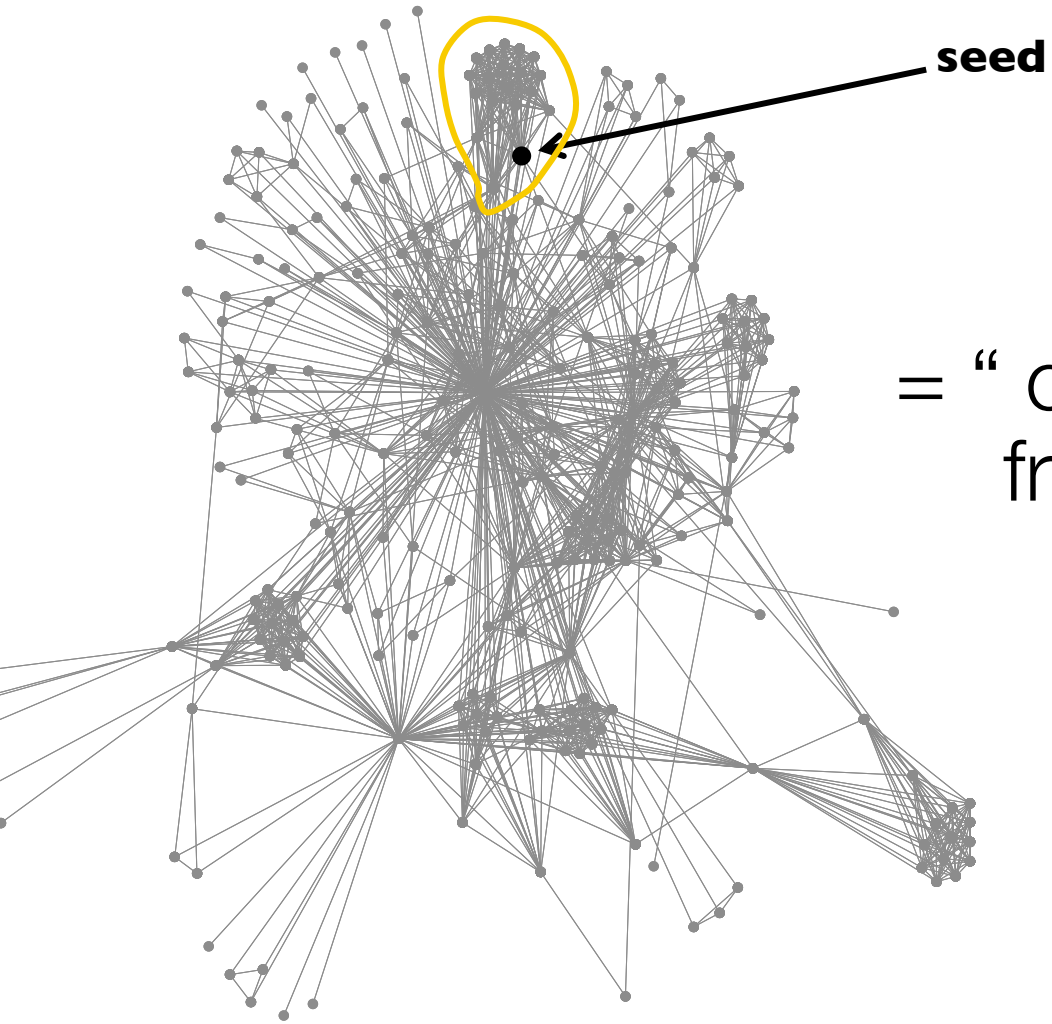
Seek the largest values in the diffusion vector;
weak accuracy, because identifying the largest values is the goal, not the precise values themselves
(or even the precise ranking of values).

GOAL: compute **f** with accuracy $0 \leq f_j - \hat{f}_j \leq \varepsilon d_j$ (entry-wise). Equivalent to

$$\mathbf{f} \geq \hat{\mathbf{f}} \quad \text{and} \quad \|\boldsymbol{D}^{-1}(\mathbf{f} - \hat{\mathbf{f}})\|_\infty < \varepsilon$$

Low-conductance sets

$$\text{conductance}(\,T\,) \;=\; \frac{\#\text{ edges leaving } T}{\min(\,\text{vol}(T),\ \text{vol}(G\text{-}T)\,)}$$



$$\text{vol}(S) = \sum_{v \in S} d(v)$$

= " chance a random step from inside T exits T "

Use a diffusion for good conductance sets

1. Approximate **f** so $\|\boldsymbol{D}^{-1}(\mathbf{f} - \hat{\mathbf{f}})\|_\infty \leq \epsilon$
2. Scale by **D**,
3. Then "sweep" for best conductance set.

Sweep:

1. Sort diffusion vector so $f_1/d(1) \geq f_2/d(2) \geq \cdots$
2. Consider the sweep sets S(j) = {1,2,…,j}
3. Return the set S(j) with the best conductance.

Diffusions used for conductance

Personalized PageRank (PPR) $$\mathbf{f} = \sum_{k=0}^{\infty} \alpha^k \boldsymbol{P}^k \tilde{\mathbf{s}}$$

Heat Kernel (HK) $$\mathbf{f} = \sum_{k=0}^{\infty} \frac{t^k}{k!} \boldsymbol{P}^k \tilde{\mathbf{s}}$$

Diffusions used for conductance

Personalized PageRank (PPR) $$\mathbf{f} = \sum_{k=0}^{\infty} \alpha^k \boldsymbol{P}^k \tilde{\mathbf{s}}$$

Heat Kernel (HK) $$\mathbf{f} = \sum_{k=0}^{\infty} \frac{t^k}{k!} \boldsymbol{P}^k \tilde{\mathbf{s}}$$

Time-dependent PageRank (TDPR)

$$\mathbf{f} = \sum_{k=0}^{\infty} \left[(1-\alpha)\alpha^k \left(1 - e^{-\gamma} \sum_{r=0}^{k} \frac{\gamma^r}{r!} \right) + e^{-\gamma} \frac{\alpha^k \gamma^k}{k!} \right] \boldsymbol{P}^k \mathbf{s}$$

Comes from $$\mathbf{x}'(t) = (1-\alpha)\mathbf{s} - (\boldsymbol{I} - \alpha \boldsymbol{P})\mathbf{x}(t)$$

$$\mathbf{x}(0) = \mathbf{s}$$

Diffusions used for conductance

Personalized PageRank (PPR)

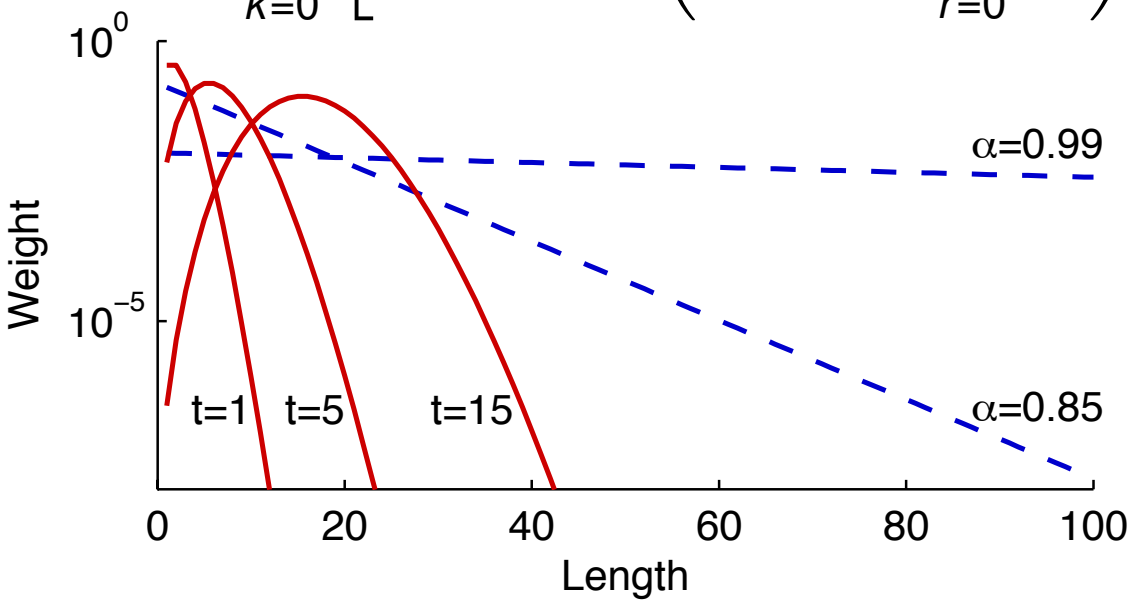
$$\mathbf{f} = \sum_{k=0}^{\infty} \alpha^k \boldsymbol{P}^k \tilde{\mathbf{s}}$$

Heat Kernel (HK)

$$\mathbf{f} = \sum_{k=0}^{\infty} \frac{t^k}{k!} \boldsymbol{P}^k \tilde{\mathbf{s}}$$

Time-dependent PageRank (TDPR)

$$\mathbf{f} = \sum_{k=0}^{\infty} \left[(1-\alpha)\alpha^k \left(1 - e^{-\gamma} \sum_{r=0}^{k} \frac{\gamma^r}{r!} \right) + e^{-\gamma} \frac{\alpha^k \gamma^k}{k!} \right] \boldsymbol{P}^k \mathbf{s}$$



Various diffusions explore different aspects of graphs.

Diffusions used for conductance

Personalized PageRank (PPR) $$\mathbf{f} = \sum_{k=0}^{\infty} \alpha^k \boldsymbol{P}^k \tilde{\mathbf{s}}$$

Heat Kernel (HK) $$\mathbf{f} = \sum_{k=0}^{\infty} \frac{t^k}{k!} \boldsymbol{P}^k \tilde{\mathbf{s}}$$

Time-dependent PageRank (TDPR)

$$\mathbf{f} = \sum_{k=0}^{\infty} \left[(1-\alpha)\alpha^k \left(1 - e^{-\gamma} \sum_{r=0}^{k} \frac{\gamma^r}{r!} \right) + e^{-\gamma} \frac{\alpha^k \gamma^k}{k!} \right] \boldsymbol{P}^k \mathbf{s}$$

Use other matrices, too: $$\mathbf{f} = \sum_{k=0}^{\infty} c_k \boldsymbol{L}^k \mathbf{s}$$

(Various weightings and scalings

of the Laplacian have been explored, [Ghosh et al. '14])

Diffusions: conductance & algorithms

	good conductance	fast algorithm
PR	Local Cheeger Inequality [Andersen,Chung,Lang 06]	[Andersen Chung Lang 06] "PPR-push" is $O(1/(\varepsilon(1-\alpha)))$
HK	Local Cheeger Inequality [Chung '07]	[Kloster, Gleich '14] "HK-push" is $O(e^t C/\varepsilon)$
TDPR	Open question	[Avron, Horesh '15] Constant-time heuristically
Gen Diff	[Ghosh et al. '14] on L; open question for general f	In revision!

Weak convergence for PageRank

Approximating a solution to $(\boldsymbol{I} - \alpha \boldsymbol{P})\mathbf{x} = \tilde{\mathbf{s}}$

residual and error satisfy $\mathbf{r}^{(k)} = \tilde{\mathbf{s}} - (\boldsymbol{I} - \alpha \boldsymbol{P})\mathbf{x}^{(k)}$

$$(\boldsymbol{I} - \alpha \boldsymbol{P})^{-1}\mathbf{r}^{(k)} = (\mathbf{x} - \mathbf{x}^{(k)})$$

$$\|\mathbf{x} - \mathbf{x}^{(k)}\| \leq \|(\boldsymbol{I} - \alpha \boldsymbol{P})^{-1}\| \|\mathbf{r}^{(k)}\|$$

for any sub-multiplicative matrix norm $\| \ \|$. Scale by $\mathbf{D}$!

Weak convergence for PageRank

Approximating a solution to $(\boldsymbol{I} - \alpha \boldsymbol{P})\mathbf{x} = \tilde{\mathbf{s}}$

residual and error satisfy $\mathbf{r}^{(k)} = \tilde{\mathbf{s}} - (\boldsymbol{I} - \alpha \boldsymbol{P})\mathbf{x}^{(k)}$

$$(\boldsymbol{I} - \alpha \boldsymbol{P})^{-1}\mathbf{r}^{(k)} = (\mathbf{x} - \mathbf{x}^{(k)})$$

$$\|\mathbf{x} - \mathbf{x}^{(k)}\| \leq \|(\boldsymbol{I} - \alpha \boldsymbol{P})^{-1}\| \|\mathbf{r}^{(k)}\|$$

for any sub-multiplicative matrix norm $\| \ \|$. Scale by $\mathbf{D}$!

$$\boldsymbol{D}^{-1}\mathbf{r}^{(k)} = \boldsymbol{D}^{-1}\tilde{\mathbf{s}} - (\boldsymbol{I} - \alpha \boldsymbol{P}^T)\boldsymbol{D}^{-1}\mathbf{x}^{(k)}$$

$$(\boldsymbol{I} - \alpha \boldsymbol{P}^T)^{-1}\boldsymbol{D}^{-1}\mathbf{r}^{(k)} = \boldsymbol{D}^{-1}(\mathbf{x} - \mathbf{x}^{(k)})$$

$$\|\boldsymbol{D}^{-1}(\mathbf{x} - \mathbf{x}^{(k)})\| \leq \|(\boldsymbol{I} - \alpha \boldsymbol{P}^T)^{-1}\| \|\boldsymbol{D}^{-1}\mathbf{r}^{(k)}\|$$

This requires A is symmetric:

$$\boldsymbol{D}^{-1}\boldsymbol{P} = \boldsymbol{D}^{-1}(\boldsymbol{A}\boldsymbol{D}^{-1}) = (\boldsymbol{D}^{-1}\boldsymbol{A})\boldsymbol{D}^{-1} = \boldsymbol{P}^T\boldsymbol{D}^{-1}$$

Weak coordinate relaxation

Approximating a solution to $(\boldsymbol{I} - \alpha \boldsymbol{P})\mathbf{x} = \tilde{\mathbf{s}}$

Residual and error satisfy

$$\|\boldsymbol{D}^{-1}(\mathbf{x} - \mathbf{x}^{(k)})\|_\infty \leq \|(\boldsymbol{I} - \alpha \boldsymbol{P}^T)^{-1}\|_\infty \|\boldsymbol{D}^{-1}\mathbf{r}^{(k)}\|_\infty$$

$$\|\boldsymbol{D}^{-1}(\mathbf{x} - \mathbf{x}^{(k)})\|_\infty \leq \frac{1}{1-\alpha} \max\{r_j/d_j\}$$

Weak coordinate relaxation

Approximating a solution to $(\boldsymbol{I} - \alpha \boldsymbol{P})\mathbf{x} = \tilde{\mathbf{s}}$

Residual and error satisfy

$$\|\boldsymbol{D}^{-1}(\mathbf{x} - \mathbf{x}^{(k)})\|_\infty \leq \|(\boldsymbol{I} - \alpha \boldsymbol{P}^T)^{-1}\|_\infty \|\boldsymbol{D}^{-1}\mathbf{r}^{(k)}\|_\infty$$

$$\|\boldsymbol{D}^{-1}(\mathbf{x} - \mathbf{x}^{(k)})\|_\infty \leq \frac{1}{1-\alpha} \max\{r_j/d_j\}$$

Contrast with 1-norm version: here simply track residual entries (degree normalized) that exceed a threshold.

This suggests a new method of choosing the coordinate j.

The rest of the update operation stays the same.

Weak coordinate relaxation, operation

Approximating a solution to $(\boldsymbol{I} - \alpha \boldsymbol{P})\mathbf{x} = \tilde{\mathbf{s}}$

Initial solution and residual: $\mathbf{x}^{(0)} = 0, \mathbf{r}^{(0)} = (1 - \alpha)\mathbf{s}$

Iterative update: a queue stores big entries: $r_j \geq \varepsilon d_j$

- pick top entry off Q(**r**), j.

Weak coordinate relaxation

Approximating a solution to $(\boldsymbol{I} - \alpha \boldsymbol{P})\mathbf{x} = \tilde{\mathbf{s}}$

Initial solution and residual: $\mathbf{x}^{(0)} = 0, \mathbf{r}^{(0)} = (1 - \alpha)\mathbf{s}$

Iterative update: a queue stores big entries: $r_j \geq \varepsilon d_j$

- pick top entry off Q(**r**), j.
- update solution: $x_j^{(k+1)} = x_j^{(k)} + r_j$
- update residual:

$$r_i^{(k+1)} = \begin{cases} 0, & \text{if } i = j \\ r_i^{(k)} + r_j \alpha / d_j, & \text{if } i \sim j \\ r_i^{(k)}, & \text{else} \end{cases}$$

Weak coordinate relaxation

Approximating a solution to $(\boldsymbol{I} - \alpha \boldsymbol{P})\mathbf{x} = \tilde{\mathbf{s}}$

Initial solution and residual: $\mathbf{x}^{(0)} = 0, \mathbf{r}^{(0)} = (1 - \alpha)\mathbf{s}$

Iterative update: a queue stores big entries: $r_j \geq \varepsilon d_j$

- pick top entry off Q(**r**), j.
- update solution: $x_j^{(k+1)} = x_j^{(k)} + r_j$
- update residual:

$$r_i^{(k+1)} = \begin{cases} 0, & \text{if } i = j \\ r_i^{(k)} + r_j \alpha / d_j, & \text{if } i \sim j \\ r_i^{(k)}, & \text{else} \end{cases}$$

- for $i \sim j$: $r_i \geq \varepsilon d_i$, add r_i to Q(**r**) if not present

Weak coordinate relaxation, work bound

1. Every update is on an entry in Q(**r**) satisfying $\varepsilon d_i \leq r_i$

2. Sum of updates is $\sum_{t=1} r_{i(t)} = \sum_{k=1}^{n} \hat{f}_k \leq 1$

3. Total work is $\sum_{t=1} d_{i(t)}$

Weak coordinate relaxation, work bound

1. Every update is on an entry in Q(**r**) satisfying $\varepsilon d_i \leq r_i$

2. Sum of updates is $\sum_{t=1} r_{i(t)} = \sum_{k=1}^{n} \hat{f}_k \leq 1$

3. Total work is $\sum_{t=1} d_{i(t)}$

All together: $\varepsilon \sum_{t=1} d_{i(t)} = \sum_{t=1} \varepsilon d_{i(t)} \leq \sum_{t=1} r_{i(t)} \leq 1$

Work is bounded by $1/((1-\alpha)\varepsilon)$
a constant independent of the graph size!

Andersen, Chung, and Lang, 2006

Weak coordinate relaxation, work bound

1. Every update is on an entry in Q(**r**) satisfying $\varepsilon d_i \leq r_i$

2. Sum of updates is $\sum_{t=1} r_{i(t)} = \sum_{k=1}^{n} \hat{f}_k \leq 1$

3. Total work is $\sum_{t=1} d_{i(t)}$

All together: $\varepsilon \sum_{t=1} d_{i(t)} = \sum_{t=1} \varepsilon d_{i(t)} \leq \sum_{t=1} r_{i(t)} \leq 1$

Work is bounded by $1/((1-\alpha)\varepsilon)$
a constant independent of the graph size!

(comes from $\mathbf{r}^{(0)} = (1-\alpha)\mathbf{s}$)

Weak coordinate relaxation remarks

- [Ghosh et al ’14] proved Cheeger inequalities for related diffusions that use weighted Laplacians
- Is there a related Cheeger inequality, and a constant-time algorithm, for the degree-normalized Katz diffusion?
- Is there a “best” set of diffusion coefficients for identifying particular structures?
- Can we improve on the sweep procedure? Or bound its performance?
  - [Kenter et al. ‘15] introduced a randomized subroutine that improves on sweep in certain conditions

Related work: weak coordinate relaxation

- Deterministic coordinate descent
  - [Andersen, Chung, Lang ’06] Local Graph Partitioning
  - [Andersen, Lang ’06] Communities from Seed Sets
  - [Kloster, Gleich ’14] Heat Kernel clustering
  - [Kloumann, Kleinberg ’14] Community membership from seed
  - [Ghosh et al. ’14] Interplay between dynamics and networks
  - [Avron, Horesh ’15] Time-Dependent PageRank clustering

Brief outline

Introduction & Application 20-30 min
Coordinate Relaxation for Strong Convergence 30 min
Break 15 min

Katz Diffusion 10 min
Weak Convergence for PageRank 20 min
Monte Carlo methods 15 min
Break 15 min

Implicit Regularization 25 min
Discussion 15 min

Monte Carlo methods for diffusion vectors

Monte Carlo motivation

Benefits of MC over deterministic?

- strong convergence method drastically slows down as it operates on nodes of large degree
- weak convergence method gets no accuracy when it encounters nodes of large degree
- MC avoids out-link accesses best

Monte Carlo method for Matrix inversion

[Forsyth & Liebler, 1950]
Matrix Inversion by a Monte Carlo method:

Want $(\boldsymbol{B}^{-1})_{ij}$, so design a game such that the expected value is exactly $(\boldsymbol{B}^{-1})_{ij}$. It has inspired other work:

Monte Carlo method for Matrix inversion

[Forsyth & Liebler, 1950]
Matrix Inversion by a Monte Carlo method:

Want $(\boldsymbol{B}^{-1})_{ij}$, so design a game such that the expected value is exactly $(\boldsymbol{B}^{-1})_{ij}$. It has inspired other work:

- [K. Avrachenkov '05] MC methods in PageRank
- [Fogaras et al. '05] Fully scaling personalized PageRank
- [Das Sarma et al. '08] Estimating PageRank on graph streams
- [Bahmani '10] Fast incremental Personalized PageRank
- [Bahmani '10] PageRank & MapReduce
- [Borgs '12] Sublinear PageRank
- [Chung, Simpson WAW13] Solving systems w/ heat kernel

Designing a game for PageRank

We want a specific entry of say, $\mathbf{f}_i$.

$$\mathbf{f} = (1-\alpha)\sum_{k=0}^{\infty}\alpha^k \boldsymbol{P}^k \mathbf{e}_j$$

GOAL: design a random process for producing $\hat{\mathbf{f}}$ so that the expected value of each entry is the true value.

Designing a game for PageRank

We want a specific entry of say, $\mathbf{f}_i$.

$$\mathbf{f} = (1-\alpha)\sum_{k=0}^{\infty}\alpha^k \boldsymbol{P}^k \mathbf{e}_j$$

GOAL: design a random process for producing $\hat{\mathbf{f}}$ so that the expected value of each entry is the true value.

Observe that

$$\mathbf{f}_i = (1-\alpha)\sum_{k=0}^{\infty}\alpha^k (\boldsymbol{P}^k)_{ij}$$

$$= (1-\alpha)\sum_{k=0}^{\infty}\alpha^k \left(\sum_{p_k(i,j)}\sum_{i1\in N(j)}\sum_{i2\in N(i1)}\cdots\sum_{ik\in N(i(k-1))} P_{ik,i(k-1)}\cdots P_{i2,i1}P_{i1,j}\right)$$

where $p_k(i,j)$ is the set of all k-walks from j to i (and $i_k = i$).

Designing a game for PageRank

We'll convert this so it looks like an expected value:

$$\mathbf{f}_i = (1-\alpha)\sum_{k=0}^{\infty}\alpha^k\left(\sum_{p_k(i,j)}\sum_{i1\in N(j)}\sum_{i2\in N(i1)}\cdots\sum_{ik\in N(i(k-1))}P_{ik,i(k-1)}\cdots P_{i2,i1}P_{i1,j}\right)$$

Note that P(ik, i(k-1)) ... P(i1,j), is the probability of taking a specific walk, $w_k(i,j)$.

Designing a game for PageRank

We'll convert this so it looks like an expected value:

$$\mathbf{f}_i = (1-\alpha)\sum_{k=0}^{\infty}\alpha^k\left(\sum_{p_k(i,j)}\sum_{i1\in N(j)}\sum_{i2\in N(i1)}\cdots\sum_{ik\in N(i(k-1))}P_{ik,i(k-1)}\cdots P_{i2,i1}P_{i1,j}\right)$$

Note that P(ik, i(k-1)) ... P(i1,j), is the probability of taking a specific walk, $w_k(i,j)$. We can rewrite...

$$\mathbf{f}_i = (1-\alpha)\sum_{k=0}^{\infty}\alpha^k\left(\sum_{w_k(i,j)\in p_k(i,j)}\mathbb{P}(w_k(i,j))\right)$$

$$= \sum_{k=0}^{\infty}\left(\sum_{w_k(i,j)\in p_k(i,j)}(1-\alpha)\alpha^k\mathbb{P}(w_k(i,j))\right)$$

Designing a game for PageRank

GOAL: convert so it looks like an expected value.

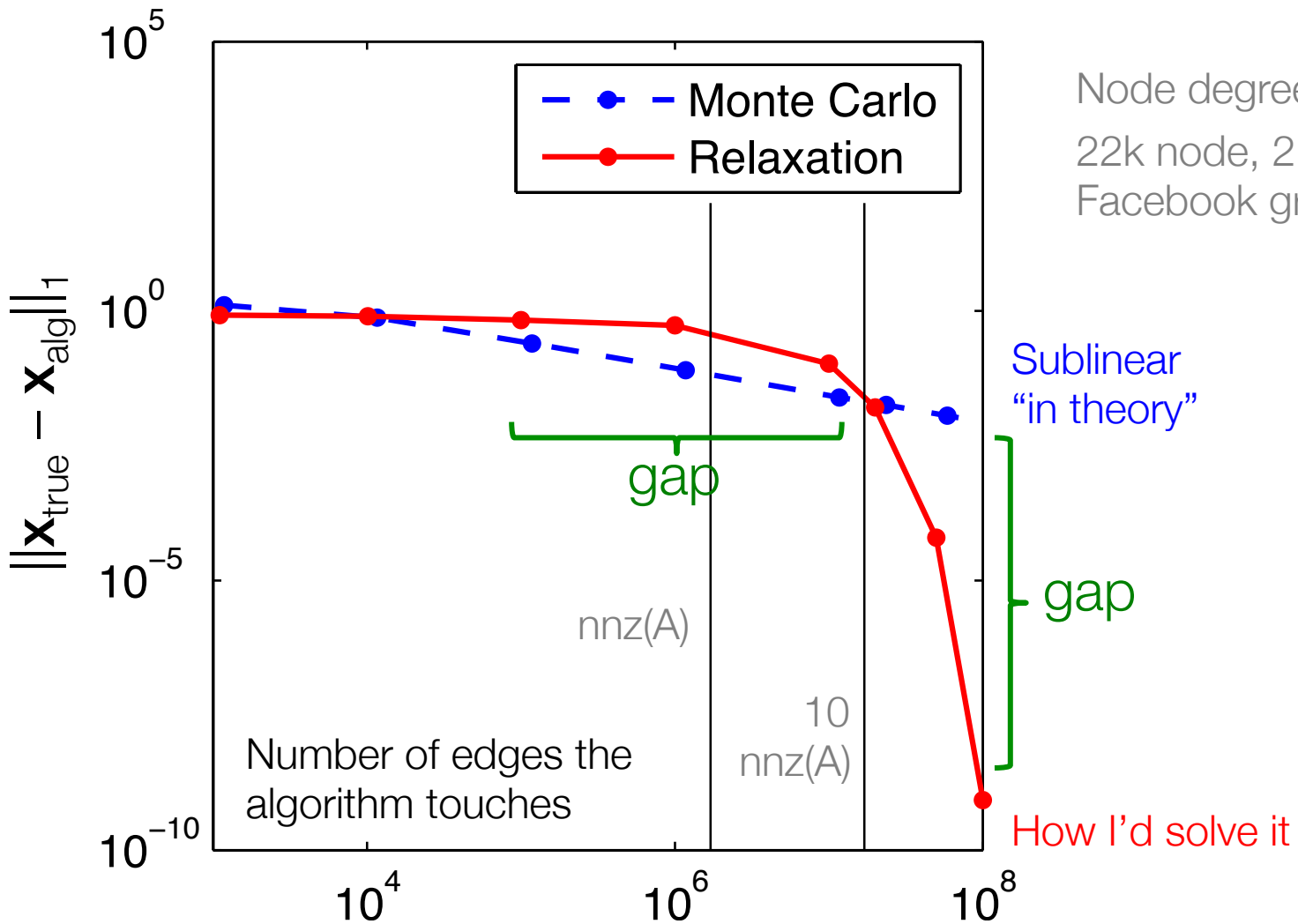
$$\mathbf{f}_i = \sum_{k=0}^{\infty} \left(\sum_{w_k(i,j) \in p_k(i,j)} (1-\alpha)\alpha^k \mathbb{P}(w_k(i,j)) \cdot 1 \right)$$

This is the expected value of:

1. Choose a length k with probability $(1-\alpha)\alpha^k$
2. Make a random k-walk from j. It lands at node i_k
3. Update solution where walk ends, i_k, by adding 1.

Hence for a single iteration of this, $\mathbb{E}(\hat{\mathbf{f}}_i) = \mathbf{f}_i$

Monte Carlo vs Deterministic



Monte Carlo Remarks

Accuracy, convergence are problematic (previous slide)

OPEN QUESTION: can we improve on number of samples / random walks required, or the accuracy attained?

Related work: Monte Carlo

- Monte Carlo methods:
  - [Forsyth Liebler ’50] fore-runner
  - [K. Avrachenkov ’05] MC methods in PageRank
  - [Fogaras et al. ’05] Fully scaling personalized PageRank
  - [Das Sarma et al. ’08] Estimating PageRank on graph streams
  - [Bahmani ’10] Fast incremental Personalized PageRank
  - [Bahmani ’10] PageRank & MapReduce
  - [Borgs ’12] Sublinear PageRank
  - [Chung, Simpson WAW13] Solving systems w/ heat kernel
- Hybrid monte carlo / coordinate relaxation
  - [Lofgren et al. 2014], node-to-node PPR estimate

Brief outline

Introduction & Application 20-30 min
Coordinate Relaxation for Strong Convergence 30 min
Break 15 min

Katz Diffusion 10 min
Weak Convergence for PageRank 20 min
Monte Carlo methods 15 min
Break 15 min

Implicit Regularization 25 min
Discussion 15 min

Statistical regularization

Best? known instance: sparsity and the Lasso

$$\text{Least squares} \qquad \underset{\mathbf{x}}{\text{minimize}} \quad \|\boldsymbol{A}\mathbf{x} - \mathbf{b}\|_2^2$$

$$\text{Lasso} \qquad \underset{\mathbf{x}}{\text{minimize}} \quad \|\boldsymbol{A}\mathbf{x} - \mathbf{b}\|_2^2 + \lambda\|\mathbf{x}\|_1$$

The Lasso solution tends to produce *sparse* solutions.

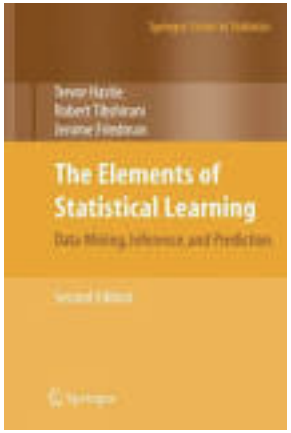
- … Candès and Tao formalized the relationship with the sparsest solution
- Prevents overfitting/overtraining on a given sample
- Related to Bayesian priors

Statistical regularization

Least squares $\quad \underset{\mathbf{x}}{\text{minimize}} \quad \|\boldsymbol{A}\mathbf{x} - \mathbf{b}\|_2^2$

Lasso $\quad \underset{\mathbf{x}}{\text{minimize}} \quad \|\boldsymbol{A}\mathbf{x} - \mathbf{b}\|_2^2 + \lambda\|\mathbf{x}\|_1$

Choose the regularizer to counter a given noise type



See Hastie, Tibshirani, Friedman 2009
The Elements of Statistical Learning: Data Mining, Inference, and Prediction

Implicit regularization

1. Run an algorithm procedure
2. Show that your algorithm implicitly is tolerant to a type of noise.

Implementing Regularization Implicitly Via Approximate Eigenvector Computation

Michael W. Mahoney MMAHONEY@CS.STANFORD.EDU
Department of Mathematics, Stanford University, Stanford, CA 94305

Lorenzo Orecchia ORECCHIA@EECS.BERKELEY.EDU
Computer Science Division, UC Berkeley, Berkeley, CA 94720

Consider a graph diffusion (of our type) these diffusions implicitly regularize the resulting solution vectors to be tolerant to noise.

Mahoney & Orecchia 2011

An example theorem

SPECTRAL CLUSTERING

$$\begin{aligned} \text{minimize} \quad & \textstyle\sum_{ij} \mathcal{L}_{ij} X_{ij} \\ \text{subject to} \quad & \text{trace}(\boldsymbol{X}) = 1 \\ & \boldsymbol{X} \succeq 0 \end{aligned}$$

REGULARIZED SPECTRAL CLUSTERING

$$\begin{aligned} \text{minimize} \quad & \textstyle\sum_{ij} \mathcal{L}_{ij} X_{ij} + \lambda F(\boldsymbol{X}) \\ \text{subject to} \quad & \text{trace}(\boldsymbol{X}) = 1 \\ & \boldsymbol{X} \succeq 0 \end{aligned}$$

Let $F(\boldsymbol{X}) = \text{trace}(\boldsymbol{X} \log \boldsymbol{X}) - \text{trace}(\boldsymbol{X})$
then the solution of regularized spectral is
$\boldsymbol{X} = C \exp(-(1/\lambda)\mathcal{L})$

You do nothing special!
Using the heat kernel implicitly regularizes solutions against noise characterized by the generalized entropy function

Mahoney & Orecchia 2011

More examples

SPECTRAL CLUSTERING

$$\begin{aligned} \text{minimize} \quad & \textstyle\sum_{ij} \mathcal{L}_{ij} X_{ij} \\ \text{subject to} \quad & \operatorname{trace}(\boldsymbol{X}) = 1 \\ & \boldsymbol{X} \succeq 0 \end{aligned}$$

REGULARIZED SPECTRAL CLUSTERING

$$\begin{aligned} \text{minimize} \quad & \textstyle\sum_{ij} \mathcal{L}_{ij} X_{ij} + \lambda F(\boldsymbol{X}) \\ \text{subject to} \quad & \operatorname{trace}(\boldsymbol{X}) = 1 \\ & \boldsymbol{X} \succeq 0 \end{aligned}$$

PageRank

Let $F(\boldsymbol{X}) = \log\det \boldsymbol{X}$
then the solution of regularized spectral is
$\boldsymbol{X} = C(\boldsymbol{I} - \alpha\boldsymbol{\mathcal{L}})^{-1}$

Truncation

Let $F_p(\boldsymbol{X}) = 1/p \operatorname{trace}(\boldsymbol{X}^p)$
then the solution of regularized spectral is
$\boldsymbol{X} = C(\boldsymbol{\mathcal{A}})^{q-1}$ where $1/p + 1/q = 1$

These results should be true up to degree normalization on the solution.

Our question

Why does the “push method” have such incredible empirical utility?

Answer

Gleich & Mahoney, ICML 2014.
Anti-differentiating approximation algorithms, a case study with min-cuts, spectral, and flow.

Algorithmic Anti-differentiation

Understanding how and why heuristic procedures

- Early stopping
- Truncating small entries
- etc

are actually *algorithms* for implicit objectives.

The ideal world

Given Problem P

Derive solution characterization C

Show algorithm A finds a solution where C holds

Profit?

Given “min-cut”

Derive “max-flow is equivalent to min-cut”

Show push-relabel solves max-flow

Profit!

(The ideal world)’

Given Problem P

Derive solution approx. characterization C’

Show algorithm A’ quickly finds a solution where C’ holds

Profit?

Given “sparest-cut”

Derive Rayleigh-quotient approximation

Show power-method finds a good Rayleigh-quotient

Profit?

The real world?

Given Task P

Hack around until you find something useful

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

Profit!

Given "find-communities"

Hack around ??? (hidden) ???

Write paper presenting "three matvecs finds real-world communities"

Profit!

Algorithmic Anti-differentiation

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

Understand why H works

Show heuristic H solves P'

Guess and check until you find something H solves

Derive characterization of heuristic H

Given "find-communities"

Hack around

Write paper presenting "three matvecs finds real-world communities"

Profit!

e.g. Mahoney & Orecchia

Algorithmic Anti-differentiation

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 X, what objective does it optimize?

In an unconstrained case, this is just "anti-differentiation!"

Our question

Why does the “push method” have such incredible empirical utility?

Answer

Gleich & Mahoney, ICML 2014.
Anti-differentiating approximation algorithms, a case study with min-cuts, spectral, and flow.

The O(correct) answer

1. PageRank related to Laplacian
2. Laplacian related to cuts
3. Andersen, Chung, Lang provides the "right" bounds and "localization"

Now the θ(correct) answer?

A deeper insight into the relationship

Intellectually indebted to …

Chin, Mądry, Miller & Peng [2013]

Orecchia & Zhu [2014]

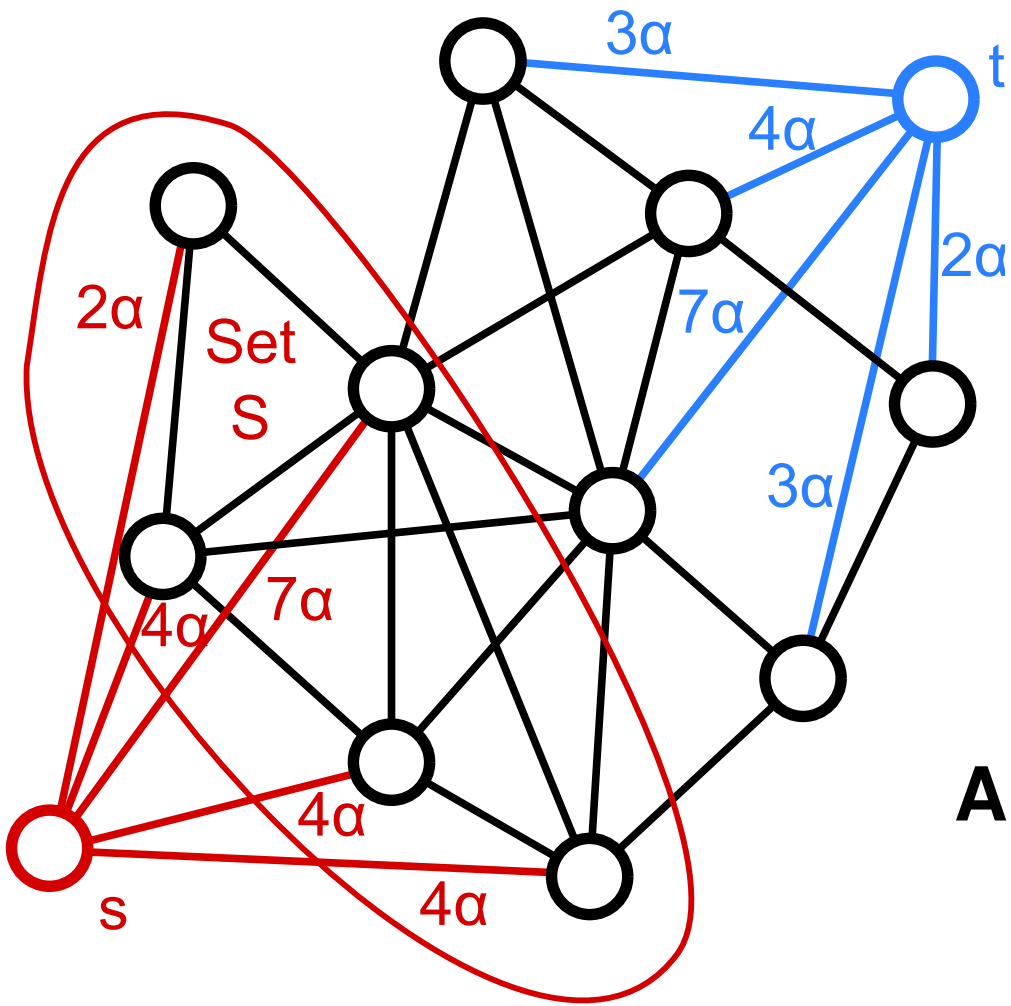
The s-t min-cut problem

Unweighted incidence matrix

Diagonal capacity matrix

$$\begin{array}{ll} \text{minimize} & \|\mathbf{B}\mathbf{x}\|_{C,1} = \sum_{ij \in E} C_{i,j} |x_i - x_j| \\ \text{subject to} & x_s = 1, x_t = 0, \mathbf{x} \geq 0. \end{array}$$

The localized cut graph

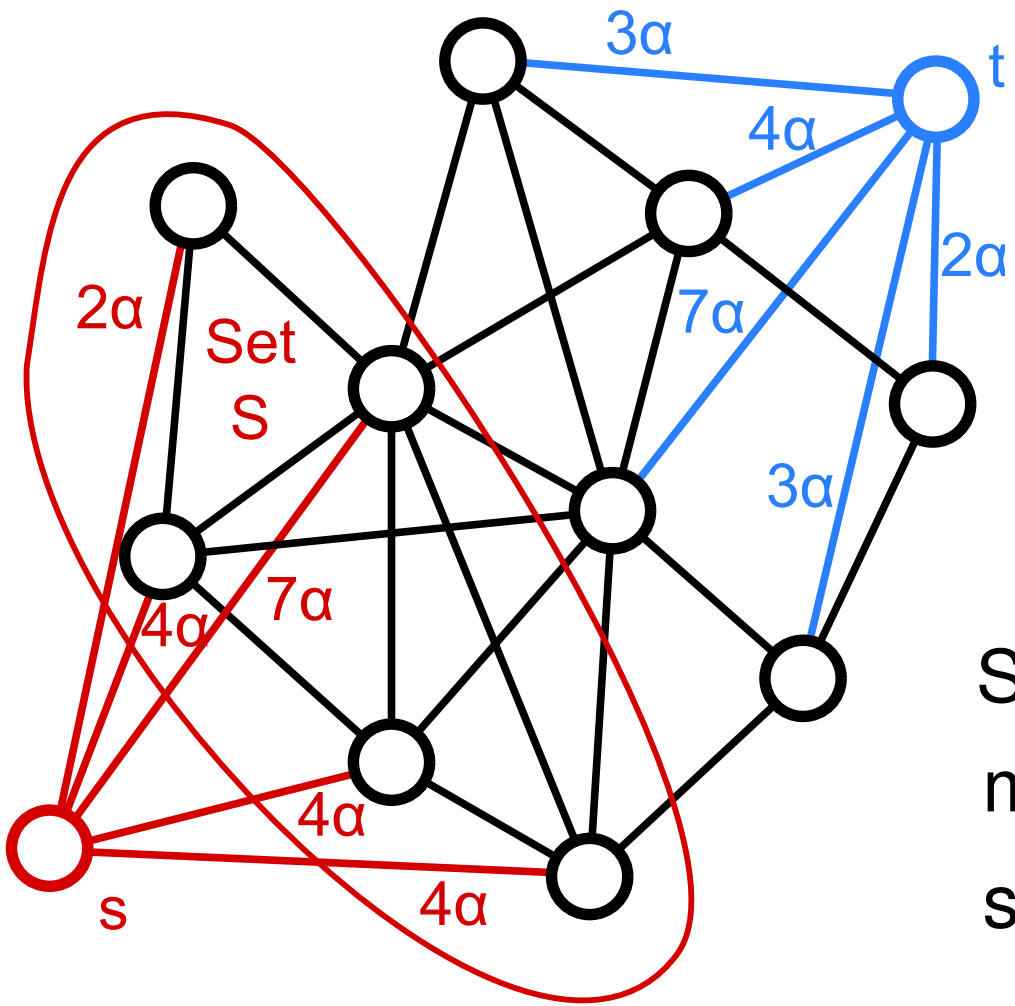


Connect s to vertices in S with weight $\alpha \cdot$ degree
Connect t to vertices in $\bar{S}$ with weight $\alpha \cdot$ degree

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



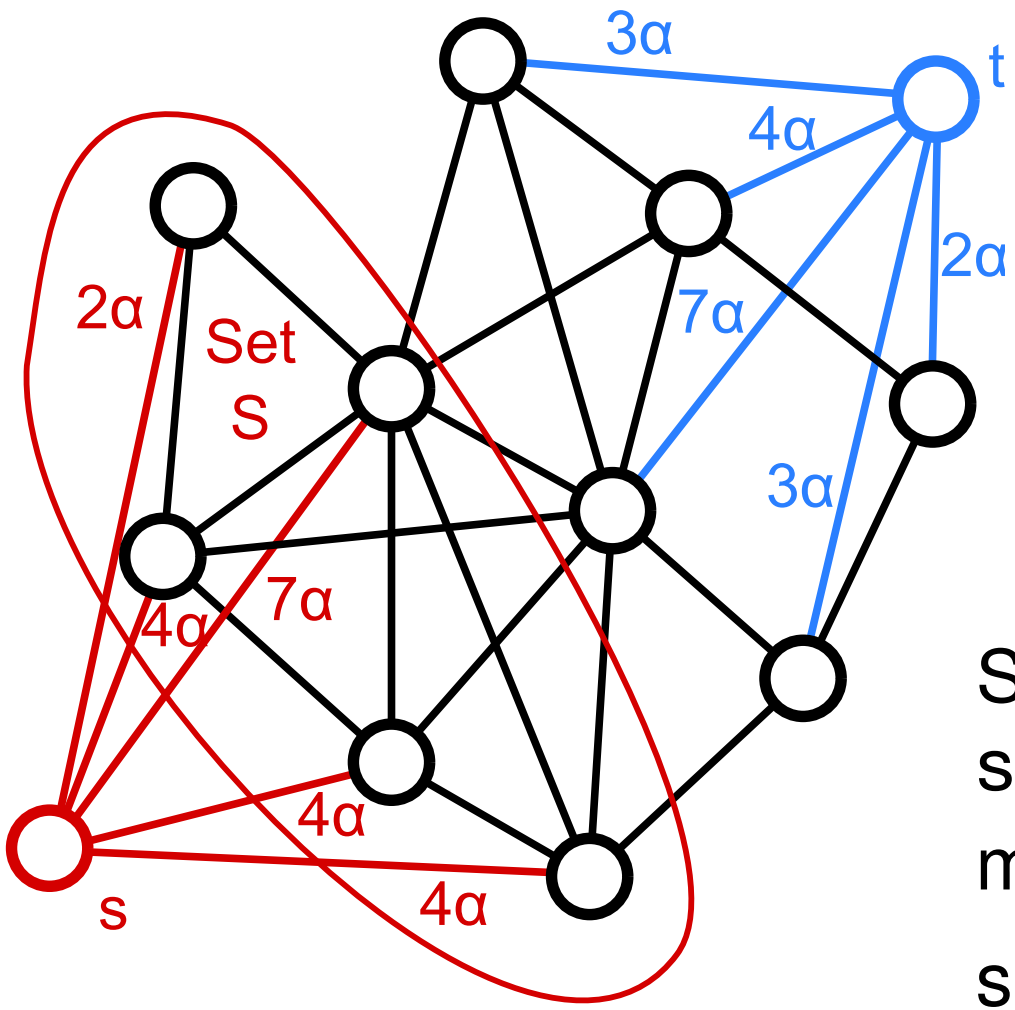
Connect s to vertices in S with weight $\alpha \cdot$ degree
Connect t to vertices in $\bar{S}$ with weight $\alpha \cdot$ degree

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

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



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

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

subject to $x_s = 1, x_t = 0$

s-t min-cut → PageRank

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:

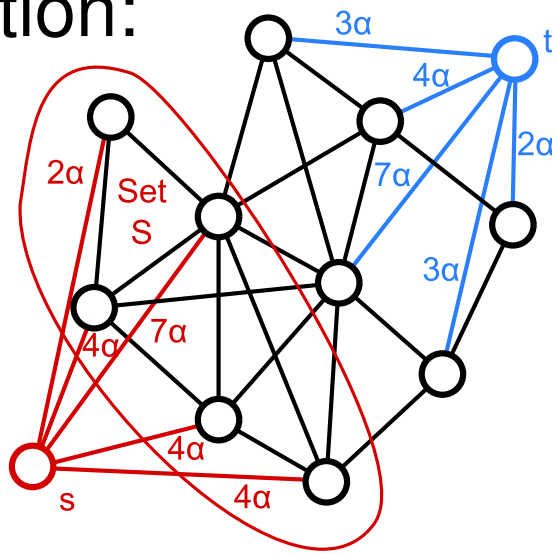
$$\begin{array}{ll} \text{minimize} & \|\mathbf{B}_S\mathbf{x}\|_{C(\alpha),2} \\ \text{subject to} & x_s = 1, x_t = 0. \end{array}$$

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.

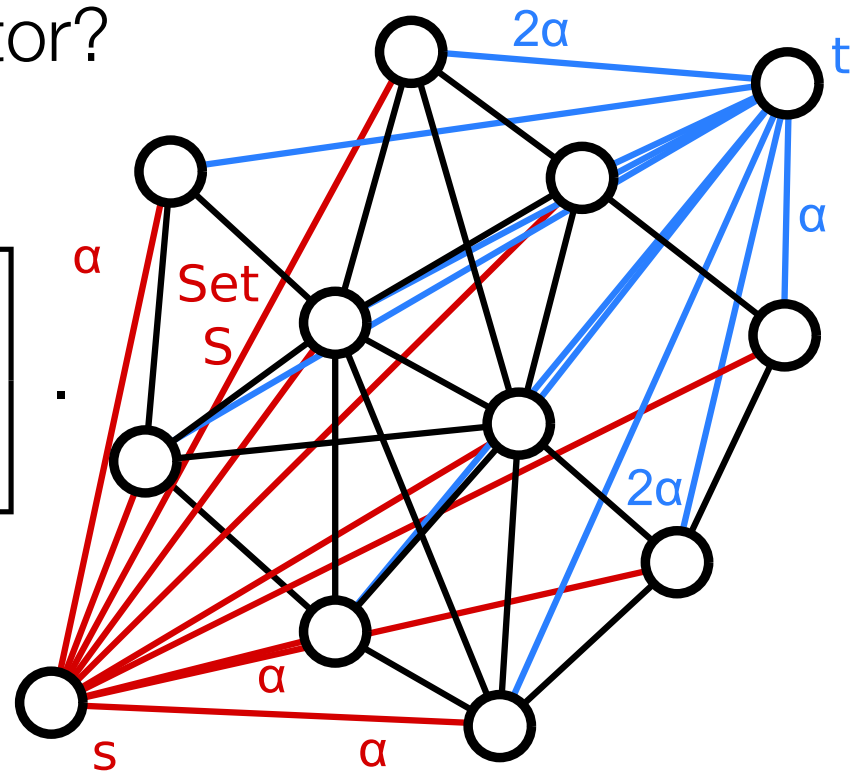


PageRank → s-t min-cut

That equivalence works if **v** is degree-weighted.

What if **v** is the uniform vector?

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



And beyond …

$$\begin{bmatrix} 0 & \mathbf{e}_S^T & 0 \\ \mathbf{e}_S & \theta\mathbf{A} & \mathbf{e}_{\bar{S}} \\ 0 & \mathbf{e}_{\bar{S}} & 0 \end{bmatrix}. \qquad (\mathbf{I} + \theta\mathbf{L})\mathbf{x} = \mathbf{e}_S$$

Easy to cook up interesting diffusion-like problems and adapt them to this framework. In particular, Zhou et al. (2004) gave a semi-supervised learning diffusion we study soon.

The 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 (as Kyle mentioned!)

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. $\mathbf{r}_i^{(k+1)} = \begin{cases} \tau d_j \rho & i = j \\ r_i^{(k)} + \beta(r_j - \tau d_j \rho)/d_j & i \sim j \\ r_i^{(k)} & \text{otherwise} \end{cases}$
5. $k \leftarrow k + 1$

Back to the push method

Let $\mathbf{x}$ 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"

Some reflections on algorithmic anti-differentiating

Differentiation
Given f(x), computing g(x) = f'(x) analytically usually isn't too hard.

Anti-differentiation
Given f(x), computing F(x) where f(x) = F'(x) can be very hard and or impossible

Algorithms solve for the KKT conditions (e.g. f(x) = 0)

Algorithmic anti-differentiation finds the objective that the algorithm solves (e.g. minimize F(x))

For simple optimization, this analogy is precise.

In general, this is very hard or impossible.
(3-4 years for PageRank)

Your question?

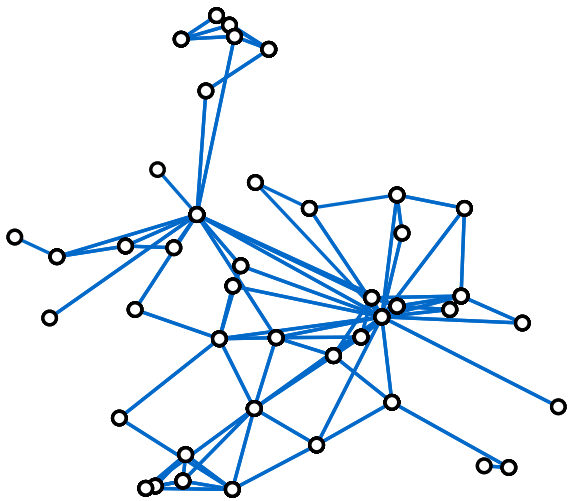
So what? Why does this matter?

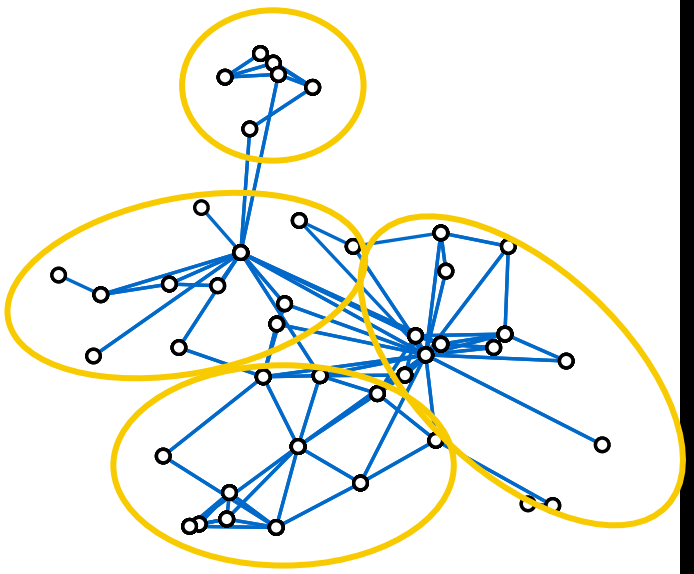
Answer

Gleich & Mahoney, KDD 2015
Using local spectral methods to robustify graph-based learning.

The graph-based data analysis pipeline

1	0	0	0	1	0	0	1	
0	1	0	1	0	0	1	1	
0	1	0	1	0	0	0	1	
1	0	0	0	0	0	1	1	
1	1	0	1	1	1	0	1	
1	0	1	1	0	0	0	1	
1	0	1	1	1	0	1	0	
1	1	1	1	0	1	0	0	
1	1	1	0	0	1	1	1	
1	1	0	1	1	1	1	1	





Raw data

- Relationships
- Images
- Text records
- Etc.

Convert to a graph

- Nearest neighs
- Kernels
- 2-mode to 1-mode
- Etc.

Algorithm/Learning

- Important nodes
- Infer features
- Clustering
- Etc.

“Noise” in the initial data modeling decisions

Explicit graphs are those that are given to a data analyst.

“A social network”

- Known spam accounts included?
- Users not logged in for a year?
- Etc.

A type of noise

Constructed graphs are built based on some other primary data.

“nearest neighbor graphs”

- K-NN or ε-NN
- Thresholding correlations to zero

Often made for computational convenience! (Graph too big.)

A different type a noise!

Labeled graphs occur in information diffusion/propagation

“function prediction”

- Labeled nodes
- Labeled edges
- Some are wrong

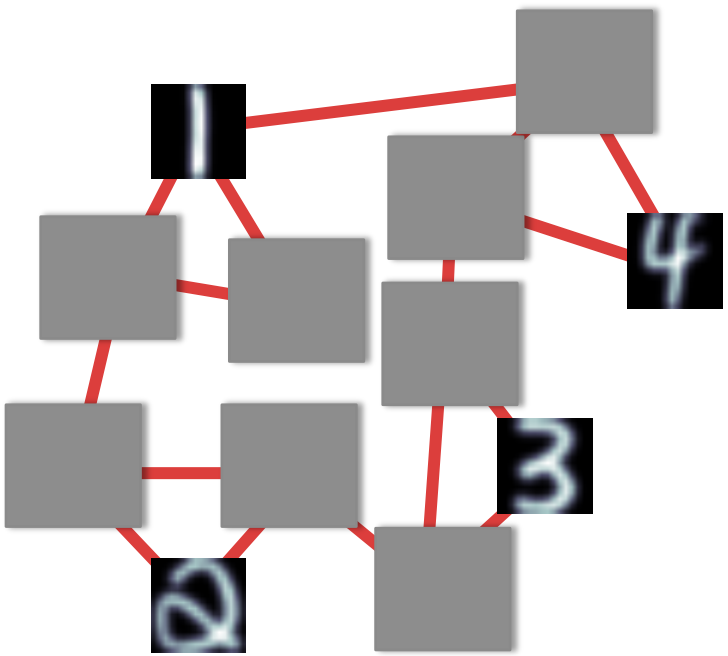
A direct type of noise!

Do these decisions matter?

Our experience Yes! Dramatically so!

Semi-supervised graph-based learning

Given a graph, and a few labeled nodes, predict the labels on the rest of the graph.

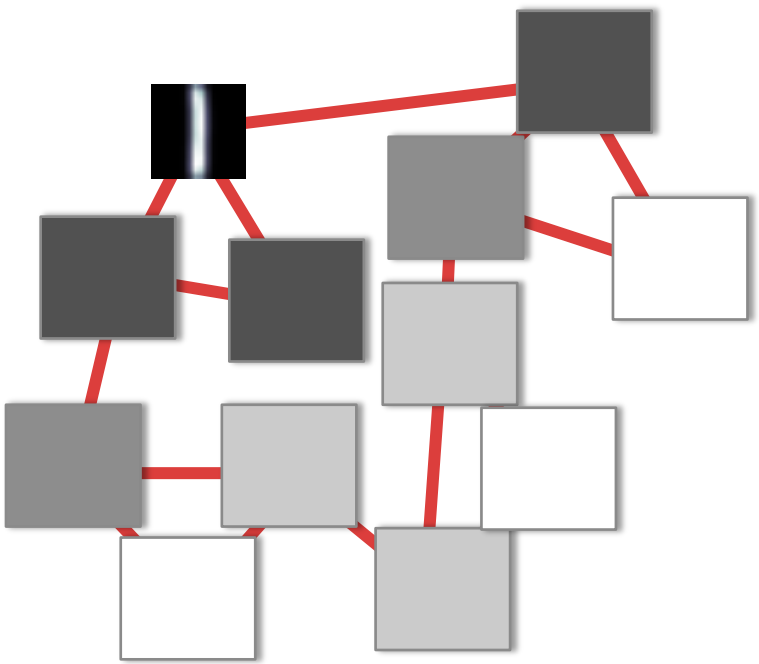


Algorithm

1. Run a diffusion for each label (possibly with neg. info from other classes)
2. Assign new labels based on the value of each diffusion

Semi-supervised graph-based learning

Given a graph, and a few labeled nodes, predict the labels on the rest of the graph.

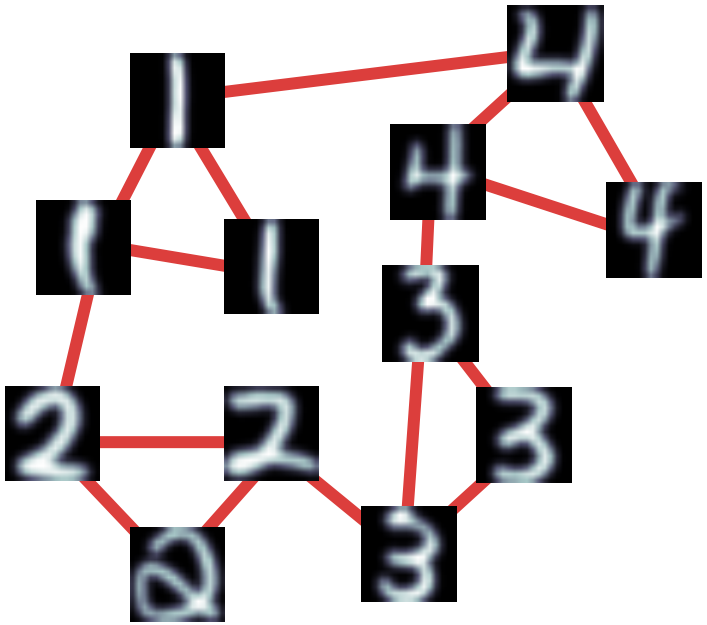


Algorithm

1. Run a diffusion for each label (possibly with neg. info from other classes)
2. Assign new labels based on the value of each diffusion

Semi-supervised graph-based learning

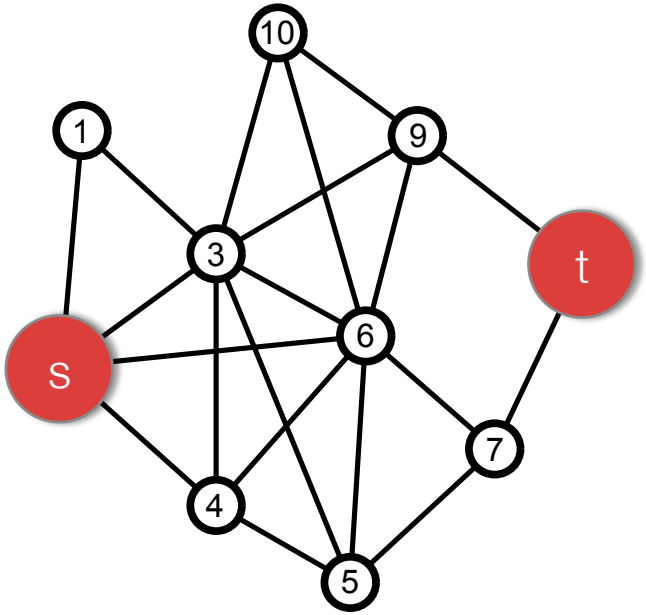
Given a graph, and a few labeled nodes, predict the labels on the rest of the graph.



Algorithm

1. Run a diffusion for each label (possibly with neg. info from other classes)
2. Assign new labels based on the value of each diffusion

The diffusions proposed for semi-supervised learning are s,t-cut minorants



In the unweighted case, solve via max-flow.

In the weighted case, solve via network simplex or industrial LP.

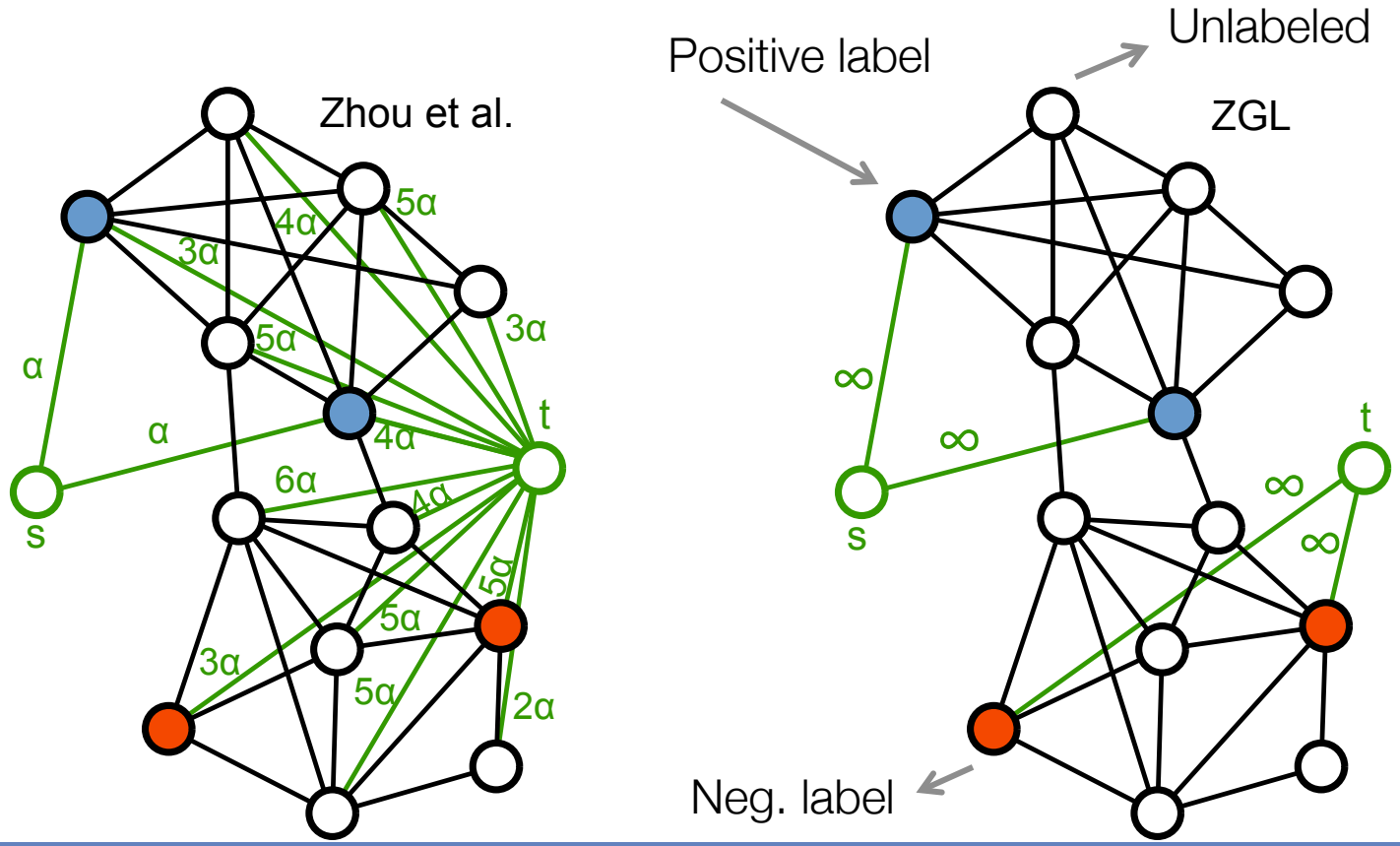
$$\begin{array}{ll} \text{minimize} & \sum_{ij \in E} C_{i,j} |x_i - x_j| \\ \text{subject to} & x_s = 1, x_t = 0. \end{array}$$

MINCUT LP

$$\begin{array}{ll} \text{minimize} & \sqrt{\sum_{ij \in E} C_{i,j} |x_i - x_j|^2} \\ \text{subject to} & x_s = 1, x_t = 0. \end{array}$$

Spectral minorant – lin. sys.

Representative cut problems

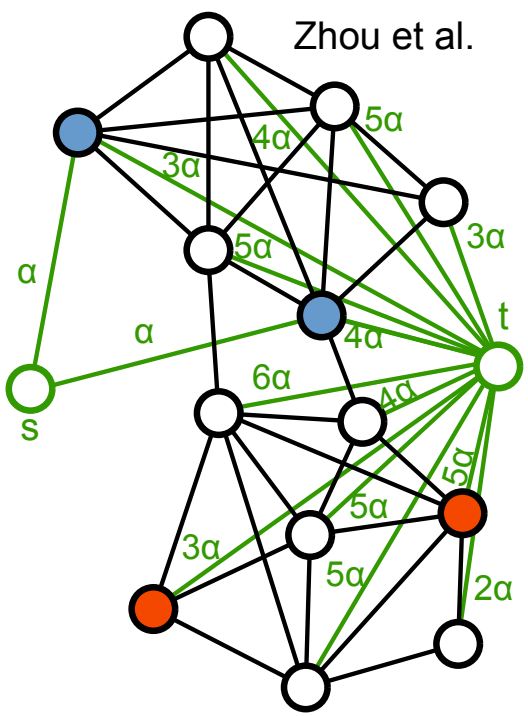


These help our intuition about the solutions
All spectral minorants are linear systems.

Zhou et al. NIPS 2003; Zhu et al., ICML 2003;
Andersen Lang, SODA 2008; Joachims, ICML 2003

Implicit regularization views on the Zhou et al. diffusion

The Mahoney-Orecchia-Vishnoi (MOV) vector is a localized variation on the Fiedler vector to find a small conductance set nearby a seed set.



$$\text{minimize} \quad \sqrt{\sum_{ij \in E} C_{i,j} |x_i - x_j|^2}$$

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

RESULT

The spectral minorant of Zhou is equivalent to the weakly-local MOV solution.

PROOF

The two linear systems are the same (after working out a few equivalences).

IMPORTANCE

We'd expect Zhou to be "more robust"

A scalable, localized algorithm for Zhou et al's diffusion.

$$\begin{array}{ll} \text{minimize} & \sqrt{\sum_{ij \in E} C_{i,j} |x_i - x_j|^2} \\ \text{subject to} & x_s = 1, x_t = 0. \end{array}$$

RESULT

We can use a variation on coordinate descent methods related to the Andersen-Chung-Lang PUSH procedure to solve Zhou's diffusion in a scalable manner.

PROOF. See Gleich-Mahoney ICML '14

IMPORTANCE (1)

We should be able to make Zhou et al. scale.

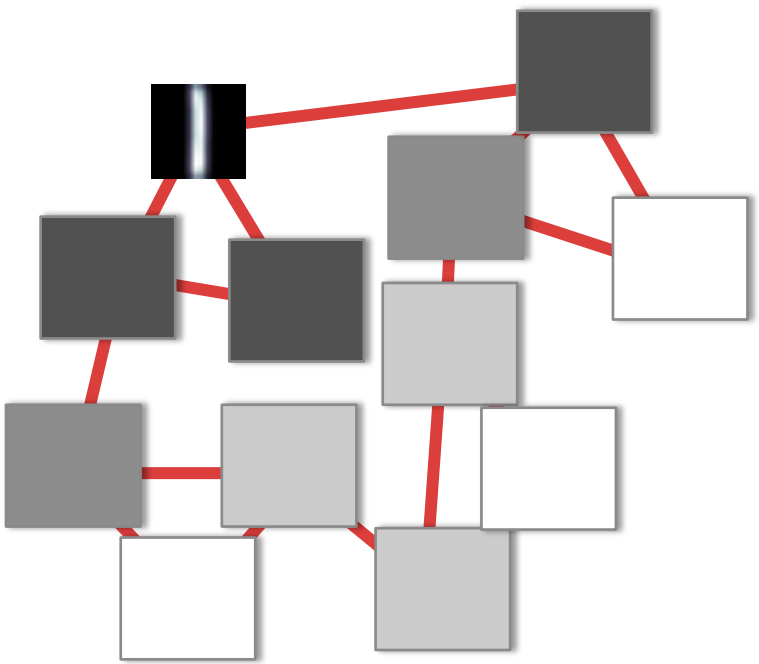
IMPORTANCE (2)

Using this algorithm adds another implicit regularization term that should further improve robustness!

$$\begin{array}{ll} \text{minimize} & \boxed{\sum_{ij \in E} C_{i,j} |x_i - x_j|^2} + \boxed{\tau \sum_{i \in V} d_i x_i} \\ \text{subject to} & x_s = 1, x_t = 0, x_i \geq 0. \end{array}$$

Semi-supervised graph-based learning

Given a graph, and a few labeled nodes, predict the labels on the rest of the graph.



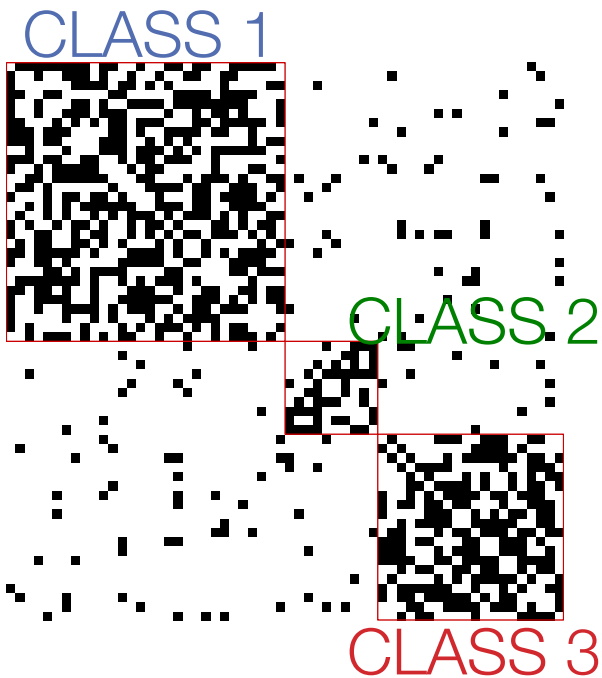
Algorithm

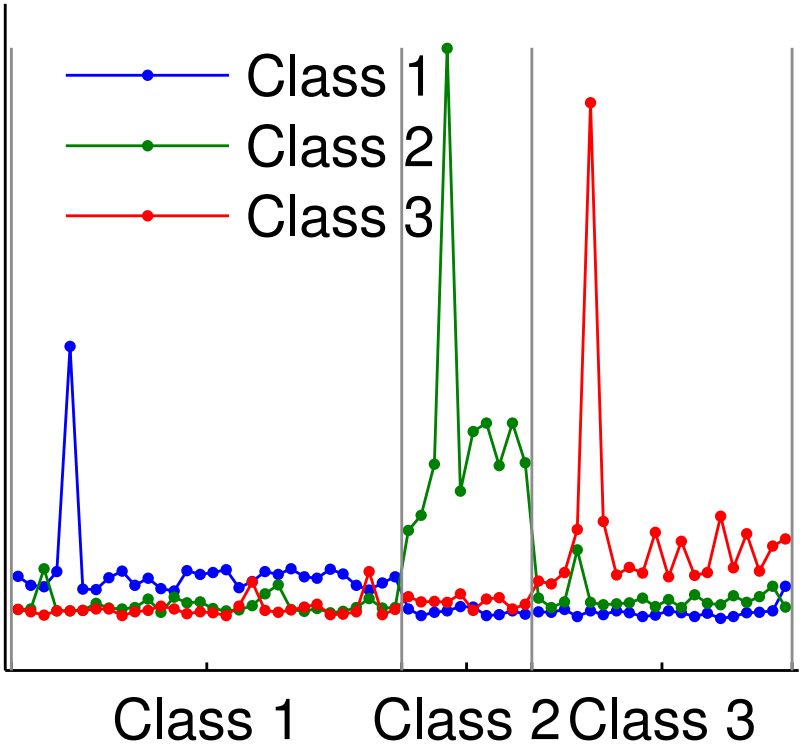
1. Run a di
   each label (po
   with neg. info from
   other classes)
2. Assign new labels based on the value of each diffusion

New interpretation of the diffusions and more scalable (and robust!) versions!

Traditional rounding methods for SSL are value-based

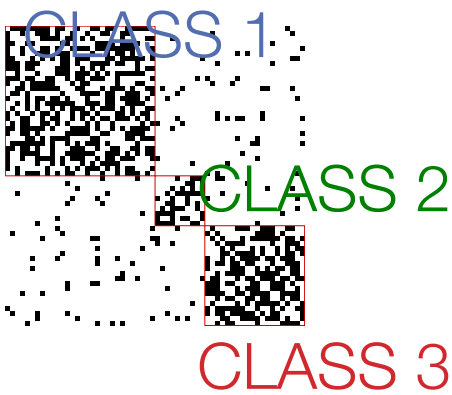
Zhou's diffusion

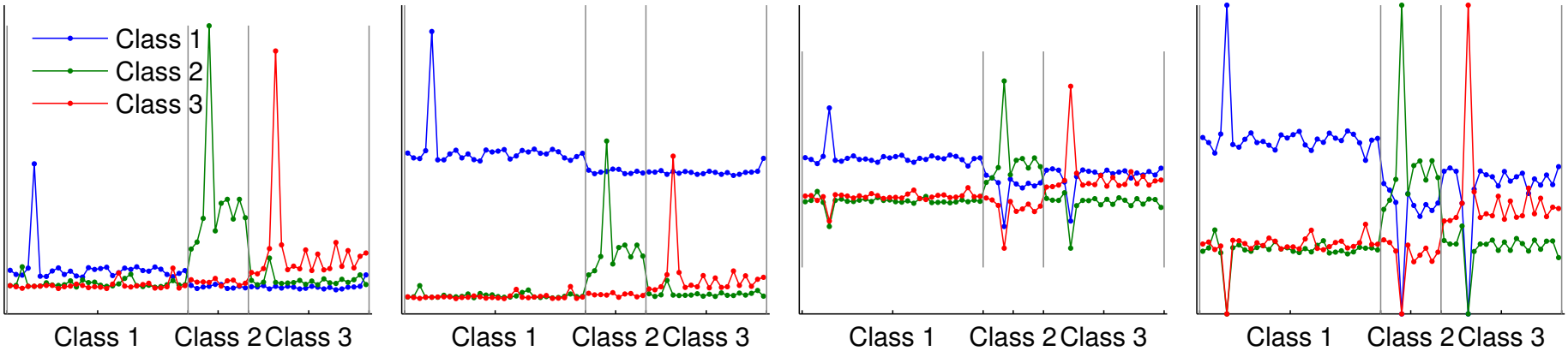




VALUE-BASED
Use the largest value of the diffusion to pick the label.

But value based rounding doesn't work for all diffusions





(b) Zhou et al., $l = 3$ (c) Andersen-Lang, $l = 3$ (d) Joachims, $l = 3$ (e) ZGL, $l = 3$

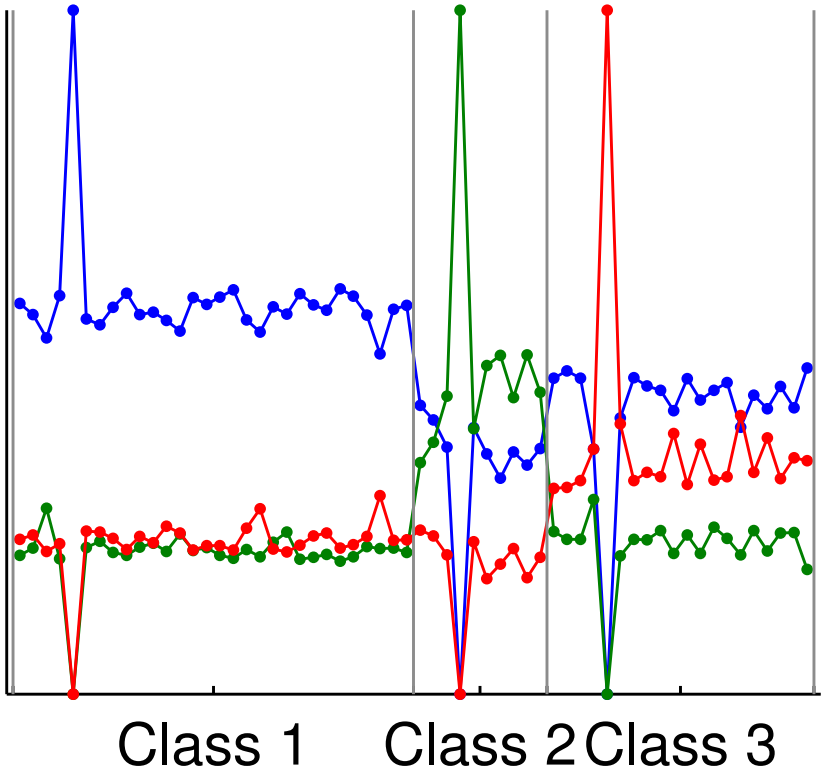
VALUE-BASED rounding fails for most of these diffusions

BUT

There is still a signal there!

Adding more labels doesn't help either, see the paper for those details

Rank-based rounding is far more robust.



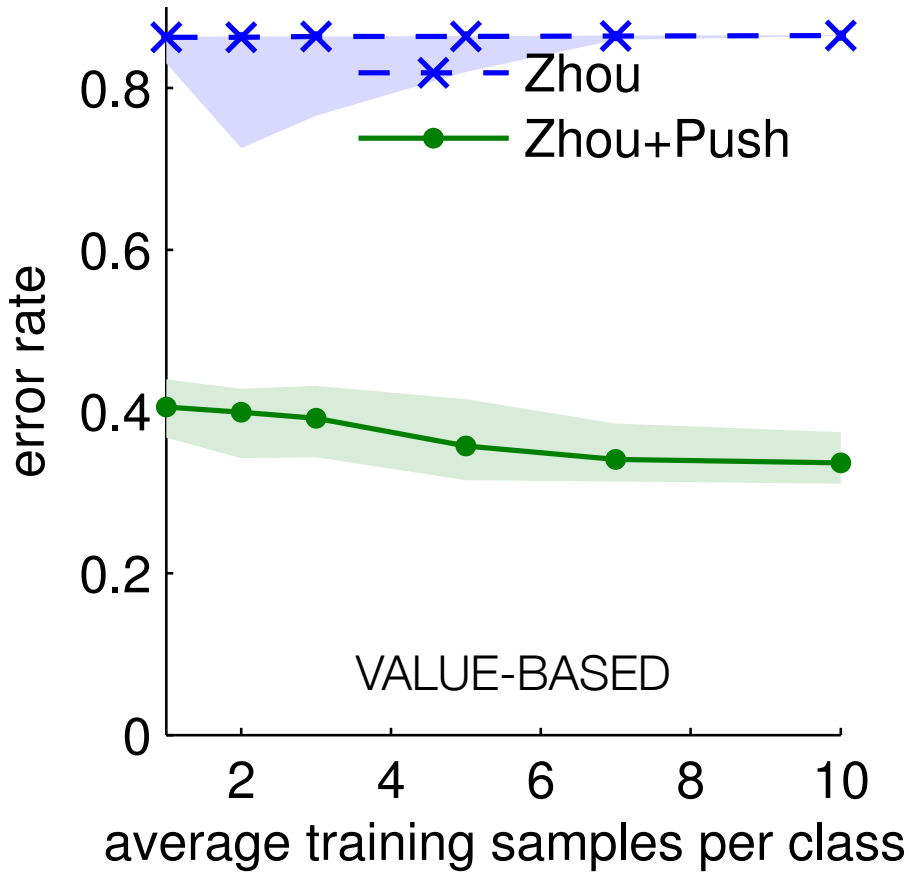
NEW IDEA
Look at the RANK of the item in each diffusion instead of it's VALUE.

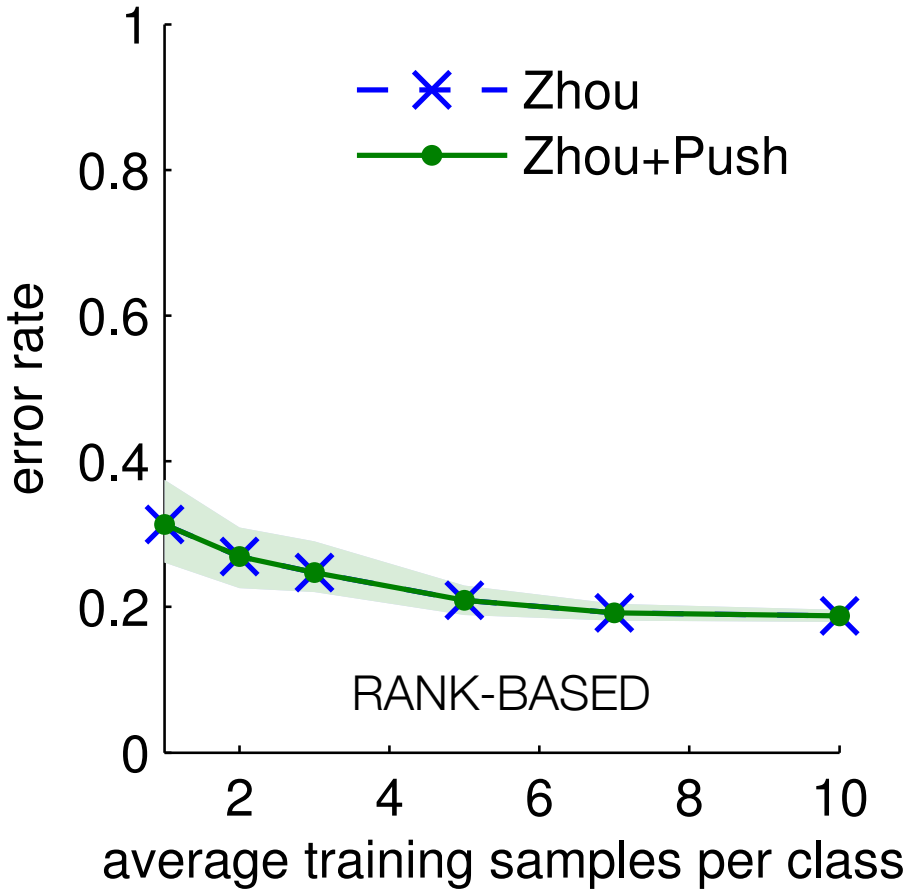
JUSTIFICATION
Based on the idea of sweep-cut rounding in spectral methods (use the order induced by the eigenvector, not its values)

IMPACT
Much more robust rounding to labels

Rank-based rounding has a big impact on a real-study.

We used the digit prediction task out of Zhou's paper and added just a bit of noise as label errors and switched parameters.





Solutions Paths

One benefit of the weak convergence algorithms…

A direct decomposition is a black box:
Feed in input, get output.

In contrast, the iterative nature of “push” means running the algorithm is essentially “watching” the diffusion process occur.

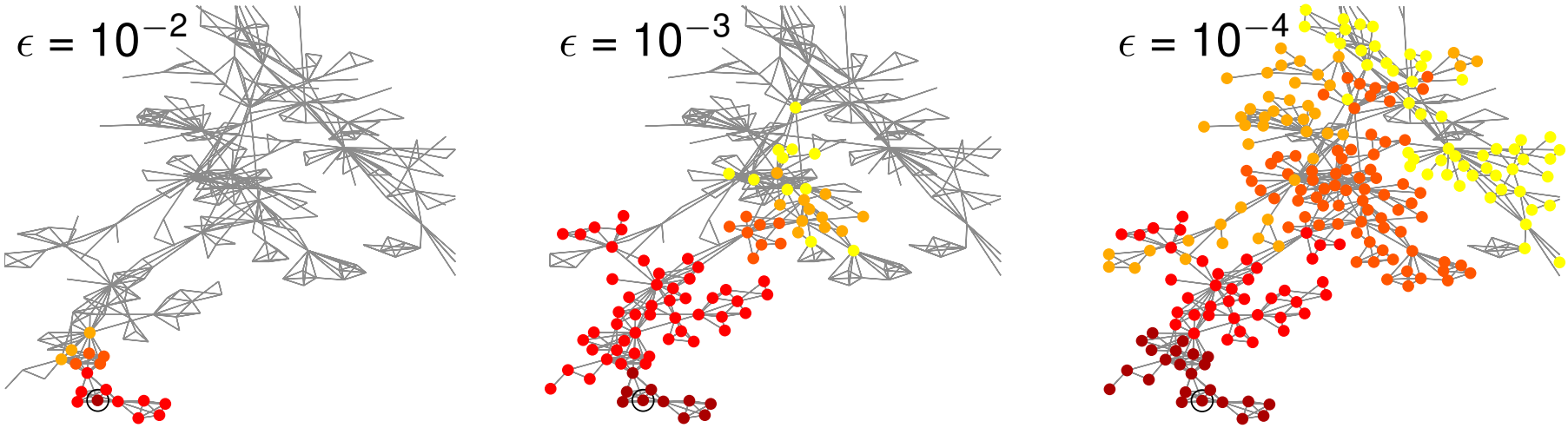
We get more information this way!

Solutions Paths

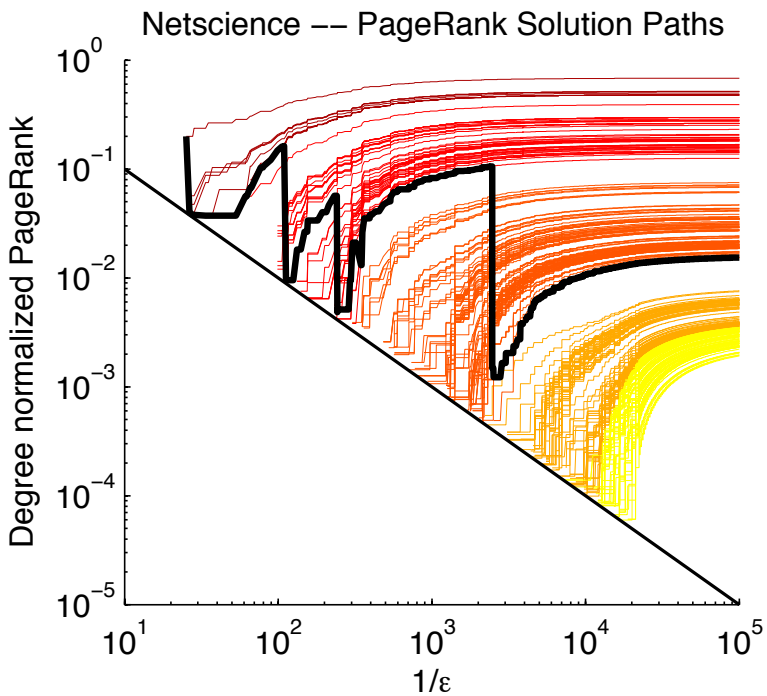
One benefit of the weak convergence algorithms…

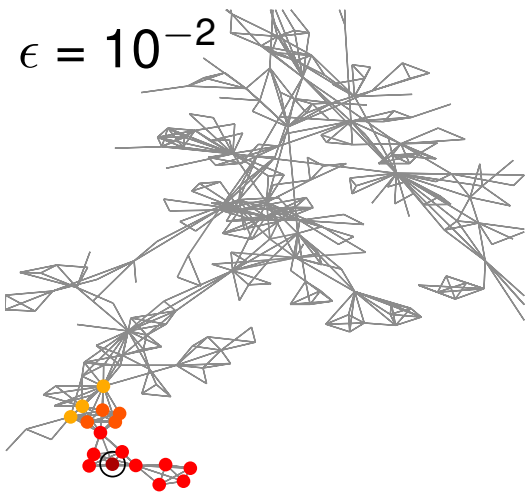
A direct decomposition is a black box:
Feed in input, get output.

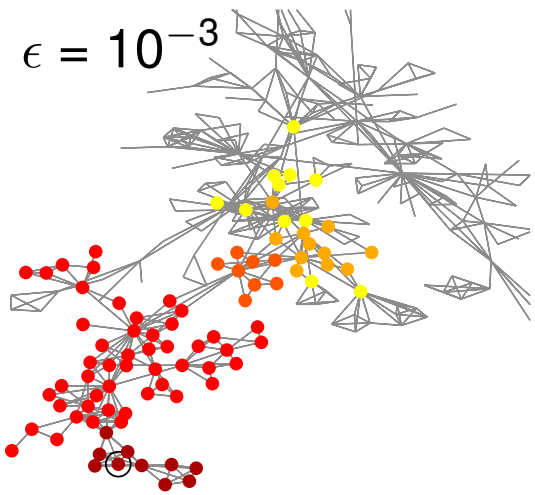
In contrast, the iterative nature of “push” means running the algorithm is essentially “watching” the diffusion process occur.

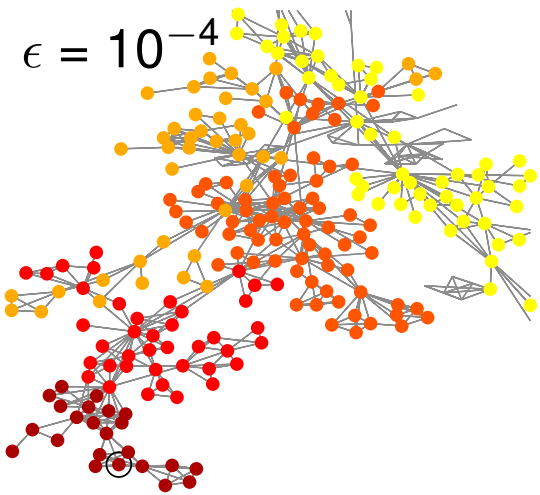


Solutions Paths





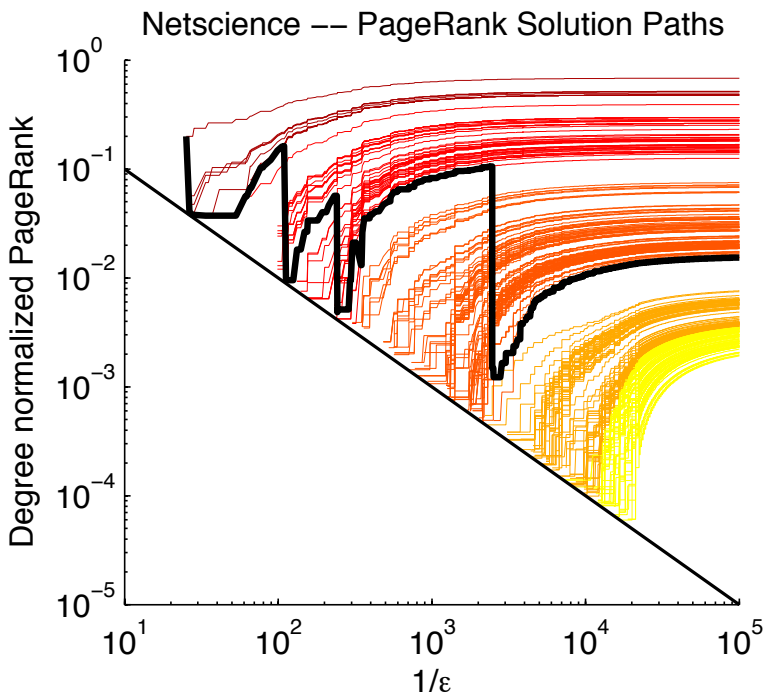


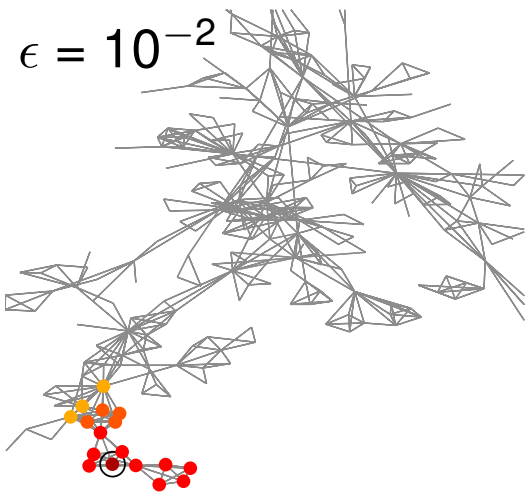


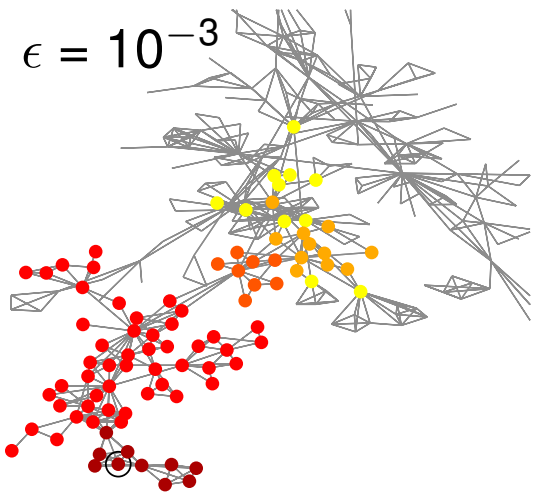
Solutions Paths

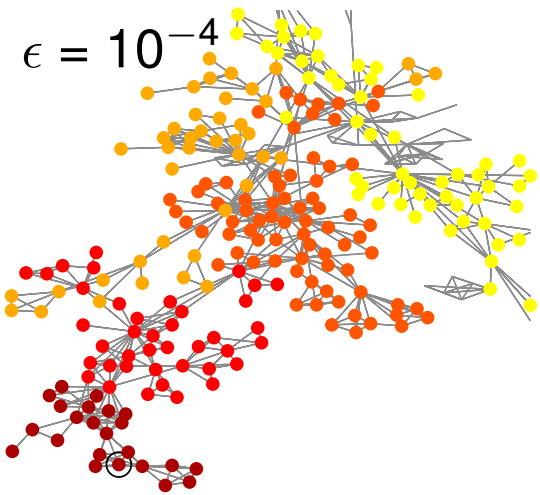
Each curve is a node. Its value increases as ε goes to 0.

Thick black line shows set of best conductance.





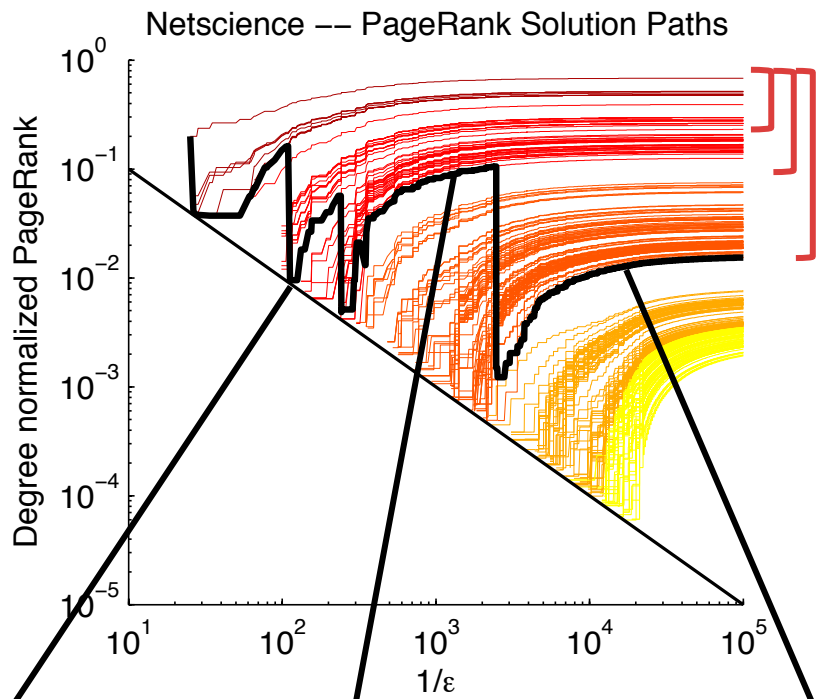




Solutions Paths

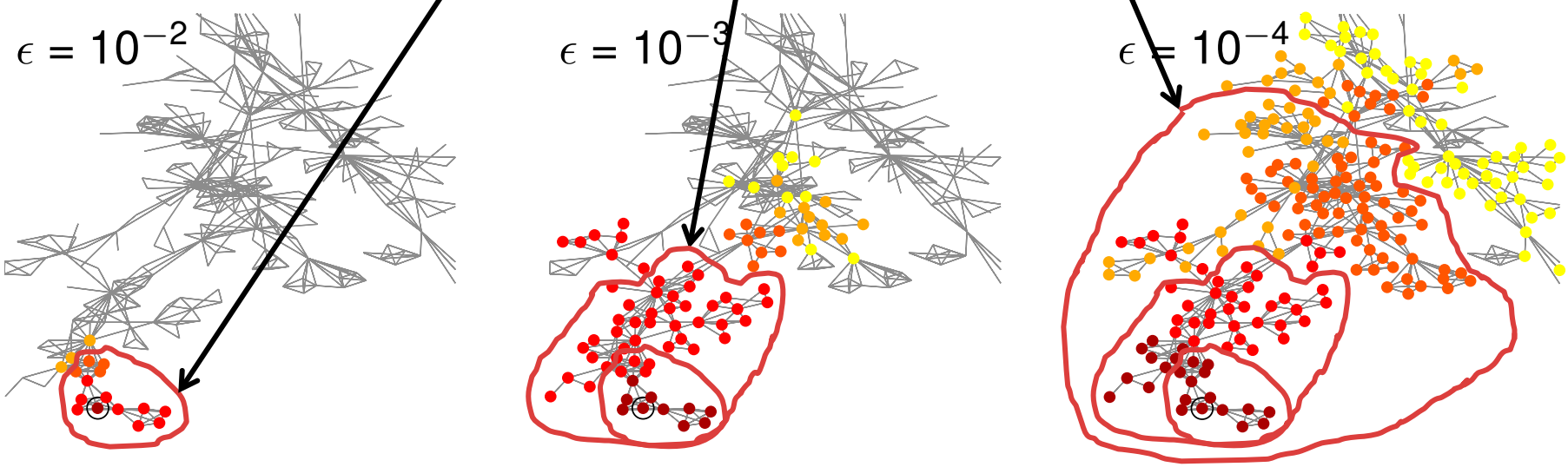
Each curve is a node. Its value increases as ε goes to 0.

Thick black line shows set of best conductance.



Bundles of curves are good clusters

Paths identify *nested* clusters



Solutions Paths

Locate nested, good-conductance sets
that a single diffusion + sweep could miss.

Can be done efficiently because the constant-time approach to computing diffusions enables efficient storage and analysis of the push process

Total Paths work (for PageRank): $O\left(\frac{1}{\epsilon(1-\alpha)}\right)^2$
Still efficient!

Recent directions

1. Diffusions in time-dependent networks

- Grindrod et al. 2011 (Katz); Gleich &Rossi, 2014 (PageRank); Grindrod & Higham, 2014 (Katz again…)

Open issues

1. Parameter selection
2. Tuned diffusions