Gaps between theory practice in

large scale matrix computations for networks

David F. Gleich
Assistant Professor
Computer Science
Purdue University

Networks as matrices

OBSERVATION OF PLAY ACTIVITIES IN A NURSERY SCHOOL

HELEN BOTT

This paper aims to chronicle our experience over two years in the attempt to formulate certain principles of method for observing and analyzing play activity of young children. We are convinced that findings in

Bott, Genetic Psychology Manuscripts, 1928

Networks as matrices

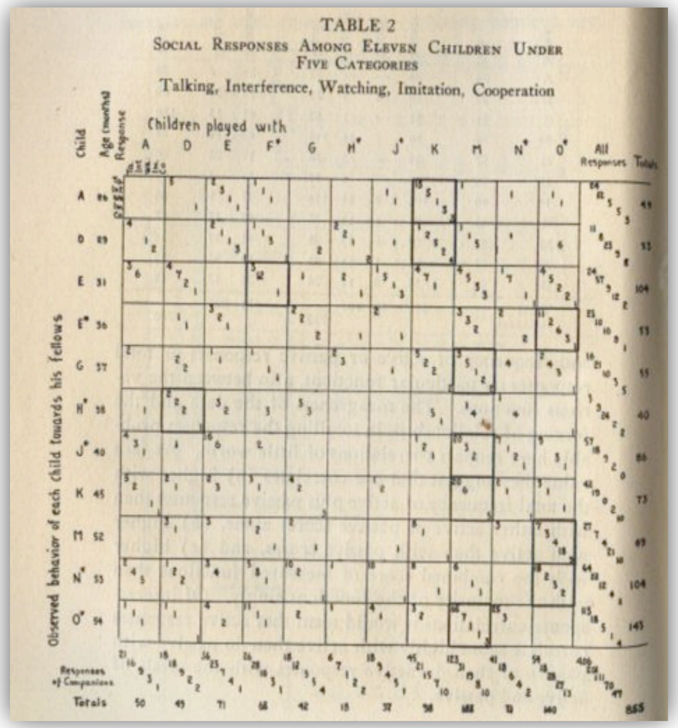


TABLE 2

SOCIAL RESPONSES AMONG ELEVEN CHILDREN UNDER FIVE CATEGORIES

Talking, Interference, Watching, Imitation, Cooperation

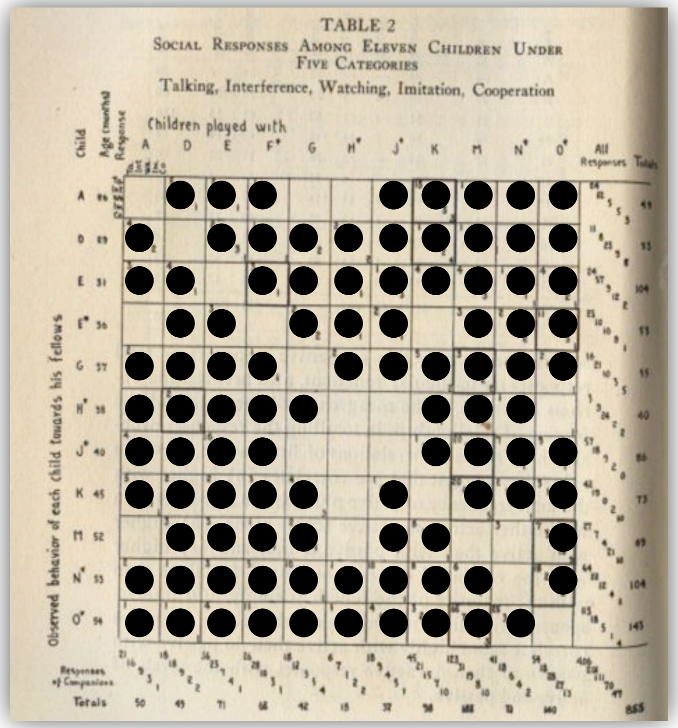
Children played with

Child	A	D	E	F*	G	H*	J*	K	M	N*	O*	All Responses	Totals
A													49
D													53
E													104
F*													53
G													55
H*													40
J*													86
K													73
M													69
N*													104
O*													145
Totals	50	49	71	68	42	15	37	58	188	73	140		865

Observed behavior of each child towards his fellows

Responses of Companions

Networks as matrices



Networks as matrices

$$\boldsymbol{A} = \begin{pmatrix} & \bullet & \bullet & \bullet & & & \bullet & \bullet & \bullet & \bullet & \bullet \\ \bullet & & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet \\ \bullet & \bullet & & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet \\ & \bullet & \bullet & & \bullet & \bullet & \bullet & & \bullet & \bullet & \bullet \\ \bullet & \bullet & \bullet & \bullet & & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet \\ \bullet & \bullet & \bullet & \bullet & \bullet & & & \bullet & \bullet & \bullet & \\ \bullet & \bullet & \bullet & \bullet & & & & \bullet & \bullet & \bullet & \bullet \\ \bullet & \bullet & \bullet & \bullet & \bullet & & \bullet & & \bullet & \bullet & \bullet \\ & \bullet & \bullet & \bullet & \bullet & & \bullet & \bullet & & & \bullet \\ \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & & \bullet \\ \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \end{pmatrix}$$

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

Gene Golub

Charles van Loan

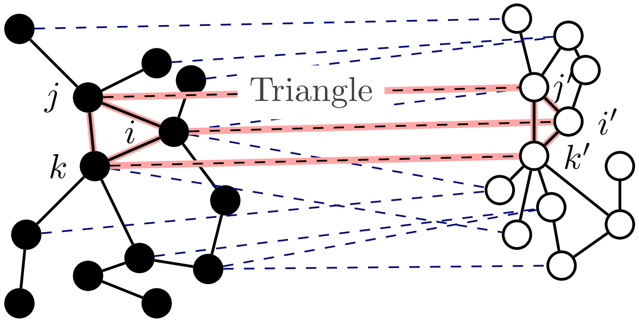
Matrix computations in a red-pill

Solve a problem **better** by exploiting its structure

My research

Models and algorithms for high performance matrix and network computations on data

Network alignment



Big data methods too

Massive matrix computations

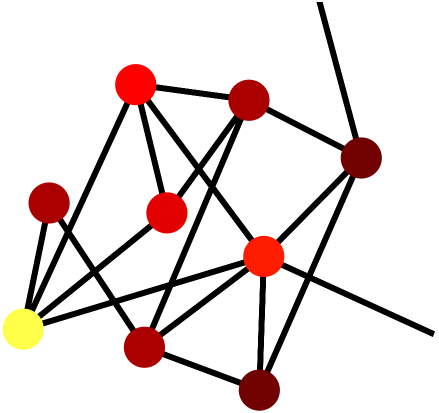
$$\mathbf{Ax} = \mathbf{b}$$

$$\min \|\mathbf{Ax} - \mathbf{b}\|$$

$$\mathbf{Ax} = \lambda \mathbf{x}$$

on multi-threaded and distributed architectures

Fast & Scalable Network analysis



One canonical problem

$$(\boldsymbol{I} - \alpha \boldsymbol{A}^T \boldsymbol{D}^{-1})\mathbf{x} = \mathbf{f}$$

$\boldsymbol{A}$	adjacency matrix	
$\boldsymbol{D}$	degree matrix	
α	*regularization*	
$\mathbf{f}$	"prior" or "givens"	

PageRank

Personalized PageRank

Semi-supervised learning on graphs

Protein function prediction

Gene-experiment association

Network alignment

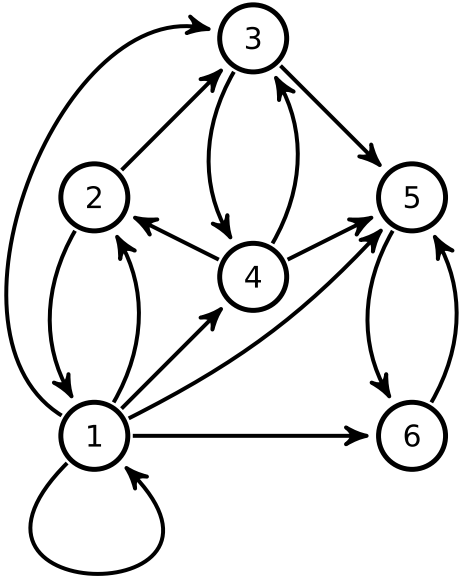
Food webs

One canonical problem

$$(\boldsymbol{I} - \alpha \boldsymbol{A}^T \boldsymbol{D}^{-1})\mathbf{x} = \mathbf{f}$$

Vahab - clustering
Karl - clustering
Art – prediction
Jen - prediction
Sebastiano – ranking/centrality

An example on a graph



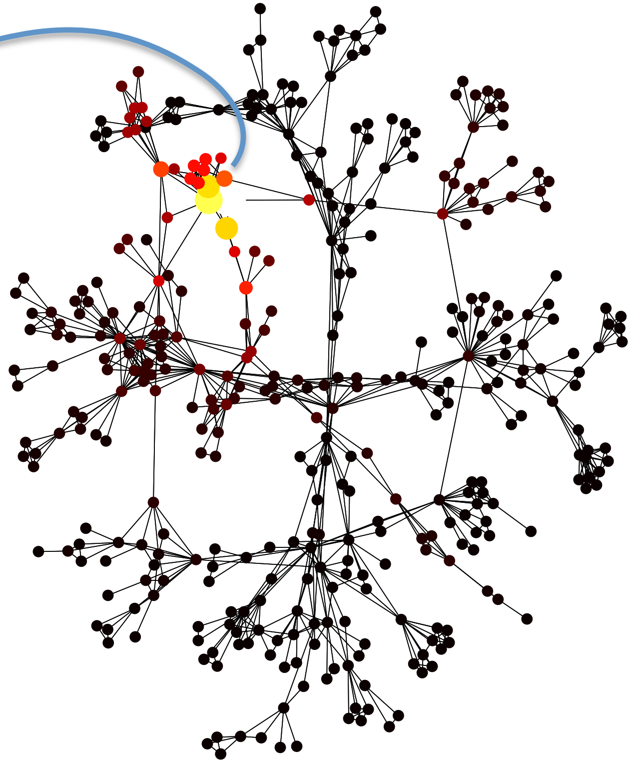
$$(\boldsymbol{I} - \alpha \boldsymbol{A}^T \boldsymbol{D}^{-1})\mathbf{x} = \mathbf{f}$$

$$\left(\boldsymbol{I} - \alpha \overbrace{\begin{bmatrix} 1/6 & 1/2 & 0 & 0 & 0 & 0 \\ 1/6 & 0 & 0 & 1/3 & 0 & 0 \\ 1/6 & 1/2 & 0 & 1/3 & 0 & 0 \\ 1/6 & 0 & 1/2 & 0 & 0 & 0 \\ 1/6 & 0 & 1/2 & 1/3 & 0 & 1 \\ 1/6 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}} \right) \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \end{bmatrix}$$

non-singular linear system ($\alpha < 1$), non-negative inverse, works with weights, directed & undirected graphs, weights that don't sum to less than 1 in each column, …

An example on a bigger graph

f has a single one here



Newman's netscience graph
379 vertices
1828 non-zeros

"zero" on most of the nodes

A special case

“one column” or “one node”

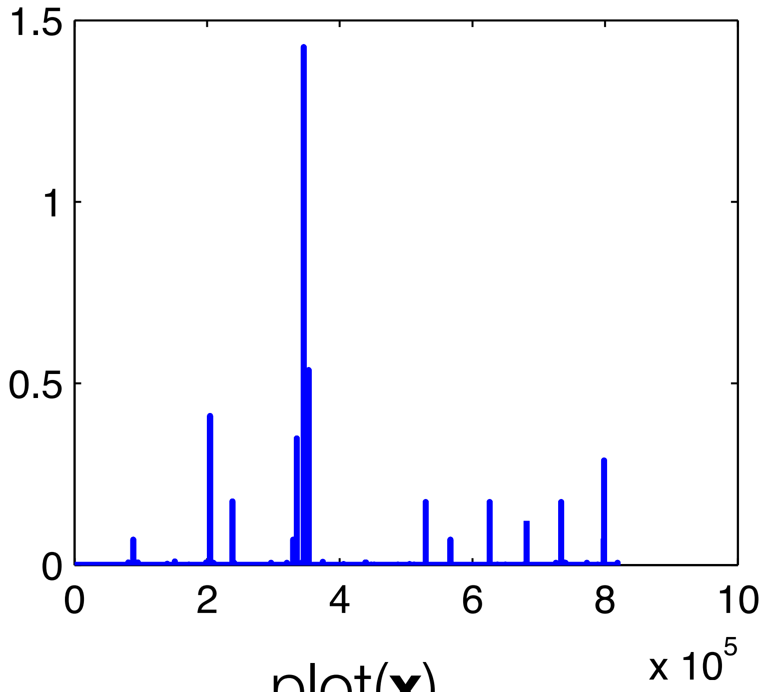
$$(\boldsymbol{I} - \alpha \boldsymbol{A}^T \boldsymbol{D}^{-1})\mathbf{x} = \mathbf{e}_i$$

$$\mathbf{x} = \text{ column } i \text{ of } (\boldsymbol{I} - \alpha \boldsymbol{A}^T \boldsymbol{D}^{-1})^{-1}$$

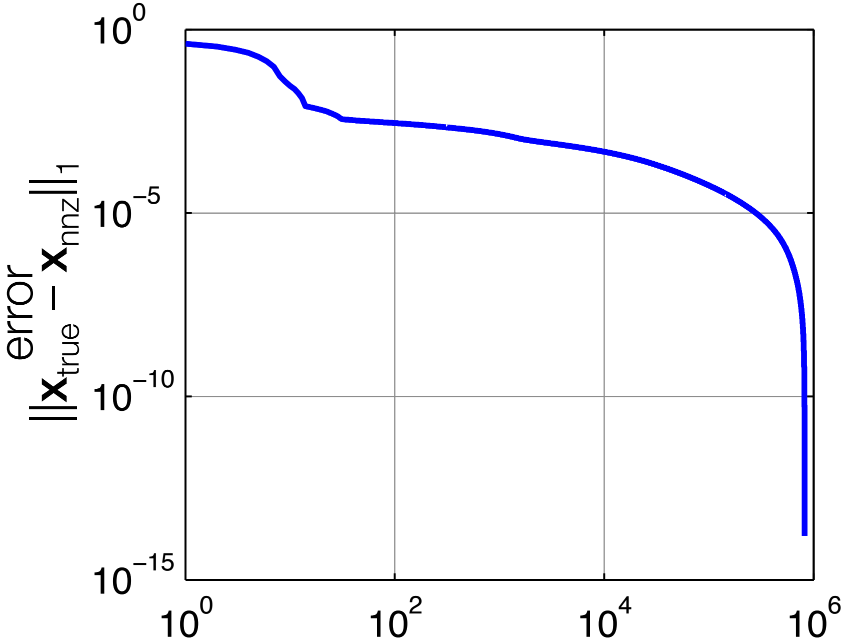
localized solutions

An example on a bigger graph

Crawl of flickr from 2006 ~800k nodes, 6M edges, alpha=1/2



plot(**x**)



Complexity is complex

- Linear system – $O(n^3)$
- Sparse linear sys. (undir.) – $O(m \log(m)^{\gamma})$
  where γ is a function of latest result on solving SDD systems on graphs
- Neumann series – $O(m \log(\alpha)/\log(tol))$

Matrix Inversion by a Monte Carlo Method[1]

Forsythe and Liebler, 1950

The following unusual method of inverting a class of matrices was devised by J. VON NEUMANN and S. M. ULAM. Since it appears not to be in print, an exposition may be of interest to readers of *MTAC*. The method is remarkable in that it can be used to invert a class of n-th order matrices (see final paragraph) with only n^2 arithmetic operations in addition to the scanning and discriminating required to play the solitaire game. The method therefore appears best suited to a human computer with a table of random digits and no calculating machine. Moreover, the method lends itself fairly well to obtaining a single element of the inverse matrix without determining the rest of the matrix. The term "Monte Carlo" refers to mathematical sampling procedures used to approximate a theoretical distribution [see *MTAC*, v. 3, p. 546].

Monte Carlo methods for PageRank

K. Avrachenkov et al. 2005. Monte Carlo methods in PageRank
Fogaras et al. 2005. Fully scaling personalized PageRank.
Das Sarma et al. 2008. Estimating PageRank on graph streams.
Bahmani et al. 2010. Fast and Incremental Personalized PageRank
Bahmani et al. 2011. PageRank & MapReduce
Borgs et al. 2012. Sublinear PageRank

Complexity – “O(log |V|)”

Gauss-Seidel and Gauss-Southwell

Methods to solve $\boldsymbol{A}\,\mathbf{x} = \mathbf{b}$

Update $\quad \mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \rho_j \mathbf{e}_j \quad$ such that $\quad [\boldsymbol{A}\mathbf{x}^{(k+1)}]_j = [\mathbf{b}]_j$

In words “Relax” or “free” the *j*th coordinate of your solution vector in order to satisfy the *j*th equation of your linear system.

Gauss-Seidel repeatedly cycle through j = 1 to n

Gauss-Southwell use the value of j that has the highest magnitude residual

$$\mathbf{r}^{(k)} = \mathbf{b} - \mathbf{A}\mathbf{x}^{(k)}$$

Matrix computations in a red-pill

Solve a problem **better** by exploiting its structure

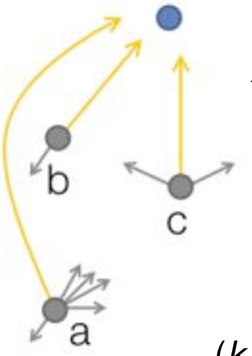
Gauss-Seidel/Southwell for PageRank

w/ access to in-links & degs.

PageRankPull

j = blue node

Solve for $x_j^{(k+1)}$

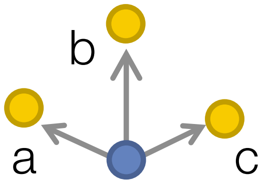


$$x_j^{(k+1)} - \alpha x_a^{(k)}/6 - \alpha x_b^{(k)}/2 - \alpha x_c^{(k)}/3 = f_j$$

$$x_j^{(k+1)} - \alpha \sum_{i \to j} x_i^{(k)}/deg_i = f_j$$

w/ access to out-links

PageRankPush



j = blue node

Let

$$\mathbf{r}^{(k)} = \mathbf{f} + \alpha \mathbf{A}^T \mathbf{D}^{-1} \mathbf{x}^{(k)} - \mathbf{x}^{(k)}$$

then $x_j^{(k+1)} = x_j^{(k)} + r_j$

Update $\mathbf{r}^{(k+1)}$ $\quad r_j^{(k+1)} = 0$

$$r_a^{(k+1)} = r_a^{(k)} + \alpha r_j^{(k)}/3$$
$$r_b^{(k+1)} = r_b^{(k)} + \alpha r_j^{(k)}/3$$
$$r_c^{(k+1)} = r_c^{(k)} + \alpha r_j^{(k)}/3$$

Python code for PPR Push

```
# initialization
# graph is a set of sets
# eps is stopping tol
# 0 < alpha < 1
x = dict()
r = dict()
sumr = 0.
for (node,fi) in f.items():
  r[node] = fi
  sumr += fi
```

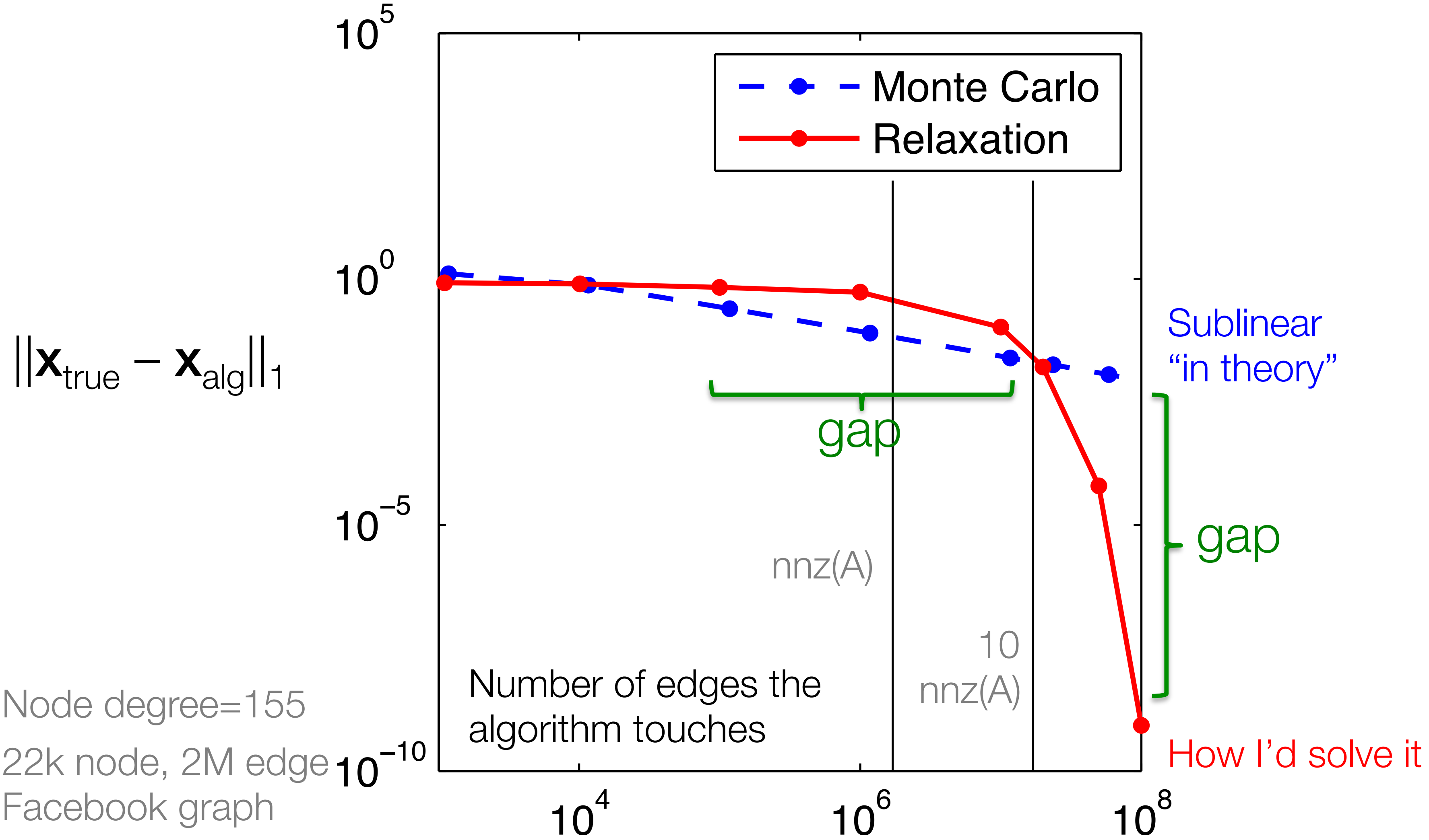
```
# main loop
while sumr > eps/(1-alpha):
  j = max(r.iteritems,
             key=(lambda x: r[x])
  rj = r[j]
  x[j] += rj
  r[j] = 0
  sumr -= rj
  deg = len(graph[j])
  for i in graph[j]:
    if i not in r: r[i] = 0.
    r[j] += alpha/deg*rj
    sumr += alpha/deg*rj
```

If $\mathbf{f} \geq 0$, this terminates when $\|\mathbf{x}_{\text{true}} - \mathbf{x}_{\text{alg}}\|_1 < \varepsilon$

Relaxation methods for PageRank

Arasu et al. 2002, PageRank computation and the structure of the web
Jeh and Widom 2003, Scaling personalized PageRank
McSherry 2005, A unified framework for PageRank acceleration
Andersen et al. 2006, Local PageRank
Berkhin 2007, Bookmark coloring algorithm for PageRank

Complexity – “O(|E|)”



Node degree=155

22k node, 2M edge Facebook graph

Matrix computations in a red-pill

Solve a problem **better** by exploiting its structure

Some unity?

Theorem (Gleich and Kloster, 2013 arXiv:1310.3423)

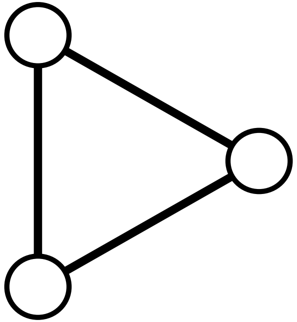
Consider solving personalized PageRank using the Gauss-Southwell relaxation method **in a graph with a Zipf-law in the degrees with exponent p=1 and max-degree d**, then the work involved in getting a solution with 1-norm error ε is

$$\left.\text{work}^* = O\left((1/\varepsilon)^{\frac{1}{1-\alpha}} d(\log d)^2\right)\right\}\text{ Improve this?}$$

* (the paper currently bounds $\exp(\mathbf{A}\,\mathbf{D}^{-1})\,\mathbf{e}_i$ but analysis yields this bound for PageRank)

** (this bound is not very useful, but it justifies that this method isn't horrible in theory)

There is more structure



The one ring.

(See C. Seshadhri's talk for the reference)

Further open directions

Nice to solve Unifying convergence results for Monte Carlo and relaxation on large networks to have provably efficient, practical algs.

– Use triangles? Use preconditioning?

A curiosity Is there any reason to use a Krylov method?

– Staple of matrix computations, $\mathbf{A} \to \mathbf{A}\mathbf{V}_{k+1} = \mathbf{V}_k \mathbf{H}_k$ with $\mathbf{H}_k$ small

BIG gap Can we get algorithms with “top k” or “ordering” convergence?

– See Bahmani et al. 2010; Sarkar et al. 2008 (Proximity Search)

Important? Are the useful, tractable multi-linear problems on a network?

– e.g. triangles for network alignment; e.g. Kannan’s planted clique problem.

Supported by NSF CAREER 1149756-CCF