SAND2010-8758P

FAST KATZ AND COMMUTERS

Quadrature Rules and Sparse Linear Solvers for Link Prediction Heuristics

David F. Gleich
Sandia National Labs

la/opt seminar
October 14th 2010

With Pooya Esfandiar, Chen Grief, Laks V. S. Lakshmanan, and Byung-Won On

MAIN RESULTS –SLIDE ONE

A – adjacency matrix

L – Laplacian matrix

Katz score :

$$K_{i,j} = [(\mathbf{I} - \alpha \mathbf{A})^{-1}]_{i,j}, i \neq j$$

Commute time:

$$C_{i,j} = L^{+}_{i,i} + L^{+}_{j,j} - 2L^{+}_{i,j}$$

MAIN RESULTS – SLIDE TWO

For Katz Compute one $K_{i,j}$ fast

Compute top $K_{i,:}$ fast

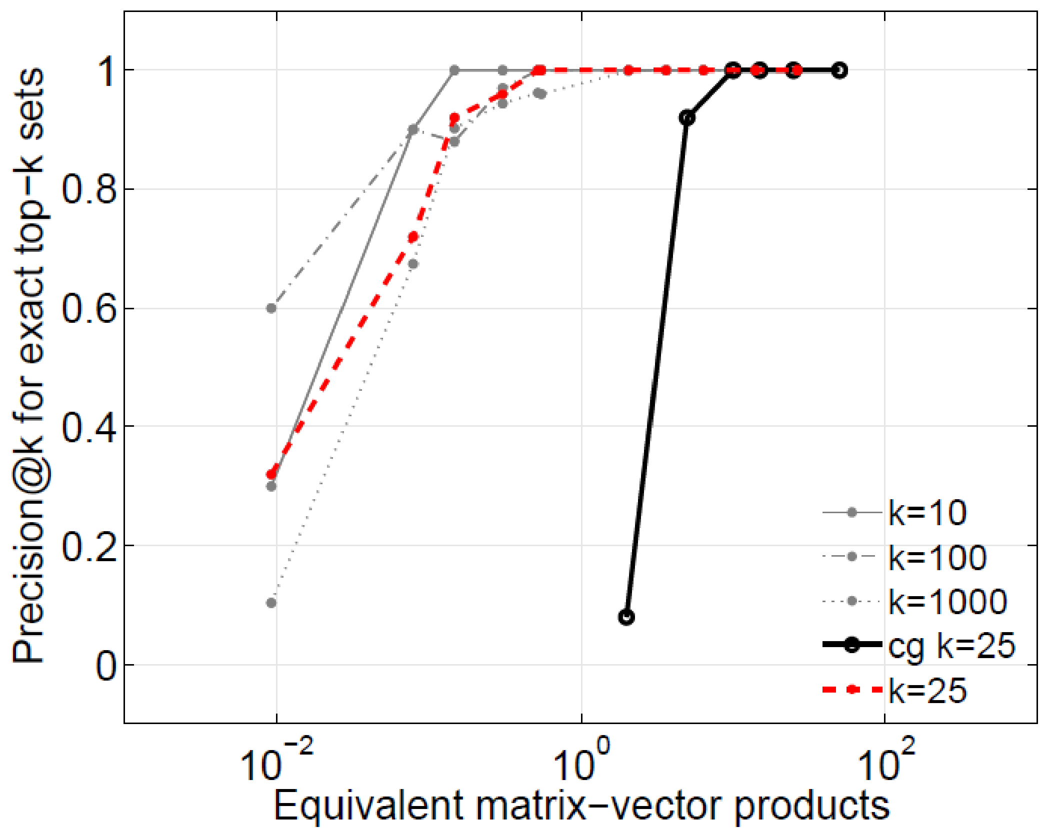
For Commute

Compute one $C_{i,j}$ fast

For almost commute

Compute top $F_{i,:}$ fast

MAIN RESULTS – SLIDE THREE



OUTLINE

Why study these measures?

Katz Rank and Commute Time

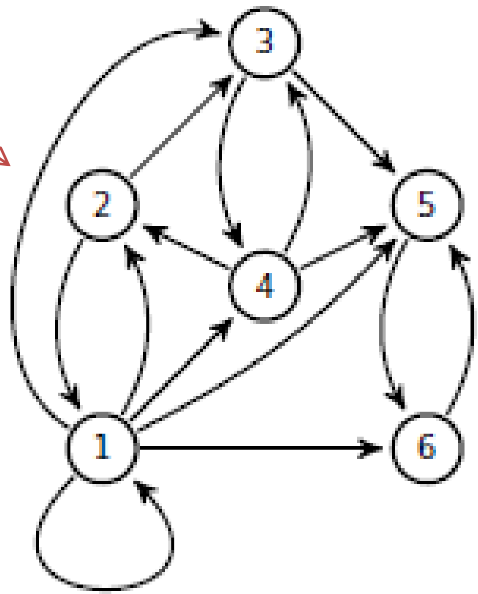
How else do people compute them?

Quadrature rules for pairwise scores

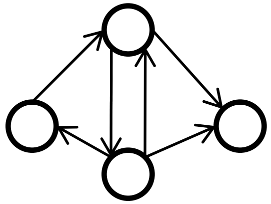
Sparse linear systems solves for top-k

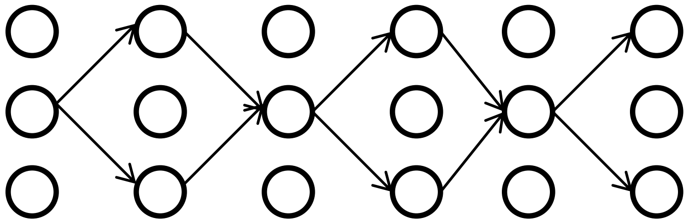
As many results as we have time for…

WHY? LINK PREDICTION



~~Neighborhood based~~





Path based

Liben-Nowell and Kleinberg 2003, 2006 found that path based link prediction was more efficient

NOTE

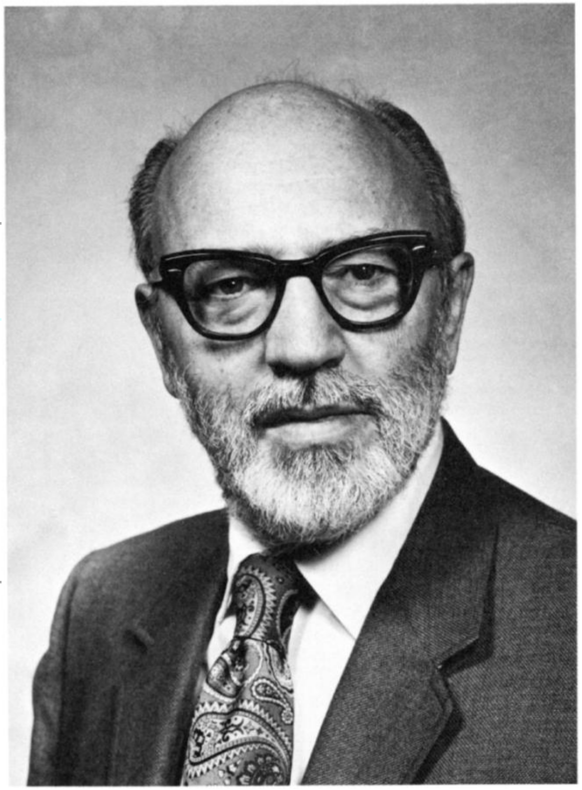
All graphs are undirected

All graphs are connected

Leo Katz (statistician)

From Wikipedia, the free encyclopedia

Leo Katz (born 29 November 1914 in Detroit - died 6 May 1976) was an American statistician. In 1953, he wrote a paper that already outlined the algorithm today known as PageRank[1].



References

1. ^ Katz, Leo. "A new status index derived from soc
Psychometrika, 18 (1953), 39-43

External links

- http://www.jstor.org/pss/3213364 Obituary

NOT QUITE, WIKIPEDIA

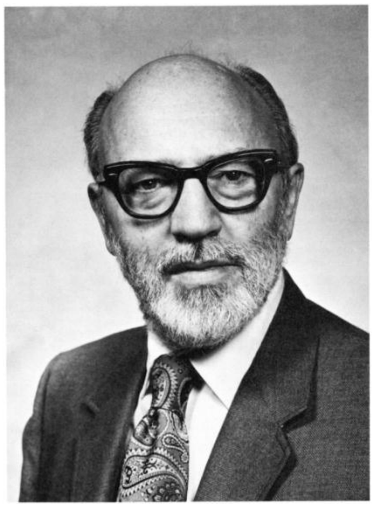
$\mathbf{A}$: adjacency, $\mathbf{P} = \mathbf{D}^{-1}\mathbf{A}$: random walk

PageRank $(\mathbf{I} - \alpha \mathbf{P}^T)^{-1}\mathbf{e}$

Katz $(\mathbf{I} - \alpha \mathbf{A}^T)^{-1}\mathbf{e}$

These are equivalent if G has constant degree

WHAT KATZ ACTUALLY SAID



"we assume that each link independently has the same probability of being effective" …

"we conceive a constant α , depending on the group and the context of the particular investigation, which has the force of a probability of effectiveness of a single link. A k-step chain then, has probability α^k of being effective."

"We wish to find the column sums of the matrix"

$$T = aC + a^2C^2 + \cdots + a^kC^k + \cdots = (I - aC)^{-1} - I.$$

Leo Katz 1953, A New Status Index Derived from Sociometric Analysis, Psychometria 18(1):39-43

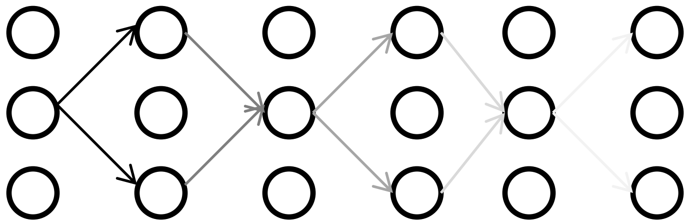
A MODERN TAKE

The Katz score (node-based) is

$$t_i = \sum_j \sum_{k=1}^{\infty} \alpha^k \text{nwalks}_{i,j}^{(k)}$$

The Katz score (edge-based) is

$$K_{i,j} = \sum_{k=1}^{\infty} \alpha^k \text{nwalks}_{i,j}^{(k)}$$



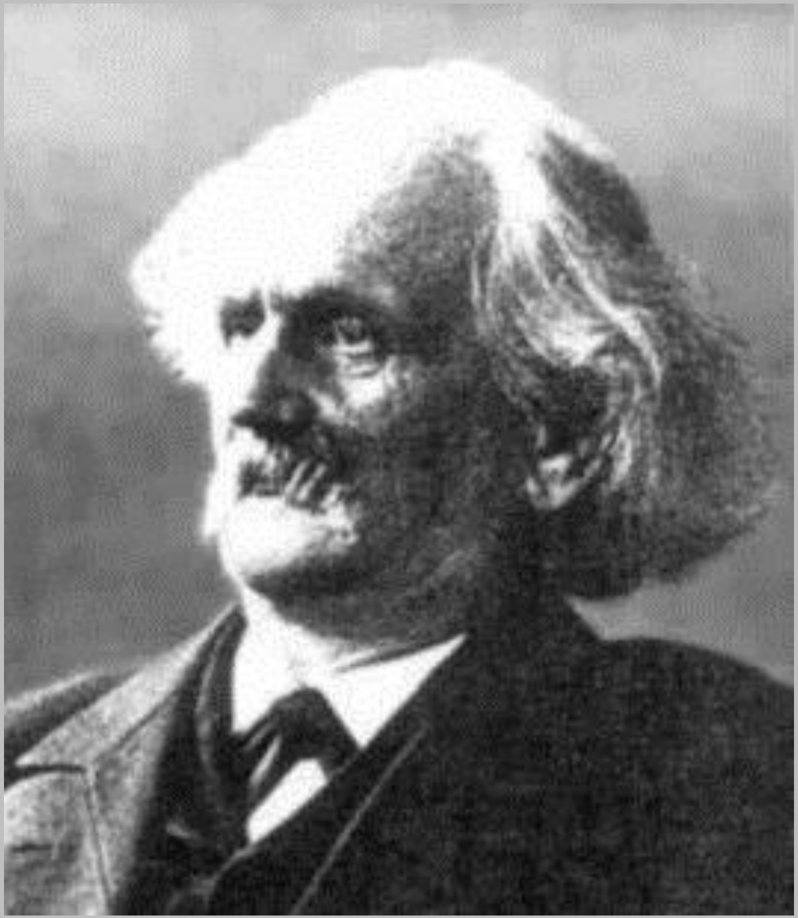
RETURNING TO THE MATRIX

$$nwalks_{i,j}^{(k)} = A_{i,j}^{k}$$

$$\mathbf{K} = \alpha\mathbf{A} + \alpha^2\mathbf{A}^2 + \alpha^3\mathbf{A}^3 + \ldots$$

Carl Neumann $(\mathbf{I}\text{-}\ \mathbf{A})^{-1} = \sum_{k=0}^{\infty} \mathbf{A}^k$

$$\mathbf{K} = (\mathbf{I} - \alpha\mathbf{A})^{-1} - \mathbf{I}$$



Carl Neumann

I've heard the Neumann series called the "von Neumann" series more than I'd like! In fact, the von Neumann kernel of a graph should be named the "Neumann" kernel!

Wikipedia page

PROPERTIES OF KATZ'S MATRIX

$\mathbf{K}$ is symmetric

$\mathbf{K}$ exists when $\alpha \neq 1/\lambda(\mathbf{A})$

$(\mathbf{I} - \alpha\mathbf{A})$ is sym. pos. def. when $\alpha < 1/\|A\|_2$

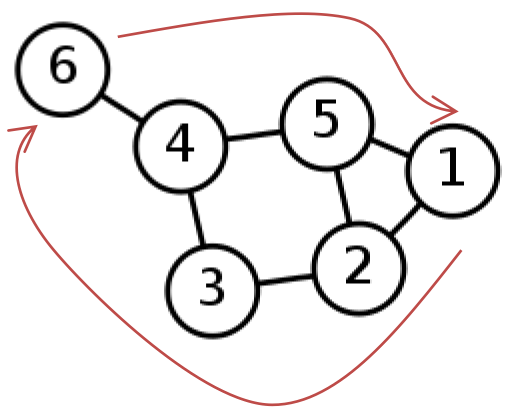
Note that $\alpha <$ 1/max-degree suffices

COMMUTE TIME

Consider a uniform random walk on a graph $\mathbf{P} = \mathbf{D}^{-1}\mathbf{A}$

Also called the hitting time from node i to j, or the first transition time

$$C_{i,j} = E[T \mid X_0 = i, \ldots, X_T = j] + \mathrm{E}[\mathrm{T} \mid X_0 = j, \ldots, X_T = i]$$



SKIPPING DETAILS

$\mathbf{L} = \mathbf{D} - \mathbf{A}$: graph Laplacian

$$C_{i,j} = \mathrm{Vol}(G)(\mathbf{e}_i - \mathbf{e}_j)^T \mathbf{L}^+ (\mathbf{e}_i - \mathbf{e}_j)$$

$\mathbf{L}\mathbf{e} = 0$ is the only null-vector

WHAT DO OTHER PEOPLE DO?

1) Just work with the linear algebra formulations

2) For Katz, Truncate the Neumann series as a few (3-5) terms *(I'm searching for this ref.)*

3) Use low-rank approximations from EVD(A) or EVD(L)

4) For commute, use Johnson-Lindenstrauss inspired random sampling

5) Approximately decompose into smaller problems

Liben-Nowll and Kleinberg CIKM2003, Acar et al. ICDM2009, Spielman and Srivastava STOC2008, Sarkar and Moore UAI2007

THE PROBLEM

All of these techniques are preprocessing based because most people's goal is to compute *all* the scores.

We want to avoid preprocessing the graph.

There are a few caveats here! i.e. one could solve the system instead of looking for the matrix inverse

WHY NO PREPROCESSING?

The graph is constantly changing
as I rate new movies.

WHY NO PREPROCESSING?

Top-k predicted "links" are movies to watch!

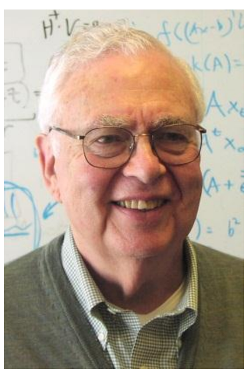
NETFLIX

Pairwise scores give user similarity

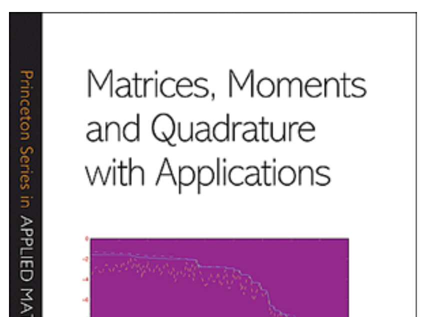
PAIRWISE ALGORITHMS

Katz $\mathbf{e}_i^T(\mathbf{I} - \alpha\mathbf{A})^{-1}\mathbf{e}_j$

Commute $(\mathbf{e}_i - \mathbf{e}_j)^T\mathbf{L}^+(\mathbf{e}_i - \mathbf{e}_j)$



Golub and Meurant to the rescue!



MMQ - THE BIG IDEA

Quadratic form $\mathbf{v}^T f(\mathbf{E})\mathbf{v}$ — Think f(x) = x^{-1}

$\Downarrow$

Weighted sum $\sum_{i=1}^{n} w_i^2 f(\lambda_i)$ — A is s.p.d. use EVD

$\Downarrow$

Stieltjes integral $\int_a^b f(\lambda) dw(\lambda)$ — "A tautology"

$\Downarrow$

Quadrature approximation $\sum_{j=1}^{k} f(\eta_j)\omega_j$

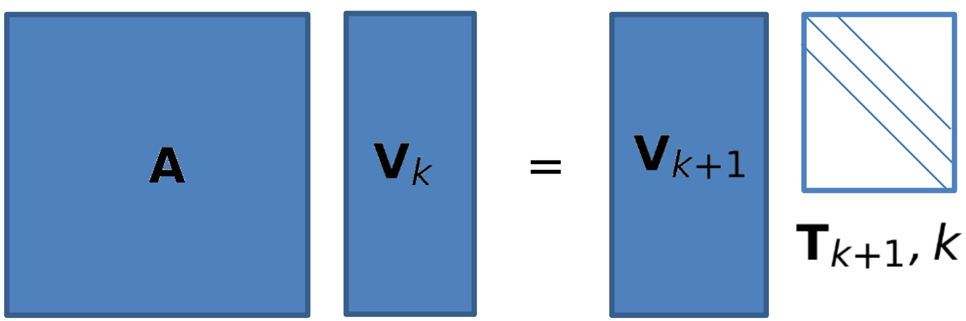
$\Downarrow$

Matrix equation $\mathbf{e}_1^T f(\mathbf{T}_k)\mathbf{e}_1$ — Lanczos

LANCZOS

A, **b** , k-steps of the Lanczos method produce

$$\mathbf{A}\mathbf{V}_k = \mathbf{V}_{k+1}\mathbf{T}_{k+1,k} \quad \text{and} \quad \mathbf{V}_k\mathbf{e}_1 = \rho\mathbf{b}$$



PRACTICAL LANCZOS

Only need to store the last 2 vectors in $\mathbf{V}_k$

Updating requires O(matvec) work

$\mathbf{V}_k$ is not orthogonal

MMQ PROCEDURE

Goal b = $\mathbf{u}^T \mathbf{E}^{-1} \mathbf{u}$

Given l < $\lambda(\mathbf{E}) < u$

1. Run k-steps of Lanczos on $\mathbf{E}$ starting with $\mathbf{u}$
2. Compute $\mathbf{T}_u$, $\mathbf{T}$ with an additional eigenvalue at u , set $b_u = \mathbf{e}_1^T \mathbf{T}_u^{-1} \mathbf{e}_1$

   Correspond to a Gauss-Radau rule, with u as a prescribed node
3. Compute $\mathbf{T}_l$, $\mathbf{T}$ with an additional eigenvalue at l , set $b_l = \mathbf{e}_1^T \mathbf{T}_l^{-1} \mathbf{e}_1$

   Correspond to a Gauss-Radau rule, with l as a prescribed node
4. Output $[b_l, b_u]$ as lower and upper bounds on b

PRACTICAL MMQ

Increase k to become more accurate

Bad eigenvalue bounds yield worse results

$\mathbf{T}_u$ and $\mathbf{T}_l$ are easy to compute

$\mathbf{T}_u^{-1}$ not required, we can iteratively update it's LU factorization

PRACTICAL MMQ

Algorithm 1 Computing Score Bounds

Input: $E, u, a < \lambda_{\min(E)}, b > \lambda_{\max(E)}, k$

Output: $\underline{b_k} \le u^T E^{-1} u \le \overline{b_k}$

Initial step: $h_{1} = 0$, $h_0 = u$, $\omega_1 = u^T E u$, $\gamma_1 = \|(E - \omega_1 I)u\|$, $b_1 = \omega_1^{-1}$, $d_1 = \omega_1$, $c_1 = 1$, $\overline{d_1} = \omega_1 - a$, $\underline{d_1} = \omega_1 - b$, $h_1 = \frac{(E-\omega_1 I)u}{\gamma_1}$.

for $j = 2, \ldots k$ **do**

$\quad \omega_j = h_{j-1}^T E h_{j-1}$

$\quad \widetilde{h_j} = (E - \omega_j I)h_{j-1} - \gamma_{j-1} h_{j-2}$

$\quad \gamma_j = \|\widetilde{h_j}\|$

$\quad h_j = \frac{\widetilde{h_j}}{\gamma_j}$

$\quad b_j = b_{j-1} + \frac{\gamma_{j-1}^2 c_{j-1}^2}{d_{j-1}(\omega_j d_{j-1} - \gamma_{j-1}^2)}$

$\quad d_j = \omega_j - \frac{\gamma_{j-1}^2}{d_{j-1}}$; $c_j = c_{j-1} \frac{\gamma_{j-1}}{d_{j-1}}$

$\quad \overline{d_j} = \omega_j - a - \frac{\gamma_{j-1}^2}{\overline{d_{j-1}}}$; $\underline{d_j} = \omega_j - b - \frac{\gamma_{j-1}^2}{\underline{d_{j-1}}}$

$\quad \overline{\omega_j} = a + \frac{\gamma_j^2}{\overline{d_j}}$; $\underline{\omega_j} = b + \frac{\gamma_j^2}{\underline{d_j}}$

$\quad \overline{b_j} = b_j + \frac{\gamma_j^2 c_j^2}{d_j(\overline{\omega_j} d_j - \gamma_j^2)}$; $\overline{b_j} = b_j + \frac{\gamma_j^2 c_j^2}{d_j(\underline{\omega_j} d_j - \gamma_j^2)}$

ONE LAST STEP FOR KATZ

Katz $\mathbf{e}_i^T(\mathbf{I} - \alpha\mathbf{A})^{-1}\mathbf{e}_j$

$\mathbf{u} \neq \mathbf{v}$ ☹

$$\mathbf{u}^T f(\mathbf{E})\mathbf{v} = 1/4[(\mathbf{u}+\mathbf{v})^T f(\mathbf{E})(\mathbf{u}+\mathbf{v})$$
$$- (\mathbf{u}\text{-}\mathbf{v})^T f(\mathbf{E})(\mathbf{u}-\mathbf{v})]$$

TOP-K ALGORITHM FOR KATZ

Approximate $\mathbf{x}$

$$(\mathbf{I} - \alpha \mathbf{A})\mathbf{x} = \mathbf{e}_i$$

where $\mathbf{A}$ is sparse

Keep $\mathbf{x}$ sparse too

Ideally, don't "*touch*" all of $\mathbf{A}$

INSPIRATION - PAGERANK

Approximate $\mathbf{x}$

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

where $\mathbf{P}$ is sparse

Keep $\mathbf{x}$ sparse too? YES!

Ideally, don't "*touch*" all of $\mathbf{A}$? YES!

McSherry WWW2005, Berkhin 2007, Anderson et al. FOCS2008 – Thanks to Reid Anderson for telling me McSherry did this too.

THE ALGORITHM - MCSHERRY

For $\mathbf{Ex} = \mathbf{b}$

Start with the Richardson iteration

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \omega(\mathbf{b} - \mathbf{Ex}^{(k)})$$

Rewrite

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \omega\mathbf{r}^{(k)}$$

Richardson converges if $\rho(\mathbf{I} - \mathbf{A}) < 1$

THE ALGORITHM

Note $\mathbf{b}$ is sparse.

If $\mathbf{x}^{(0)} = 0$, then $\mathbf{r}^{(0)}$ is sparse.

Idea

only add one component of $\mathbf{r}^{(k)}$ to $\mathbf{x}^{(k)}$

THE ALGORITHM

For $\mathbf{E}\mathbf{x} = \mathbf{b}$

Init: $vx^{(0)} = 0, \mathbf{r}^{(0)} = \mathbf{b}$

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \omega r_j^{(k)} \mathbf{e}_j$$

$$\mathbf{r}^{(k+1)} = \mathbf{r}^{(k)} + \omega r_j^{(k)} \mathbf{E}\mathbf{e}_j$$

How to pick j ?

THE ALGORITHM FOR KATZ

For $(\mathbf{I} - \alpha\mathbf{A})\mathbf{x} = \mathbf{e}_i$

Init: $\mathbf{x}^{(0)} = 0, \mathbf{r}^{(0)} = \mathbf{e}_1$

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + r_j^{(k)}\mathbf{e}_j$$

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

Pick j as max $r_j^{(k)}$

Storing the non-zeros of the residual in a heap makes picking the max log(n) time. See Anderson et al. FOCS2008 for more

CONVERGENCE?

If you pick j as the maximum element, we can show this is convergent if Richardson converges. This proof requires **E** to be symmetric positive definite.

RESULTS - DATA

Graph	Nodes	Edges
dblp	93,156	178,145
arxiv	86,376	517,563
flickr	513,969	3,190,452

All unweighted, connected graphs

RESULTS – KATZ ALPHAS

Easy α

$$1/(10\|\mathbf{A}\|_1 + 10)$$

Hard α

$$1/(\|\mathbf{A}\|_2 + 1)$$

PAIRWISE RESULTS

Katz upper and lower bounds

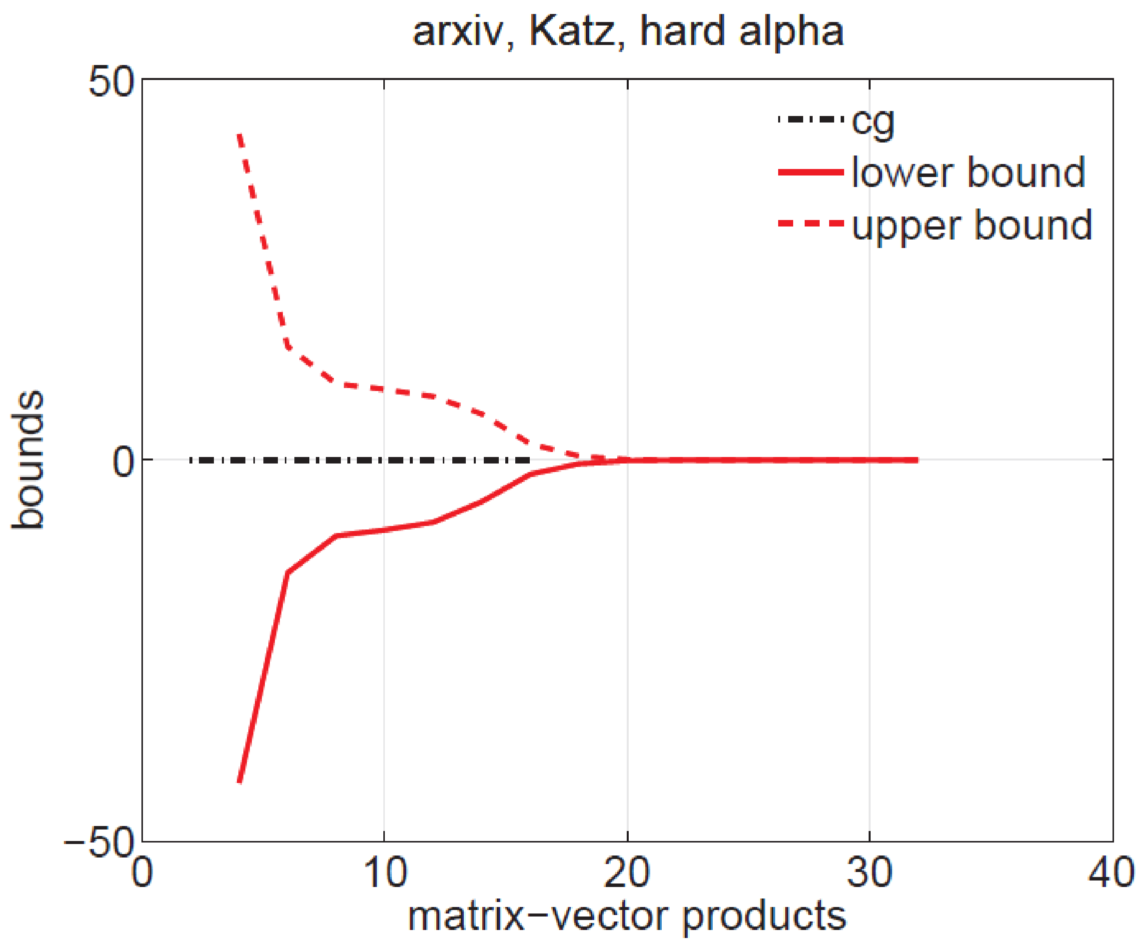
Katz error convergence

Commute-time upper and lower bounds

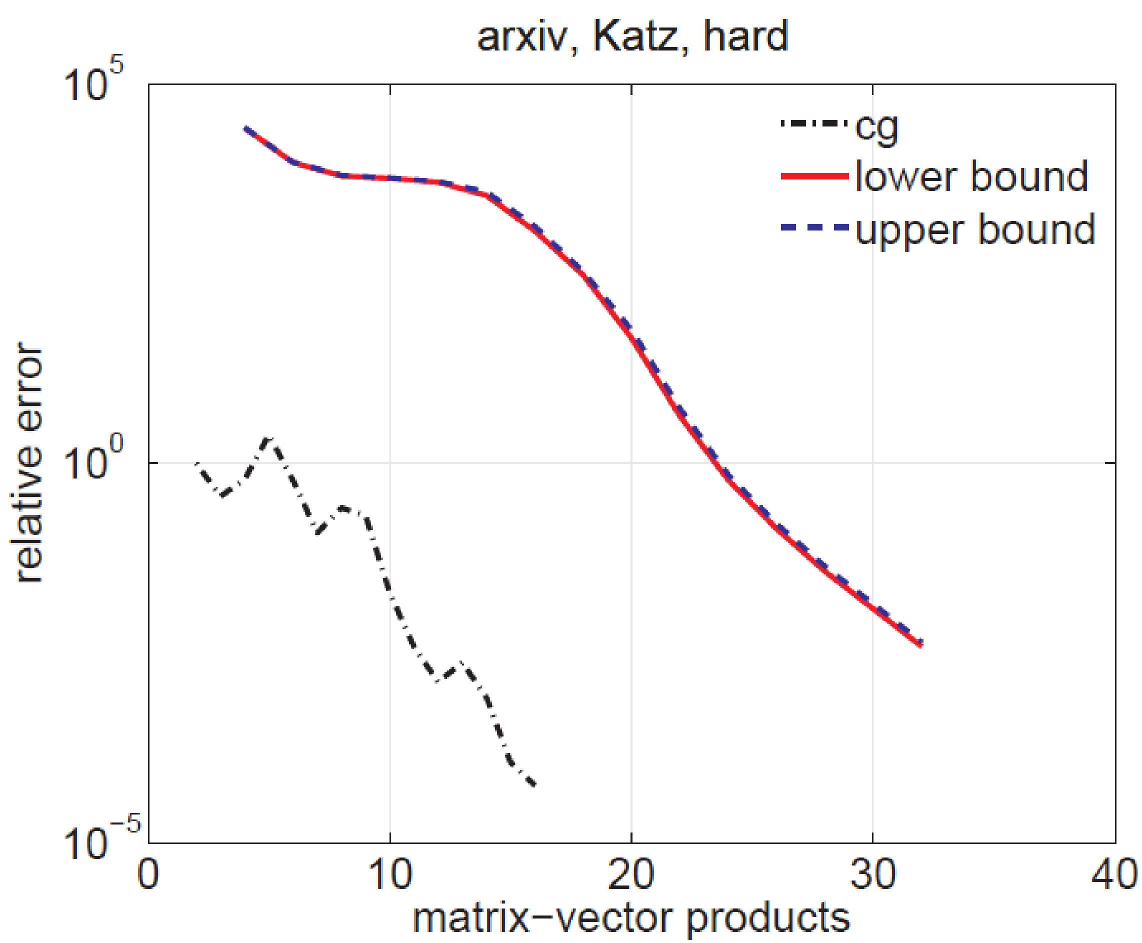
Commute-time error convergence

For the arXiv graph here

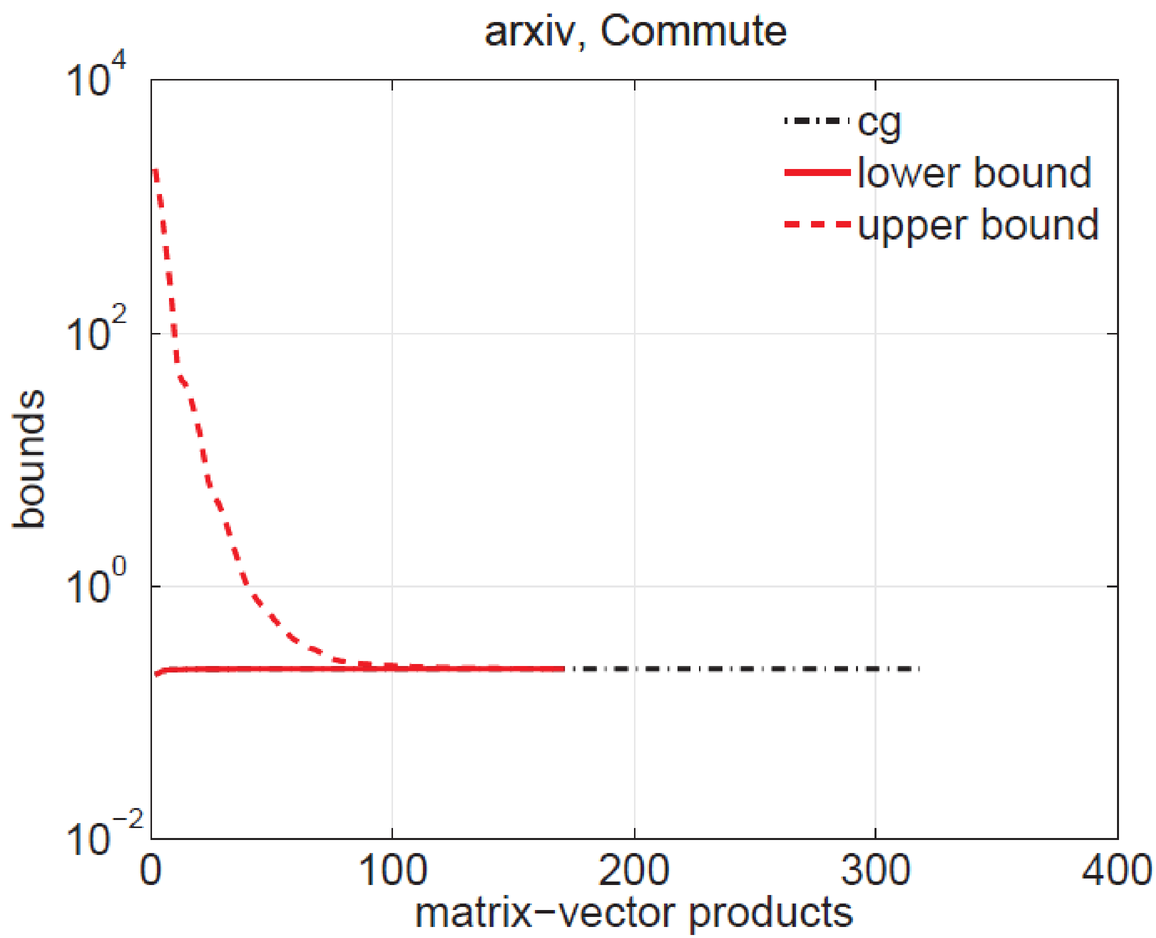
KATZ BOUND CONVERGENCE



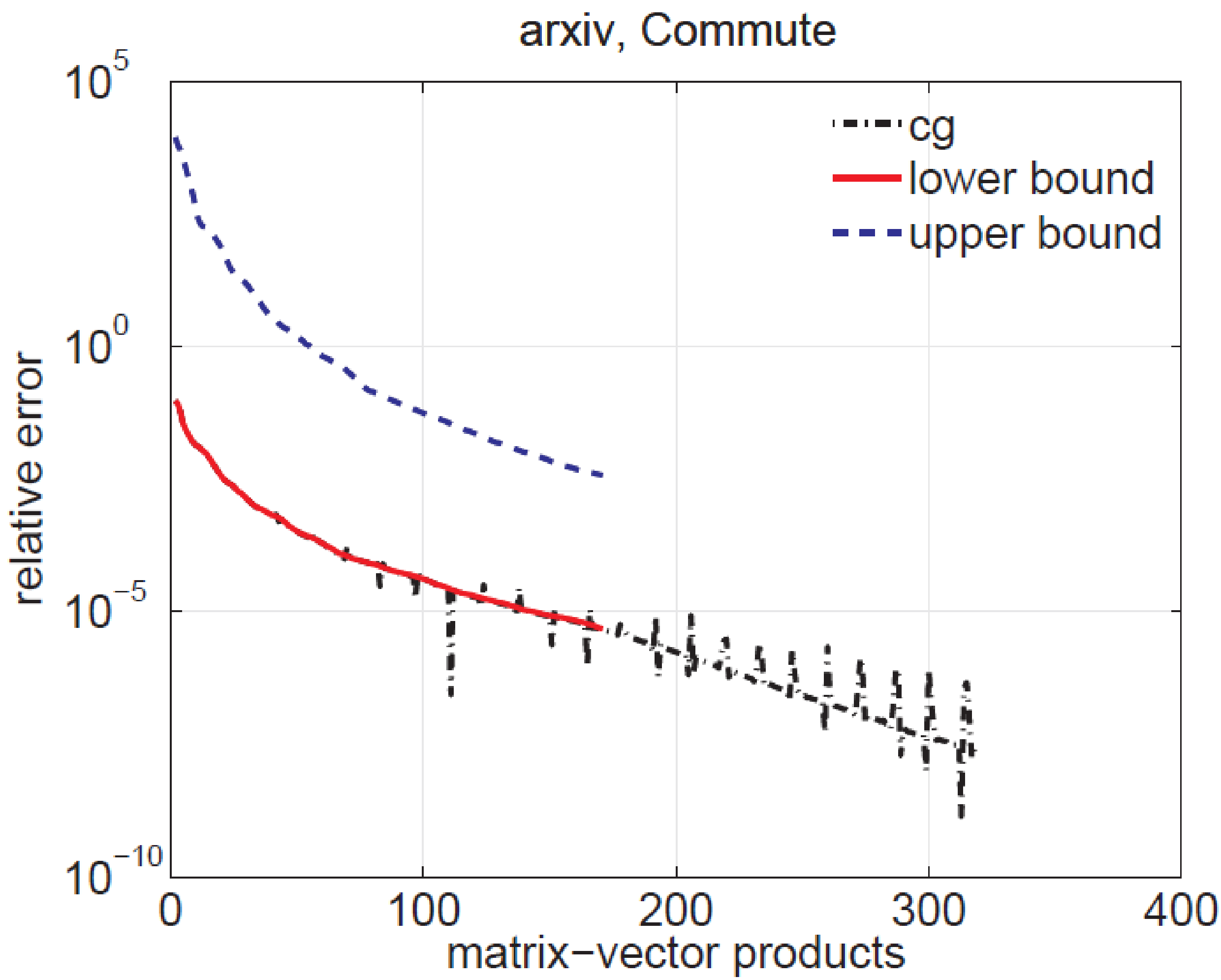
KATZ ERROR CONVERGENCE



COMMUTE BOUND CONVERG.



COMMUTE ERROR CONVERG.



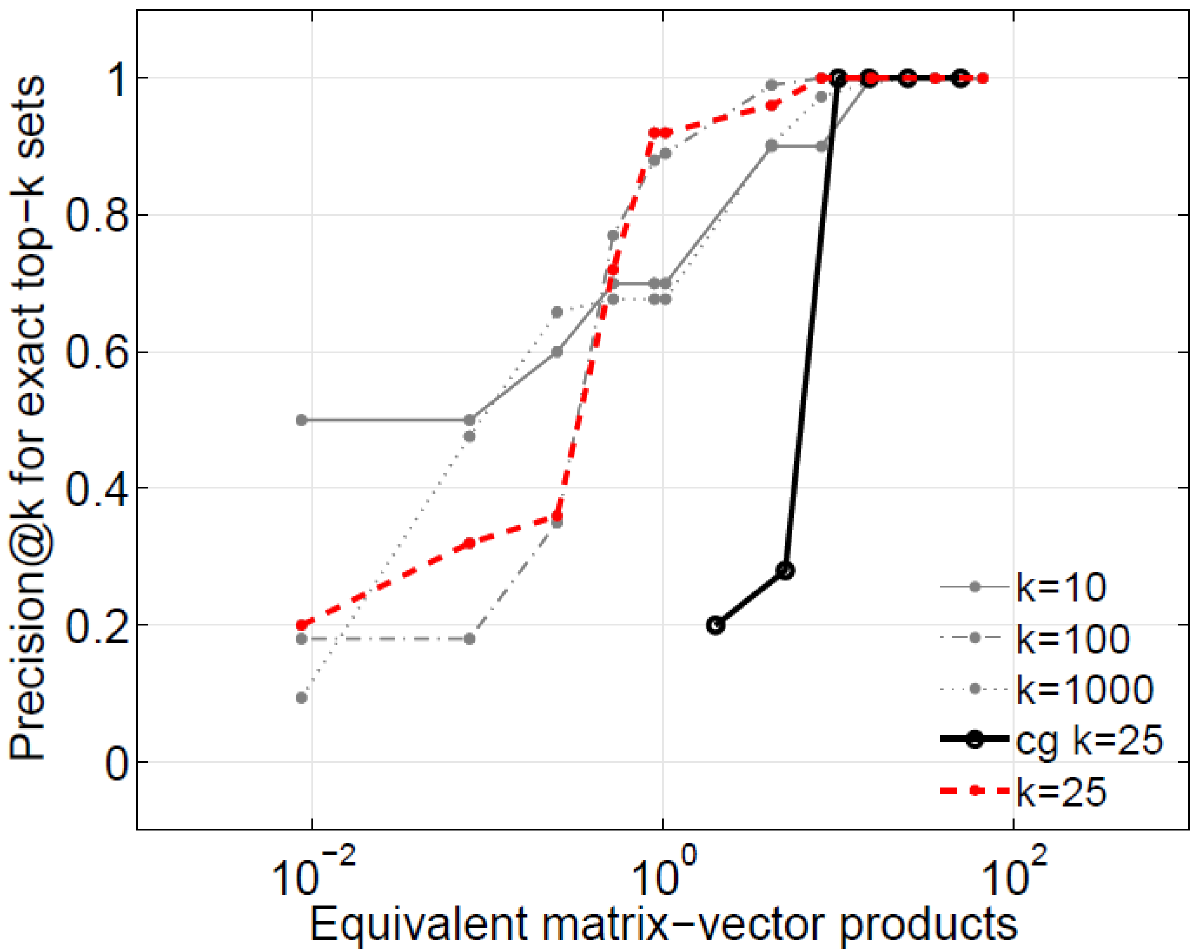
TOP-K RESULTS

Katz set convergence

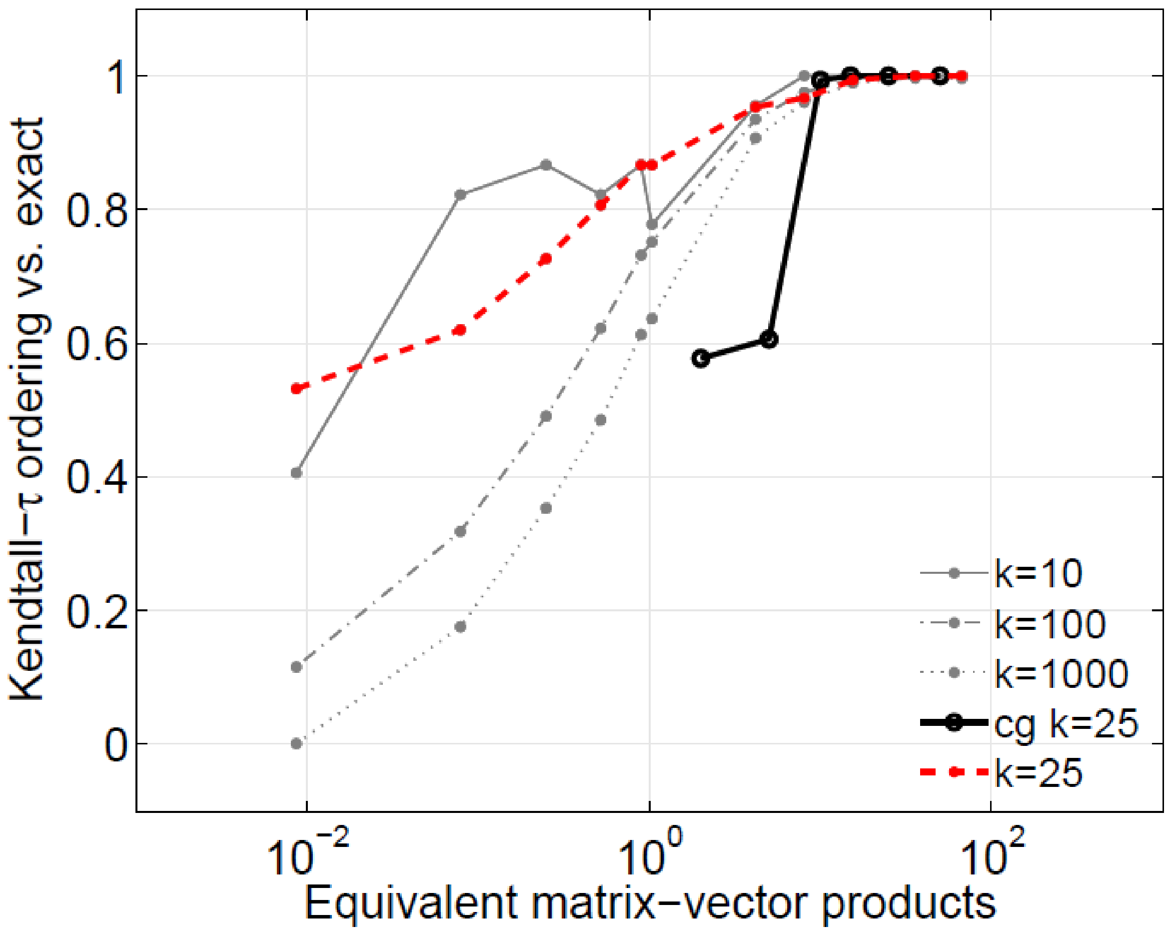
Katz order convergence

For arXiv graph

KATZ SET CONVERGENCE



KATZ ORDER CONVERGENCE



CONCLUSIONS

These algorithms are faster than many alternatives.

For pairwise commute, stopping criteria are simpler

For top-k, we often need less than 1 matvec for good enough results

WARTS

Stopping criteria on our top-k algorithm can be a bit hairy

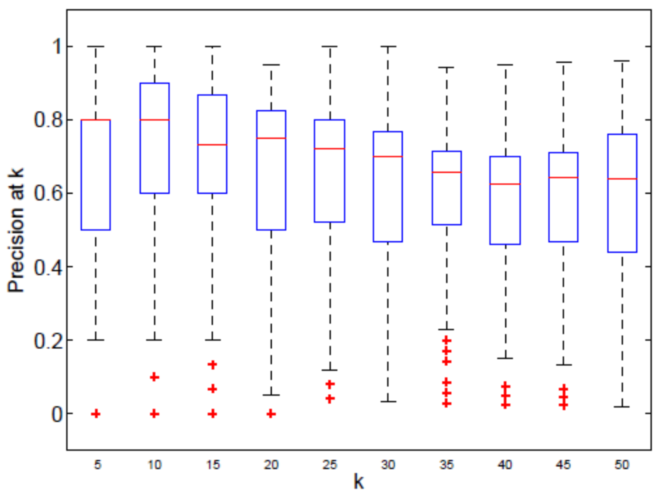
The top-k approach doesn’t work right for commute time

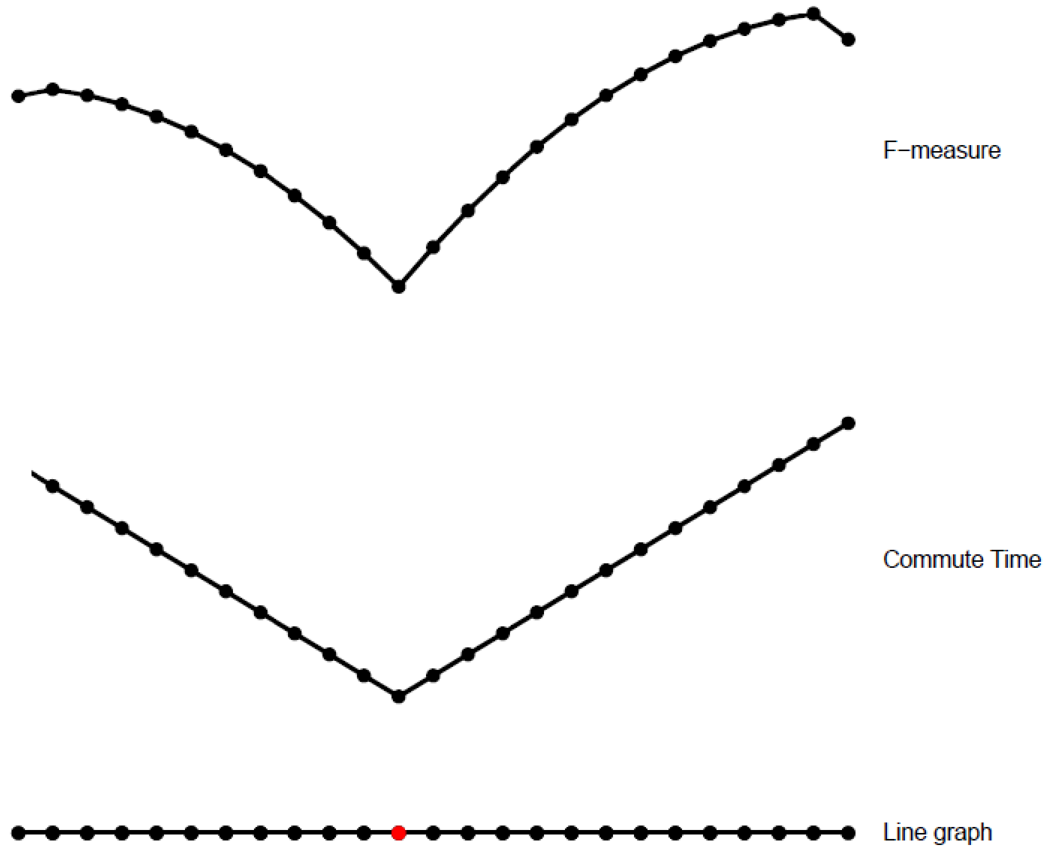
TODO

Try on netflix data! ☺

Explore our “almost commute measure more”

F-MEASURE


Precision at k
1
0.8
0.6
0.4
0.2
0
5 10 15 20 25 30 35 40 45 50
k


F−measure
Commute Time
Line graph

By AngryDogDesign on DeviantArt

Preprint available by request

Slides should be online soon

Code is online already
stanford.edu/~dgleich/
publications/2010/codes/fast-katz