# Sublinear Algorithms for Personalized PageRank, with Applications

Ashish Goel

Joint work with Peter Lofgren; Sid Banerjee; C Seshadhri

# Personalized PageRank

Assume a directed graph with *n* nodes and *m* edges

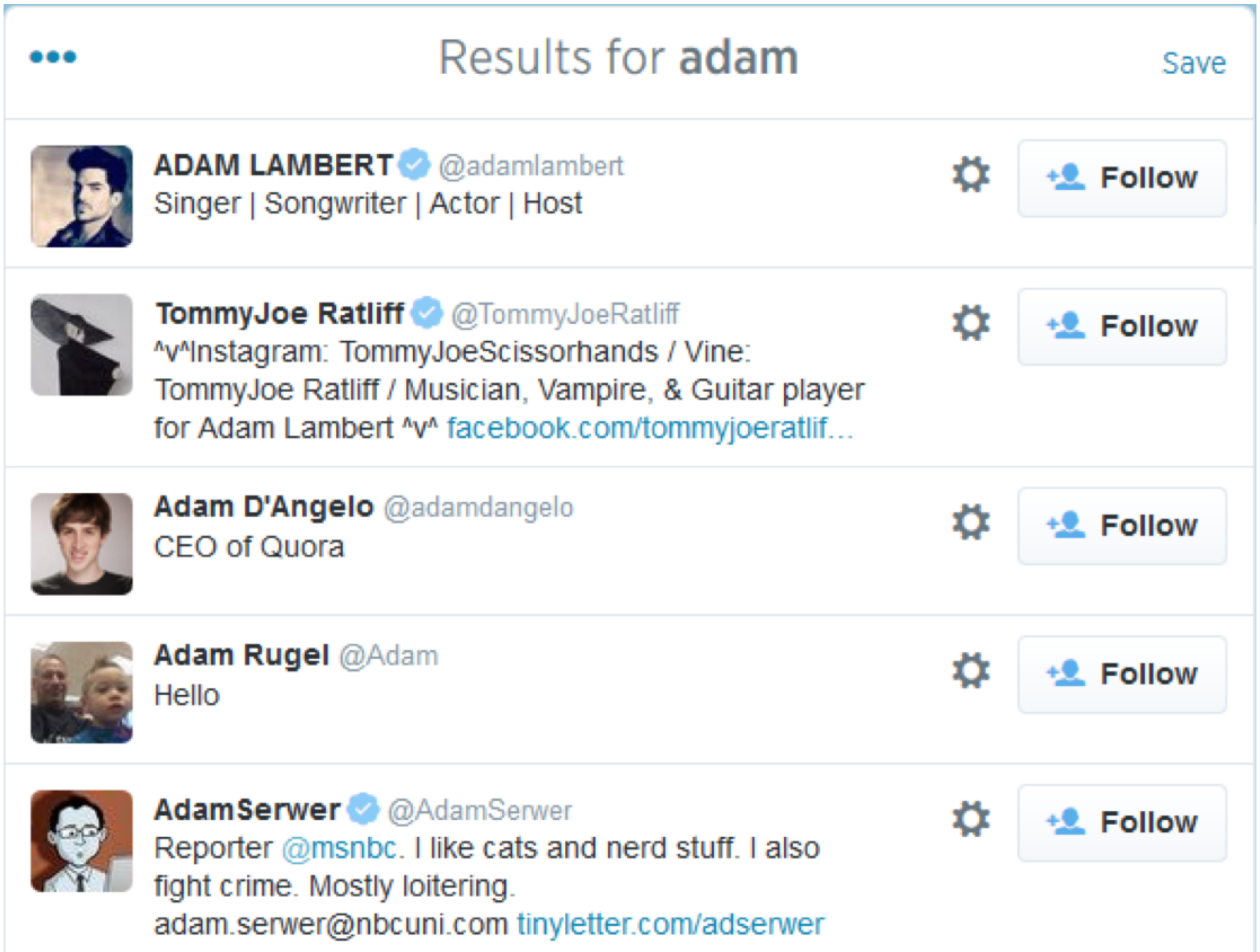
Given source $s$, target $t$, and stopping probability $\alpha$

- Start a random walk from $s$
- At each step, stop with probability $\alpha$, else continue

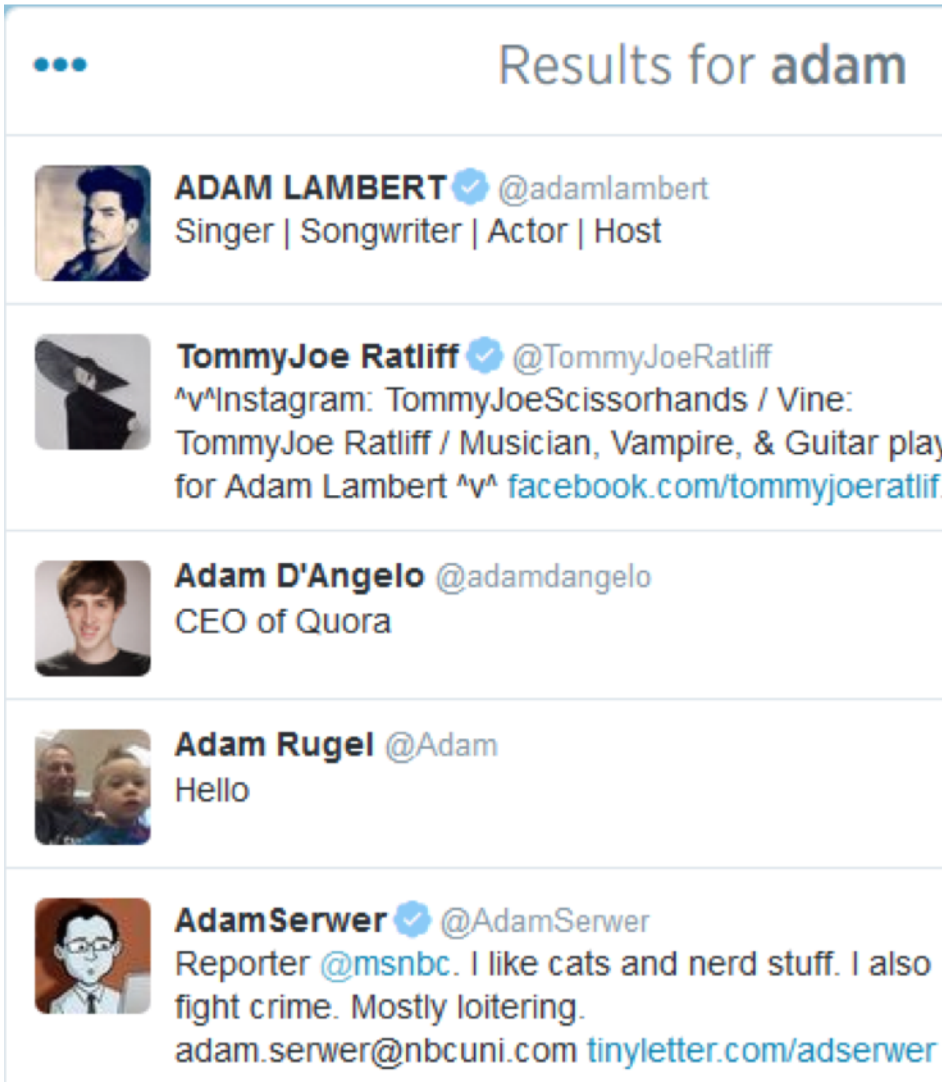
Then Personalized PageRank from $s$ to $t$ is

$$\pi_s(t) = \mathbb{P}[\text{Walk from } s \text{ stops at } t]$$

# Motivation: Personalized Search



# Motivation: Personalized Search

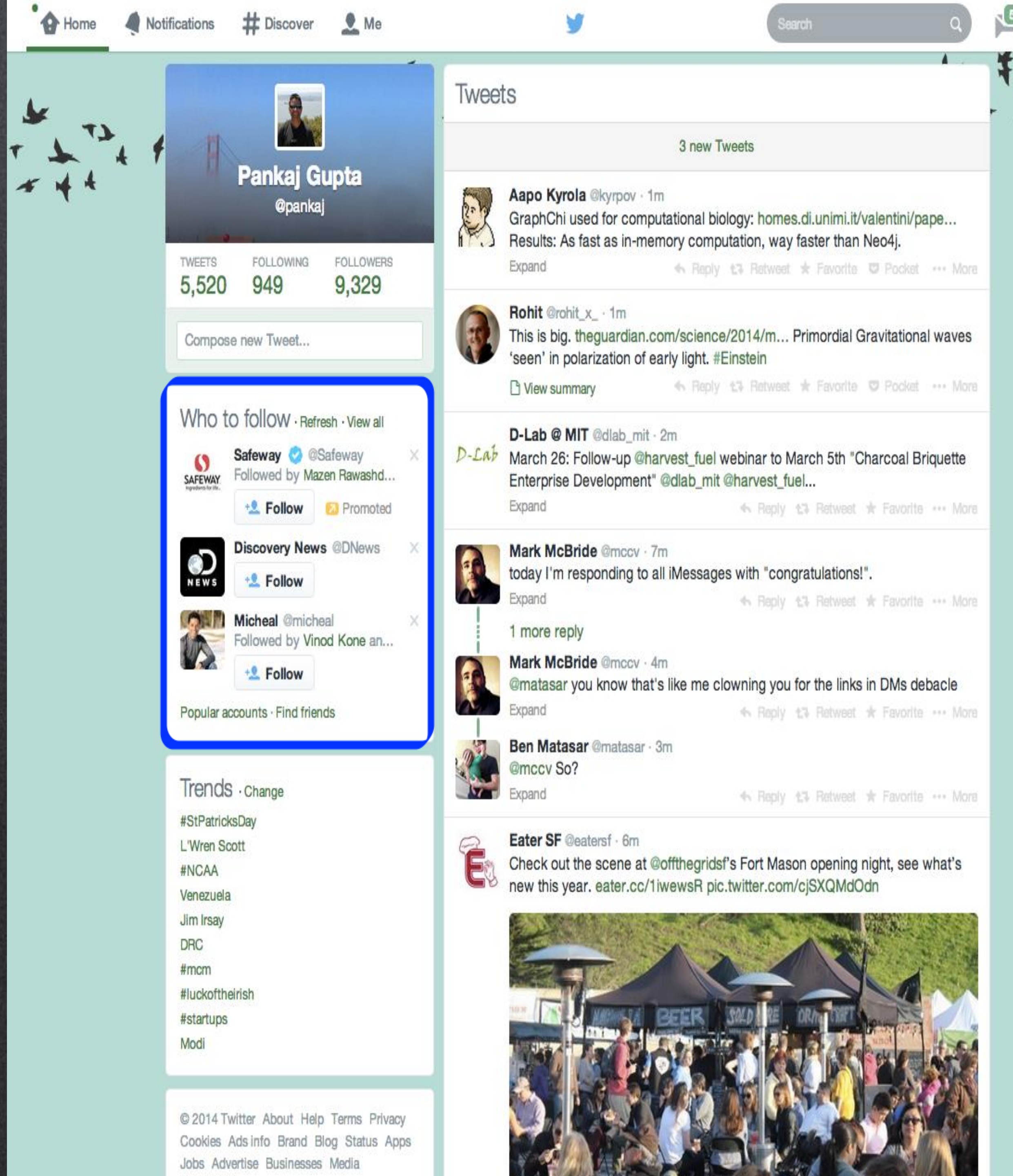



## Re-ranked by PPR

| Name | Description |
|---|---|
| Adam Messinger | CTO @twitter |
| Adam D'Angelo (webpage) | CEO of Quora |
| Adam Satariano (webpage) | Technology Reporter, Bloomberg News satariano.adam[at]http://t.co/g3neLm2J |
| Adam Steltzner | Rocket scientist, intermittent gardener, master of mars, and dangerous dinner guest. Co-founder of Adam and Trisha's dog and baby farm. |
| Adam Rugel (webpage) | Hello |
| AdamSerwer (webpage) | Reporter @msnbc. I like cats and nerd stuff. I also fight crime. Mostly loitering. adam.serwer@nbcuni.com https://t.co/WkKdyjSHcP |

# Applications

- Personalized Web Search
  [Haveliwala, 2003]
- Product Recommendation
  [Baluja, et al, 2008]
- Friend Recommendation (SALSA)
  [Gupta et al, 2013]

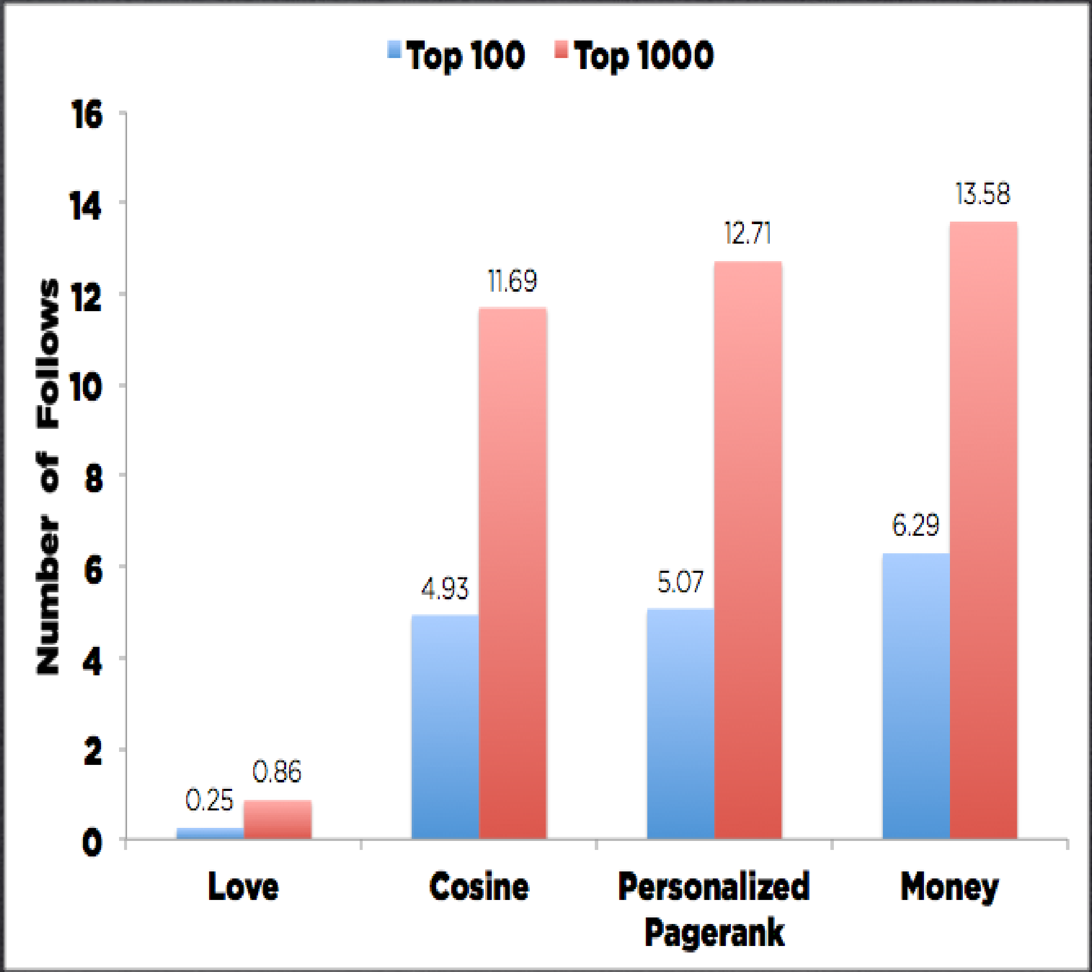

Home
Notifications
Discover
Me
Search
Pankaj Gupta
@pankaj
TWEETS
5,520
FOLLOWING
949
FOLLOWERS
9,329
Compose new Tweet...
Who to follow · Refresh · View all
Safeway @Safeway
Followed by Mazen Rawashd...
Follow
Promoted
Discovery News @DNews
Follow
Micheal @micheal
Followed by Vinod Kone an...
Follow
Popular accounts · Find friends
Trends · Change
#StPatricksDay
L'Wren Scott
#NCAA
Venezuela
Jim Irsay
DRC
#mcm
#luckoftheirish
#startups
Modi
© 2014 Twitter About Help Terms Privacy
Cookies Ads info Brand Blog Status Apps
Jobs Advertise Businesses Media
Tweets
3 new Tweets
Aapo Kyrola @kyrpov · 1m
GraphChi used for computational biology: homes.di.unimi.it/valentini/pape...
Results: As fast as in-memory computation, way faster than Neo4j.
Expand
Reply Retweet Favorite Pocket More
Rohit @rohit_x_ · 1m
This is big. theguardian.com/science/2014/m... Primordial Gravitational waves 'seen' in polarization of early light. #Einstein
View summary
Reply Retweet Favorite Pocket More
D-Lab @ MIT @dlab_mit · 2m
March 26: Follow-up @harvest_fuel webinar to March 5th "Charcoal Briquette Enterprise Development" @dlab_mit @harvest_fuel...
Expand
Reply Retweet Favorite More
Mark McBride @mccv · 7m
today I'm responding to all iMessages with "congratulations!".
Expand
Reply Retweet Favorite More
1 more reply
Mark McBride @mccv · 4m
@matasar you know that's like me clowning you for the links in DMs debacle
Expand
Reply Retweet Favorite More
Ben Matasar @matasar · 3m
@mccv So?
Expand
Reply Retweet Favorite More
Eater SF @eatersf · 6m
Check out the scene at @offthegridsf's Fort Mason opening night, see what's new this year. eater.cc/1iwewsR pic.twitter.com/cjSXQMdOdn
BEER

## A Dark Test for Twitter's People Recommendation System

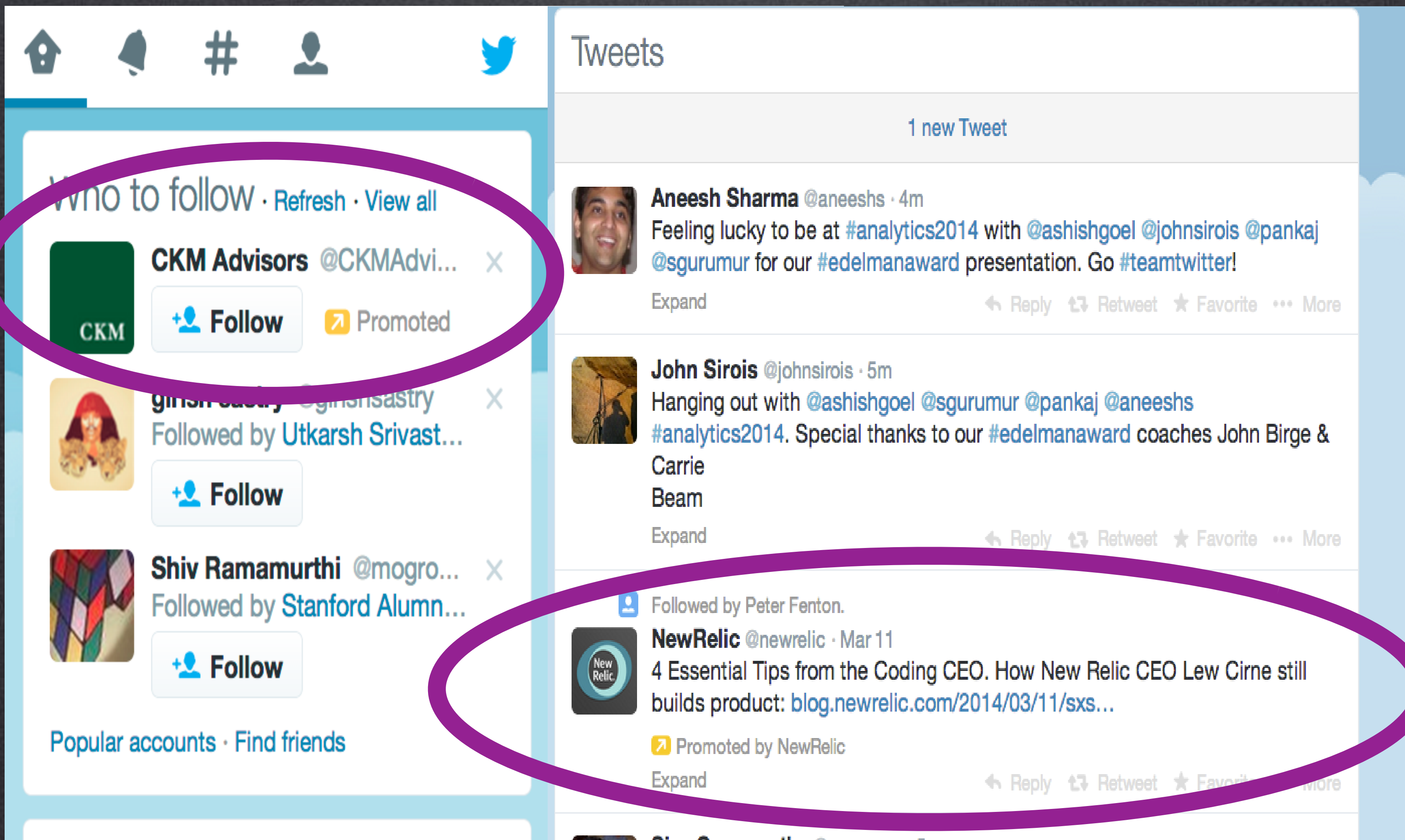
Run various algorithms to predict follows, but don't display the results. Instead, just observe how many of the top predictions get followed organically

(Money = Personalized PageRank on a bipartite graph; Love = HITS)

[Bahmani, Chowdhury, Goel; 2010]

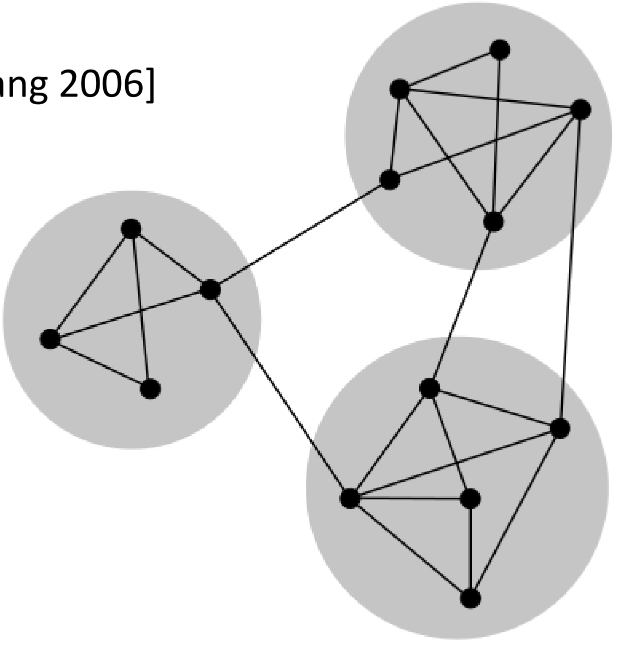


# Promoted Tweets and Promoted Accounts



# Applications

- Community Detection
  - Personalized PageRank
    [Yang, Lescovec 2015],[Andersen, Chung, Lang 2006]



# Estimation Goal

Given $G$, $\alpha$, start node $s$, target $t$, and min probability $\delta$, estimate

$$\pi_s(t)$$

within relative error $\epsilon$ when $\pi_s(t) \geq \delta$

Want only $\pi_s(t)$ not entire $\pi_s \in \mathbb{R}^n$

Since average of $\pi_s$ is $\frac{1}{n}$ we set $\delta = \frac{1}{n}$.

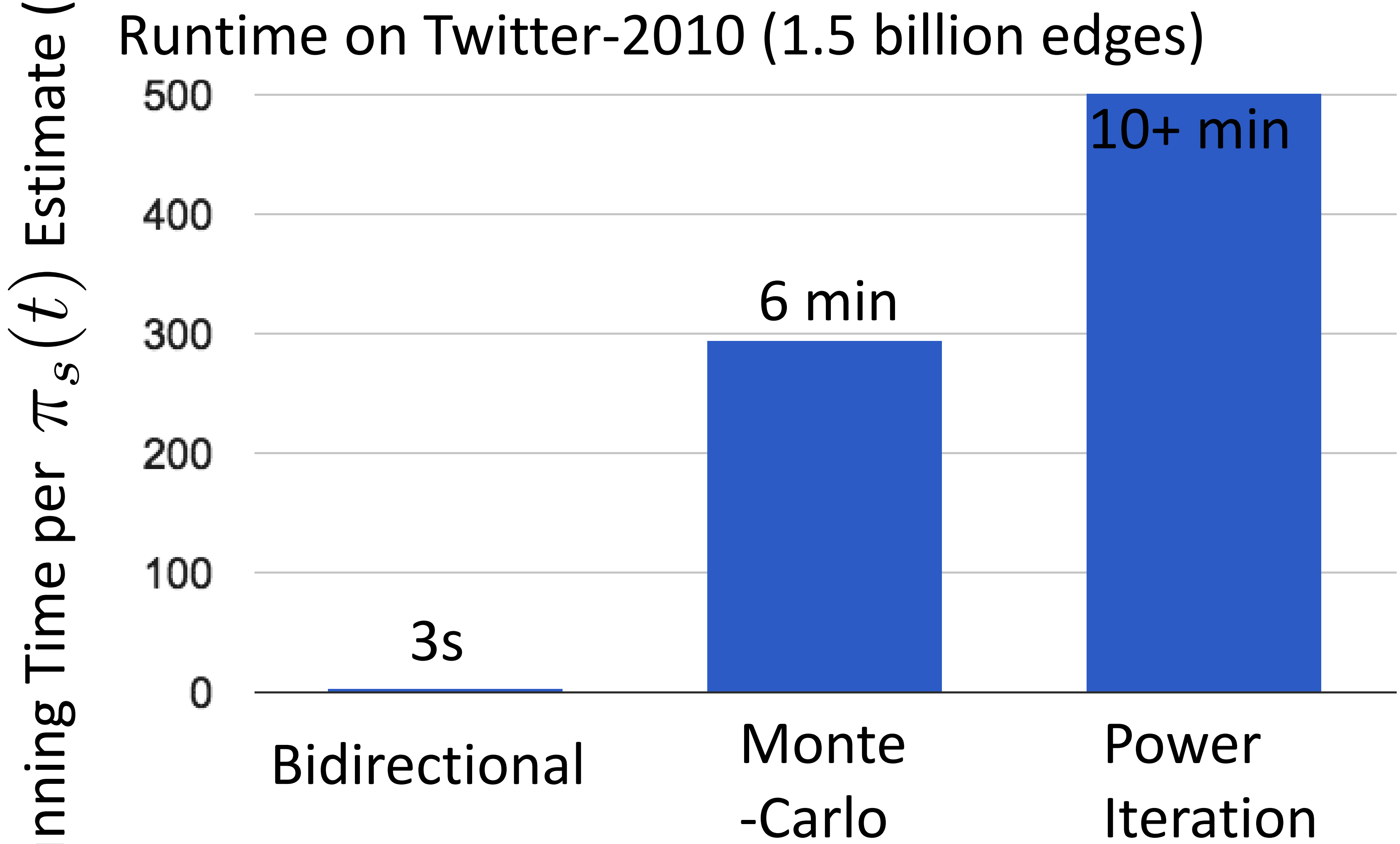
# The Challenge

- Every user has different score vector: Full pre-computation: $O(n^2)$
- Computing from scratch previously took $\Omega(n)$ time—several minutes on Twitter-2010

# Previous Algorithms Summary

- Monte-Carlo: Sample random walks.
- (Local) Power-Iteration: Iteratively improve estimates based on recursive equation

# Results Preview (Experiment)




Mean relative error set to ≈10% for all algorithms.

# Results Preview (Theory)

- Task: estimate $\pi_s(t)$ of size $\frac{1}{n}$ within relative error $\epsilon$
- Previous Algorithms:
  - Monte Carlo: $\Omega\left(\frac{n}{\epsilon^2}\right) \leftarrow$ # Nodes
  - Power Iteration/ Local Update: $\Omega(m) \leftarrow$ # Edges
- Bidirectional Estimator for average target: $\tilde{O}\left(\frac{\sqrt{m}}{\epsilon}\right)$

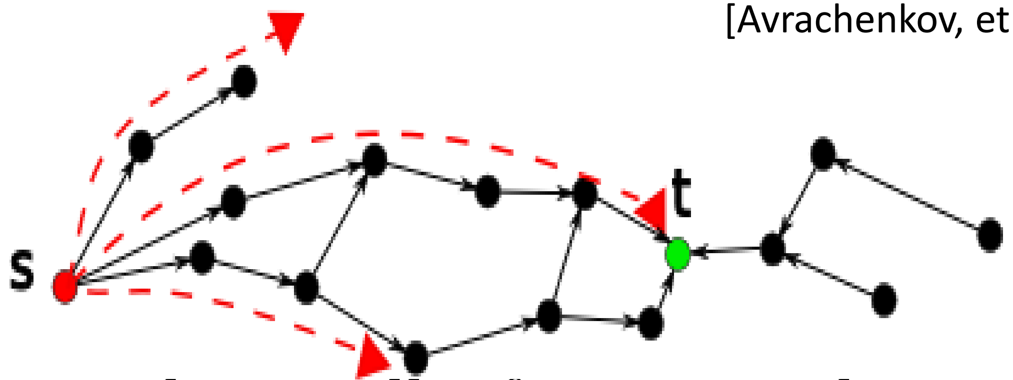
On Twitter-2010, *n*=40M, *m*=1.5B, $\sqrt{m}$ =40K

# Generalizations

- Arbitrary starting distributions.
  Uniform ⇒ Global PageRank
  in average time $\tilde{O}(\sqrt{m})$
- Other Walk Length Distributions like Heat Kernel (used in community detection [Kloster, Gleich 2014],[Chung 2007]): Our estimator is 100x faster on 4 graphs
- Arbitrary Discrete Markov Chain

# Previous Algorithm: Monte-Carlo

[Avrachenkov, et al 2007]




Sample random walks from $s$, and return estimate

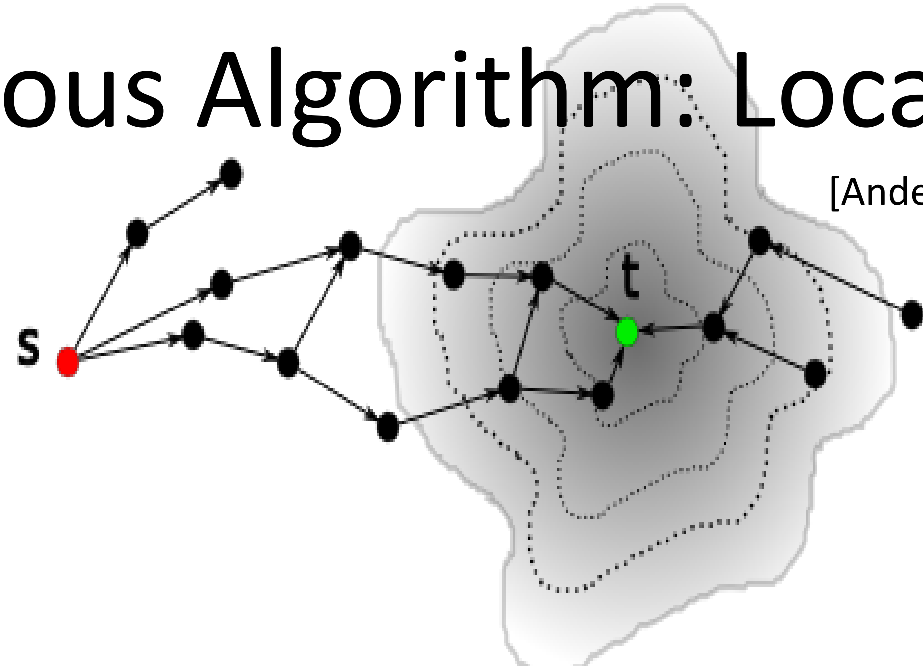
$$\hat{\pi}_s(t) = \text{Fraction of walks ending at } t$$

Running time for $\epsilon$ relative error if $\pi_s(t) \geq \delta$:

$$\Theta\left(\frac{1}{\epsilon^2 \delta}\right)$$

# Previous Algorithm: Local Update

[Andersen, et al 2007]




- Computes $\pi_s(t)$ from *all s* to a *single t*
- Works from t backwards along edges, updating Personalized PageRank estimates locally.
- Running time for average *t*:

$$\approx \frac{\bar{d}}{\delta}$$

Average Degree

Additive Error

# Local Update Background

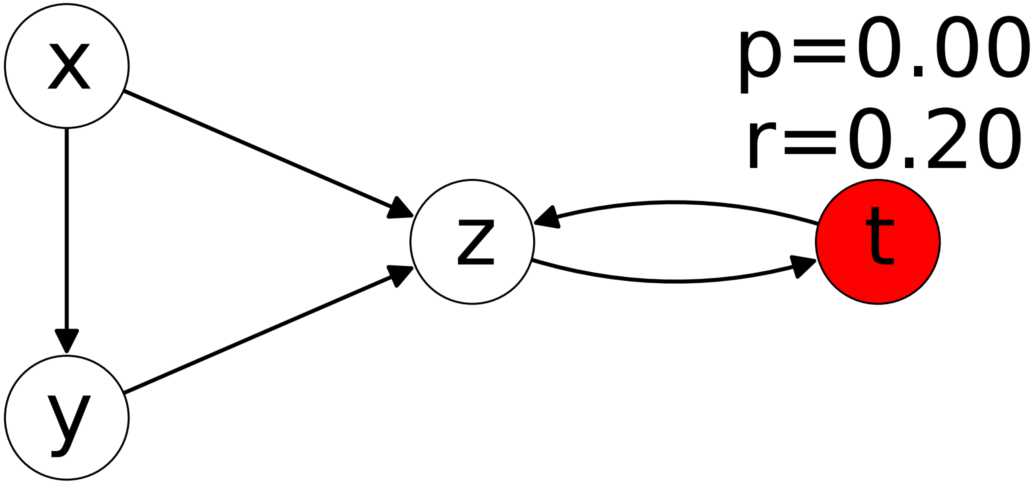
Recursive Definition:

$$\square_s(t) = \underset{0.2}{\square}\mathbb{1}_{[t=s]} + (\underbrace{1-\square}_{0.8})\frac{1}{d_s}\sum_{v \in N^{out}(s)} \square_v(t)$$

# Local Update Example

$$p^t(v) : \text{estimates } \pi_v(t)$$
$$r^t(v) : \text{residual value to be pushed back}$$

After 0 iterations

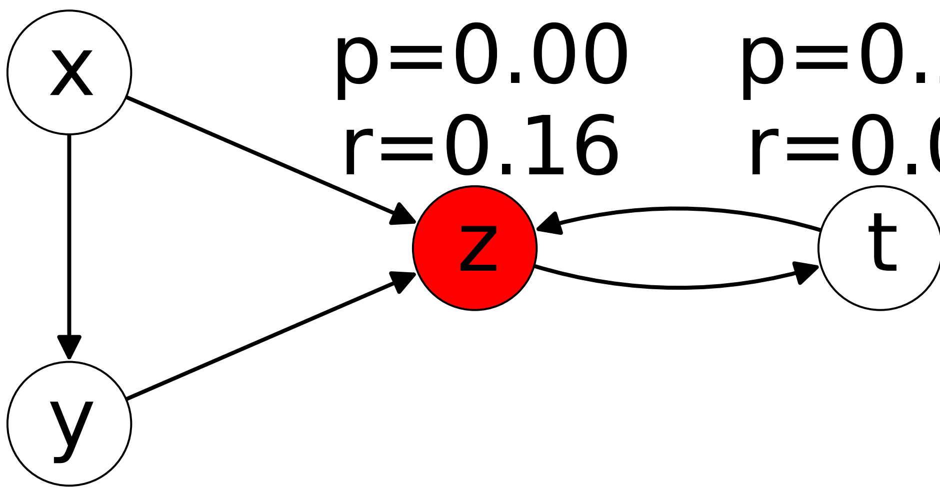


# Local Update Example

$p^t(v)$ : estimates $\pi_v(t)$

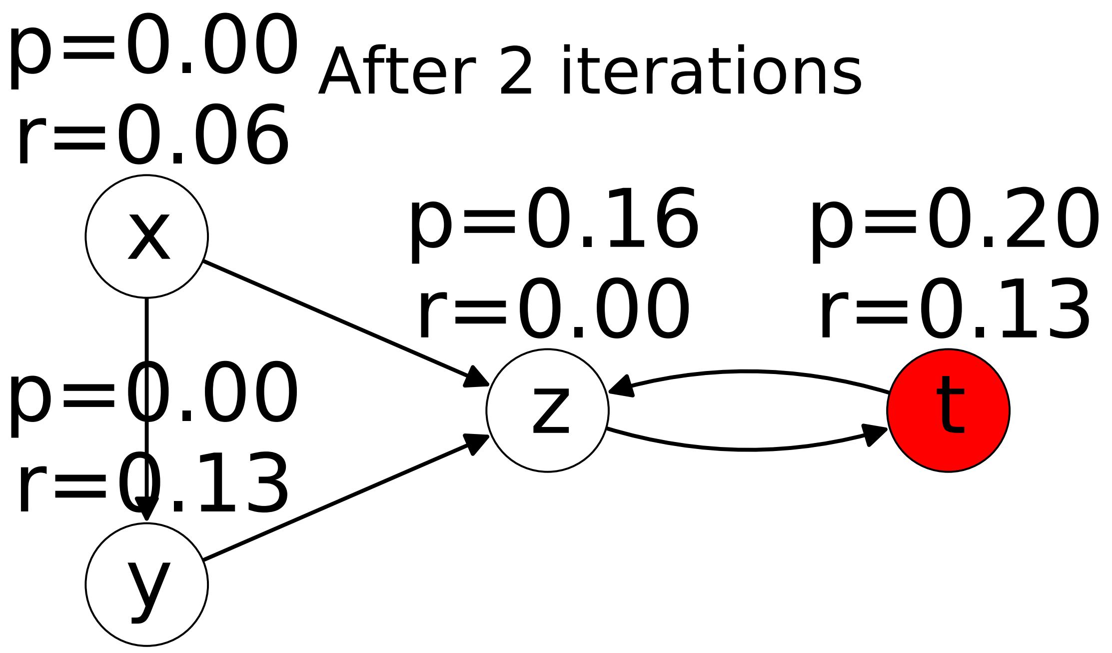
$r^t(v)$ : residual value to be pushed back

After 1 iteration



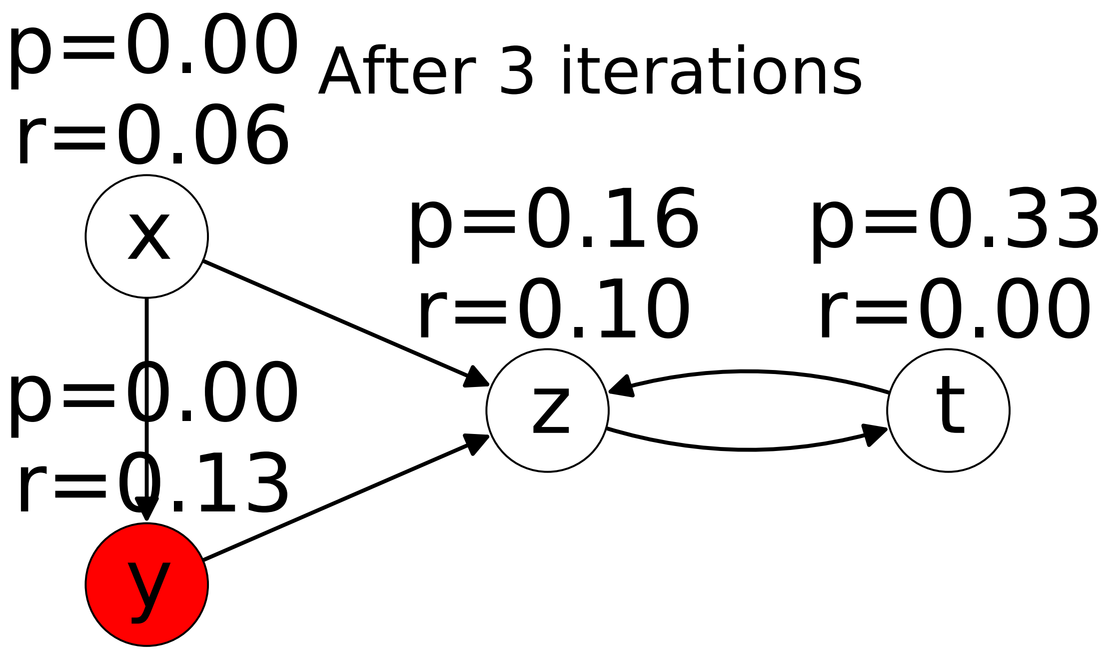
# Local Update Example

$p^t(v)$ : estimates $\pi_v(t)$
$r^t(v)$ : residual value to be pushed back



# Local Update Example

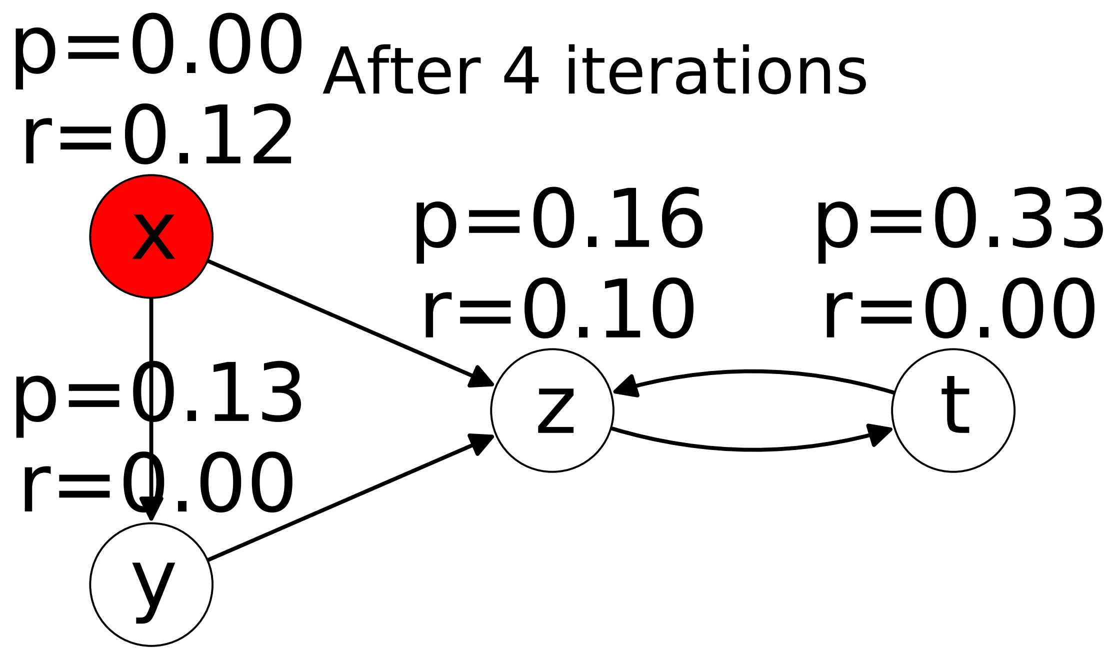
$p^t(v)$ : estimates $\pi_v(t)$
$r^t(v)$ : residual value to be pushed back



# Local Update Example

$p^t(v)$ : estimates $\pi_v(t)$

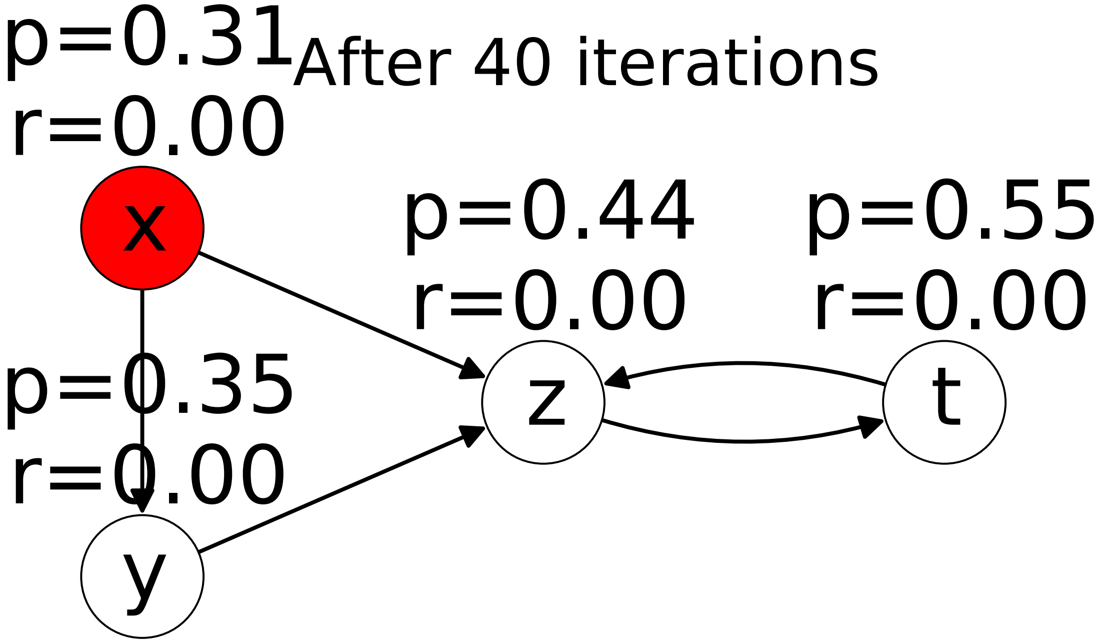
$r^t(v)$ : residual value to be pushed back



# Local Update Example

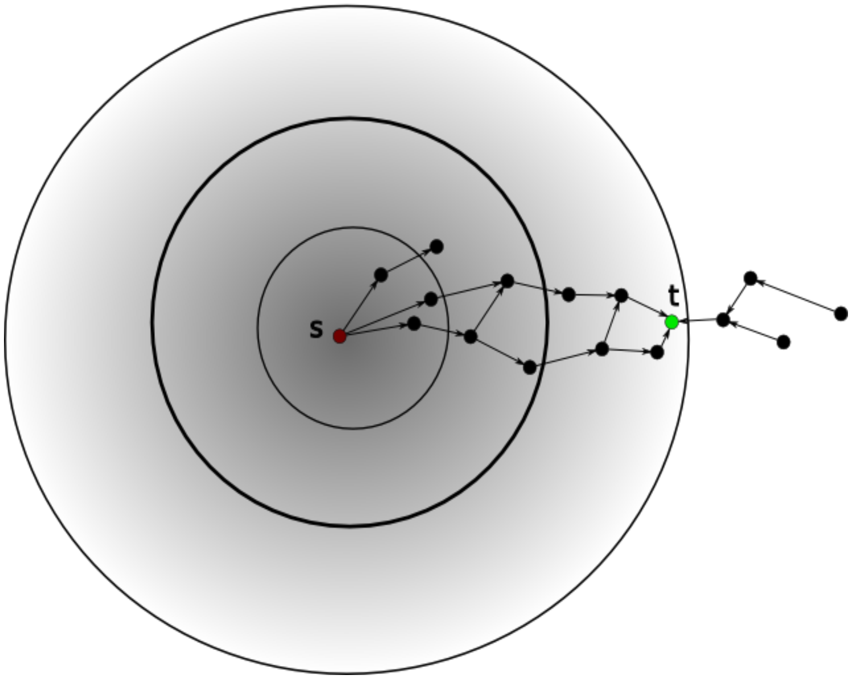
$p^t(v)$ : estimates $\pi_v(t)$

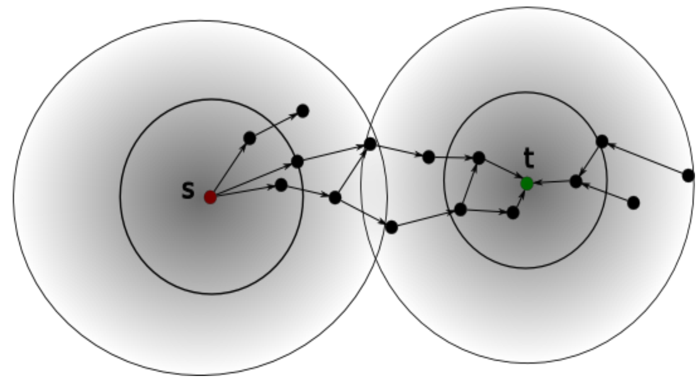
$r^t(v)$ : residual value to be pushed back




Given $r_{\max}$, continue until $\forall v, r^t(v) < r_{\max}$.

# Analogy: Bidirectional Shortest Path




$$d^\ell$$

$$2d^{\ell/2} = O\left(\sqrt{d^\ell}\right)$$

# Bidirectional Estimation

The estimates p and residuals r satisfy a loop invariant [Anderson, et al 2007]:

$$\pi_s(t) = p^t(s) + \underbrace{\frac{1}{\alpha} \sum_{v \in V} \pi_s(v) \underbrace{r^t(v)}_{< r_{\max}}}_{< r_{\max}}$$

Reinterpret the residuals as an expectation!

$$\pi_s(t) = p^t(s) + \alpha^{-1} \mathop{\mathbb{E}}_{v \sim \pi_s} \underbrace{[r^t(v)]}_{< r_{\max}}$$

# Bidirectional-PPR Algorithm

Given $(s, t)$, run local update to threshold $r_{\max}$ to get $(p^t, r^t)$

Sample random walks from $s$ to form

$$\tilde{\pi}_s(v) = \text{Fraction of walks stopping at } v$$

Return

$$p^t(s) + \alpha^{-1} \sum_v \tilde{\pi}_s(v) r^t(v)$$

# Number of samples

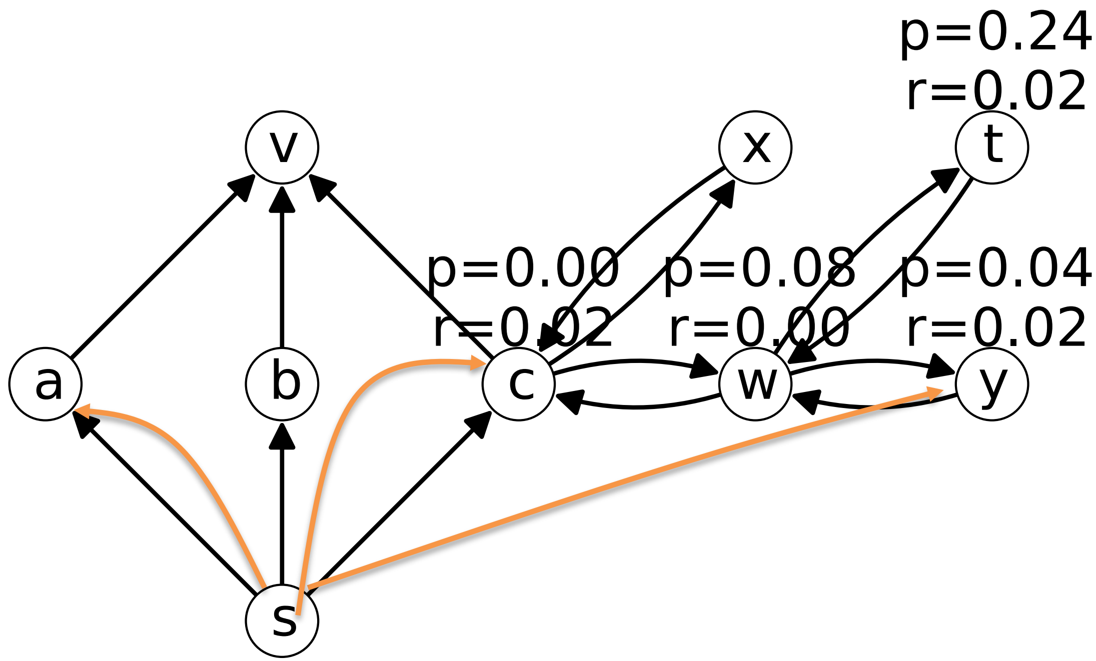
Every walk gives a sample, with

- Maximum value $r_{max}$
- Expected value at least ±

Number of walks needed to get a (1 +/- ²)-approximation with high probability =

$$\tilde{O}\left(\frac{r_{max}}{\epsilon^2\delta}\right) = \tilde{O}\left(\frac{nr_{max}}{\epsilon^2}\right)$$

# Bidirectional-PPR Example




$$\pi_s(t) = p^t(s) + \alpha^{-1} \sum_v \tilde{\pi}_s(v) r^t(v)$$

$$\approx \frac{1}{3} 0.10 + \frac{1}{3} 0.09$$

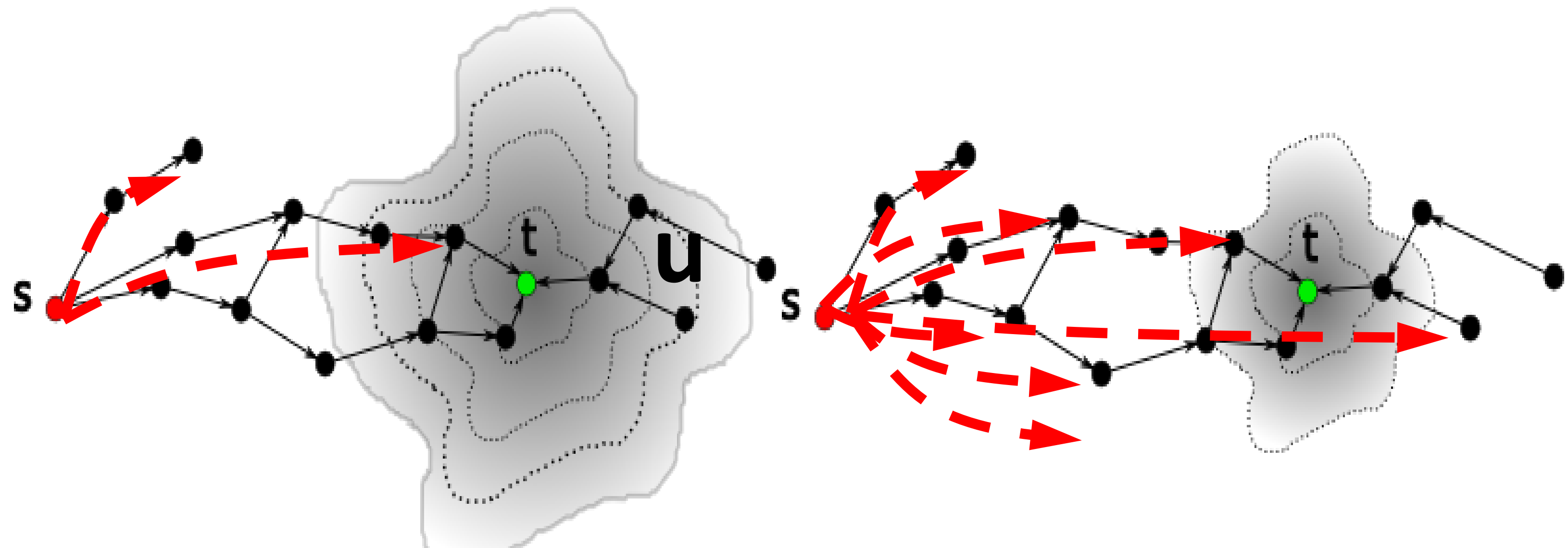
# Theoretical Results

Bidirectional-PPR estimates $\pi_s(t)$ of size $\frac{1}{n}$ within relative error $\epsilon$ (with high probability).

Average Running Time ($t$ chosen uniformly):

$$\tilde{O}\left(\frac{\sqrt{m}}{\epsilon}\right)$$

# Forward vs Reverse Work Trade-off




More Reverse Push
Fewer Walks

More Walks
Fewer Reverse Pushes

$$\text{Average Running time} = \tilde{O}\left(\underbrace{\frac{n \cdot r_{\max}}{\epsilon^2}}_{\text{Forward Walks}} + \underbrace{\frac{\bar{d}}{r_{\max}}}_{\text{Reverse Pushes}}\right)$$

# Theoretical Results

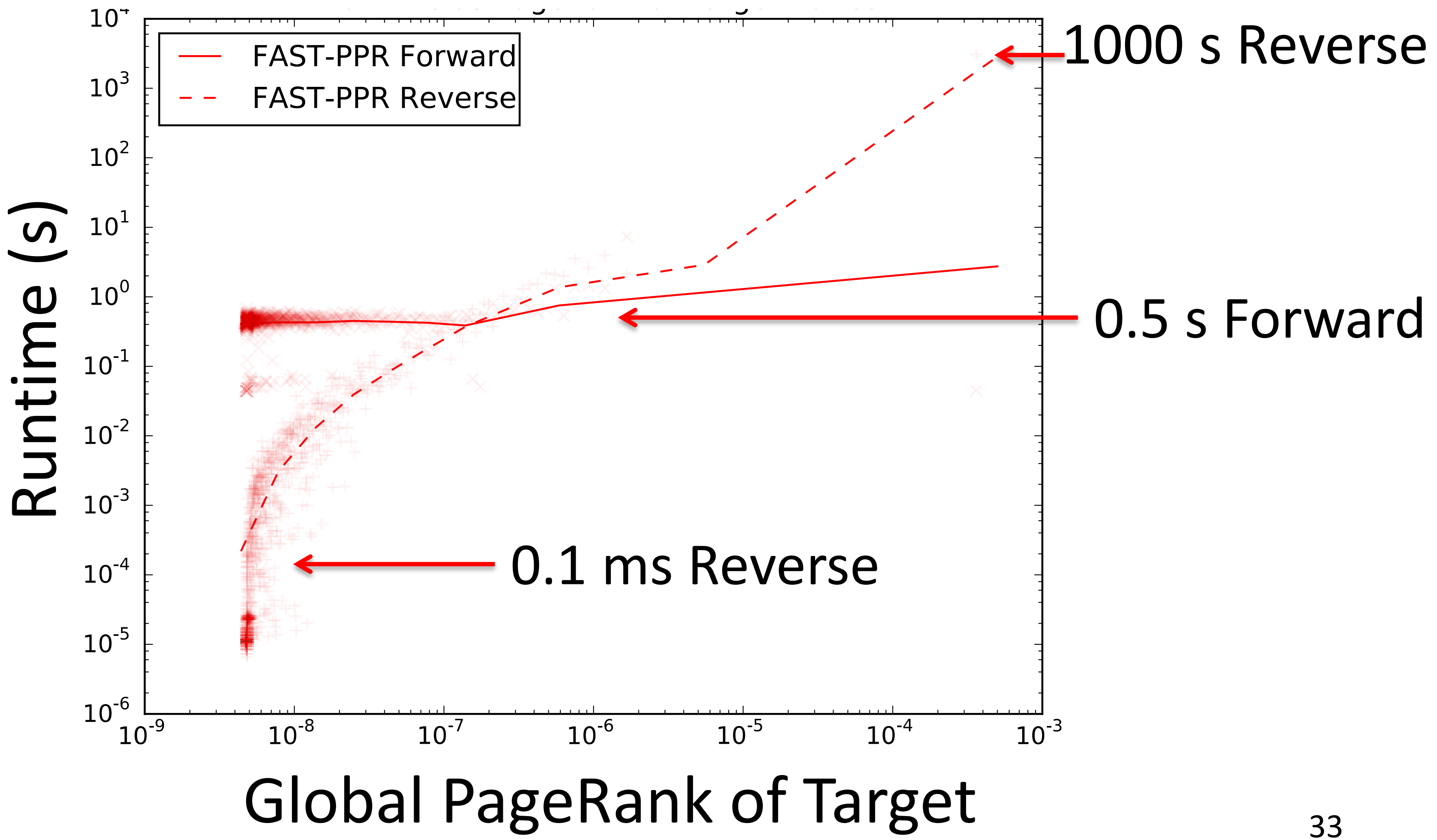
Given $O\left(\frac{\sqrt{m}}{\epsilon}\right)$ pre-computation and storage per node, worst case running time is

$$\tilde{O}\left(\frac{\sqrt{m}}{\epsilon}\right)$$

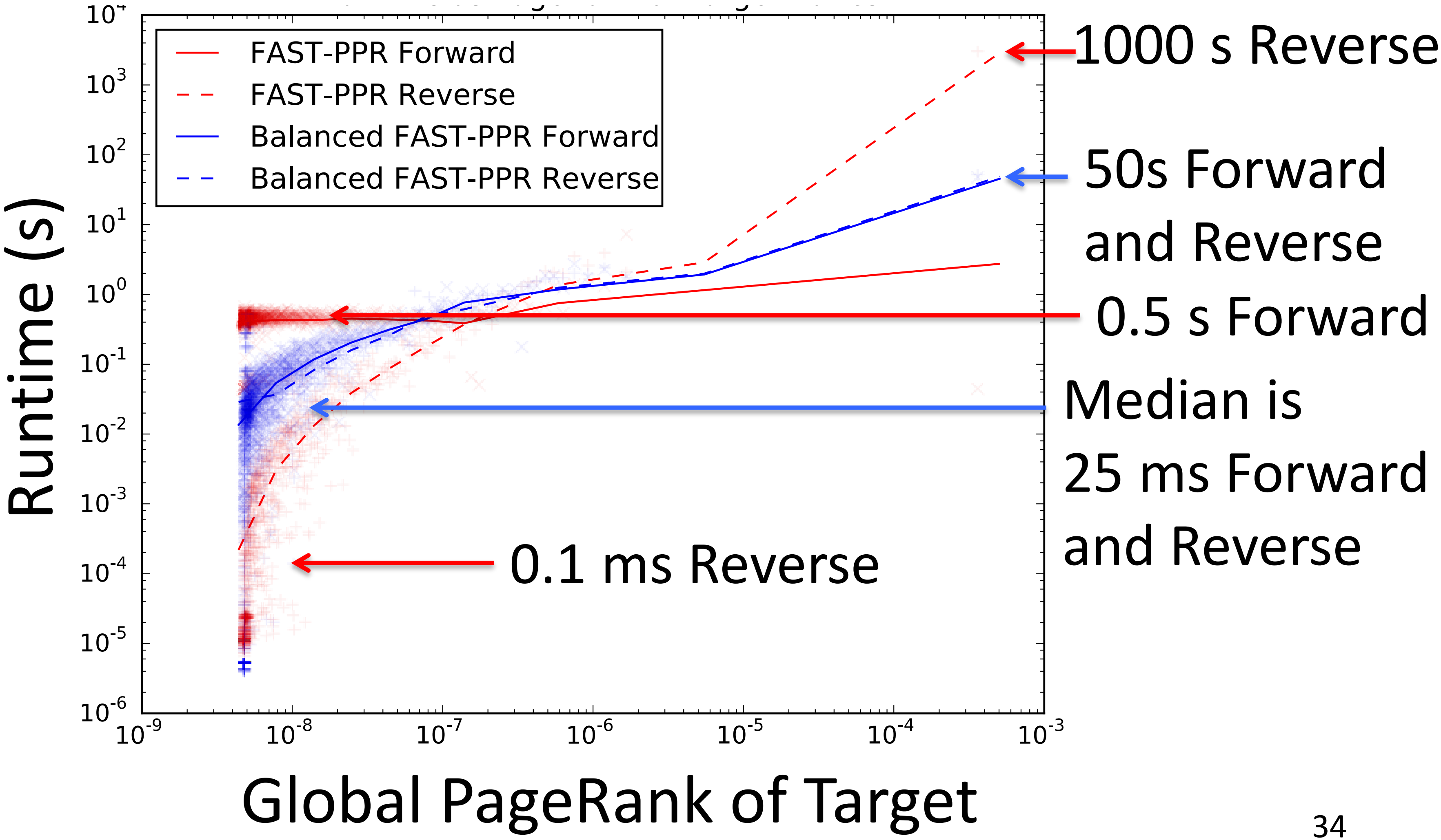
Time-Space Trade-off

[Lofgren, Banerjee, Goel, Seshadhri 2014; Lofgren, Banerjee, Goel 2015]

# Problem: Unbalanced Forward and Reverse Runtime



# Heuristic: Balancing Forward and Reverse Runtime



# Experiments

| Dataset | Type | # Nodes | # Edges |
|---|---|---|---|
| DBLP-2011 | undirected | 986K | 6.7M |
| Pokec | directed | 1.6M | 30.6M |
| LiveJournal | undirected | 4.8M | 69M |
| Orkut | undirected | 3.1M | 117M |
| Twitter-2010 | directed | 42M | 1.5B |
| UK-2007-05 | directed | 106M | 3.7B |

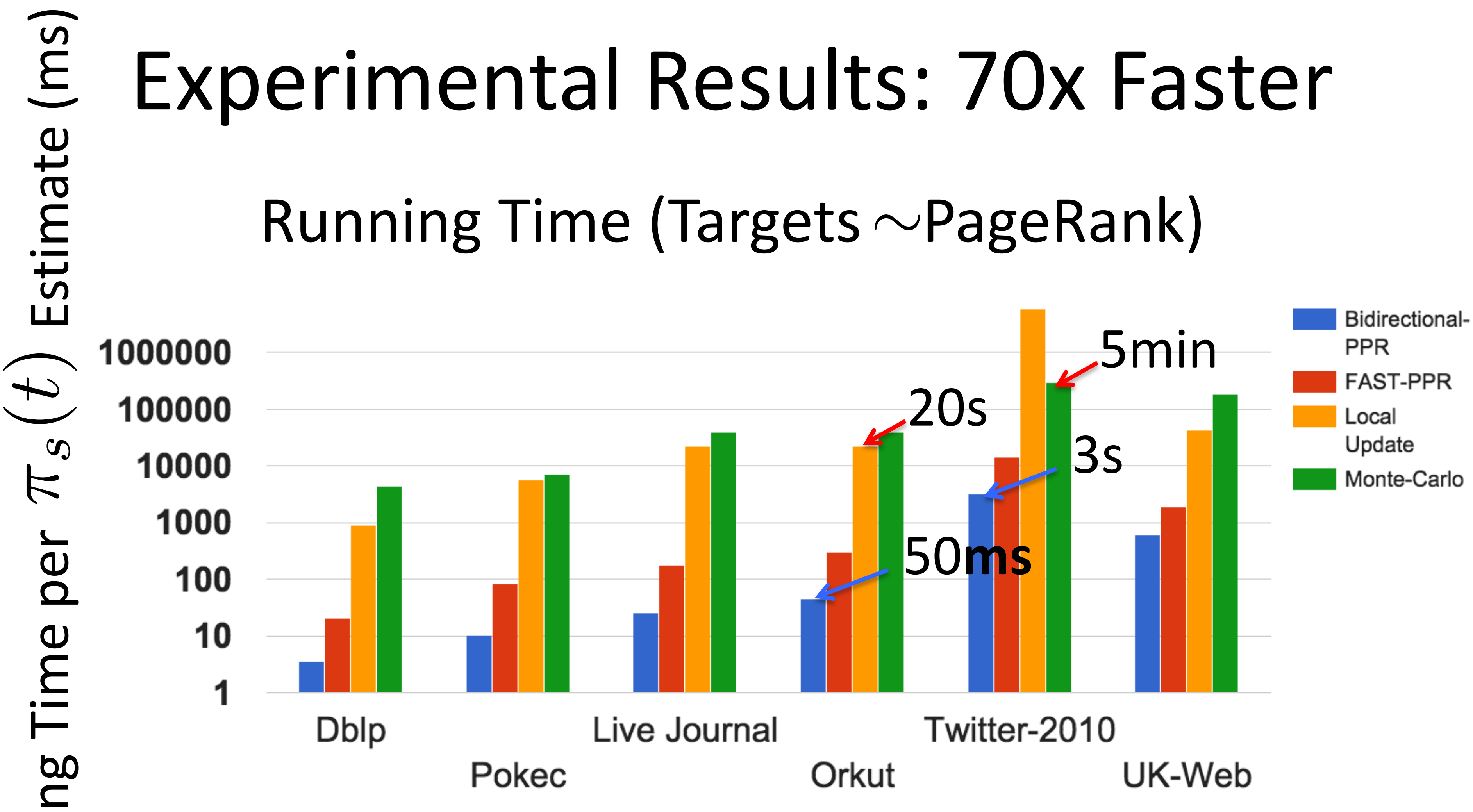
Teleport prob. $\alpha = 0.2$

Minimum ppr $\delta = \frac{4}{n}$

Parameters chosen so mean empirical relative error $\approx 10\%$

$s \sim$ uniform random, $t \sim$ global PageRank

# Experimental Results: 70x Faster




Mean relative error set to ≈10% for all algorithms.

# Alternative Estimator for Undirected Graphs

Key property: $\pi_s(t) = \frac{d_t}{d_s}\pi_t(s)$

We push forwards from s, and take random walks from t.

Result: We can estimate $\pi_s(t)$ larger than $\frac{d_t}{m}$ in worst-case time

$\lim_{\alpha \to 0} \pi_s(t)$

$$\tilde{O}\left(\frac{\sqrt{m}}{\epsilon}\right)$$

with relative error $\epsilon$ with high probability.

# Alternative Algorithm for Undirected Graphs

- Loop Invariant of push-forward algorithm [Andersen, Chung, Lang, 2006]

$$\pi_s(t) = p^s(t) + \sum_{v \in V} r^s(v)\pi_v(t)$$

- Use symmetry, and then interpret as expectation

$$\begin{aligned}\pi_s(t) &= p^s(t) + \sum_{v} \left( r^s(v)\pi_t(v)\frac{d_t}{d_v} \right) \\ &= p^s(t) + \mathop{\mathbb{E}}_{v \sim \pi_t}\left[ \frac{r^s(v)}{d_v} d_t \right]\end{aligned}$$

# Alternative Algorithm for Undirected Graphs

- Loop Invariant of push-forward algorithm [Andersen, Chung, Lang, 2006]

$$\pi_s(t) = p^s(t) + \sum_v r^s(v)\pi_v(t)$$

$r_{max}$

- Use symmetry, a… …et as expectation

$$\begin{aligned}\pi_s(t) &= p^s(t) + \sum_v \left( r^s(v)\pi_t(v)\frac{d_t}{d_v} \right) \\ &= p^s(t) + \mathop{\mathbb{E}}_{v\sim\pi_t}\left[ \boxed{\frac{r^s(v)}{d_v}} d_t \right]\end{aligned}$$

# Running time for Undirected Graphs

$$
\begin{aligned}
\text{Running Time} &= \tilde{O}\left(\frac{1}{r_{\max}} + \frac{r_{\max} d_t}{\delta \epsilon^2}\right) \\
&= \tilde{O}\left(\frac{1}{\epsilon}\sqrt{\frac{d_t}{\delta}}\right) \\
&= \tilde{O}\left(\frac{\sqrt{m}}{\epsilon}\right) \ \text{if } \delta = d_t/m
\end{aligned}
$$

# Open Problems

- Get rid of the dependence on degree, to get an amortized bound of $O(\pm^{1/2})$
- Get a worst-case bound of $O(m^{1/2})$ for directed graphs under the condition that the target has a high global PageRank
- Find sharding and sampling algorithms that preserve Personalized PageRank (eg. a sparsifier for Personalized PageRank?)
- Build an index around Personalized PageRank to enable network based Personalized Search

# Open Problems

- Get rid of the dependence on degree, to get an amortized bound of $O(\pm^{1/2})$
- Get a worst-case bound of $O(m^{1/2})$ for directed graphs under the condition that the target has a high global PageRank
- Find sharding and sampling algorithms that preserve Personalized PageRank (eg. a sparsifier for Personalized PageRank?)
- Build an index around Personalized PageRank to enable network based Personalized Search

# Personalized Search Problem

Given

- A network with nodes (with keywords) and edges (weighted, directed)—Twitter
- A query, filtering nodes to a set T— “People named Adam”
- A user s (or distribution over nodes) —me

Rank the approximate top-k targets by $\pi_s(t)$

# Personalized Search Problem

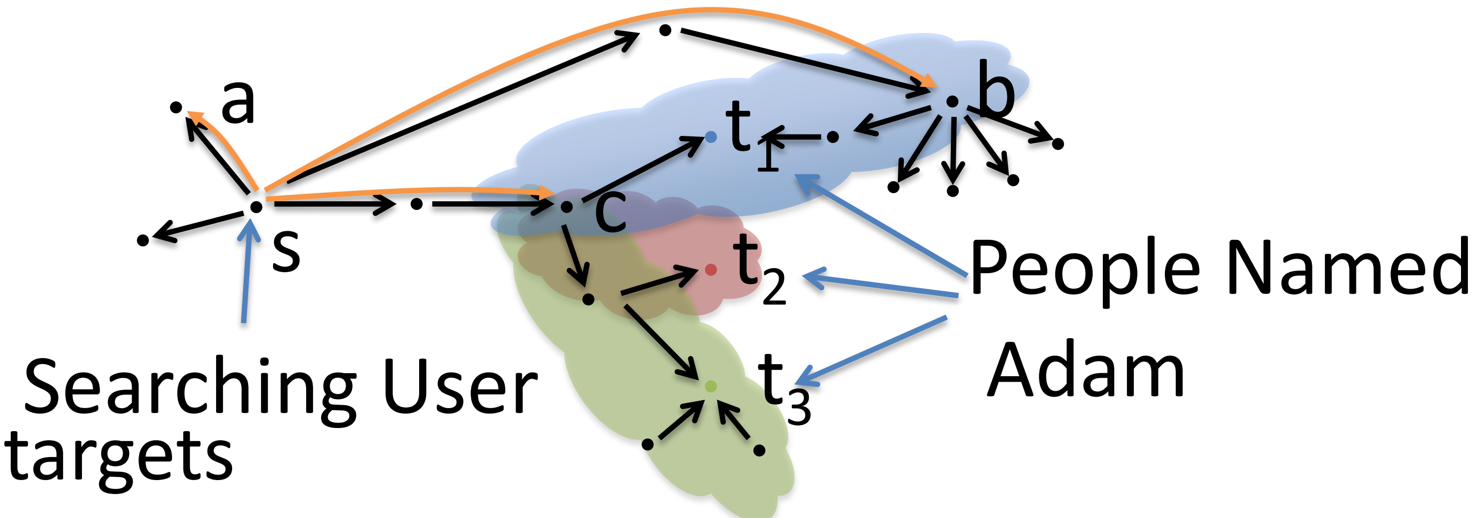
Baselines:

- Monte Carlo: Needs many walks to find enough samples within T unless T is very large
- Bidirectional-PPR to each *t*: Slow unless T is small

Challenge: Can we efficiently find top-k for any size of T?

Idea: Modify Bidirectional-PPR to sample $t \in T$ in proportion to $\pi_s(t)$

# Personalized Search Example




Expand targets

Random walks

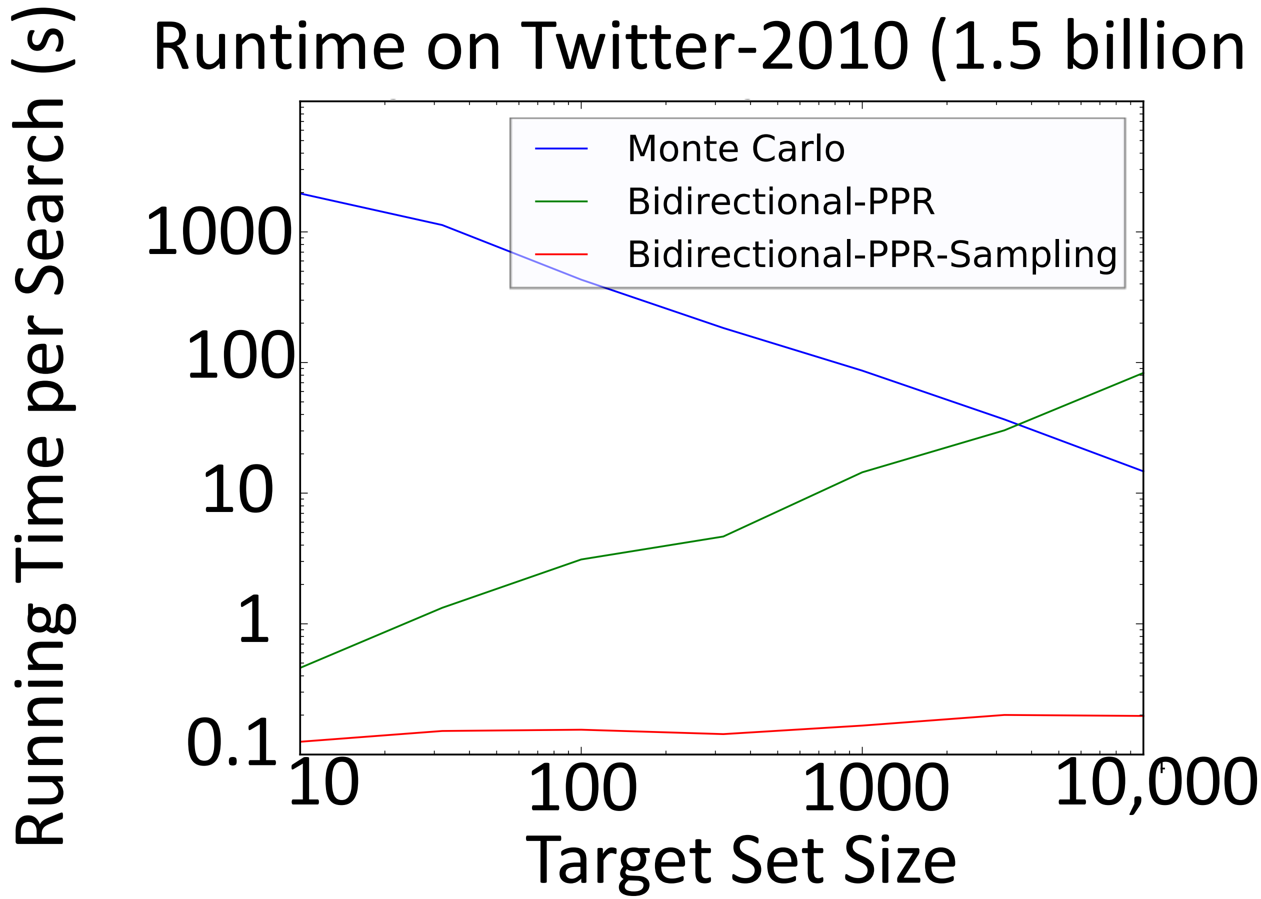
To sample a target:

layer 1: sample (a,b,c) w.p. (0, 10%, 90%)

Layer 2: b→ $t_1$

c→ sample ($t_1$, $t_2$, $t_3$) w.p (56%, 22%, 22%)

# Personalized Search Running Time




Precision@3 set to 90% for all algorithms.

Significant Pre-computation (3-30MB per keyword)

# Personalized Search Result

Theorem: Using $O\left(n\sqrt{m}\right)$ storage, we can sample a target $n_s$ times from a distribution approximating $\pi_s(t)|t \in T$ in time

$$\tilde{O}\left(\frac{\sqrt{m}}{\epsilon} + n_s\right)$$

In Experiments, $n_s = \frac{\sqrt{m}}{\epsilon}$

# Demo

CiteSeer$^x$ 5M

entropy computer network

Include Citations

Results **1 - 10** of **1,124,238** Next 10 →

## Entropy and Partial Differential Equations

by Lawrence C. Evans - *AMERICAN MATHEMATICAL SOCIETY, VOLUME* , *1998*

"... ..."

Abstract - Cited by 762 (3 self) - Add to MetaCart

## A Maximum-Entropy-Inspired Parser

by Eugene Charniak , *1999*

"... We present a new parser for parsing down to Penn tree-bank style parse trees that achieves 90.1% average precision/recall for sentences of length 40 and less, and 89.5% for sentences of length 100 and less when trained and tested on the previously established [5,9,10,15,17] "stan- dard" se ..."

Abstract - Cited by 878 (18 self) - Add to MetaCart

## A Maximum Entropy approach to Natural Language Processing

by Adam L. Berger, Stephen A. Della Pietra , Vincent J. Della Pietra - *COMPUTATIONAL LINGUISTICS* , *1996*

"... The concept of maximum entropy can be traced back along multiple threads to Biblical times. Only recently, however, have computers become powerful enough to permit the widescale application of this concept to real world problems in statistical estimation and pattern recognition. In this paper we des ..."

Abstract - Cited by 1152 (5 self) - Add to MetaCart

Google

entropy computer network

Scholar

About 529,000 results (**0.05** sec)

Articles

Case law

My library

Any time

Since 2015

Since 2014

Since 2011

Custom range...

Sort by relevance

Sort by date

☑ include patents

☑ include citations

Create alert

A quantitative evaluation technique of attack effect of **computer network** based on **network entropy**

Y ZHANG, M XIAN, G WANG - Journal of China Institute of, 2004 - en.cnki.com.cn

An attack effect evaluation model of **computer network** based on **network entropy** was proposed in this paper. The paper put forward the concept of **network entropy** from the point of view of the security characteristic change of **computer network** after it was attacked. The **...**

Cited by 33 Related articles Cite Save More

What is **computer** security?

M Bishop - Security & Privacy, IEEE, 2003 - ieeexplore.ieee.org

**...** 67 MATT BISHOP University of California, Davis **Computer** and **network** security, or cybersecurity, are critical issues. But merely protecting the systems that hold data about citizens, corporations, and government agencies it is not enough. The infra- **...**

Cited by 1827 Related articles All 14 versions Cite Save More

[CITATION] … **Computer Network** Intrusion Protection Center, GSCAS, Beijing 100039); Fuzzy Risk Assessment of **Entropy**-weight Coefficient Method Applied in **Network** …

Z Dongmei - **Computer** Engineering, 2004

Cited by 17 Related articles Cite Save

Detecting anomalies in **network** traffic using maximum **entropy** estimation

Y Gu, A McCallum, D Towsley - Proceedings of the 5th ACM SIGCOMM …, 2005 - dl.acm.org

**...** [3] ENDACE. http://www.endace.com. [4] GU, Y., MCCALLUM, A., AND TOWSLEY, D. Detecting anomalies in **network** traffic using maximum **entropy**. Tech. rep., Department of **Computer** Science, UMASS, Amherst, 2005. [5] INTEL CORP. Intel ixp 1200 **network** processor, 2000. **...**

Cited by 286 Related articles All 44 versions Cite Save

# Demo

Task: Find applications of entropy in computer networking.

[personalizedsearchdemo.com](personalizedsearchdemo.com)

Keyword: entropy Personalize to Author Name: Nick McKeown Exclude Author's papers: Search

20 of 17193 Results -- Monte Carlo

**Encode-then-encipher encryption: How to exploit nonces or redundancy in plaintexts for efficient cryptography**

1.42E-5

by Mihir Bellare, Phillip Rogaway

We investigate the following approach to symmetric encryption: first encode the message via some keyless transform, and then encipher the encoded message, meaning apply a permutation FK based on a shared key K. We provid ...

---

**Entropy Of ATM Traffic Streams: A Tool For Estimating QoS Parameters**

8.0E-6

by N.G. Duffield, J.T. Lewis, Neil O'Connell, Raymond Russell, Fergal Toomey , *1995*

this paper, we are concerned with the components of cell-loss and cell-delay which are attributable to a single buffer of finite size. The QoS parameters we are concerned with are: ...

---

**Thermodynamic Probability Theory: Some Aspects Of Large Deviations**

5.9E-6

by J.T. Lewis, C.-E. Pfister , *1993*

this paper. The probability measures which are studied in the theory of large deviations are the distributions of random variables taking values in a topological space X, so that they are measures on a topological space ...

---

**LeZi-Update: An Information-Theoretic Approach to Track Mobile Users in PCS Networks**

5.5E-6

by Amiya Bhattacharya, Sajal K. Das , *1999*

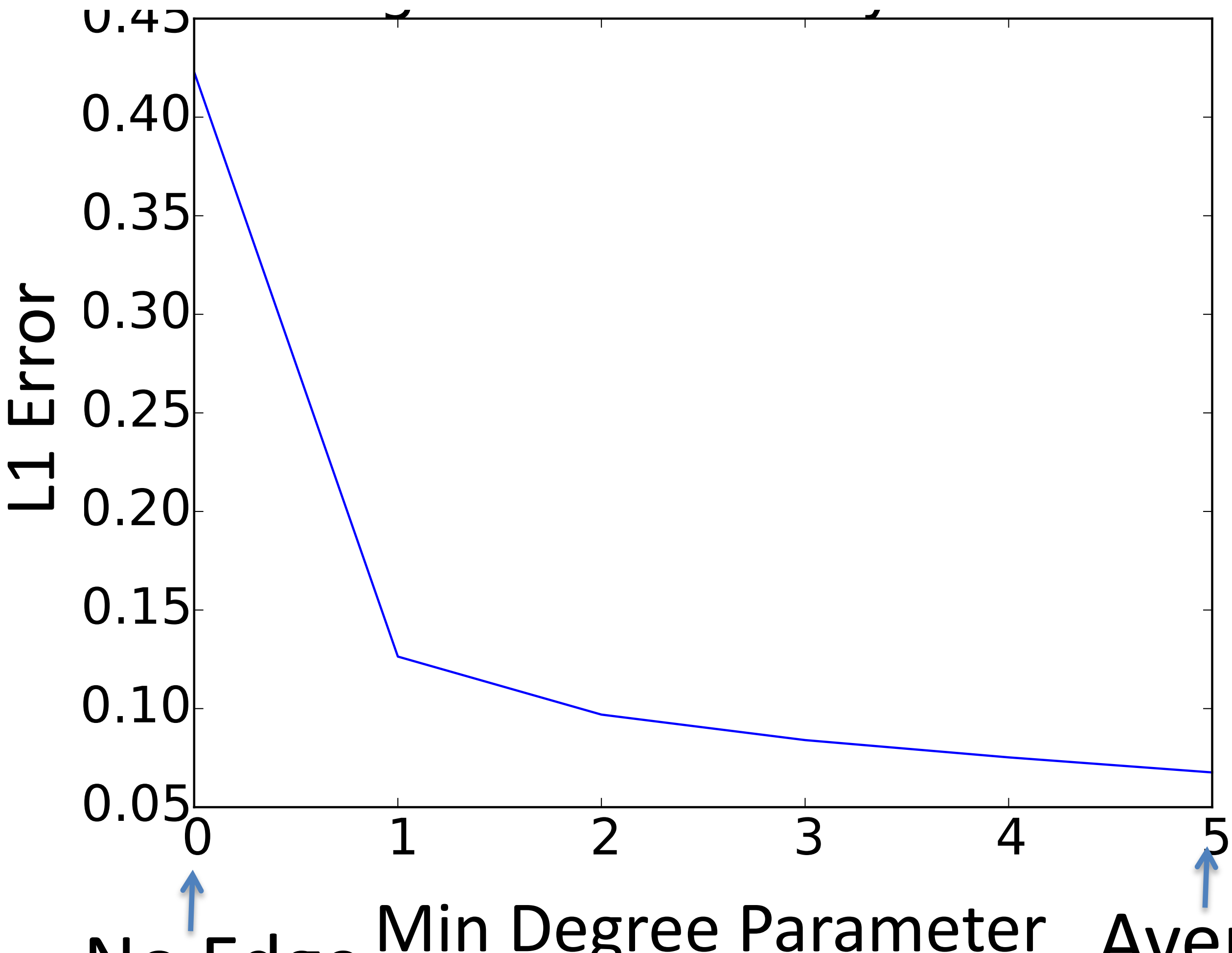
The complexity of the mobility tracking problem in a cellular environment has been characterized under an informationtheoretic framework. Shannon's entropy measure is identified as a basis for comparing user mobility mod ...

---

# Distributed PageRank

- Problem: Computing PageRank on graph too large for one machine.
- Algorithm:
  - Shard edges randomly,
  - compute on each machine
  - average results
- Basic idea: Duplicate edges from low-degree nodes. Gives an unbiased* estimator.

# Sharded Global PageRank Accuracy on Twitter-2010




No Edge Duplication

Average 2.1x Duplication