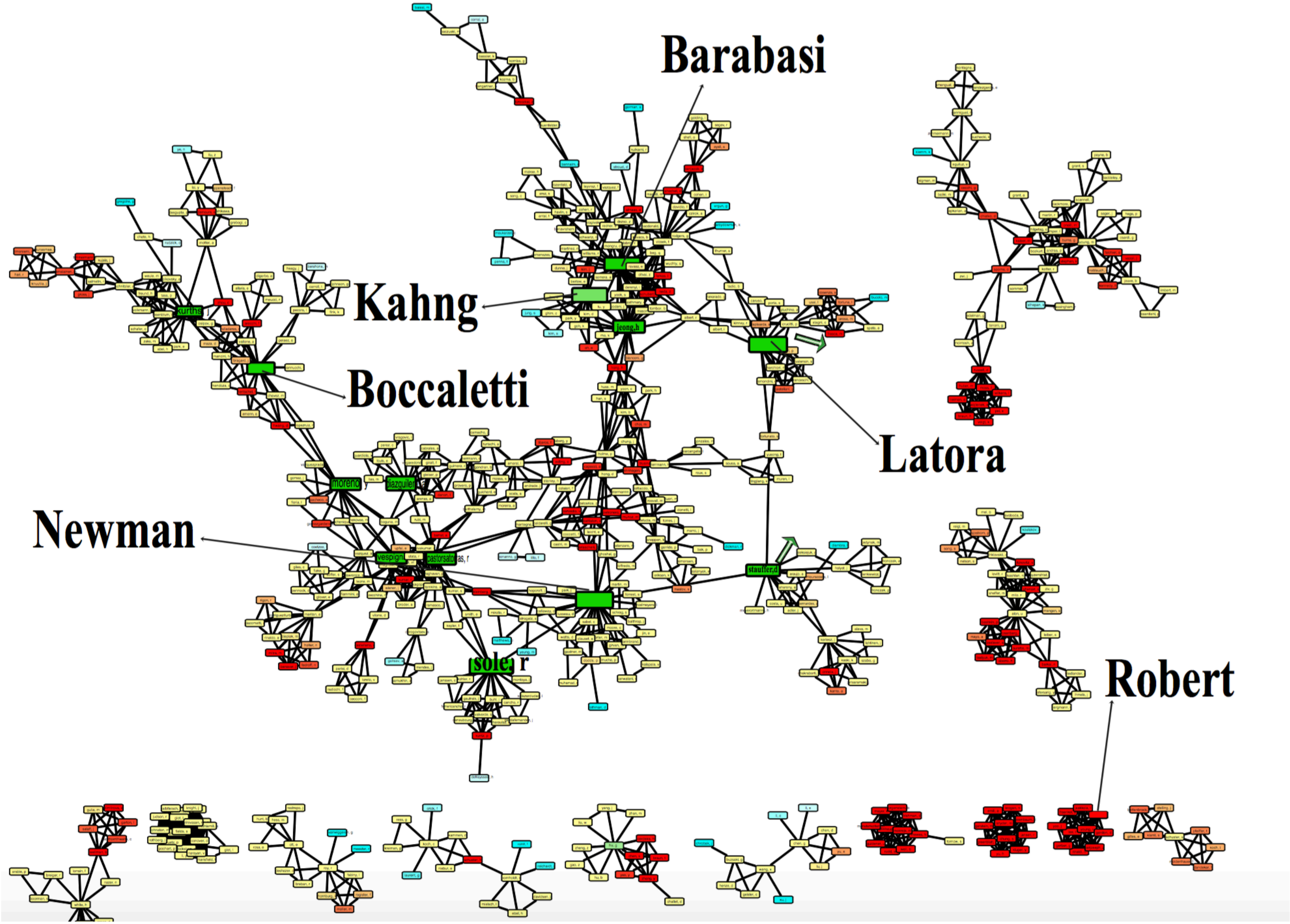
Panther: Fast Top-K Similarity Search on Large Networks

Jing Zhang[1], Jie Tang[1], Cong Ma[1], Hanghang Tong[2], Yu Jing[1], and Juanzi Li[1]

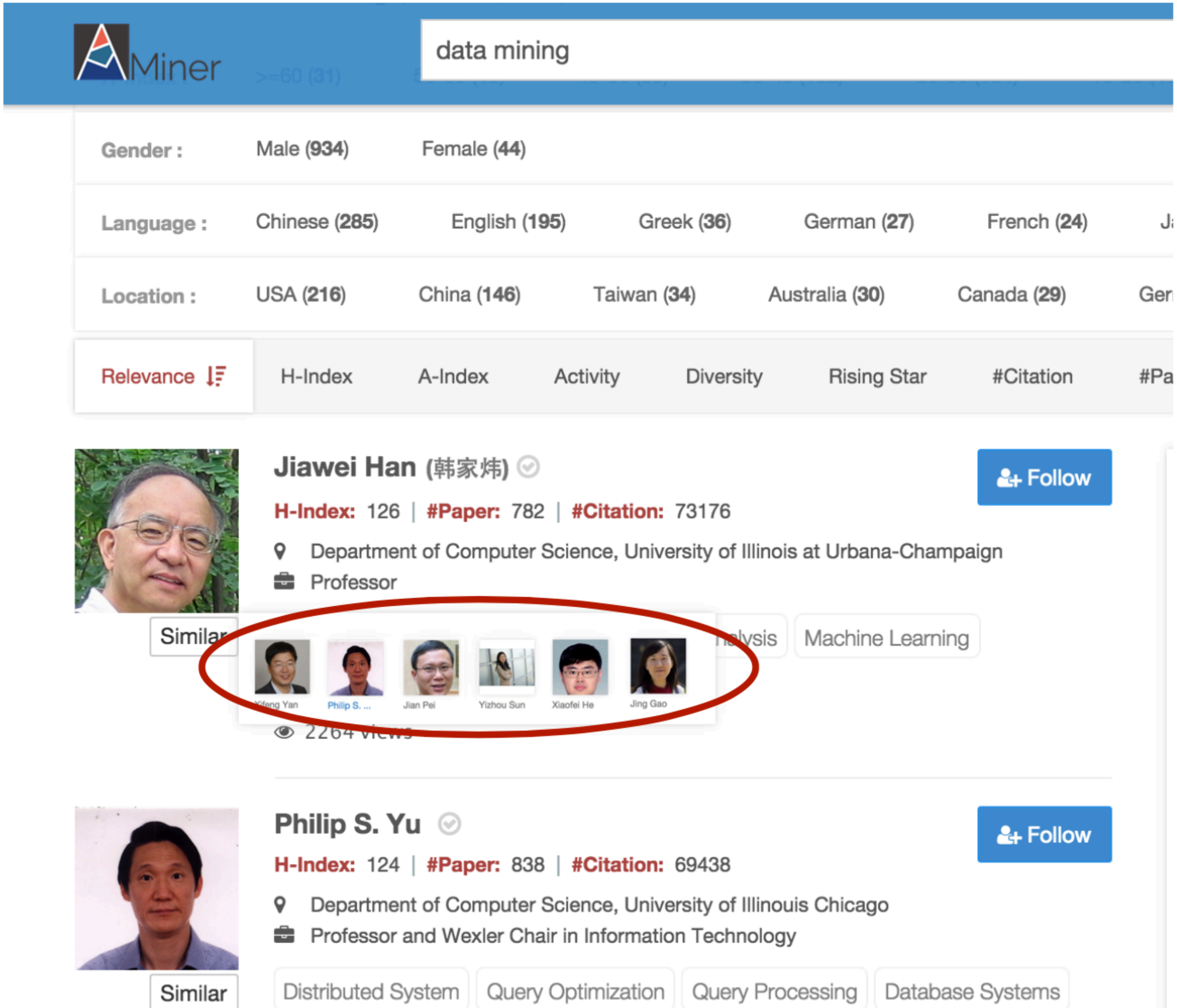
[1]Department of Computer Science and Technology
Tsinghua University

[2]School of Computing, Informatics, and Decision Systems Engineering
Arizona State University

Who are Similar with Barabási?



Similar Authors in Aminer



Related Work and Challenges

1

Share many direct/indirect common neighbors.

2

Disconnected, but share similar structure.

Method	Time Complexity	Space Complexity
SimRank [kdd'02]	O(lN^2d^2)	O(N^2)
TopSim [ICDE'12]	O(NTd^T)	O($N+M$)
RWR [KDD'04]	O(lN^2d)	O(N^2)
RoleSim [KDD'11]	O(lN^2d^2)	O(N^2)
ReFex [KDD'11]	O($N+l(fM+Nf^2)$)	O($N+Mf$)

- Find top-K similar vertices for any vertex in a network
- d: average degree, f: feature number, T: path length

Challenges

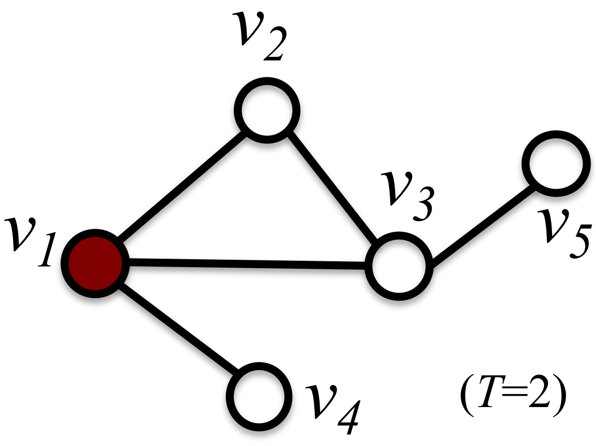
C1 : How to design a similarity method that applies to both similarities?
C2: Computational efficiency challenge.

Our Approach: Panther

1 Path Similarity

Intuition: two vertices are similar if they frequently appear on the same paths.

$$S_{ps}(v_i, v_j) = \frac{\sum_{p \in P(v_i, v_j)} w(p)}{\sum_{p \in \Pi} w(p)}$$



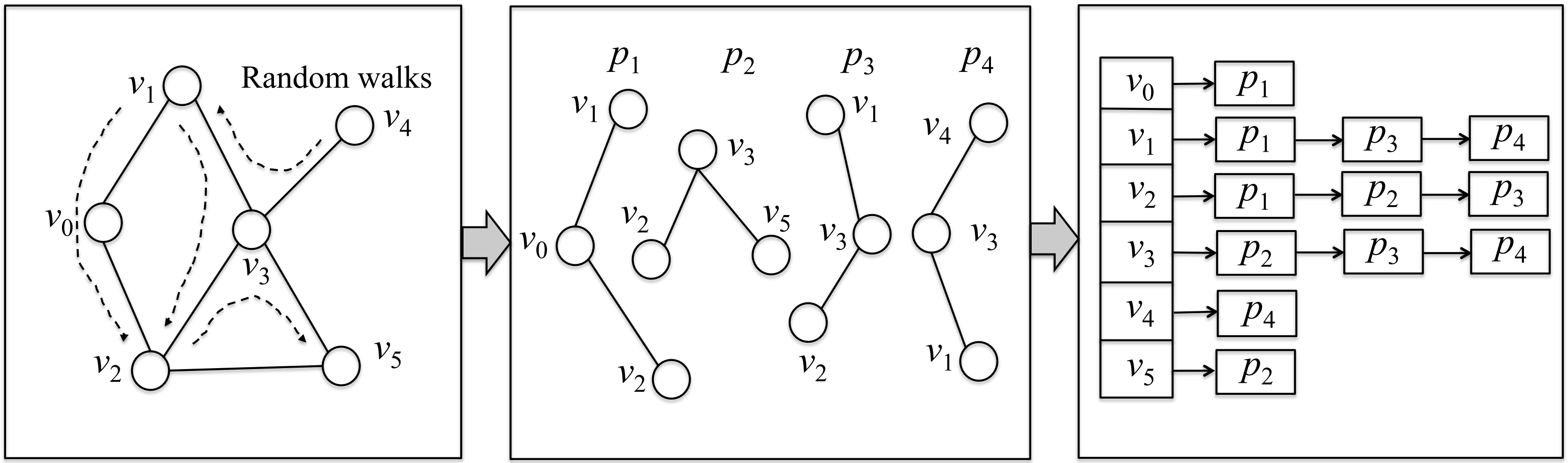
- A path is a T-length sequence of vertices $p = (v_1, \cdots, v_{T+1})$.
- Π is all the T-paths in G.
- Path weight:

$$w(p) = \prod_{i=1, j=i+1}^{T} t_{ij}. \quad t_{ij} = \frac{w_{ij}}{\sum_{v_k \in \mathcal{N}(v_i)} w_{ik}}$$

$S_{ps}(v_1, v_2)=0.37$,
$S_{ps}(v_1, v_3)=0.42$,
$S_{ps}(v_1, v_4)=0.39$,
$S_{ps}(v_1, v_5)=0.09$.

Panther$_{ps}$

Basic idea: random path sampling



(a) Input network (b) Random paths (c) Vertex-to-path index

Simplified path similarity:

$$S_{ps}(v_i, v_j) = \frac{|P(v_i, v_j)|}{R}$$

O(RT) $\Rightarrow$ O(dT)

Theoretical Analysis

- How many random paths shall we sample?

Domain and range set

Upper bound of range set's VC dimension

Distribution

Theorem 1

1 2 3

Let $\mathcal{R}$ be a range set on a domain $\mathcal{D}$, with $VC(\mathcal{R}) \leq d$, and let ϕ be a distribution on $\mathcal{D}$. Given $\varepsilon, \delta \in (0,1)$, let S be a set of $|S|$ points sampled from $\mathcal{D}$ according to ϕ, with

$$|S| = \frac{c}{\varepsilon^2}(d + \ln\frac{1}{\delta}),$$

where c is a universal positive constant. Then S is a ε-approximation to $(\mathcal{R}, \phi)$ with probability of at least $1 - \delta$.

Required sample size

Theoretical Analysis

- Domain: *Π*
- Range set: $\mathcal{R}_G = \{P_{v_i,v_j} : v_i, v_j \in V\}$
- VC bound: $VC(\mathcal{R}_G) \leq \log_2 \binom{T}{2} + 1$
- Distribution: $\phi(p) = \text{prob}(p) = \frac{w(p)}{\sum_{p \in \Pi} w(p)}$
- Path similarity $\frac{\sum_{p \in P_{v_i,v_j}} w(p)}{\sum_{p \in P} w(p)}$ is $\phi(P_{v_i,v_j})$
- Conclusion

$$R = \frac{c}{\varepsilon^2}(\log_2 \binom{T}{2} + 1 + \ln \frac{1}{\delta})$$

  – *R* random paths can guarantee *ε* and 1−*δ*.

Details

Proof of $VC(\mathcal{R}_G) \le \log_2 \binom{T}{2} + 1$

Assume $VC(\mathcal{R}_G) = l$ and $l > \log_2 \binom{T}{2} + 1$

A set Q of size l can be shattered by R_G

A 1-1 corresponding between each subset in Q and each range P_i in R_G

$|\{P_i | p \in P_i \text{ and } P_i \in \mathcal{R}_G\}| = 2^{l-1}$

$\mathcal{R}_G = \{P_{v_i,v_j} : v_i, v_j \in V\}$

A path belongs only to the ranges w.r.t a pair of vertices in the path

$\binom{T}{2} < 2^{l-1}$

$|\{P_i | p \in P_i \text{ and } P_i \in \mathcal{R}_G\}| = \binom{T}{2} < 2^{l-1}$

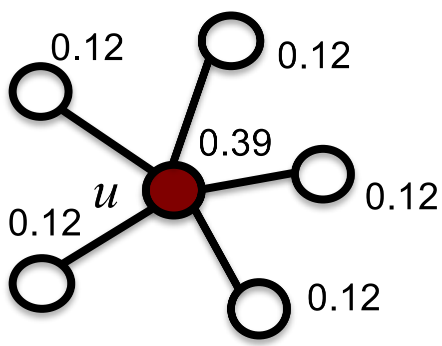
Contradiction

Vector Similarity and Panther$_{vs}$

- Limitation of path similarity: bias to close neighbors.
- **Vector Similarity**: the probability distributions of a vertex linking to all other vertices are similar if their topology structures are similar.
- **Panther$_{vs}$** :Use top-*D* path similarities calculated by Panther$_{ps}$ to represent a vector:

$$\theta(v_i) = (S_{ps}(v_i, v_{(1)}), S_{ps}(v_i, v_{(2)}), \ldots, S_{ps}(v_i, v_{(D)}))$$

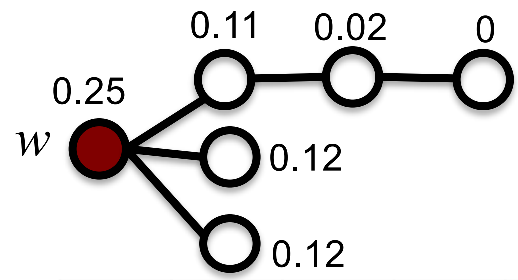
$$S_{vs}(v_i, v_j) = \frac{1}{\|\theta(v_i) - \theta(v_j)\|}$$



(T=2)

0.39	0.12	0.12	0.12	0.12	0.12

0.13	0.13	0.04	0	0	0



0.25	0.12	0.12	0.11	0.02	0

$S_{vs}(u,w)=0.27 > S_{vs}(u,v)=0.16$

Time Complexity

Method	Time Complexity	Space Complexity
SimRank	$O(IN^2d^2)$	$O(N^2)$
TopSim	$O(NTd^T)$	$O(N+M)$
RWR	$O(IN^2d)$	$O(N^2)$
RoleSim	$O(IN^2d^2)$	$O(N^2)$
ReFex	$O(N+I(fM+Nf^2))$	$O(N+Mf)$
Panther$_{ps}$	$O(\boxed{RTc}+\boxed{NdT})$	$O(\boxed{RT}+\boxed{Nd})$
Panther$_{vs}$	$O(RTc+NdT+\boxed{Nc})$	$O(RT+Nd+\boxed{ND})$

Random path

Vertex-to-path index

Kd-tree

Random path sampling

Top-k similarity search for any vertex

Build and query kd-tree

Experiments

Evaluation Aspects

- Efficiency Performance
- Accuracy Performance
- Parameter Sensitivity Analysis

Efficiency Performance

Tencent network

Preprocessing time + top-*k* similarity search time

|V|	|E|	RWR [(KDD'04]	TopSim [ICDE'12]	RoleSim [KDD'11]	ReFex [KDD'11]	$Panther_{ps}$	$Panther_{vs}$
6,523	10,000	+7.79hr	+38.58m	+37.26s	3.85s+0.07s	0.07s+0.26s	0.99s+0.21s
25,844	50,000	+>150hr	+11.20hr	+12.98m	26.09s+0.40s	0.28s+1.53s	2.45s+4.21s
48,837	100,000		+30.94hr	+1.06hr	2.02m+0.57s	0.58s+3.48s	5.30s+5.96s
169,209	500,000		+>120hr	+>72hr	17.18m+2.51s	8.19s+16.08s	27.94s+24.17s
230,103	1,000,000				31.50m+3.29s	15.31s+30.63s	49.83s+22.86s
443,070	5,000,000				24.15hr+8.55s	50.91s+2.82m	4.01m+1.29m
702,049	10,000,000				>48hr	2.21m+6.24m	8.60m+6.58m
2,767,344	50,000,000					15.787m+1.36hr	1.60hr+2.17hr
5,355,507	100,000,000					44.09m+4.50hr	5.61hr+6.47hr
26,033,969	500,000,000					4.82hr+25.01hr	32.90hr+47.34hr
51,640,620	1,000,000,000					13.32hr+80.38hr	98.15hr+120.01hr

Can scale up to handle 1 billion edges

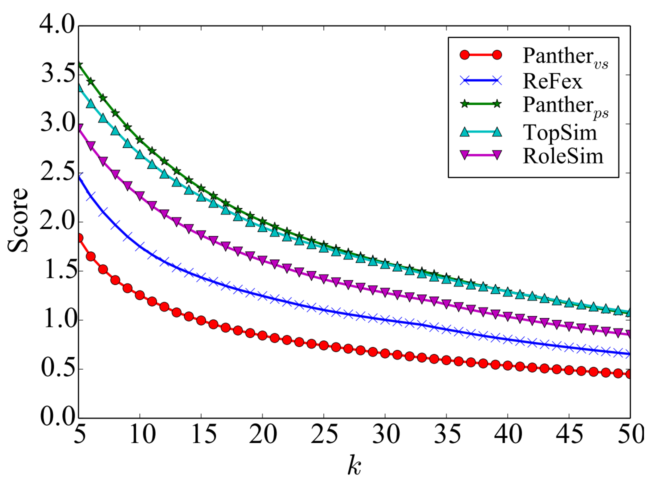
270X speed up

390X speed up

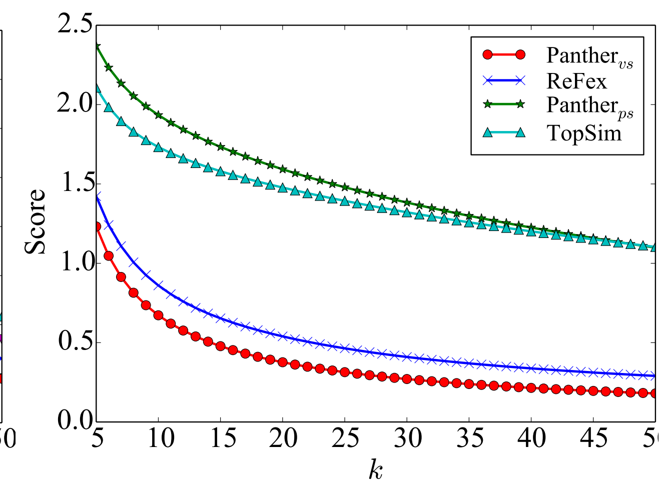
❖ T=5, c=0.5, $\varepsilon=\sqrt{1/|E|}$ and δ=0.1, R=16,609,640

Accuracy Performance of Panther$_{ps}$

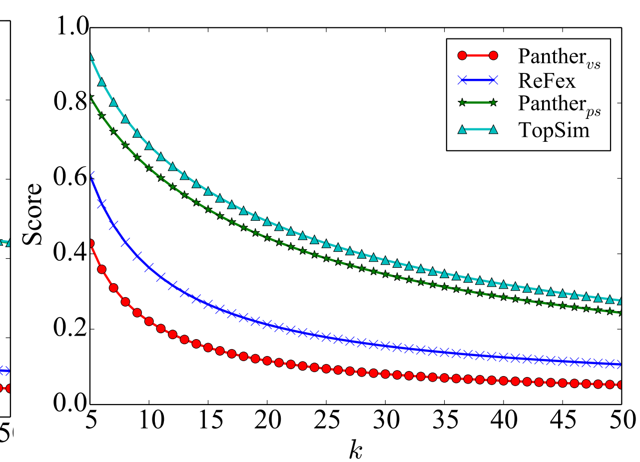
- Evaluate how Panther$_{ps}$ can approximate common neighbors.
- The score represents the improvement over a random method.



KDD



Twitter

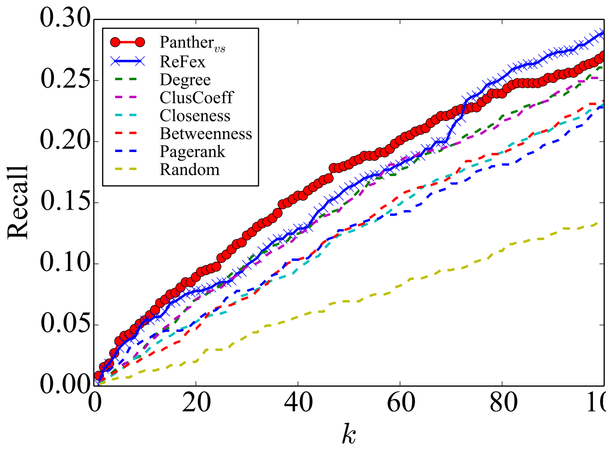


Mobile

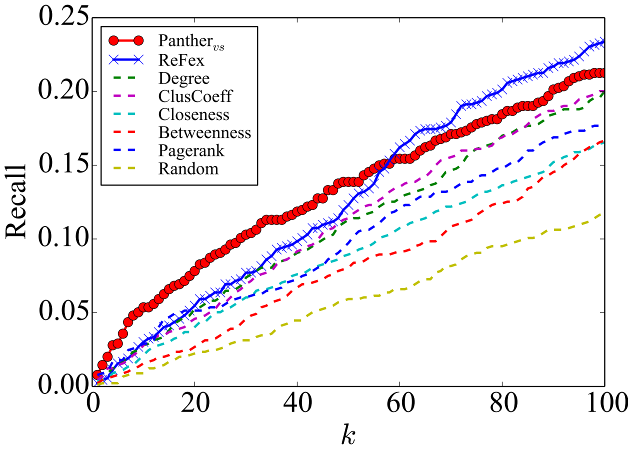
- Co-author networks: |V|=3K, |E| = 7K.
- Twitter network: |V| = 100K, |E| = 500K.
- Mobile network: |V| = 200K, |E| = 200K.

Accuracy Performance of Panther$_{vs}$

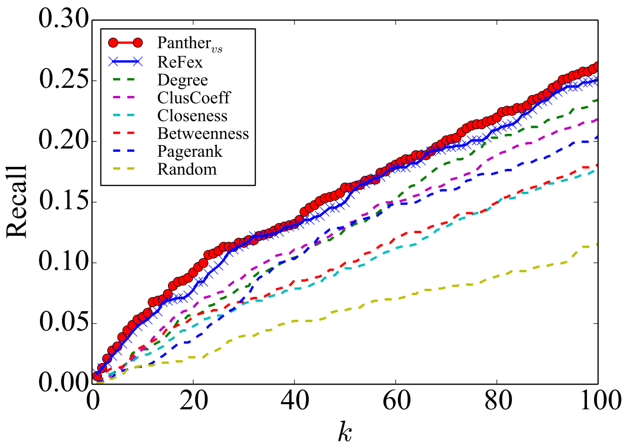
- Identity Resolution
  - Assume the same authors in different networks of the same domain are similar to each other.
- Settings
  - Given any two co-author networks, e.g., KDD and ICDM, if the top-*k* similar vertices from ICDM consists of the query author from KDD, we say that the method hits a correct instance.



KDD-ICDM



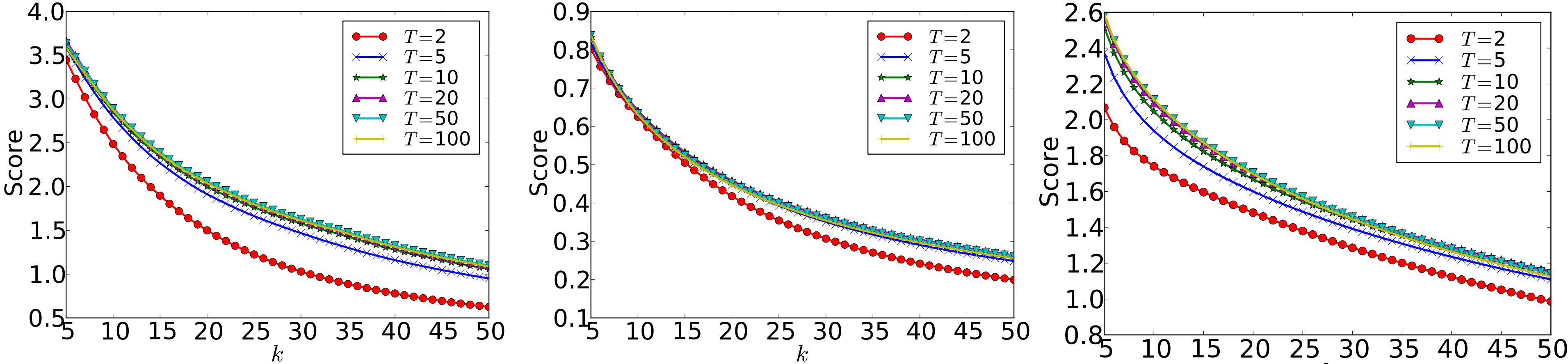
SIGIR-CIKM



SIGMOD-ICDE

Parameter Analysis: Path Length T

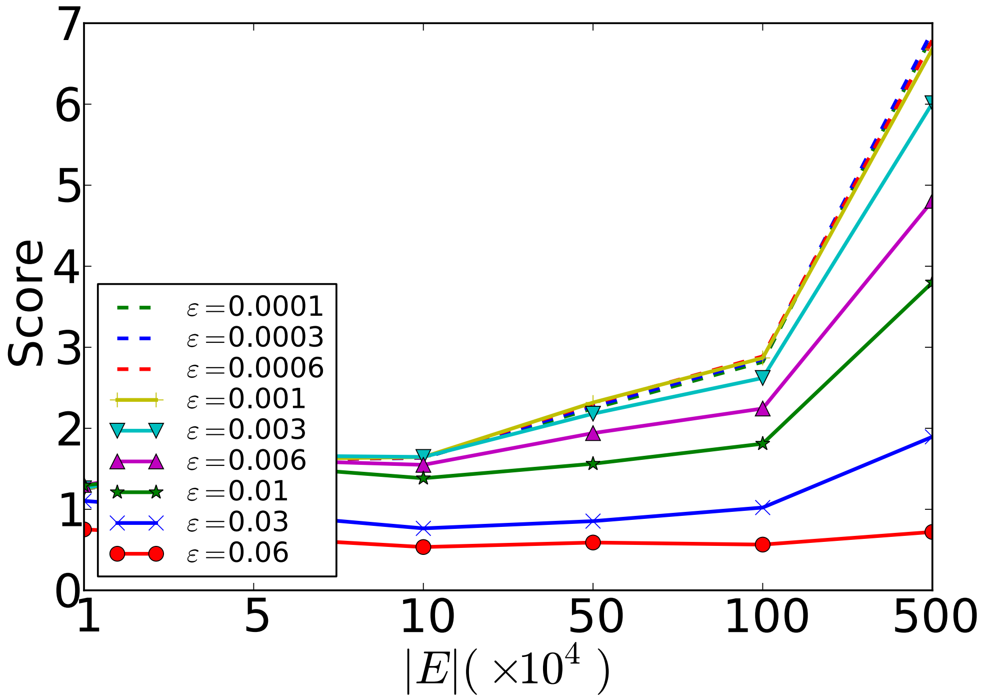
- The performance gets better when T increases.
- The performance almost becomes stable When $T \geq 5$.



KDD Mobile Twitter

Effect of path length T on the accuracy performance of Panther$_{ps}$.

Parameter Analysis: Error Bound ε



$$R = \frac{c}{\varepsilon^2}(\log_2 \binom{T}{2} + 1 + \ln \frac{1}{\delta})$$

- ❖ Tencent sub networks
- ❖ Panther$_{ps}$

- When $|E|/(1/\varepsilon)^2$ ranges from 5 to 20, scores of Panther$_{ps}$ are almost convergent.
- The value $(1/\varepsilon)^2$ is almost linearly positively correlated with the number of edges in a network.

Conclusion

- Methods:
  - Solve two similarity metrics efficiently.

- Theoretic analysis:
  - Sampling size is only related to path length given error-bound and confidence level.

- Empirical evaluations:
  - When |V| = 0.5 million and |E|=5 million, $Panther_{ps}$ achieves a 390× speed-up and $Panther_{vs}$ achieves a 270x speed-up.
  - Panther can scale up to a network with 1 billion edges.

Thank You

Code & Data:
http://aminer.org/Panther