ASAP: Towards Accurate, Stable and Accelerative Penetrating-Rank Estimation on Large Graphs

Xuefei Li[1], Weiren Yu[2], Bo Yang[3], Jiajin Le[3]

[1] Fudan University
[2] University of New South Wales
[3] Donghua University

Presented by Weiren Yu

Roadmap

1. P-Rank Overview
2. Accuracy Estimate
3. Stability Analysis
4. Algorithm on Undirected Graphs
5. Empirical Evaluation

P-Rank Overview

- Information Network (IN)
  - Physical / Conceptual entities → vertices
  - Interconnected relationships → edges
- INs form a critical component of modern information infrastructure
  - highway or urban transportation networks
  - research collaboration and publication networks
  - Biological networks
  - social networks

P-Rank Overview (cont.)

- P(enetrating)-Rank similarity
  - A new promising structural measure (CIKM'09)
  - An extension of SimRank metrics
- Basic Philosophy
  - Two entities are similar, if
    - they are referenced by similar entities
    - they reference similar entities
- Mathematical Formula

$$s(u,u)=1;$$

$$s(u,v)=\underbrace{\frac{\lambda \cdot C_{\text{in}}}{|\mathcal{I}(u)||\mathcal{I}(v)|}\sum_{i=1}^{|\mathcal{I}(u)|}\sum_{j=1}^{|\mathcal{I}(v)|} s(\mathcal{I}_i(u),\mathcal{I}_j(v))}_{\text{in-link part}}+\underbrace{\frac{(1-\lambda)\cdot C_{\text{out}}}{|\mathcal{O}(u)||\mathcal{O}(v)|}\sum_{i=1}^{|\mathcal{O}(u)|}\sum_{j=1}^{|\mathcal{O}(v)|} s(\mathcal{O}_i(u),\mathcal{O}_j(v))}_{\text{out-link part}}.$$

P-Rank Overview (cont.)

- P-Rank Computation
  - Naïve way: a fixed-point iterative paradigm

$$s^{(k+1)}(u,u) = 1.$$

$$s^{(k+1)}(u,v) = \frac{\lambda \cdot C_{\text{in}}}{|\mathcal{I}(u)||\mathcal{I}(v)|} \sum_{i=1}^{|\mathcal{I}(u)|} \sum_{j=1}^{|\mathcal{I}(v)|} s^{(k)}\left(\mathcal{I}_i(u), \mathcal{I}_j(v)\right) + \frac{(1-\lambda) \cdot C_{\text{out}}}{|\mathcal{O}(u)||\mathcal{O}(v)|} \sum_{i=1}^{|\mathcal{O}(u)|} \sum_{j=1}^{|\mathcal{O}(v)|} s^{(k)}\left(\mathcal{O}_i(u), \mathcal{O}_j(v)\right).$$

- Iterative P-Rank Properties
  - Symmetry: $s^{(k)}(a,b) = s^{(k)}(b,a)$
  - Monotonicity: $0 \le s^{(k)}(a,b) \le s^{(k+1)}(a,b) \le 1$
  - Existence & Uniqueness (0<c<1)

$$\lim_{k\to\infty} s^{(k)}(u,v) = \sup_{k\ge 0}\left\{s^{(k)}(u,v)\right\} = s(u,v)$$

Motivations

- Despite the convergence of P-Rank iteration, a precise P-Rank accuracy estimation is not provided.
- P-Rank condition number is not studied, which can measure how much networks may change in proportion to small perturbation in P-Rank scoring results.
- No efficient algorithm is designed specially for computing P-Rank on undirected graphs.

Contributions

- We provide an accuracy estimation of the P-Rank convergence rate with a prescribed iterative error in the fixed number of iterations.
- We show that P-Rank is well-conditioned for small choices of the damping factors, by providing a tight stability bound for κ_∞.
- We propose a novel non-iterative $O(n^3)$-time algorithm (ASAP) for efficiently computing similarities over undirected graphs.

Roadmap

1. P-Rank Overview
2. **Accuracy Estimate**
3. Stability Analysis
4. Algorithm on Undirected Graphs
5. Empirical Evaluation

P-Rank accuracy estimation

- P-Rank iterative paradigm:

$$s^{(k+1)}(u,u) = 1.$$

$$s^{(k+1)}(u,v) = \frac{\lambda \cdot C_{\text{in}}}{|\mathcal{I}(u)||\mathcal{I}(v)|} \sum_{i=1}^{|\mathcal{I}(u)|} \sum_{j=1}^{|\mathcal{I}(v)|} s^{(k)}(\mathcal{I}_i(u), \mathcal{I}_j(v)) + \frac{(1-\lambda) \cdot C_{\text{out}}}{|\mathcal{O}(u)||\mathcal{O}(v)|} \sum_{i=1}^{|\mathcal{O}(u)|} \sum_{j=1}^{|\mathcal{O}(v)|} s^{(k)}(\mathcal{O}_i(u), \mathcal{O}_j(v)).$$

$$\lim_{k\to\infty} s^{(k)}(u,v) = \sup_{k\geq 0} \{s^{(k)}(u,v)\} = s(u,v)$$

- P-Rank accuracy estimate problem:

Given a network G, for each iteration k = 1, 2, … ,

it is to find an upper bound ϵ_k s.t.

$$|s^{(k)}(u, v) - s(u, v)| \leq \epsilon_k$$

for any vertices u and v in G.

P-Rank accuracy estimation

- Theorem 1. The P-Rank accuracy estimate problem has a tight upper bound

$$\epsilon_k = (\lambda C_{in} + (1 - \lambda) C_{out})^{k+1}$$

such that $\forall\ k=0,1,\ldots,\ \forall\ u, v \in V$

$$|s^{(k)}(u, v) - s(u,v)| \le \epsilon_k.$$

- Theorem 1 provides an a-priori estimate for the gap between iterative and exact P-Rank similarity:

$$k = \lceil \log \epsilon / \log (\lambda \cdot C_{in} + (1-\lambda) \cdot C_{out}) \rceil$$

P-Rank accuracy estimation

- Example:

  Setting $C_{in} = 0.6, C_{out} = 0.4$, $\lambda = 0.3$, $k = 5$ produces the high accuracy :

  $$\epsilon_k = (0.3 \times 0.6 + (1-0.3) \times 0.4)^{5+1} = 0.0095.$$

- The "=" in Theorem 1 can be attainable :

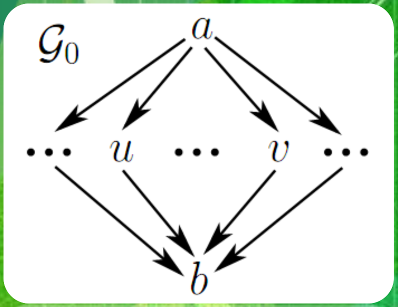
  $$s^{(0)}(u,v) = 0,$$

  $\forall$ k=1,2…

  $$s^{(k)}(u,v) = \lambda C_{in} + (1-\lambda) C_{out}.$$

  Hence, for k=0,

  $$|s(u,v) - s^{(k)}(u,v)| = (\lambda C_{in} + (1-\lambda) C_{out})^{0+1}$$



Roadmap

1. P-Rank Overview
2. Accuracy Estimate
3. **Stability Analysis**
4. Algorithm on Undirected Graphs
5. Empirical Evaluation

Stability Analysis of P-Rank

- P-Rank stability:
  - how the slight perturbation of the network affects P-Rank similarity scores s(·, ·).

- P-Rank Matrix Representation

$$q_{i,j} \triangleq \begin{cases} a_{j,i} / \sum_{j=1}^{n} a_{j,i}, & \text{if } \mathcal{I}(i) \neq \varnothing; \\ 0, & \text{if } \mathcal{I}(i) = \varnothing. \end{cases}$$

$$p_{i,j} \triangleq \begin{cases} a_{i,j} / \sum_{j=1}^{n} a_{i,j}, & \text{if } \mathcal{O}(i) \neq \varnothing; \\ 0, & \text{if } \mathcal{O}(i) = \varnothing. \end{cases}$$

$$\mathbf{S} = \lambda C_{\text{in}} \cdot \mathbf{Q} \cdot \mathbf{S} \cdot \mathbf{Q}^T + (1-\lambda) C_{\text{out}} \cdot \mathbf{P} \cdot \mathbf{S} \cdot \mathbf{P}^T + (1 - \lambda C_{\text{in}} - (1-\lambda) C_{\text{out}}) \cdot \mathbf{I}_n,$$

$$\underbrace{(\mathbf{I}_{n^2} - \lambda C_{\text{in}} (\mathbf{Q} \otimes \mathbf{Q}) - (1-\lambda) C_{\text{out}} (\mathbf{P} \otimes \mathbf{P}))}_{\triangleq \mathbf{M}} \cdot \underbrace{\text{vec}(\mathbf{S})}_{\triangleq \mathbf{s}} = \underbrace{\text{vec}(\mathbf{I}_n)}_{\triangleq \mathbf{b}}.$$

Stability Analysis of P-Rank

- P-Rank conditional number :

  Let

$$\mathbf{M} \triangleq \mathbf{I}_{n^2} - \lambda C_{in}(\mathbf{Q} \otimes \mathbf{Q}) - (1-\lambda) C_{out}(\mathbf{P} \otimes \mathbf{P}).$$

  P-Rank conditional number of G is defined as

$$\kappa_\infty(\mathcal{G}) \triangleq \|\mathbf{M}\|_\infty \cdot \|\mathbf{M}^{-1}\|_\infty$$

- κ_∞(G) measures how stable the P-Rank similarity score is to the changes in the link structure of the network G.

  (e.g., inserting or deleting vertices or edges)

Stability Analysis of P-Rank

- Theorem 2. Given a network G, ∀ λ∈[0,1] and ∀ C_{in}, C_{out} ∈ (0,1), P-Rank conditional number has the following tight bound:

$$\kappa_\infty(\mathcal{G}) \le \frac{1 + \lambda \cdot C_{in} + (1-\lambda) \cdot C_{out}}{1 - \lambda \cdot C_{in} - (1-\lambda) \cdot C_{out}}.$$

- Small choices of κ_∞ (G) would make P-Rank stable (well-conditioned).

(i.e., a small change ΔM in link structure to M may not cause a large change Δs in P-Rank scores).

$$\frac{\|\Delta \mathbf{s}\|_\infty}{\|\mathbf{s}\|_\infty} \le \kappa_\infty(\mathcal{G}) \cdot \frac{\|\Delta \mathbf{M}\|_\infty}{\|\mathbf{M}\|_\infty}$$

Stability Analysis of P-Rank

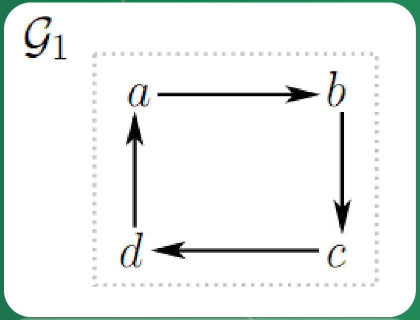
- The weighting factor λ affects κ_∞ (G) as follows:

$$\frac{\partial}{\partial \lambda}\left(\frac{1+\lambda \cdot C_{\text{in}}+(1-\lambda) \cdot C_{\text{out}}}{1-\lambda \cdot C_{\text{in}}-(1-\lambda) \cdot C_{\text{out}}}\right)=\frac{2\left(C_{\text{in}}-C_{\text{out}}\right)}{\left(1-\lambda \cdot C_{\text{in}}-(1-\lambda) \cdot C_{\text{out}}\right)^{2}},$$

  - when $C_{in} > C_{out}$ and λ ↗ ,
    a small change in G produces a large change in P-Rank, which makes P-Rank ill-conditioned.
  - when $C_{in} < C_{out}$ and λ ↗ ,
    a small change in G produces a small change in P-Rank, which makes P-Rank well-conditioned.
  - when $C_{in} = C_{ou}$, κ_∞ (G) is independent of λ.

Stability Analysis of P-Rank

- The upper bound of $\kappa_\infty(G)$ is attainable iif each vertex in G has at least one in-degree and one out-degree.



Example:

$$\kappa_\infty(G) = \| M \|_\infty \cdot \| M^{-1} \|_\infty = 1.7 \times 3.333 = 5.667;$$

$$\frac{1 + \lambda \cdot C_{\text{in}} + (1-\lambda) \cdot C_{\text{out}}}{1 - \lambda \cdot C_{\text{in}} - (1-\lambda) \cdot C_{\text{out}}} = \frac{1 + 0.5 \times 0.8 + (1 - 0.5) \times 0.6}{1 - 0.5 \times 0.8 - (1 - 0.5) \times 0.6} \doteq 5.667.$$

Roadmap

1. P-Rank Overview
2. Accuracy Estimate
3. Stability Analysis
4. **Algorithm on Undirected Graphs**
5. Empirical Evaluation

Estimating P-Rank On Undirected Graphs

- Theorem 3. For undirected networks, the P-Rank similarity problem

$$\mathbf{S} = \lambda C_{\text{in}} \cdot \mathbf{Q} \cdot \mathbf{S} \cdot \mathbf{Q}^T + (1-\lambda) C_{\text{out}} \cdot \mathbf{P} \cdot \mathbf{S} \cdot \mathbf{P}^T + (1 - \lambda C_{\text{in}} - (1-\lambda) C_{\text{out}}) \cdot \mathbf{I}_n,$$

can be solvable in $O(n^3)$ worst-case time.

Comparison:

- $O(Kn^4)$ time [CIKM 09'] via naive iterative fashion
- $O(Kn^3)$ time [EDBT 10'] via matrix iteration
- $O(n^3)$ time [this work] via non-iterative paradigm

Estimating P-Rank On Undirected Graphs

- The key idea in our optimization is to maximally use the adjacency matrix A :
  - characterizing S as a power series form

$$\mathbf{S} = \sum_{k=0}^{+\infty} f(\mathbf{A}^k)$$

$A = A^T$ for undirected graphs, implying ∃ D s.t.

$$Q = P = D \cdot A$$

  - diagonalizing A into Λ to compute A^k

Hence, calculating $f(A^k)$ reduces to computing the function on each eigenvalue for A.

Estimating P-Rank On Undirected Graphs

- Proposition. For the undirected network G with n vertices, let

$$\mathbf{D} = diag((\sum_{j=1}^{n} a_{1,j})^{-1}, \cdots, (\sum_{j=1}^{n} a_{n,j})^{-1}),$$

and

$$[U, \Lambda] = \text{eig}\,(D^{1/2}AD^{1/2})$$

Then, S′ can be computed as

$$\mathbf{S}' = \mathbf{D}^{1/2}\mathbf{U} \cdot \Psi \cdot \mathbf{U}^{T}\mathbf{D}^{1/2},$$

where

$$\Psi = (\Psi_{i,j})_{n \times n} = \left(\frac{[\mathbf{U}^{T}\mathbf{D}^{-1}\mathbf{U}]_{i,j}}{1 - (\lambda \cdot C_{in} + (1-\lambda) \cdot C_{out})\, \Lambda_{i,i}\Lambda_{j,j}} \right)_{n \times n}$$

Estimating P-Rank On Undirected Graphs

Algorithm 1: ASAP $(\mathcal{G}, \lambda, C_{\text{in}}, C_{\text{out}})$

Input : a labeled undirected network $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{A}; l)$, the weighting factor λ, and in- and out-link damping factors C_{in} and C_{out}.

Output: similarity matrix $\mathbf{S} = (s_{i,j})_{n \times n}$ with $s_{i,j}$ denoting P-Rank score between vertices i and j.

initialize the adjacency matrix $\mathbf{A}$ of $\mathcal{G}$; O(n²)

compute the diagonal matrix $\mathbf{D} = diag(d_{1,1}, d_{2,2}, \cdots, d_{n,n})$ O(m)
with its entry $d_{i,i} = (\sum_{j=1}^{n} a_{i,j})^{-1}$, if $\sum_{j=1}^{n} a_{i,j} \neq 0$; and $d_{i,i} = 0$, otherwise;

compute the auxiliary matrix $\mathbf{T} = \mathbf{D}^{1/2} \cdot \mathbf{A} \cdot \mathbf{D}^{1/2}$ O(n²)

decompose $\mathbf{T}$ into the diagonal matrix $\mathbf{\Lambda} = diag(\Lambda_{1,1}, \Lambda_{2,2}, \cdots, \Lambda_{n,n})$ and the orthogonal $\mathbf{U}$ via QR factorization *s.t.* $\mathbf{T} = \mathbf{U} \cdot \mathbf{\Lambda} \cdot \mathbf{U}^T$; O(n³)

compute the auxiliary matrix $\mathbf{\Gamma} = (\Gamma_{i,j})_{n \times n} = \mathbf{U}^T \cdot \mathbf{D}^{-1} \cdot \mathbf{U}$ and $\mathbf{V} = \mathbf{D}^{1/2} \cdot \mathbf{U}$ O(n³+n²)
and the constant $C = \lambda C_{\text{in}} + (1 - \lambda) C_{\text{out}}$;

compute the matrix $\mathbf{\Psi} = (\psi_{i,j})_{n \times n}$ whose entry $\psi_{i,j} = \Gamma_{i,j} / (1 - C \cdot \Lambda_{i,i} \cdot \Lambda_{j,j})$; O(n²)

compute the P-Rank similarity matrix $\mathbf{S} = (1 - C) \cdot \mathbf{V} \cdot \mathbf{\Psi} \cdot \mathbf{V}^T$; O(n³)

return $\mathbf{S}$;

The total time complexity of ASAP is bounded by $O(n^3)$.

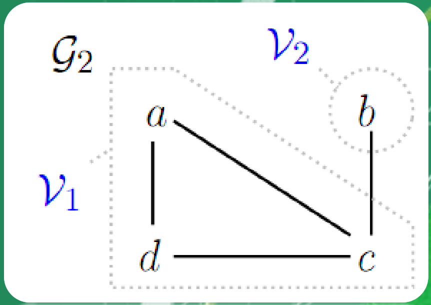
Estimating P-Rank On Undirected Graphs

- Running Example for ASAP:

Consider an undirected G_2 with

vertex set $V = V_1 \cup V_2 = \{a, c, d\} \cup \{b\}$

edge set $E = \{(a, c), (a, d), (c, d), (b, c)\}$.



$$\overset{①}{\Longrightarrow} \mathbf{A} = \begin{pmatrix} 0&0&1&1\\ 0&0&1&0\\ 1&1&0&1\\ 1&0&1&0 \end{pmatrix} \overset{②}{\Longrightarrow} \mathbf{D} = \{\text{Using (16)}\} = \begin{pmatrix} .5&&&0\\ &1&&\\ &&.333&\\ 0&&&.5 \end{pmatrix} \overset{③}{\Longrightarrow} \mathbf{Q} = \mathbf{P} = \mathbf{DA} = \begin{pmatrix} 0&0&.5&.5\\ 0&0&1&0\\ .333&.333&0&.333\\ .5&0&.5&0 \end{pmatrix}$$

$$\overset{④}{\Longrightarrow} \mathbf{\Lambda} = \text{eigval}(\mathbf{D}^{1/2}\mathbf{A}\mathbf{D}^{1/2}) = \begin{pmatrix} -.729&&&0\\ &-.5&&\\ &&.229&\\ 0&&&1 \end{pmatrix} \quad \mathbf{U} = \text{eigvec}(\mathbf{D}^{1/2}\mathbf{A}\mathbf{D}^{1/2}) = \begin{pmatrix} -.244&.707&.436&.5\\ -.583&0&-.732&.354\\ .736&0&-.290&.612\\ -.244&-.707&.436&.5 \end{pmatrix}$$

$$\overset{⑤}{\Longrightarrow} \mathbf{\Gamma} \triangleq \mathbf{U}^T\mathbf{D}^{-1}\mathbf{U} = \begin{pmatrix} 2.201&0&-.640&.656\\ 0&2&0&0\\ -.640&0&1.549&.081\\ .656&0&.081&2.25 \end{pmatrix} \overset{⑥}{\Longrightarrow} \mathbf{\Psi} = \{\text{Using Eq.(18)}\} = \begin{pmatrix} 3.231&0&-.582&.457\\ 0&2.353&0&0\\ -.582&0&1.599&.094\\ .457&0&.094&5.625 \end{pmatrix}$$

$$\overset{⑦}{\Longrightarrow} \mathbf{S} = \{\text{Using Eq.(17)}\} = \begin{pmatrix} .627&.225&.134&.156\\ .225&.770&.067&.225\\ .134&.067&.615&.134\\ .156&.225&.134&.627 \end{pmatrix}$$

Roadmap

1. P-Rank Overview
2. Accuracy Estimate
3. Stability Analysis
4. Algorithm on Undirected Graphs
5. **Empirical Evaluation**

Experimental Evaluation

- Dataset

DBLP Data	1998-1999	1998-2001	1998-2003	1998-2005	1998-2007
n	1,525	3,208	5,307	7,984	10,682
m	5,929	13,441	24,762	39,399	54,844

Real-life.

DBLP (co-authorships among scientists from 1998 to 2007)

The papers published on 6 conferences are picked up

(“ICDE”,“VLDB”, “SIGMOD”,“WWW”, “SIGIR”, “KDD”).

Synthetic.

Using a C++ boost generator to produce graphs

with vertices ranging from 100K to 1M

and edges being randomly chosen

Algorithms.

(i) Iter: conventional P-Rank algorithm [CIKM ’09]

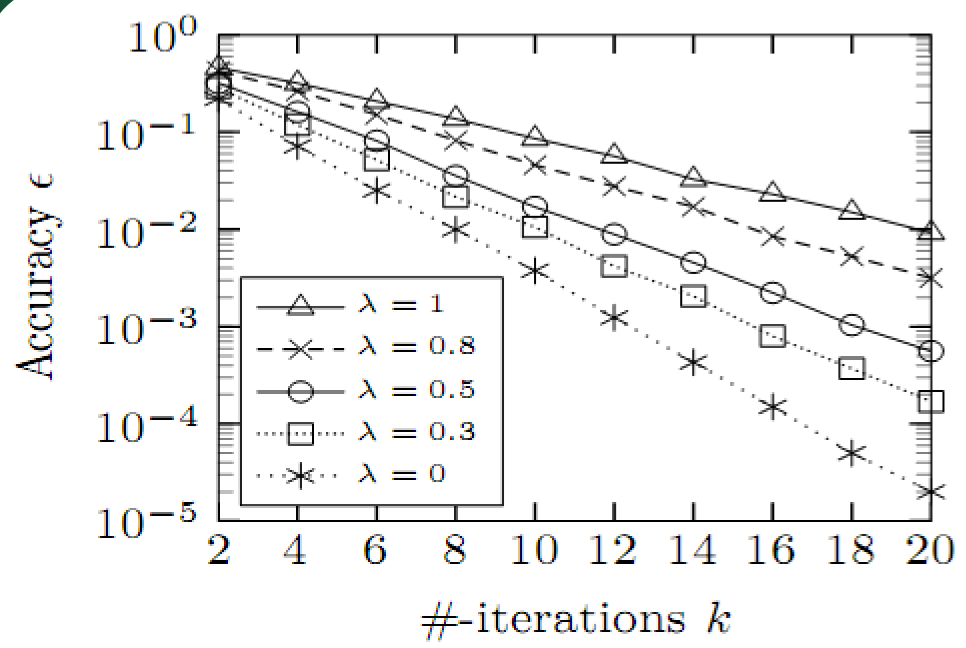
with the radius-based pruning technique

(ii)Memo: the memoization-based algorithm [VLDB J. ’10]

(iii)AUG : SimRank algorithm [WAIM ’10] on undirected graphs.

Experimental Results

- P-Rank Accuracy

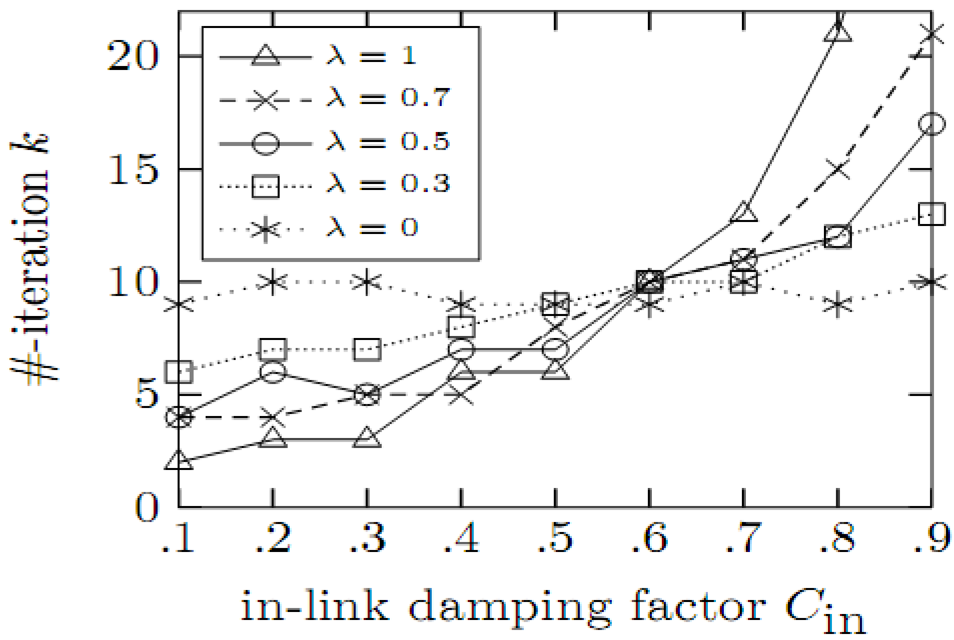


(a) #-iteration k *w.r.t.* accuracy ϵ

For each fixed λ, the downward lines for P-Rank iterations reveal an exponential accuracy as k increases, as expected in Theorem 1.

Experimental Results

- P-Rank Accuracy

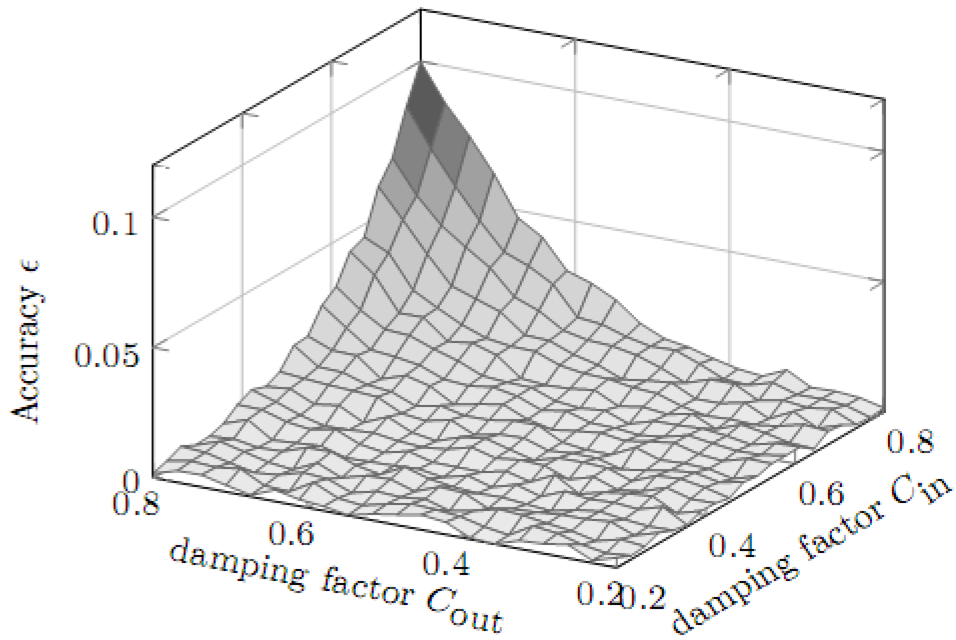


(b) damping factor C_{in} *w.r.t.* k

When $0 < \lambda \leq 1$, k shows a general increased tendency as C_{in} is growing. This tells us that small choices of damping factors may reduce the number of iterations required for a fixed accuracy.

Experimental Results

- P-Rank Accuracy

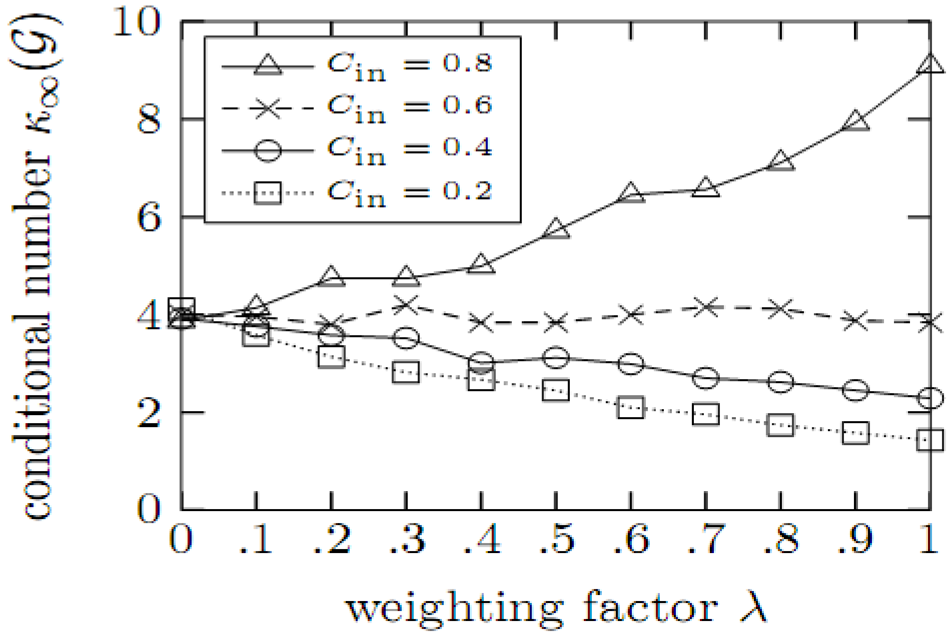


(c) damping factors C_{in}, C_{out} *w.r.t.* ϵ

The residual becomes huge only when C_{in} and C_{out} are both increasing to 1; and the iterative P-Rank is accurate when C_{in} and C_{out} are less than 0.6. This explains why small choices of damping factors are suggested in P-Rank iteration.

Experimental Results

- P-Rank Stability

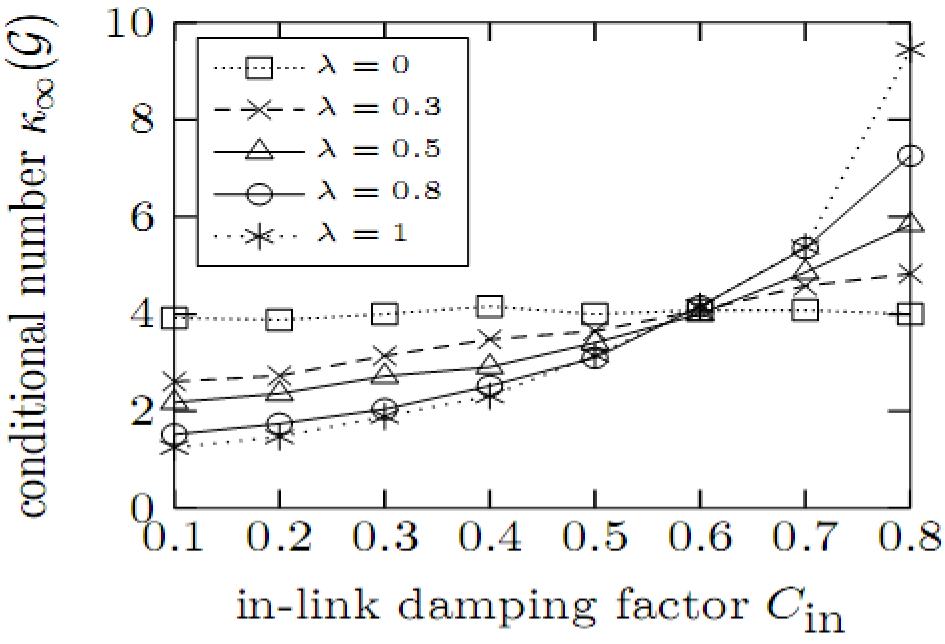


(a) weighting factor λ *w.r.t.* κ_∞

Increasing λ induces a large P-Rank conditional number when $C_{in} > 0.6$.
When $C_{in} < 0.6$, $\kappa_\infty(G)$ is decreased as λ grows.

Experimental Results

- P-Rank Stability

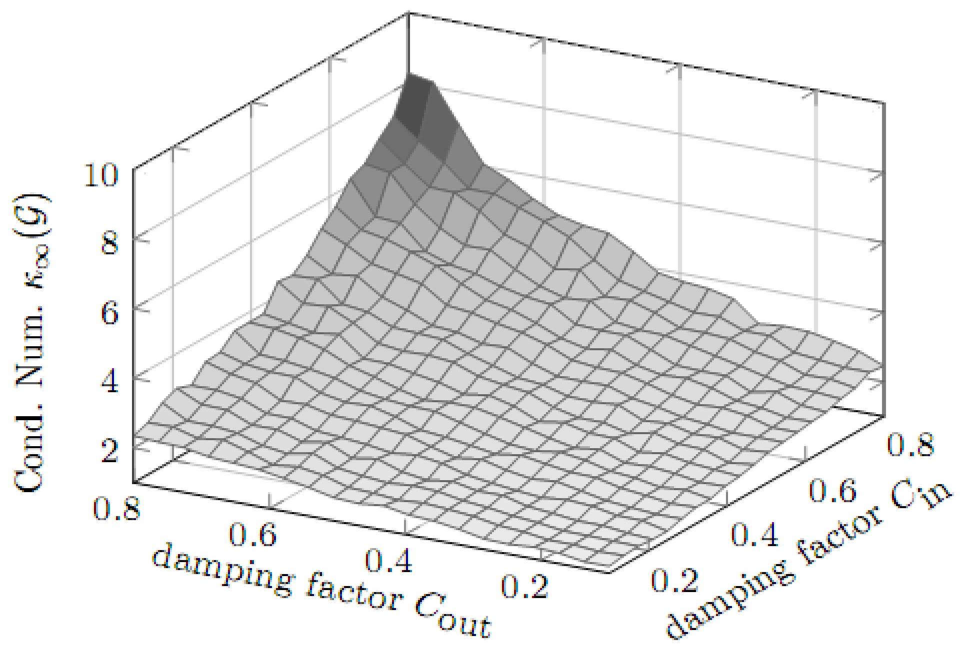


(b) damping factor C_{in} *w.r.t.* κ_∞

When λ = 0, the curve approaches to a horizontal line. These indicate that varying C_{in} as λ = 0 has no effect on the stability κ_∞ of P-Rank, for in this case only the contribution of out-links is considered for computing P-Rank similarity.

Experimental Results

- P-Rank Stability

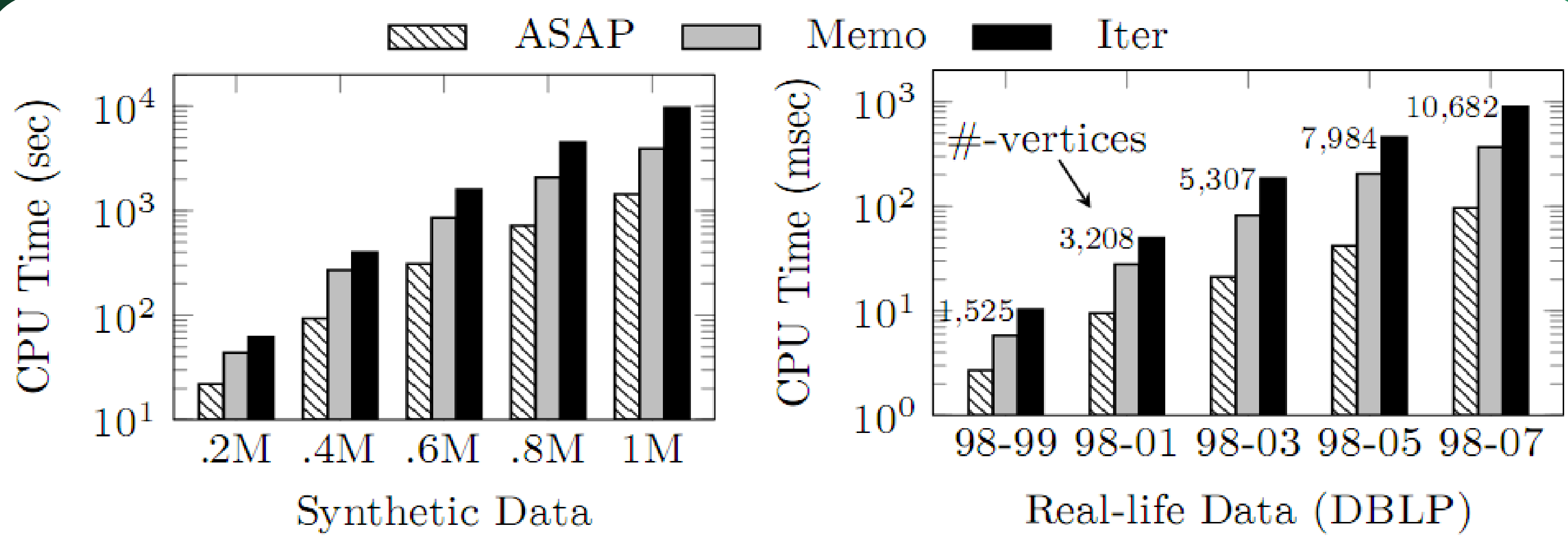


(c) damping factors C_{in}, C_{out} w.r.t. κ_∞

The result demonstrates that P-Rank is comparatively stable when both C_{in} and C_{out} are small (less than 0.6). When C_{in} and C_{out} → 1, P-Rank is ill-conditioned since small perturbations in similarity computation may cause P-Rank scores drastically altered.

Experimental Results

- P-Rank Time Efficiency

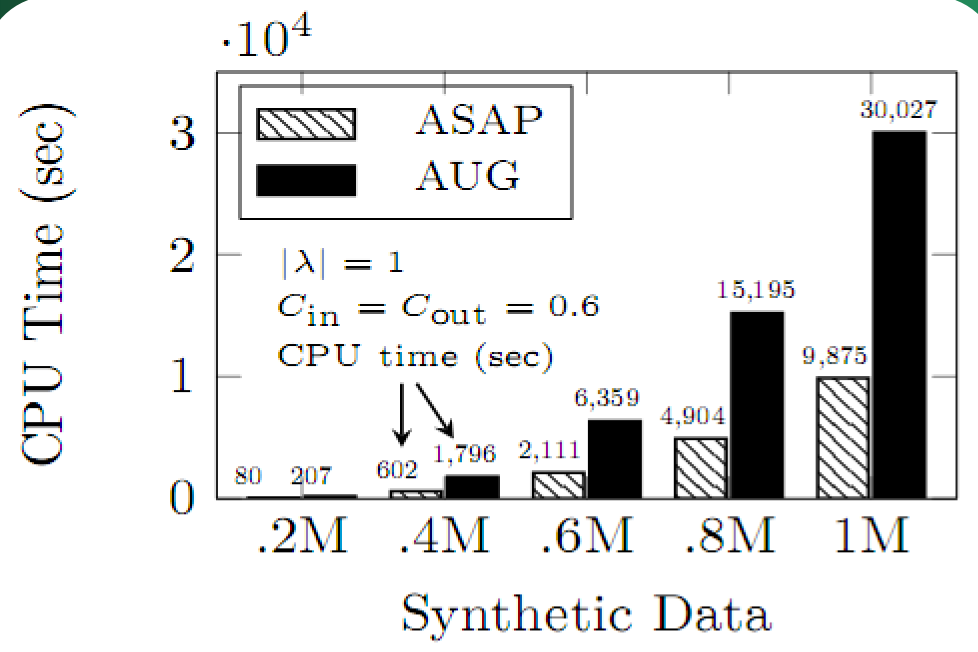


(d) #-vertices n *w.r.t.* CPU time over synthetic and real-life data

In all cases, ASAP performed the best, by taking advantage of its non-iterative paradigm.

Experimental Results

- P-Rank Time Efficiency



(e) ASAP *v.s.* AUG on synthetic data

ASAP runs approx. 3x faster than AUG because after eigen-decomposition, AUG still requires extra iterations to be performed in the small eigen-subspace, which takes a significant amount of time, whereas ASAP can straightforwardly compute similarities in terms of eigenvectors with no need for iterations, and therefore takes less time.

Conclusions

- An accuracy estimate has been proposed for the P-Rank iterative paradigm, by finding out the exact number of iterations needed to attain a given accuracy.

- The notion of P-Rank conditional number was introduced based on P-Rank matrix representation. A tight bound of P-Rank conditional number was provided to show how the weighting factor and the damping factors affect the P-Rank stability.

- An $O(n^3)$-time algorithm has been devised to deal with the P-Rank optimization problem over undirected networks.

Thank You !

Q / A ?