SSDBM 2012

On the Efficiency of Estimating Penetrating Rank on Large Graphs

Weiren Yu[1], Jiajin Le[2], Xuemin Lin[1], Wenjie Zhang[1]

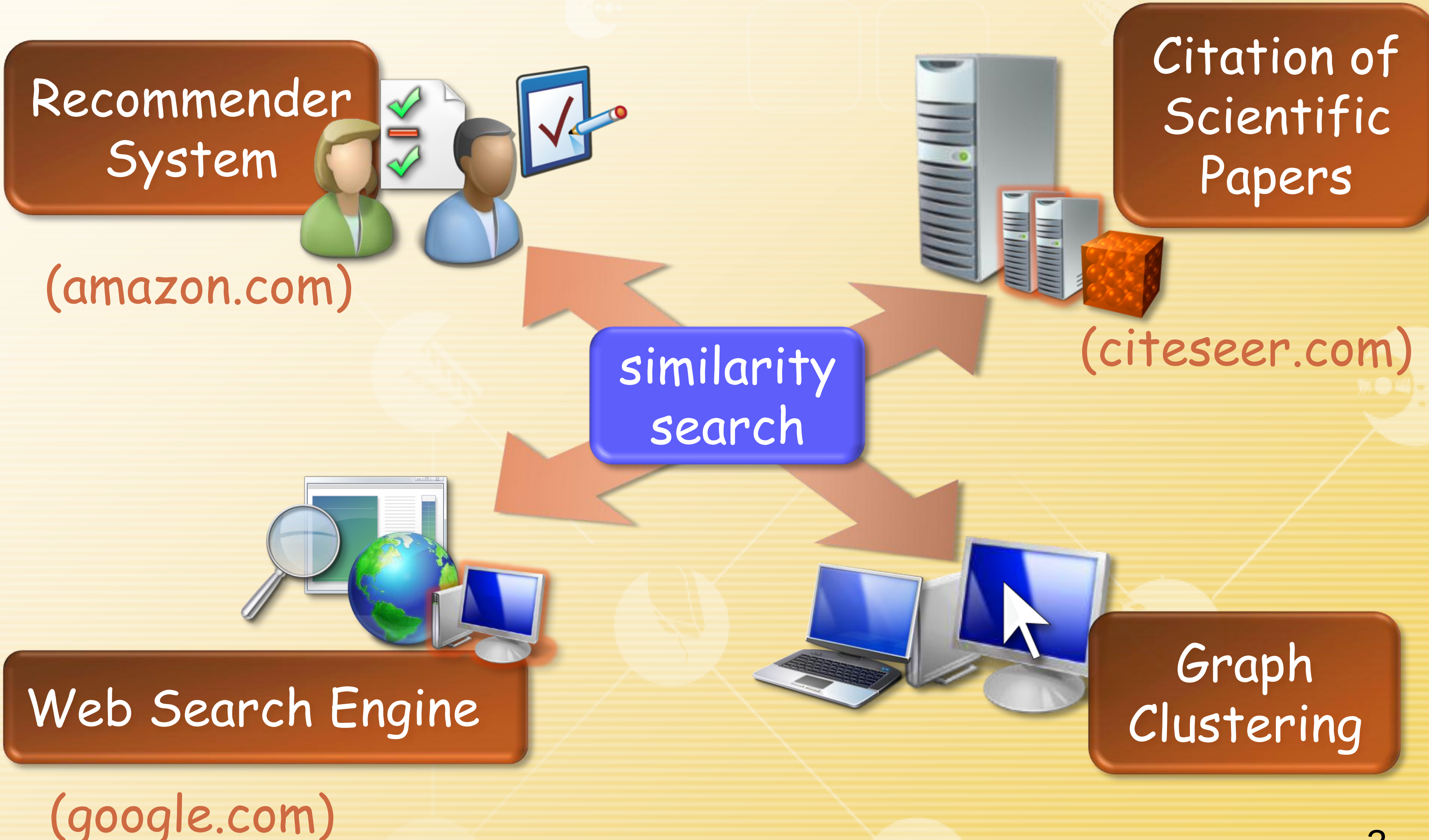
[1] University of New South Wales & NICTA, Australia

[2] Donghua University, China

Contents

1. Introduction

- Many applications require a measure of “similarity” between objects.

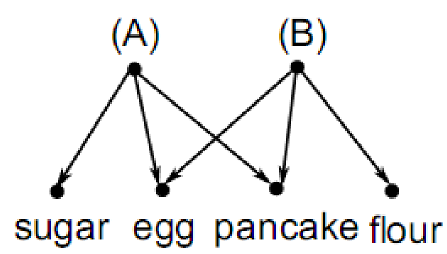


P-Rank : A New Link-based Similarity Measure

- Structural Similarity Measure
  - PageRank [Page et. al, 1999]
  - SimRank [Jeh and Widom, KDD 02]
- P(enetrating)-Rank similarity
  - A new promising structural measure [Zhao et. al. , CIKM 09]
  - An extension of SimRank metrics
- Basic Philosophy
  - Two entities are similar, if

    (1) they are referenced by similar entities

    (2) they reference similar entities



$\mathrm{SimRank}(A, B) = 0$

$\text{P-Rank}(A, B) > 0$

P-Rank Overview

- Features
  - Avoiding *"limited information problem"* of SimRank

    --- By taking account of both in- and out-links
  - Defined recursively and is computed iteratively
  - Applicable to any domain with object-to-object relationships

- Challenges
  - Costly to compute P-Rank on large graphs
    - Naïve Iteration $O(Kn^4)$ [Zhao et. al. , CIKM 09]
    - Partial Sums Amortization $O(Kn^3)$ [Lizorkin et. al. , PVLDB 08]
  - Hard to estimate the error for P-Rank approximation
    - Radius- and category-based Pruning Rule $O(Kd^2n^2)$

      [Zhao et. al. , CIKM 09]

P-Rank Formulation

- Mathematical Formula

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

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

- Iterative Paradigm

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

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

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

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

Contributions

- Characterizing P-Rank as two forms
  - matrix inversion --- deterministic optimization
  - power series --- probabilistic computation
- Deterministic optimization (off-line)
  - eliminating neighborhood structure redundancy
  - quadratic-time with an error bound
- Probabilistic computation (on-line)
  - a sampling approach
  - linear-time with controlled accuracy

P-Rank Matrix Form

- Iterative Form

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

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

- Matrix Form

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

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

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

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

P-Rank is a Linear Matrix Equation

- Key Observation

$$vec(\mathbf{A}\cdot\mathbf{X}\cdot\mathbf{B}) = (\mathbf{B}^T\otimes\mathbf{A})\cdot vec(\mathbf{X})$$

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

$$(\mathbf{I}-\mathbf{M})^{-1}\mathbf{b} \qquad \boxed{\mathbf{x} = \mathbf{M}\cdot\mathbf{x}+\mathbf{b}} \qquad \mathbf{x} = \sum_{i=0}^{\infty}\mathbf{M}^i\cdot\mathbf{b}$$

$$\mathbf{b} = vec(\mathbf{I}_n) \qquad \mathbf{x} = vec(\mathbf{s}) \qquad \mathbf{M} = \lambda\cdot C_{\text{in}}\cdot(\mathbf{Q}\otimes\mathbf{Q}) + (1-\lambda)\cdot C_{\text{out}}\cdot(\mathbf{P}\otimes\mathbf{P})$$

Two Representations of P-Rank Solution

- Matrix Inversion Form

$$vec(\mathbf{S}) = \underbrace{\left[\mathbf{I}_{n^2} - \lambda C_{in}(\mathbf{Q} \otimes \mathbf{Q}) - (1-\lambda) C_{out}(\mathbf{P} \otimes \mathbf{P})\right]^{-1}}_{\text{huge size !!!}} \cdot vec(\mathbf{I}_n).$$

- Power Series Form

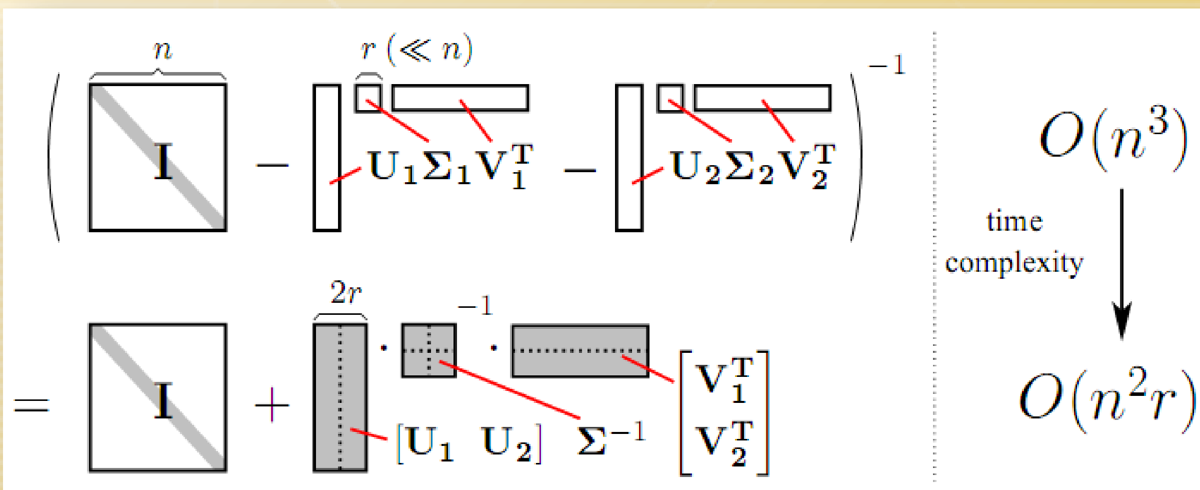
$$vec(\mathbf{S}) = \sum_{i=0}^{\infty} \left[\lambda \cdot C_{in} \cdot (\mathbf{Q} \otimes \mathbf{Q}) + (1-\lambda) \cdot C_{out} \cdot (\mathbf{P} \otimes \mathbf{P})\right]^{i} \cdot vec(\mathbf{I}_n).$$

P-Rank Deterministic Optimization

- Basic Idea
  - Most real-world graphs are low-rank and sparse.

$$vec(\mathbf{S}) = [\mathbf{I}_{n^2} - \underbrace{\lambda C_{in}(\mathbf{Q} \otimes \mathbf{Q})}_{\mathbf{U}_1\mathbf{\Sigma}_1\mathbf{V}_1^T} - \underbrace{(1-\lambda)C_{out}(\mathbf{P} \otimes \mathbf{P})}_{\mathbf{U}_2\mathbf{\Sigma}_2\mathbf{V}_2^T}]^{-1} \cdot vec(\mathbf{I}_n) \,.$$

  - Extending Woodbury matrix identity



P-Rank Deterministic Optimization

- P-Rank can be solved as follows.

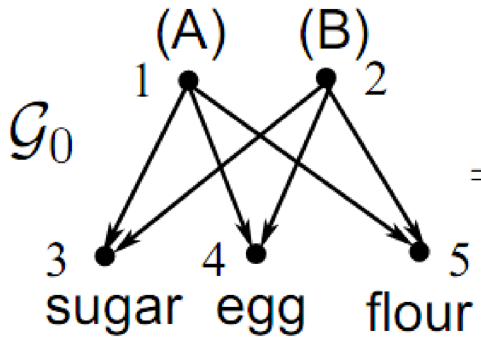
$$vec(\mathbf{S}) = \left(\tilde{\mathbf{U}}_{\mathbf{Q}} \ \tilde{\mathbf{U}}_{\mathbf{P}} \right) \boldsymbol{\Sigma} \begin{pmatrix} \tilde{\mathbf{V}}_{\mathbf{Q}}^T \\ \tilde{\mathbf{V}}_{\mathbf{P}}^T \end{pmatrix} vec\left(\mathbf{I}_n\right) + vec\left(\mathbf{I}_n\right)$$

$$\boldsymbol{\Sigma} = \begin{pmatrix} \frac{1}{\lambda C_{\text{in}}} \tilde{\boldsymbol{\Sigma}}_{\mathbf{Q}}^{-1} - \tilde{\mathbf{V}}_{\mathbf{Q}}^T \tilde{\mathbf{U}}_{\mathbf{Q}} & -\tilde{\mathbf{V}}_{\mathbf{Q}}^T \tilde{\mathbf{U}}_{\mathbf{P}} \\ -\tilde{\mathbf{V}}_{\mathbf{P}}^T \tilde{\mathbf{U}}_{\mathbf{Q}} & \frac{1}{(1-\lambda) C_{\text{out}}} \tilde{\boldsymbol{\Sigma}}_{\mathbf{P}}^{-1} - \tilde{\mathbf{V}}_{\mathbf{P}}^T \tilde{\mathbf{U}}_{\mathbf{P}} \end{pmatrix}^{-1}$$

where a tilde denotes the self-Kronecker product of a matrix, e.g., $\tilde{\mathbf{U}}_{\mathbf{Q}} = \mathbf{U}_{\mathbf{Q}} \otimes \mathbf{U}_{\mathbf{Q}}$

- Complexity
  - O $(rn^2 + r^6)$ time, O$(r \cdot \max\{r^3, n\})$ space

Example



$$\xRightarrow{①} \mathbf{A} = \begin{pmatrix} 0&0&1&1&1 \\ 0&0&1&1&1 \\ 0&0&0&0&0 \\ 0&0&0&0&0 \\ 0&0&0&0&0 \end{pmatrix} \xRightarrow{②} \mathbf{Q} = \begin{pmatrix} 0&0&0&0&0 \\ 0&0&0&0&0 \\ \frac{1}{2}&\frac{1}{2}&0&0&0 \\ \frac{1}{2}&\frac{1}{2}&0&0&0 \\ \frac{1}{2}&\frac{1}{2}&0&0&0 \end{pmatrix} \quad \mathbf{P} = \begin{pmatrix} 0&0&\frac{1}{3}&\frac{1}{3}&\frac{1}{3} \\ 0&0&\frac{1}{3}&\frac{1}{3}&\frac{1}{3} \\ 0&0&0&0&0 \\ 0&0&0&0&0 \\ 0&0&0&0&0 \end{pmatrix}$$

$$\xRightarrow{③} \mathbf{U_Q} = \begin{pmatrix} 0 \\ 0 \\ -.577 \\ -.577 \\ -.577 \end{pmatrix} \quad \mathbf{\Sigma_Q} = (1.225) \quad \mathbf{V_Q^T} = (-.707 \;\; -.707 \;\; 0 \;\; 0 \;\; 0)$$

$$\mathbf{V_P^T} = (0 \;\; 0 \;\; -.577 \;\; -.577 \;\; -.577) \quad \mathbf{\Sigma_P} = (.817) \quad \mathbf{U_P} = \begin{pmatrix} 0 \\ 0 \\ -.577 \\ -.577 \\ -.577 \end{pmatrix}$$

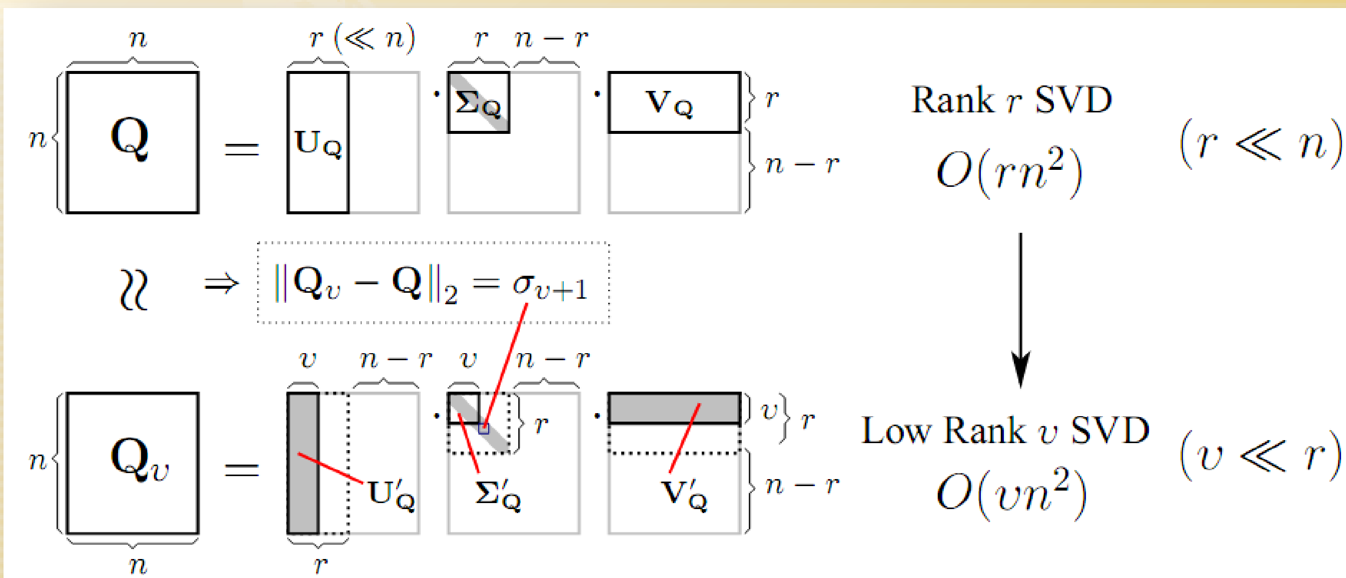
$$\xRightarrow{④} \begin{aligned} \mathbf{\Sigma}_{11} &= (3.33) \\ \mathbf{\Sigma}_{12} &= (-1) \\ \mathbf{\Sigma}_{21} &= (-1) \\ \mathbf{\Sigma}_{22} &= (5) \end{aligned} \xRightarrow{⑤} \begin{aligned} \mathbf{V}_1 &= (.383) \\ \mathbf{V}_2 &= (.277) \end{aligned} \xRightarrow{⑥} \mathbf{S} = \begin{pmatrix} .569 & .069 & 0 & 0 & 0 \\ .069 & .569 & 0 & 0 & 0 \\ 0 & 0 & .564 & .064 & .064 \\ 0 & 0 & .064 & .564 & .064 \\ 0 & 0 & .064 & .064 & .564 \end{pmatrix}$$

P-Rank Deterministic Approximation

- P-Rank matrix inversion form

$$vec(\mathbf{S}) = \left[\mathbf{I}_{n^2} - \lambda \underbrace{C_{in}(\mathbf{Q}\otimes\mathbf{Q})}_{\mathbf{U}_1\boldsymbol{\Sigma}_1\mathbf{V}_1^T} - \underbrace{(1-\lambda)\,C_{out}(\mathbf{P}\otimes\mathbf{P})}_{\mathbf{U}_2\boldsymbol{\Sigma}_2\mathbf{V}_2^T}\right]^{-1} \cdot vec(\mathbf{I}_n)\,.$$

- Reduced SVD for P-Rank Approximation



P-Rank Deterministic Approximation

- Approximation Error

$$\epsilon_v \le \frac{\lambda C_{in}\sigma_1\sigma_{v+1} + (1-\lambda)C_{out}\bar{\sigma}_1\bar{\sigma}_{v+1}}{1-\lambda C_{in}-(1-\lambda)C_{out}} r$$

e.g., WIKI 0715 $(r = 15\text{K}, \sigma_1 = 1.12, \bar{\sigma}_1 = 1.08)$

Setting $C_{\text{in}} = 0.8, C_{\text{out}} = 0.6$, and $\lambda = 0.5$

$$\epsilon_v \le \frac{0.5\times0.8\times1.12+0.5\times0.6\times1.08}{1-0.5\times0.8-0.5\times0.6} \times 10^{-7} \times 15\text{K} = 0.0039$$

- Complexity

Time: $O\left(vn^2 + v^6\right)$ with $v \le r$

Space: $O(v \cdot \max\{v^3, n\})$

P-Rank Probabilistic Computation

- Key Observation
  - P-Rank can be viewed as a geometric sum of random walks

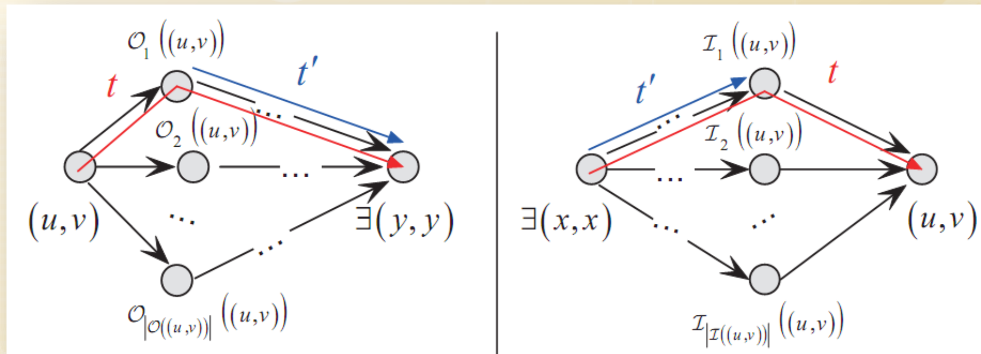
$$vec(\mathbf{S}) = \sum_{i=0}^{\infty} \left[\lambda \cdot C_{in} \cdot (\mathbf{Q} \otimes \mathbf{Q}) + (1-\lambda) \cdot C_{out} \cdot (\mathbf{P} \otimes \mathbf{P})\right]^i \cdot vec(\mathbf{I}_n)$$

  - s (u,v) represents how soon two surfers are expected to meet
- Main idea
  - utilize the first hitting time τ(u, v) of coalescing walks to estimate s (u, v)

$$s(u,v) = \mathbb{E}(\lambda \cdot C_{in}{}^{\tau_1(u,v)} + (1-\lambda) \cdot C_{out}{}^{\tau_2(u,v)})$$

P-Rank Probabilistic Computation

- Random Surfer Model
  - one-step path transformation T : t′ → t



  - length $l(t) = l(t') + 1$
  - probability

$$p\left(T\left(t'\right)\right) = \begin{cases} \frac{1}{|\mathcal{I}((u,v))|} \cdot p\left(t'\right), \; t' : \exists\left(x,x\right) \to \left(u,v\right); \\ \frac{1}{|\mathcal{O}((u,v))|} \cdot p\left(t'\right), \; t' : \left(u,v\right) \to \exists\left(y,y\right). \end{cases}$$

- Equivalence of Sampling approach

$$s\left(u,v\right) = \lambda \cdot \sum_{t:\exists(x,x)\to(u,v)} p\left(t\right) \cdot {C_{\text{in}}}^{l(t)} + \left(1-\lambda\right) \cdot \sum_{t:(u,v)\to\exists(y,y)} p\left(t\right) \cdot {C_{\text{out}}}^{l(t)}.$$

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

P-Rank Probabilistic Computation

- Complexity
  - Time $O(N \cdot n)$
  - Space $O(n + N)$

  where N : sample size, n : # of vertices

- Sample Size
  - $N \geq -2\,[(\sigma/\epsilon)^2 \log \alpha]$ suffices to ensure that

$$\Pr(|s_N - s| \geq \epsilon) < \alpha$$

  - In practice, $N \ll n$.

  e.g., on DBLP (98-07) For $n = 10K$, $\epsilon = 0.15\sigma$, $\alpha = 0.05$, we have

$$N \geq -2[0.15^{-2} \log(0.05)] = 267.$$

P-Rank Probabilistic Computation

- Error Bound

  Let $Err \triangleq \sup_{N \geq 1} \Pr\left(|\hat{s}_N - s| \geq \epsilon\right)$

  - upper bound - by Bernstein's Theorem

$$Err \leq \exp(-N\epsilon^2/(2\sigma^2))$$

  - lower bound - by Central Limit Theorem

$$Err \geq \Pr\left(\left|\frac{1}{\sqrt{N}}\sum_{i=1}^{N}\left(\frac{\hat{s}_N^{(i)} - s}{\sigma}\right)\right| \geq \frac{\epsilon\sqrt{N}}{\sigma}\right) = 2 - 2\Phi\left(\frac{\epsilon\sqrt{N}}{\sigma}\right)$$

- Relative Order Preserving

  If $s(u,v) > s(u,w) + \epsilon$, then

$$\Pr(\hat{s}_N(u,v) - \hat{s}_N(u,w) > \epsilon) \leq \exp(-N\epsilon^2/2)$$

Experiment

- Datasets
  - Synthetic data (RAND 0.5M-3.5M)
  - Real data (AMZN, DBLP, WIKI)

	0505	0601
$\lvert\mathcal{V}\rvert$	410K	402K
$\lvert\mathcal{E}\rvert$	3,356K	3,387K

Table 2: AMZN

	98-99	98-01	98-03	98-05	98-07
$\lvert\mathcal{V}\rvert$	1,525	3,208	5,307	7,984	10,682
$\lvert\mathcal{E}\rvert$	5,929	13,441	24,762	39,399	54,844

Table 3: DBLP

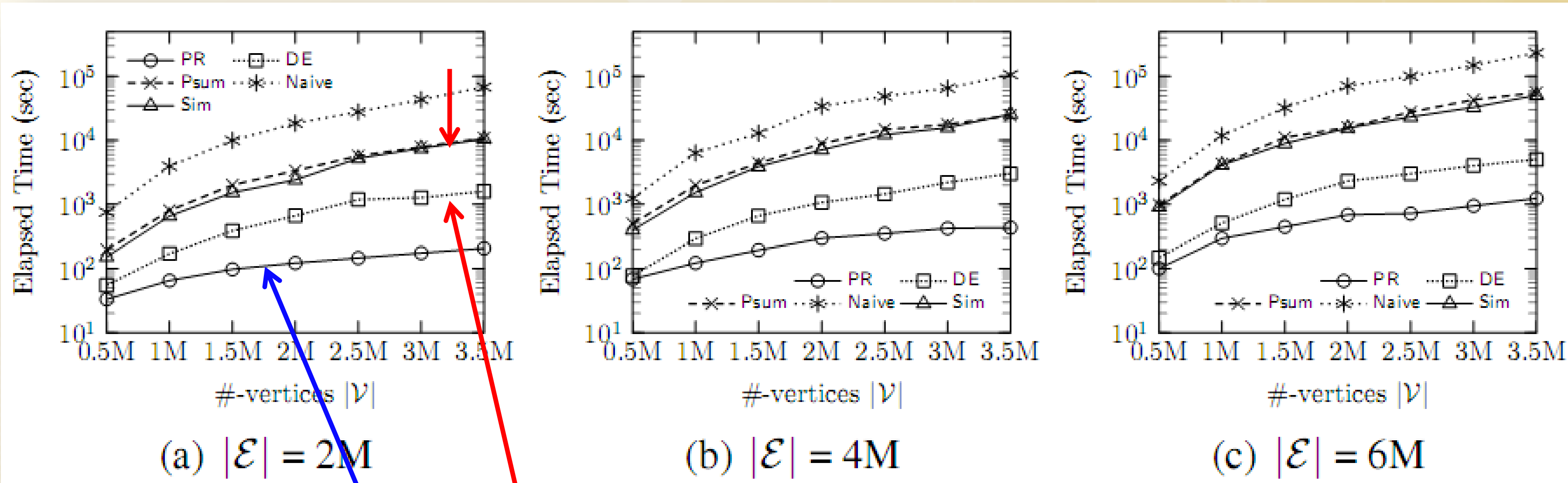
	0715	0827	0919
$\lvert\mathcal{V}\rvert$	3,088K	3,102K	3,116K
$\lvert\mathcal{E}\rvert$	1,126K	1,134K	1,142K

Table 4: WIKI

- Compared Algorithms
  - DE P-Rank, PR P-Rank
  - Naive, radius-based pruning iteration [Zhao et al, CIKM 2009]
  - Psum, leveraging a partial sum function to compute P-Rank
  - Sim, a SimRank algorithm, taking account of the evidence factor for incident vertices

Experiment (1)

Scalability on Synthetic Datasets



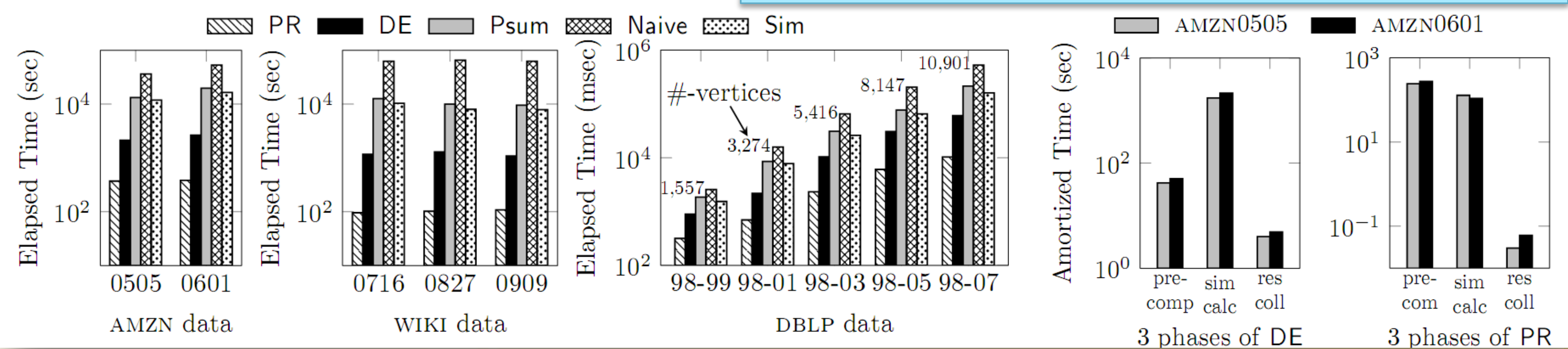
DE is one-order-of-magnitude faster

PR increases linearly with |V|

Experiment (2)

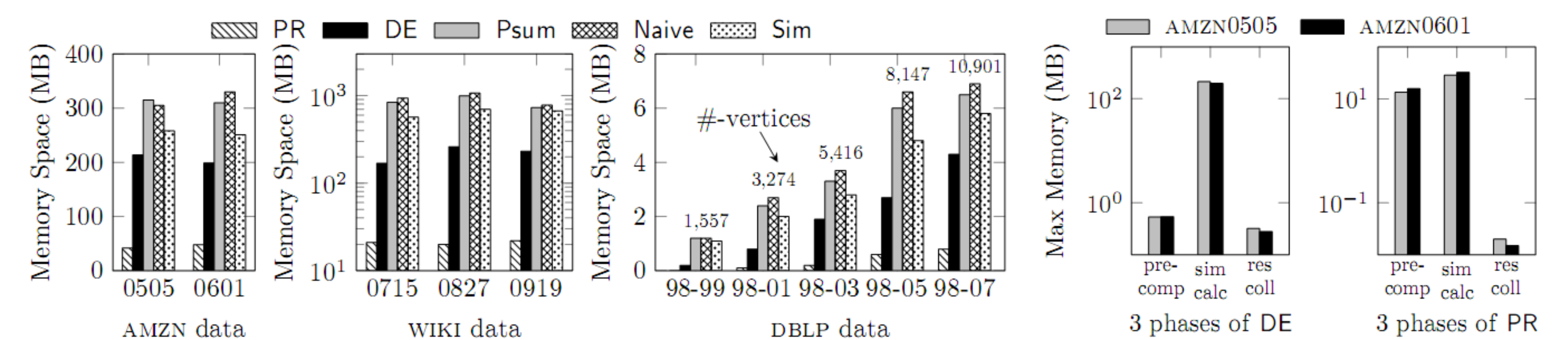
Computational Time on Real Datasets

PR outperforms the other approaches.



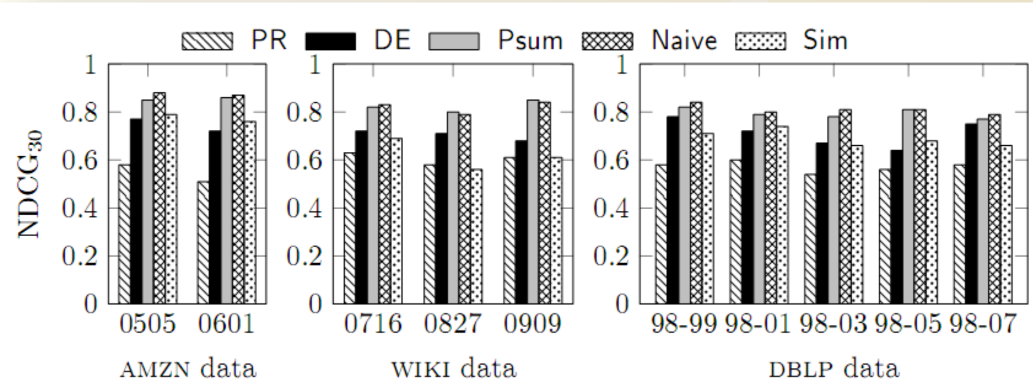
DE can cluster vertices with similar neighborhood.

Memory Space on Real Datasets



Experiment (3)

Accuracy on Real Datasets



Rank	PR	DE	Naive
1	Shivnath Babu	Shivnath Babu	Shivnath Babu
2	Chris Olston	Yingwei Cui	Yingwei Cui
3	Jun Yang	Chris Olston	Chris Olston
4	Yingwei Cui	Jun Yang	Jun Yang
5	Rajeev Motwani	Arvind Arasu	Rajeev Motwani
6	Arvind Arasu	Rajeev Motwani	Arvind Arasu
7	David J. DeWitt	Alon Y. Halevy	Utkarsh Srivastava
8	Glen Jeh	Anish Das Sarma	David J. DeWitt
9	Utkarsh Srivastava	Omar Benjelloun	Omar Benjelloun
10	Omar Benjelloun	David J. DeWitt	Alon Y. Halevy

Top-10 Co-authors of Jennifer Widom on DBLP

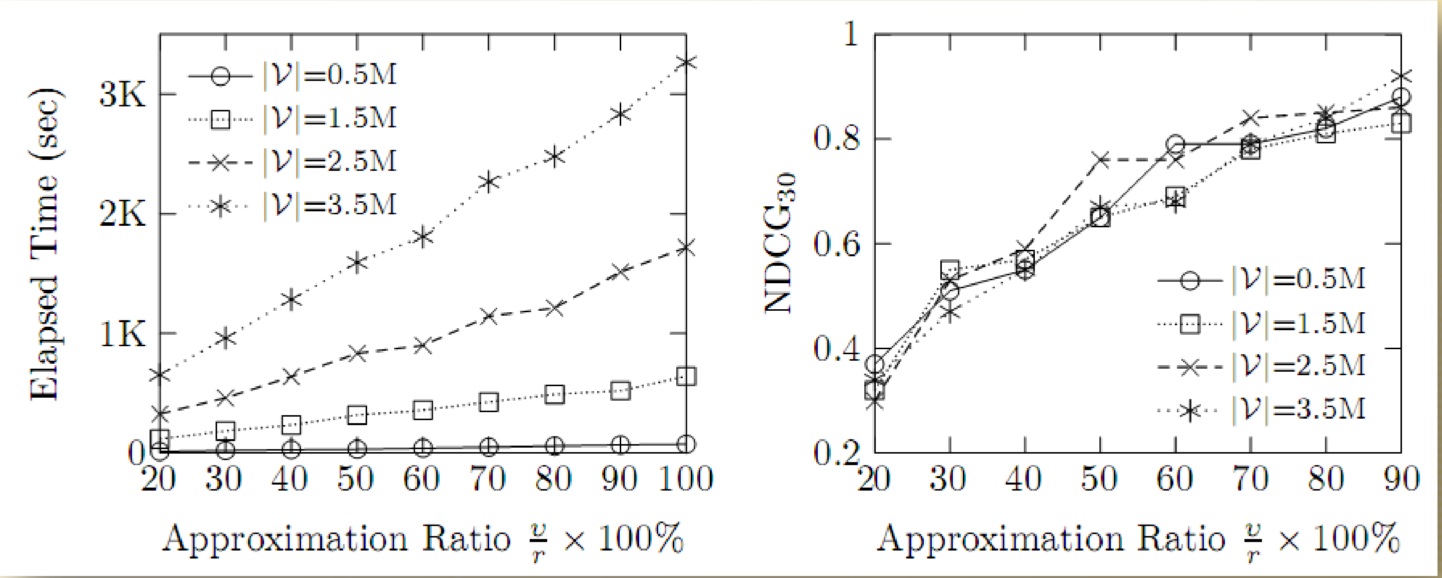
$$\mathrm{NDCG}_p = \frac{1}{\mathrm{IDCG}_p} \sum_{i=1}^{p} \left(2^{\mathrm{rank}_i} - 1\right) / \left(\log_2 (1+i)\right)$$

DE achieves higher accuracy than PR.

The accuracy of PR is not that good because a few FPTs are neglected with certain probability by sampling.

Experiment (4)

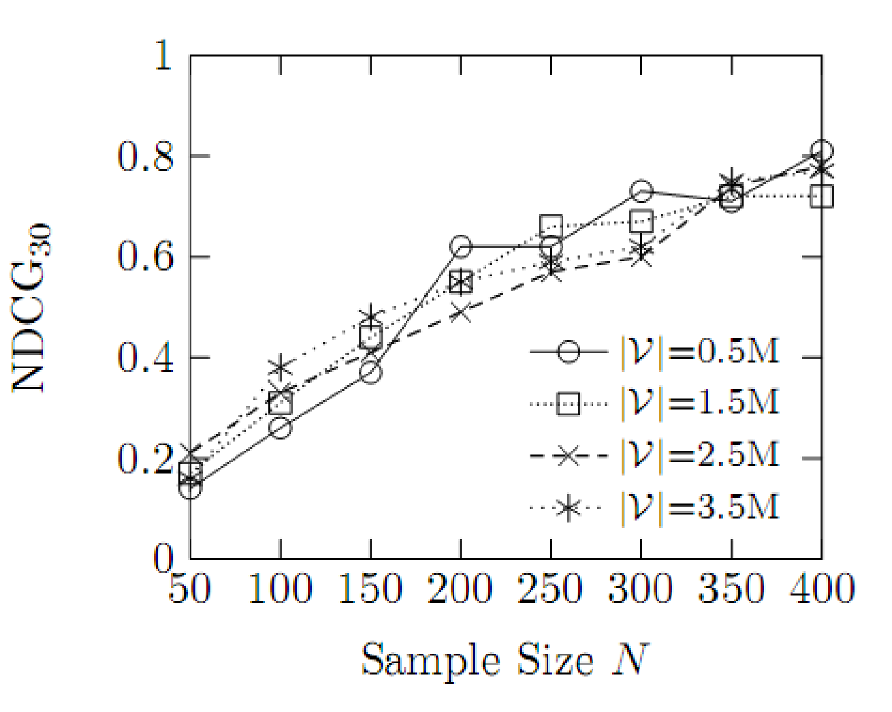
Effects of u for DE



Adding u induces smaller errors, but increases the time up to rank r.

Experiment (5)

Effects of N for PR



Adding samples of FPTs reduces errors

When N > 300, higher accuracy could be expected ($NDCG_{30}$ > 0.6)

Conclusions

- Two matrix forms are investigated to characterize P-Rank.
- Using matrix inversion form, we propose DE P-Rank to reduce the time from cubic to quadratic.
- By leveraging reduced SVD, the error estimate is obtained for P-Rank approximation.
- Using power series form, we present PR P-Rank to speed up the computation of P-Rank in linear time with controlled accuracy.

Thank You !