Fast Incremental SimRank on Link-Evolving Graphs

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

[1] University of New South Wales
[2] Imperial College London,

Outline

Overview

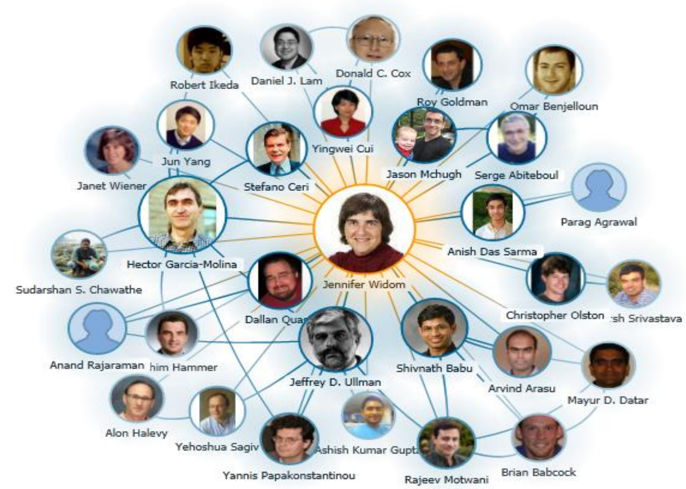
- Existing incremental method
- Our approaches
  - express ΔS as a rank-one Sylvester equation: $O(Kn^2)$
  - prune “unaffected areas” of ΔS: $O(K(nd+|AFF|))$ with $|AFF| < n^2$
- Empirical evaluations
- Conclusions

Overview

- Similarity Assessment plays a vital role in our lives.

Recommender System

Citation Graph



Collaboration Network

SimRank Overview

- SimRank
  - An appealing link-based similarity measure (KDD '02)
  - Basic philosophy

    Two vertices are similar if they are referenced by similar vertices.

- Two Forms
  - Original form (KDD '02)

$$s(a,a) = 1$$

$$s(a,b) = \frac{C}{|\mathcal{I}(a)|\,|\mathcal{I}(b)|} \sum_{j \in \mathcal{I}(b)} \sum_{i \in \mathcal{I}(a)} s(i,j)$$

similarity btw. nodes a and b

damping factor

in-neighbor set of node b

  - Matrix form (EDBT '10)

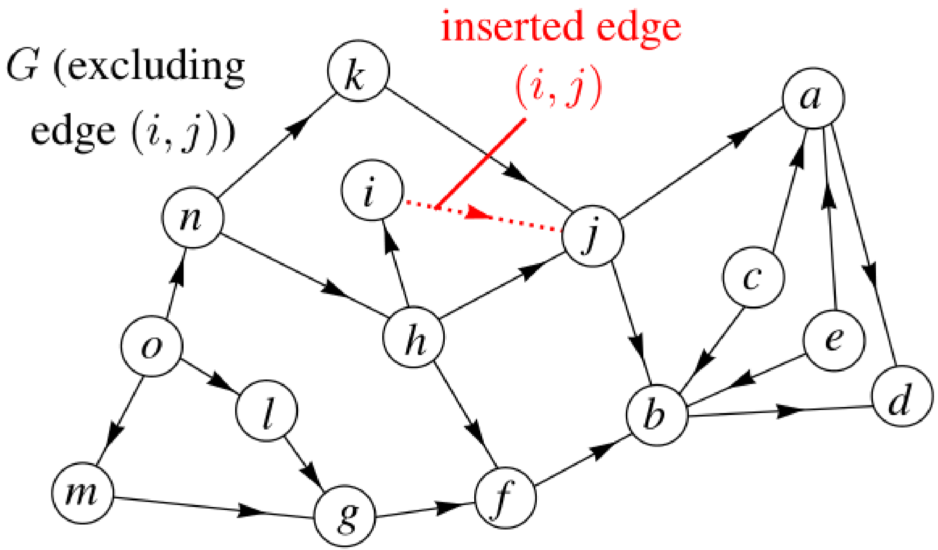
$$\mathbf{S} = C \cdot (\mathbf{Q} \cdot \mathbf{S} \cdot \mathbf{Q}^T) + (1 - C) \cdot \mathbf{I}_n$$

Existing SimRank Algorithms

- Batch Computations
  - All Pairs s(*,*)
  - Single Pair s(a,b)
  - Single Source s(*,q)
  - Similarity Join s(x,y) for all x in A, and y in B.

- Incremental Paradigms:
  - link-evolving:
    - Li et. al. [EDBT 2010] needs $O(r^4 n^2)$ time for approximation.
  - node-evolving:
    - He et al. [KDD 2010] --- GPU based

Motivation



Node-Pair	in G	in $G \cup \Delta G$	
	sim	sim_{true}	$sim_{Li\ et\ al.}$
(a, b)	0.075	0.062	0.073
(a, d)	0.000	0.006	0.002
(i, f)	**0.246**	**0.246**	**0.246**
(k, g)	**0.128**	**0.128**	**0.128**
(k, h)	**0.288**	**0.288**	**0.288**
(j, f)	0.206	0.138	0.206
(m, l)	**0.160**	**0.160**	**0.160**
(j, b)	0.000	0.030	0.001

- Li et al. [EDBT 2010] using SVD for incremental SimRank is approximate.
- When ΔG is small, the "affected areas" of ΔS are also small.

Problem (INCREMENTAL SIMRANK COMPUTATION)

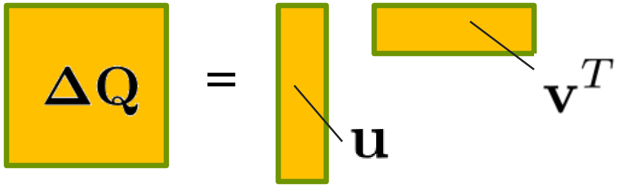
Given: G, S, ΔG, and C.

Compute: ΔS to S.

Main Idea

- For every edge update, ΔQ has a rank-one structure

$$\mathbf{\Delta Q} = \mathbf{u} \cdot \mathbf{v}^T$$



- Characterize ΔS as

$$\mathbf{\Delta S} = \mathbf{M} + \mathbf{M}^T$$, where M satisfies

$$\mathbf{M} = C \cdot \tilde{\mathbf{Q}} \cdot \mathbf{M} \cdot \tilde{\mathbf{Q}}^T + C \cdot \mathbf{u} \cdot \mathbf{w}^T$$

compute M via mat-vec multiplication

In comparison

$$\tilde{\mathbf{S}} = C \cdot \tilde{\mathbf{Q}} \cdot \tilde{\mathbf{S}} \cdot \tilde{\mathbf{Q}}^T + (1 - C) \cdot \mathbf{I}_n$$

compute S̃ via mat-mat multiplication

Mat-Mat → Mat-Vec Multiplication

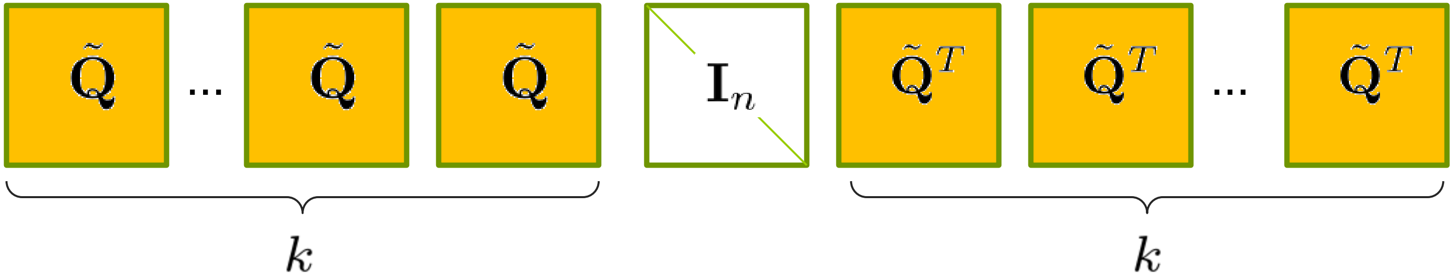
- Based on

$$\mathbf{X} = \sum_{k=0}^{\infty} \mathbf{A}^k \cdot \mathbf{C} \cdot \mathbf{B}^k \quad \Leftrightarrow \quad \mathbf{X} = \mathbf{A} \cdot \mathbf{X} \cdot \mathbf{B} + \mathbf{C}$$

we have

$$\mathbf{M} = \sum_{k=0}^{\infty} C^{k+1} \cdot \tilde{\mathbf{Q}}^k \cdot \mathbf{u} \cdot \mathbf{w}^T \cdot (\tilde{\mathbf{Q}}^T)^k,$$

$$\tilde{\mathbf{S}} = (1-C) \cdot \sum_{k=0}^{\infty} C^k \cdot \tilde{\mathbf{Q}}^k \cdot \mathbf{I}_n \cdot (\tilde{\mathbf{Q}}^T)^k.$$



Mat-Mat → Mat-Vec Multiplication

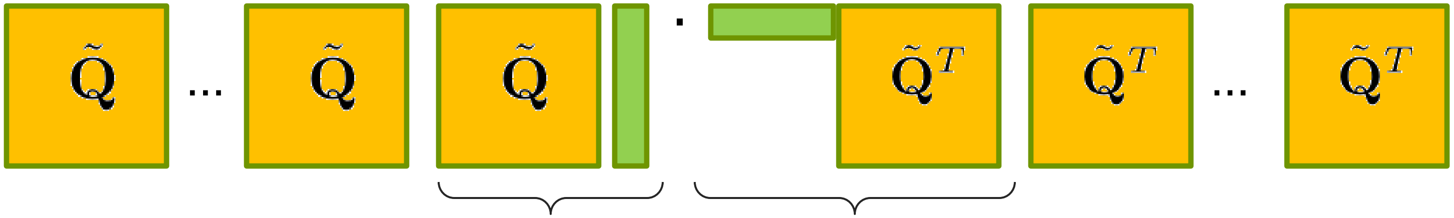
- Based on

$$\mathbf{X} = \sum_{k=0}^{\infty} \mathbf{A}^k \cdot \mathbf{C} \cdot \mathbf{B}^k \quad \Leftrightarrow \quad \mathbf{X} = \mathbf{A} \cdot \mathbf{X} \cdot \mathbf{B} + \mathbf{C}$$

we have

$$\mathbf{M} = \sum_{k=0}^{\infty} C^{k+1} \cdot \tilde{\mathbf{Q}}^k \cdot \mathbf{u} \cdot \mathbf{w}^T \cdot (\tilde{\mathbf{Q}}^T)^k,$$

$$\tilde{\mathbf{S}} = (1 - C) \cdot \sum_{k=0}^{\infty} C^k \cdot \tilde{\mathbf{Q}}^k \cdot \mathbf{I}_n \cdot (\tilde{\mathbf{Q}}^T)^k.$$



Mat-Mat → Mat-Vec Multiplication

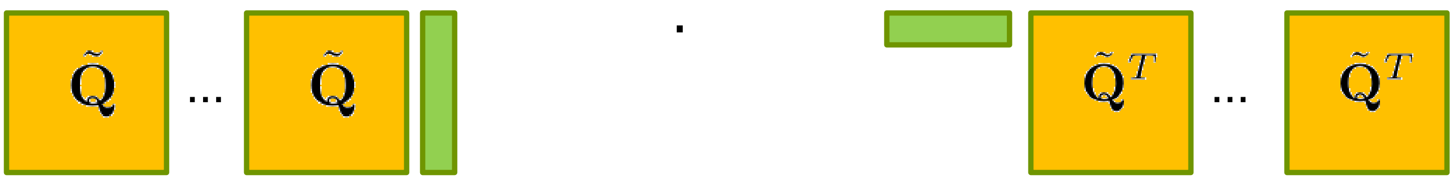
- Based on

$$\mathbf{X} = \sum_{k=0}^{\infty} \mathbf{A}^k \cdot \mathbf{C} \cdot \mathbf{B}^k \quad \Leftrightarrow \quad \mathbf{X} = \mathbf{A} \cdot \mathbf{X} \cdot \mathbf{B} + \mathbf{C}$$

we have

$$\mathbf{M} = \sum_{k=0}^{\infty} C^{k+1} \cdot \tilde{\mathbf{Q}}^k \cdot \mathbf{u} \cdot \mathbf{w}^T \cdot (\tilde{\mathbf{Q}}^T)^k,$$

$$\tilde{\mathbf{S}} = (1 - C) \cdot \sum_{k=0}^{\infty} C^k \cdot \tilde{\mathbf{Q}}^k \cdot \mathbf{I}_n \cdot (\tilde{\mathbf{Q}}^T)^k.$$



Mat-Mat → Mat-Vec Multiplication

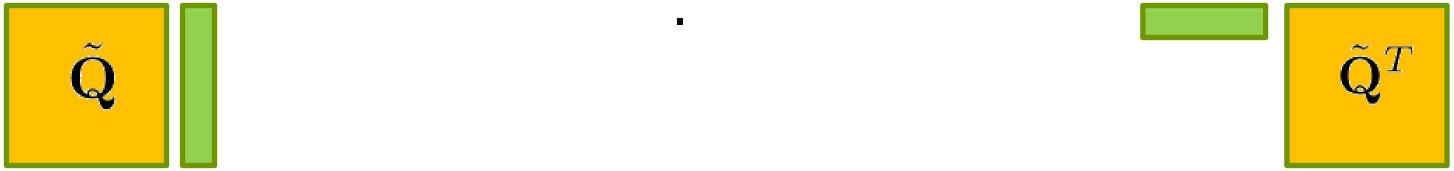
- Based on

$$\mathbf{X} = \sum_{k=0}^{\infty} \mathbf{A}^k \cdot \mathbf{C} \cdot \mathbf{B}^k \quad \Leftrightarrow \quad \mathbf{X} = \mathbf{A} \cdot \mathbf{X} \cdot \mathbf{B} + \mathbf{C}$$

we have

$$\mathbf{M} = \sum_{k=0}^{\infty} C^{k+1} \cdot \tilde{\mathbf{Q}}^k \cdot \mathbf{u} \cdot \mathbf{w}^T \cdot (\tilde{\mathbf{Q}}^T)^k,$$

$$\tilde{\mathbf{S}} = (1 - C) \cdot \sum_{k=0}^{\infty} C^k \cdot \tilde{\mathbf{Q}}^k \cdot \mathbf{I}_n \cdot (\tilde{\mathbf{Q}}^T)^k.$$



Mat-Mat → Mat-Vec Multiplication

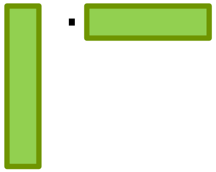
- Based on

$$\mathbf{X} = \sum_{k=0}^{\infty} \mathbf{A}^k \cdot \mathbf{C} \cdot \mathbf{B}^k \quad \Leftrightarrow \quad \mathbf{X} = \mathbf{A} \cdot \mathbf{X} \cdot \mathbf{B} + \mathbf{C}$$

we have

$$\mathbf{M} = \sum_{k=0}^{\infty} C^{k+1} \cdot \tilde{\mathbf{Q}}^k \cdot \mathbf{u} \cdot \mathbf{w}^T \cdot (\tilde{\mathbf{Q}}^T)^k,$$

$$\tilde{\mathbf{S}} = (1 - C) \cdot \sum_{k=0}^{\infty} C^k \cdot \tilde{\mathbf{Q}}^k \cdot \mathbf{I}_n \cdot (\tilde{\mathbf{Q}}^T)^k.$$



Mat-Mat → Mat-Vec Multiplication

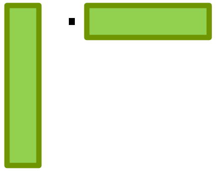
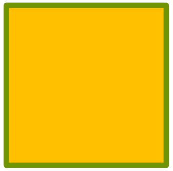
- Based on

$$\mathbf{X} = \sum_{k=0}^{\infty} \mathbf{A}^k \cdot \mathbf{C} \cdot \mathbf{B}^k \quad \Leftrightarrow \quad \mathbf{X} = \mathbf{A} \cdot \mathbf{X} \cdot \mathbf{B} + \mathbf{C}$$

we have

$$\mathbf{M} = \sum_{k=0}^{\infty} C^{k+1} \cdot \tilde{\mathbf{Q}}^k \cdot \mathbf{u} \cdot \mathbf{w}^T \cdot (\tilde{\mathbf{Q}}^T)^k,$$

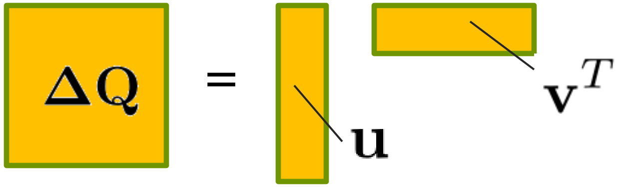
$$\tilde{\mathbf{S}} = (1 - C) \cdot \sum_{k=0}^{\infty} C^k \cdot \tilde{\mathbf{Q}}^k \cdot \mathbf{I}_n \cdot (\tilde{\mathbf{Q}}^T)^k.$$

$$\mathbf{M} = \sum_{k=0}^{\infty} C^{k+1} \cdot$$  = $$\sum_{k=0}^{\infty} C^{k+1} \cdot$$ 

Challenges

- For every edge update, ΔQ has a rank-one structure

$$\Delta \mathbf{Q} = \mathbf{u} \cdot \mathbf{v}^T$$



- Characterize ΔS as

$$\Delta \mathbf{S} = \mathbf{M} + \mathbf{M}^T$$, where M satisfies

$$\mathbf{M} = C \cdot \tilde{\mathbf{Q}} \cdot \mathbf{M} \cdot \tilde{\mathbf{Q}}^T + C \cdot \mathbf{u} \cdot \mathbf{w}^T$$

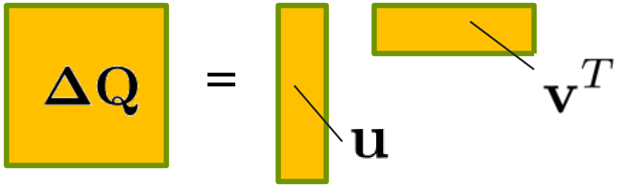
Finding u, v, w is challenging !!

~

Finding u, v

- For every edge update, ΔQ has a rank-one structure

$$\Delta \mathbf{Q} = \mathbf{u} \cdot \mathbf{v}^T$$



where

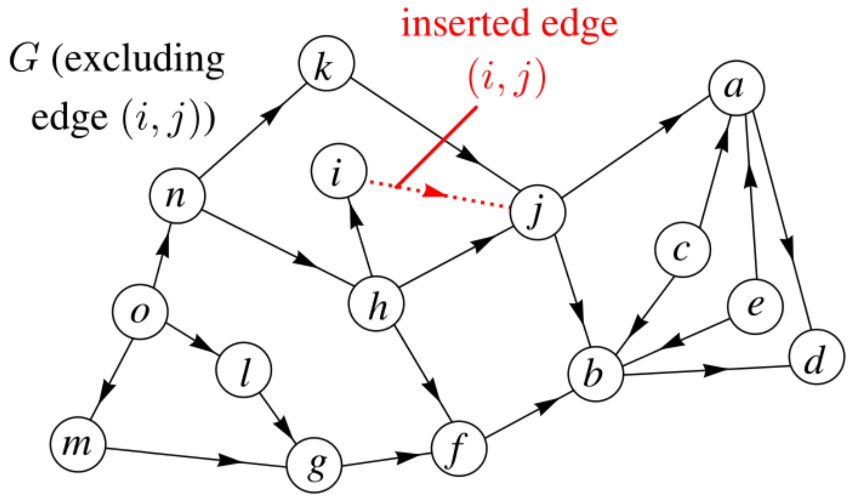
(1) for edge (i, j) insertion,

$$\mathbf{u} = \begin{cases} \mathbf{e}_j & (d_j = 0) \\ \frac{1}{d_j+1}\mathbf{e}_j & (d_j > 0) \end{cases}, \quad \mathbf{v} = \begin{cases} \mathbf{e}_i & (d_j = 0) \\ \mathbf{e}_i - [\mathbf{Q}]_{j,\star}^T & (d_j > 0) \end{cases}$$

(2) for edge (i, j) deletion,

$$\mathbf{u} = \begin{cases} \mathbf{e}_j & (d_j = 1) \\ \frac{1}{d_j-1}\mathbf{e}_j & (d_j > 1) \end{cases}, \quad \mathbf{v} = \begin{cases} -\mathbf{e}_i & (d_j = 1) \\ [\mathbf{Q}]_{j,\star}^T - \mathbf{e}_i & (d_j > 1) \end{cases}$$

Example



$$[\tilde{\mathbf{Q}}]_{j,\star} = [\,0 \;\cdots\; 0 \;\; \overset{(h)}{\tfrac{1}{3}} \;\; \tfrac{1}{3} \;\; 0 \;\; \overset{(k)}{\tfrac{1}{3}} \;\; 0 \;\cdots\; 0\,]$$

- Since the old $[\mathbf{Q}]_{j,\star} = [\,0 \;\cdots\; 0 \;\; \overset{(h)}{\tfrac{1}{2}} \;\; 0 \;\; 0 \;\; \overset{(k)}{\tfrac{1}{2}} \;\; 0 \;\cdots\; 0\,] \in \mathbb{R}^{1\times 15}$,

  after insertion: $\Delta\mathbf{Q} = \mathbf{u} \cdot \mathbf{v}^T$ with

$$\mathbf{u} = \frac{1}{d_j+1}\mathbf{e}_j = \frac{1}{3}\mathbf{e}_j = [\,0 \;\cdots\; 0 \;\; \overset{(j)}{\tfrac{1}{3}} \;\; 0 \;\cdots\; 0\,]^T \in \mathbb{R}^{15\times 1},$$

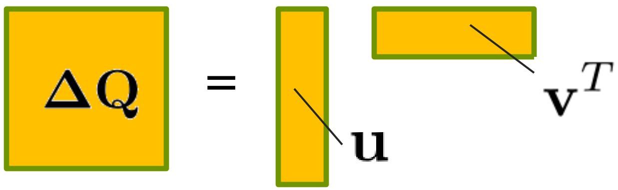
$$\mathbf{v} = \mathbf{e}_i - [\mathbf{Q}]_{j,\star}^T = [\,0 \;\cdots\; 0 \;\; \overset{(h)}{-\tfrac{1}{2}} \;\; \overset{(i)}{1} \;\; \overset{(j)}{0} \;\; \overset{(k)}{-\tfrac{1}{2}} \;\; 0 \;\cdots\; 0\,]^T \in \mathbb{R}^{15\times 1}.$$

Finding w

- For every edge update, ΔQ has a rank-one structure

$$\Delta \mathbf{Q} = \mathbf{u} \cdot \mathbf{v}^T$$

Step 1



- Characterize ΔS as

Step 3

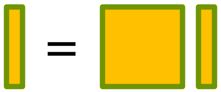
$\Delta \mathbf{S} = \mathbf{M} + \mathbf{M}^T$, where M satisfies

$$\mathbf{M} = C \cdot \tilde{\mathbf{Q}} \cdot \mathbf{M} \cdot \tilde{\mathbf{Q}}^T + C \cdot \mathbf{u} \cdot \mathbf{w}^T \qquad (1)$$

Theorem There exists $\mathbf{w} = \mathbf{y} + \frac{\lambda}{2}\mathbf{u}$ with

$$\mathbf{y} = \mathbf{Q} \cdot \mathbf{z}, \quad \text{Step 2} \quad \cdot \mathbf{z}, \quad \mathbf{z} = \mathbf{S} \cdot \mathbf{v}$$

s.t. Eq.(1) is a rank-one Sylvester Equation w.r.t. M.



$\mathbf{z} = \mathbf{S} \cdot \mathbf{v}$

$\mathbf{y} = \mathbf{Q} \cdot \mathbf{z}$

$\lambda = \mathbf{v}^T \cdot \mathbf{z}$

Complexity Analysis

No mat-mat multiplications

- Time complexity: O(Kn^2)

Step 1. Find u,v s.t. $\mathbf{\Delta Q} = \mathbf{u} \cdot \mathbf{v}^T$

$$\mathbf{u} = \begin{cases} \mathbf{e}_j & (d_j = 0) \\ \frac{1}{d_j+1}\mathbf{e}_j & (d_j > 0) \end{cases}, \quad \mathbf{v} = \begin{cases} \mathbf{e}_i & (d_j = 0) \\ \mathbf{e}_i - [\mathbf{Q}]^T_{j,\star} & (d_j > 0) \end{cases}$$

Step 2. Find w s.t. $\mathbf{M} = C \cdot \tilde{\mathbf{Q}} \cdot \mathbf{M} \cdot \tilde{\mathbf{Q}}^T + C \cdot \mathbf{u} \cdot \mathbf{w}^T$

$$\begin{aligned} &\text{initialize } \boldsymbol{\xi}_0 \leftarrow C \cdot \mathbf{u}, \quad \boldsymbol{\eta}_0 \leftarrow \mathbf{w}, \quad \mathbf{M}_0 \leftarrow C \cdot \mathbf{u} \cdot \mathbf{w}^T \\ &\text{for } k = 0, 1, 2, \cdots \\ &\quad \boldsymbol{\xi}_{k+1} \leftarrow C \cdot \tilde{\mathbf{Q}} \cdot \boldsymbol{\xi}_k, \quad \boldsymbol{\eta}_{k+1} \leftarrow \tilde{\mathbf{Q}} \cdot \boldsymbol{\eta}_k \\ &\quad \mathbf{M}_{k+1} \leftarrow \boldsymbol{\xi}_{k+1} \cdot \boldsymbol{\eta}^T_{k+1} + \mathbf{M}_k \end{aligned}$$

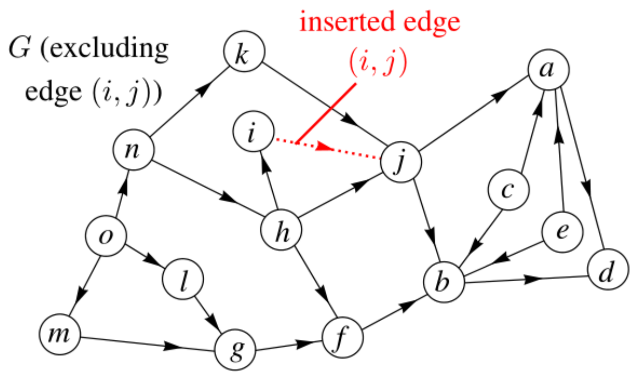
Step 3. Compute ΔS as

$$\mathbf{\Delta S} = \mathbf{M} + \mathbf{M}^T$$

Can we further improve it?

Pruning

- Key observation:
  - When link updates are small, "affected areas" in ΔS (or M) are often small as well.



	(a)	(b)	(c)	(d)	(e)	(f)	⋯	(i)	(j)	(k) ⋯ (o)
(a)	−0.005	−0.009	0	0.009					−0.009	
(b)	−0.004	−0.006	0	0.006			0		−0.007	0
(c)	0	0	0	0					0	
(d)	−0.002	−0.002	0	−0.005					0	
⋮ (i)		0					0		0	0
(j)	0.028	0.037	0	0		−0.068		−0.104	−0.060	
⋮ (o)		0					0		0	0

- Challenge:
  - How to identify only "unaffected areas" in ΔS to skip unnecessary recomputations for link update ?

Paths Aggregation

- $[\mathbf{A}^k]_{i,j}$ counts # of length-k paths from node i to j.

- $[\mathbf{S}]_{a,b}$ counts the weighted sum of paths:

$$\underbrace{a \leftarrow \circ \leftarrow \cdots \leftarrow}_{length\ \ k} \bullet \underbrace{\rightarrow \cdots \rightarrow \circ \rightarrow b}_{length\ \ k}$$

$$\mathbf{S} = C \cdot (\mathbf{Q} \cdot \mathbf{S} \cdot \mathbf{Q}^T) + (1 - C) \cdot \mathbf{I}_n$$

$$\Leftrightarrow \quad [\mathbf{S}]_{a,b} = (1 - C) \cdot \sum_{k=0}^{\infty} C^k \cdot [\mathbf{Q}^k \cdot (\mathbf{Q}^T)^k]_{a,b}$$

Q is the weighted (*i.e.*, row-normalized) matrix of A^T

Paths captured by M

$$\mathbf{\Delta S} = \mathbf{M} + \mathbf{M}^T \qquad \mathbf{M} = C \cdot \tilde{\mathbf{Q}} \cdot \mathbf{M} \cdot \tilde{\mathbf{Q}}^T + C \cdot \mathbf{u} \cdot \mathbf{w}^T$$

- Expansion of M

$$[\mathbf{M}]_{a,b} = \frac{1}{d_j+1} \Big(\underbrace{\sum_{k=0}^{\infty} C^{k+1} \cdot [\tilde{\mathbf{Q}}^k]_{a,j} [\mathbf{S}]_{i,\star} \mathbf{Q}^T \cdot [(\tilde{\mathbf{Q}}^T)^k]_{\star,b}}_{\text{Part 1}} - \underbrace{\sum_{k=0}^{\infty} C^{k} [\tilde{\mathbf{Q}}^k]_{a,j} [\mathbf{S}]_{j,\star} [(\tilde{\mathbf{Q}}^T)^k]_{\star,b}}_{\text{Part 2}}$$

$$+ \mu \underbrace{\sum_{k=0}^{\infty} C^{k+1} [\tilde{\mathbf{Q}}^k]_{a,j} [(\tilde{\mathbf{Q}}^T)^k]_{j,b}}_{\text{Part 3}} \Big)$$

- Three types of paths identified by M
  - P1: $\overbrace{\underbrace{a \leftarrow \circ \cdots \circ \leftarrow j}_{\text{length } k}}^{[\tilde{\mathbf{Q}}^k]_{a,j}} \Leftarrow \overbrace{\underbrace{i \leftarrow \circ \cdots \circ \leftarrow \bullet \rightarrow \circ \cdots \circ \rightarrow \star}_{\text{all symmetric in-link paths for node-pair } (i,\star)}}^{[\mathbf{S}]_{i,\star}} \overbrace{\rightarrow}^{\mathbf{Q}^T} \overbrace{\underbrace{\blacktriangle \rightarrow \cdots \circ \rightarrow b}_{\text{length } k}}^{[(\tilde{\mathbf{Q}}^T)^k]_{\blacktriangle,b}}$
  - P2: $\overbrace{\underbrace{a \leftarrow \circ \cdots \circ \leftarrow}_{\text{length } k}}^{[\tilde{\mathbf{Q}}^k]_{a,j}} j \overbrace{\underbrace{\leftarrow \circ \cdots \circ \leftarrow \bullet \rightarrow \circ \cdots \circ \rightarrow}_{\text{all symmetric in-link paths for node-pair } (j,\star)}}^{[\mathbf{S}]_{j,\star}} \star \overbrace{\underbrace{\rightarrow \cdots \circ \rightarrow b}_{\text{length } k}}^{[(\tilde{\mathbf{Q}}^T)^k]_{\star,b}}$
  - P3: $\overbrace{\underbrace{a \leftarrow \circ \cdots \circ \leftarrow}_{\text{length } k}}^{[\tilde{\mathbf{Q}}^k]_{a,j}} j \overbrace{\underbrace{\rightarrow \circ \cdots \circ \rightarrow b}_{\text{length } k}}^{[(\tilde{\mathbf{Q}}^T)^k]_{j,b}}$

Unaffected Areas

- Since M merely tallies these paths, node-pairs without having such paths could be safely pruned.
- Iteratively Pruning:

Let $\mathcal{F}_1 := \{b \mid b \in \mathcal{O}(y),\ \exists y,\ s.t.\ [\mathbf{S}]_{i,y} \neq 0\}$

$$\mathcal{F}_2 := \begin{cases} \varnothing & (d_j = 0) \\ \{y \mid [\mathbf{S}]_{j,y} \neq 0\} & (d_j > 0) \end{cases}$$

$$\mathcal{A}_k \times \mathcal{B}_k :=$$

$$\begin{cases} \{j\} \times (\mathcal{F}_1 \cup \mathcal{F}_2 \cup \{j\}) & (k = 0) \\ \{(a, b) | \, a \in \tilde{\mathcal{O}}(x),\ b \in \tilde{\mathcal{O}}(y),\ \exists x,\ \exists y,\ s.t.\ [\mathbf{M}_{k-1}]_{x,y} \neq 0\} & (k > 0) \end{cases}$$

Then

$$[\mathbf{M}_k]_{a,b} = 0 \quad \textit{for all } (a, b) \notin (\mathcal{A}_k \times \mathcal{B}_k) \cup (\mathcal{A}_0 \times \mathcal{B}_0)$$

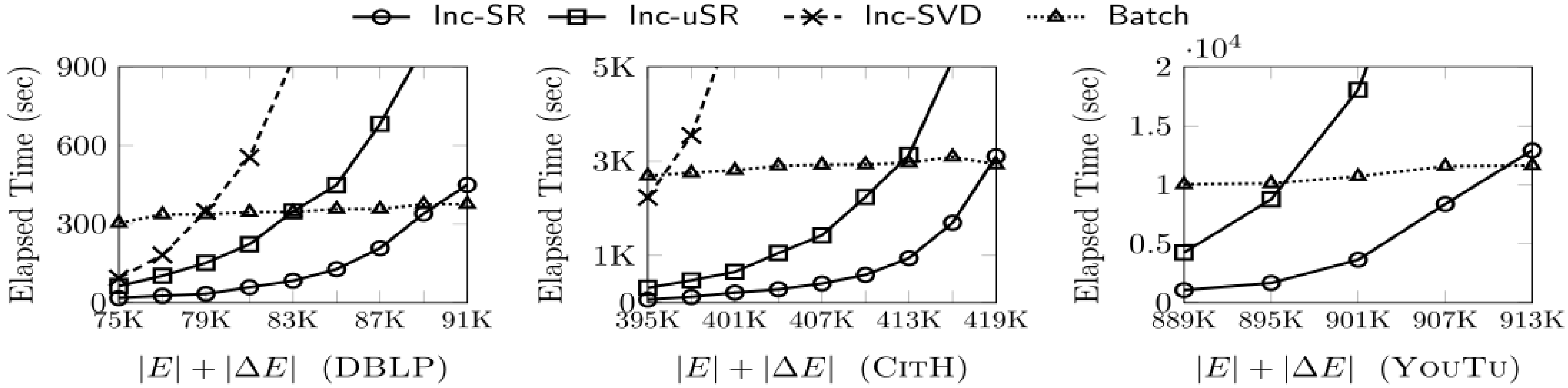
- Complexity: O(K(nd+|AFF|)) with

$$|\mathsf{AFF}| := \mathrm{avg}_{k \in [0,K]}(|\mathcal{A}_k| \cdot |\mathcal{B}_k|)$$

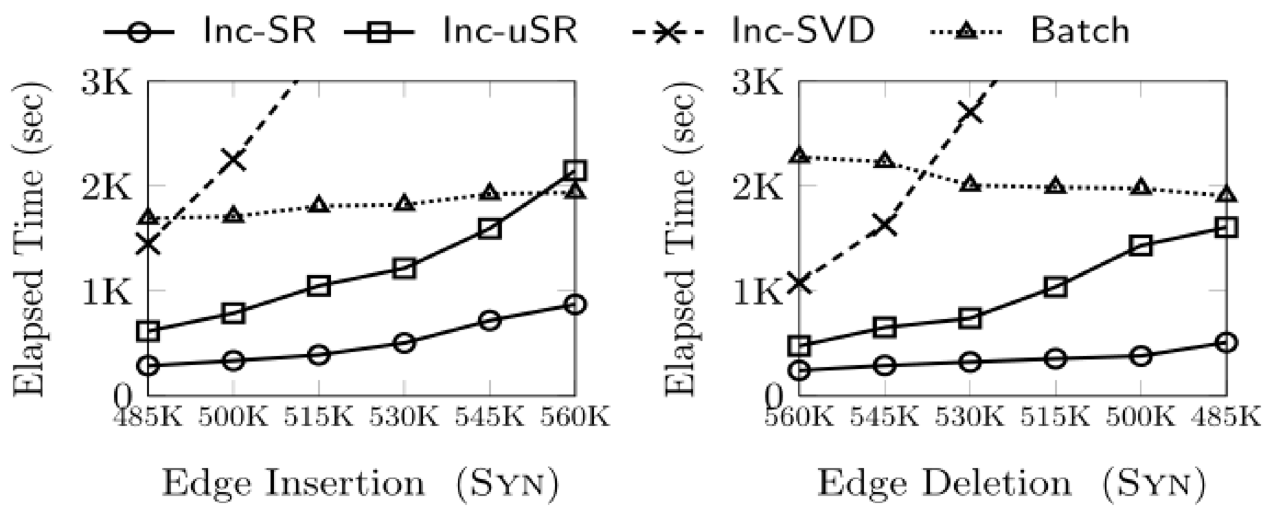
Experimental Settings

- Datasets
  - Real: DBLP, CITH, YOUTU
  - Synthetic: GraphGen generator
- Compared Algorithms
  - Inc-SR : Our Incremental SimRank with Pruning
  - Inc-uSR : Our Incremental SimRank without Pruning
  - Inc-SVD [EDBT ’10]: the best known link-update algorithm
  - Batch, the batch SimRank via fine-grained memoization
- Evaluations
  - Time Efficiency
  - Effectiveness of Pruning
  - Intermediate Memory
  - Exactness

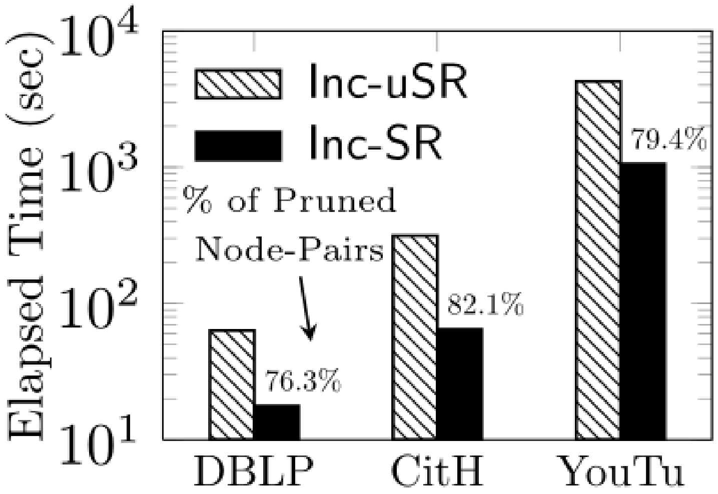
Time Efficiency



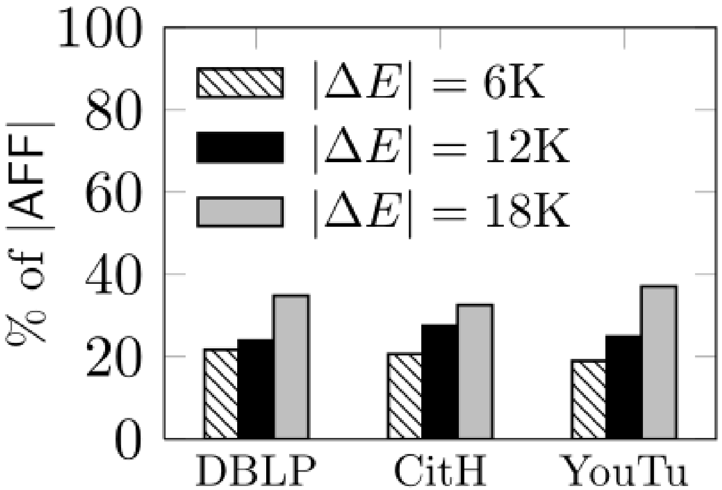
(a) Time Efficiency of Incremental SimRank on Real Data



Effectiveness of Pruning



(d) Effect of Pruning



(e) % of Affected Areas *w.r.t.* $|\Delta E|$

Intermediate Memory & Exactness

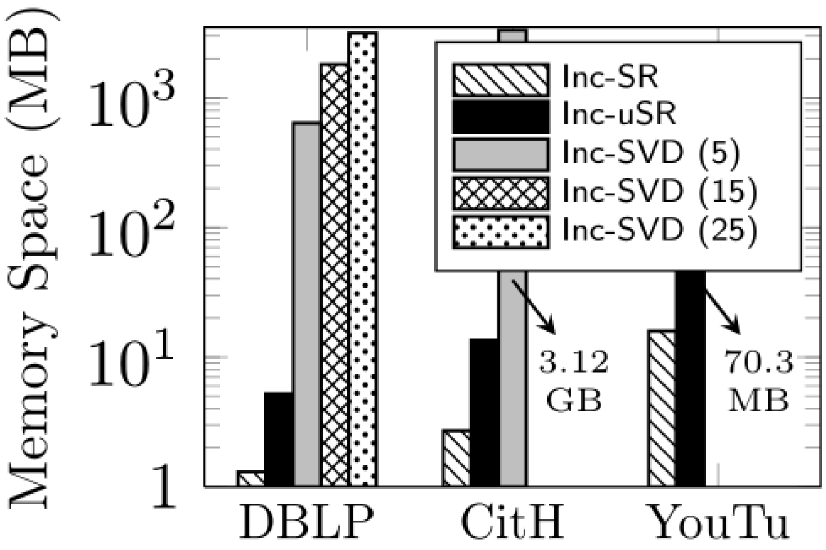


Fig. 3: Memory Space

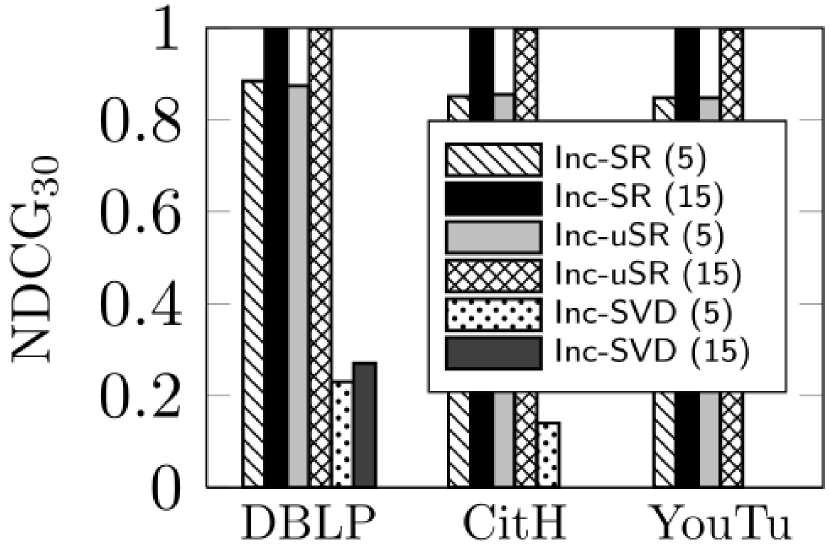


Fig. 4: $NDCG_{30}$ Exactness

Conclusions

- Two efficient methods are proposed to incrementally compute SimRank on link-evolving graphs
  - ΔS is characterized via a rank-one Sylvester equation, improving the time to $O(Kn^2)$ for every link update.
  - A pruning strategy skipping unnecessary recomputations, which further reduces the time to $O(K(nd + |AFF|))$.
- Empirical evaluations to show the superiority of our methods from several times to one order of magnitude.

Thank you!

Q/A