TPA: Fast, Scalable, and Accurate Method for Approximate Random Walk with Restart on Billion Scale Graphs

Minji Yoon, Jinhong Jung, and U Kang
Data Mining Lab
Dept. of CSE
Seoul National University

Outline

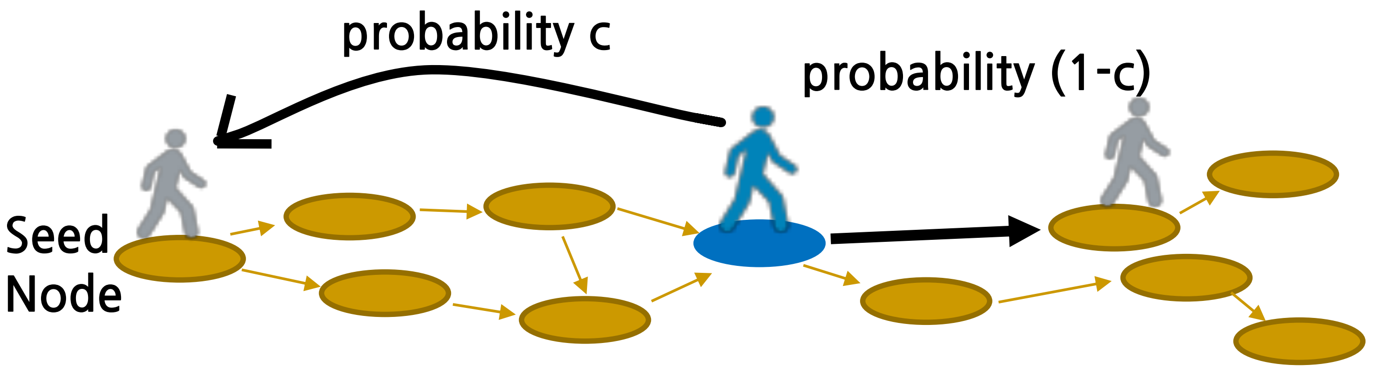
- **Problem Definition**
- Proposed Method
- Experiments
- Conclusion

Motivation

- Measuring similarity score between two nodes in a graph
  - Various applications across different domains
  - Ranking, Community detection, Link prediction, and Anomaly Detection.
- Random Walk with Restart (RWR)
  - Consider the global network from a particular user's point of view

Random Walk with Restart

- A random surfer
  - Start at seed node
  - Walk along edges with probability $(1 - c)$
  - Jump back to the seed node with probability c



Challenges

- A significant challenge on its computation
  - RWR scores are different across different seed nodes
  - Need to be recomputed for each new seed node
- Approximate RWR computation

Problem Definition
: Approximate RWR

- **Given:** adjacency matrix $\boldsymbol{A}$, seed node s, restart probability c
- **Find:** approximate RWR score vector $\mathbf{r}_{\boldsymbol{approx}}$ of exact RWR score vector $\mathbf{r}_{\boldsymbol{exact}}$ which satisfying:

$$\mathbf{r}_{\text{exact}} = (1-c)\widetilde{\mathbf{A}}^{\mathbf{T}}\mathbf{r}_{\text{exact}} + c\boldsymbol{q}_s$$

- **Input:**
  - $\widetilde{\mathbf{A}} \in \mathbb{R}^{n \times n}$: row-normalized adjacency matrix
  - $\mathbf{q}_s \in \mathbb{R}^{n \times 1}$: seed vector ($s$-th unit vector)
  - $c \in \mathbb{R}$: restart probability
- **Output**:
  - $\mathbf{r}_{\text{approx}} \in \mathbb{R}^{n \times 1}$: approximate RWR score vector

CPI: Cumulative Power Iteration

- Exact RWR computation method
- Re-interpretation of RWR
- Propagation of scores across a graph
  1) Score c is generated from the seed node
  2) At each step, scores are divided evenly into out-edges with decaying coefficient $(1 - c)$
  3) Each node accumulates scores they have received
  4) Accumulated scores become RWR score of each node

CPI: Cumulative Power Iteration

$$\mathbf{x}^{(0)} = c\mathbf{q}$$

1) Initial score c at seed node

$$\mathbf{x}^{(i)} = (1-c)\tilde{\mathbf{A}}^\top \mathbf{x}^{(i-1)} = c\left((1-c)\tilde{\mathbf{A}}^\top\right)^i \mathbf{q}$$

2) scores are divided evenly into out-edges with (1-c)

$$\mathbf{r}_{\text{CPI}} = \sum_{i=0}^{\infty} \mathbf{x}^{(i)} = c\sum_{i=0}^{\infty}\left((1-c)\tilde{\mathbf{A}}^\top\right)^i \mathbf{q}$$

3) CPI accumulate interim scores of each node to get final results

- $\mathbf{x}(i) \in \mathbb{R}^{n\times 1}$: interim score vector computed from i th iteration
- Correctness of CPI: Theorem 1
- For PageRank computation, the seed vector $\mathbf{q}$ is set to $\frac{1}{n}\mathbf{1}$

Outline

- Problem Definition
- **Proposed Method**
- Experiments
- Conclusion

TPA: Two Phase Approximation

- TPA approximates RWR scores with fast speed and high accuracy
  - CPI performs iterations until convergence
  - Divide the whole iterations in CPI into three parts as follows :

$$
\begin{aligned}
&\mathbf{r}_{\text{CPI}} \\
&= \mathbf{r}_{\text{family}} + \mathbf{r}_{\text{neighbor}} + \mathbf{r}_{\text{stranger}} \\
&= \underbrace{\mathbf{x}^{(0)} + \cdots + \mathbf{x}^{(S-1)}}_{\text{family part}} + \underbrace{\mathbf{x}^{(S)} + \cdots + \mathbf{x}^{(T-1)}}_{\text{neighbor part}} + \underbrace{\mathbf{x}^{(T)} + \cdots}_{\text{stranger part}}
\end{aligned}
$$

S : starting iteration of $r_{neighbor}$, T : starting iteration of $r_{stranger}$

TPA: Two Phase Approximation

$$\begin{aligned}
&\mathbf{r}_{\text{CPI}} \\
&= \mathbf{r}_{\text{family}} + \mathbf{r}_{\text{neighbor}} + \mathbf{r}_{\text{stranger}} \\
&= \underbrace{\mathbf{x}^{(0)} + \cdots + \mathbf{x}^{(S-1)}}_{\text{family part}} + \underbrace{\mathbf{x}^{(S)} + \cdots + \mathbf{x}^{(T-1)}}_{\text{neighbor part}} + \underbrace{\mathbf{x}^{(T)} + \cdots}_{\text{stranger part}}
\end{aligned}$$

$$\mathbf{r}_{\text{TPA}} = \mathbf{r}_{\text{family}} + \tilde{\mathbf{r}}_{\text{neighbor}} + \tilde{\mathbf{r}}_{\text{stranger}}$$

- 1st Phase: Stranger Approximation
  - Approximates $r_{stranger}$ in RWR using PageRank
- 2nd Phase: Neighbor Approximation
  - Approximates $r_{neighbor}$ using r_{family}

Stranger Approximation - Definition

- PageRank score vector p_{CPI} is represented by CPI as follows:

$$\mathbf{x}'^{(0)} = \frac{c}{n}\mathbf{1} \quad \mathbf{x}'^{(i)} = (1-c)\tilde{\mathbf{A}}^\top \mathbf{x}'^{(i-1)}$$

$$\begin{aligned}\mathbf{p}_{\text{CPI}} &= \mathbf{p}_{\text{family}} + \mathbf{p}_{\text{neighbor}} + \mathbf{p}_{\text{stranger}} \\ &= \underbrace{\mathbf{x}'^{(0)} + \cdots + \mathbf{x}'^{(S-1)}}_{\text{family part}} + \underbrace{\mathbf{x}'^{(S)} + \cdots + \mathbf{x}'^{(T-1)}}_{\text{neighbor part}} + \underbrace{\mathbf{x}'^{(T)} + \cdots}_{\text{stranger part}}\end{aligned}$$

- $r_{stranger}$ in RWR is approximated by $p_{stranger}$ in PageRank as follows:

$$\tilde{\mathbf{r}}_{\text{stranger}} = \mathbf{p}_{\text{stranger}}$$

Stranger Approximation - Intuition

- The amount of scores propagated into each node

1. # of in-edges
   - Nodes with many in-edges have many sources to receive scores
2. Distance from seed node
   - Scores are decayed by factor (1−c) as iteration progresses
   - Nodes close to the seed node take in high scores

Stranger Approximation - Intuition

- In stranger iterations
  - Scores $(x(T), x(T+1), \cdots)$ are mainly determined by # in-edges
  - **Nodes are already far from seed**
- PageRank is solely determined by arrangement of edges (= # in-edges) !!
  - Motivation of Stranger Approximation
  - Estimate stranger iterations in RWR with those in PageRank
- Precompute $\tilde{r}_{stranger}$ in preprocessing phase

TPA: Two Phase Approximation

$$\mathbf{r}_{\text{CPI}} = \mathbf{r}_{\text{family}} + \mathbf{r}_{\text{neighbor}} + \mathbf{r}_{\text{stranger}} = \underbrace{\mathbf{x}^{(0)} + \cdots + \mathbf{x}^{(S-1)}}_{\text{family part}} + \underbrace{\mathbf{x}^{(S)} + \cdots + \mathbf{x}^{(T-1)}}_{\text{neighbor part}} + \underbrace{\mathbf{x}^{(T)} + \cdots}_{\text{stranger part}}$$

$$\mathbf{r}_{\text{TPA}} = \mathbf{r}_{\text{family}} + \tilde{\mathbf{r}}_{\text{neighbor}} + \tilde{\mathbf{r}}_{\text{stranger}}$$

- 1st Phase: Stranger Approximation
  - Approximates $r_{stranger}$ in RWR using PageRank
- 2nd Phase: Neighbor Approximation
  - Approximates $r_{neighbor}$ using r_{family}

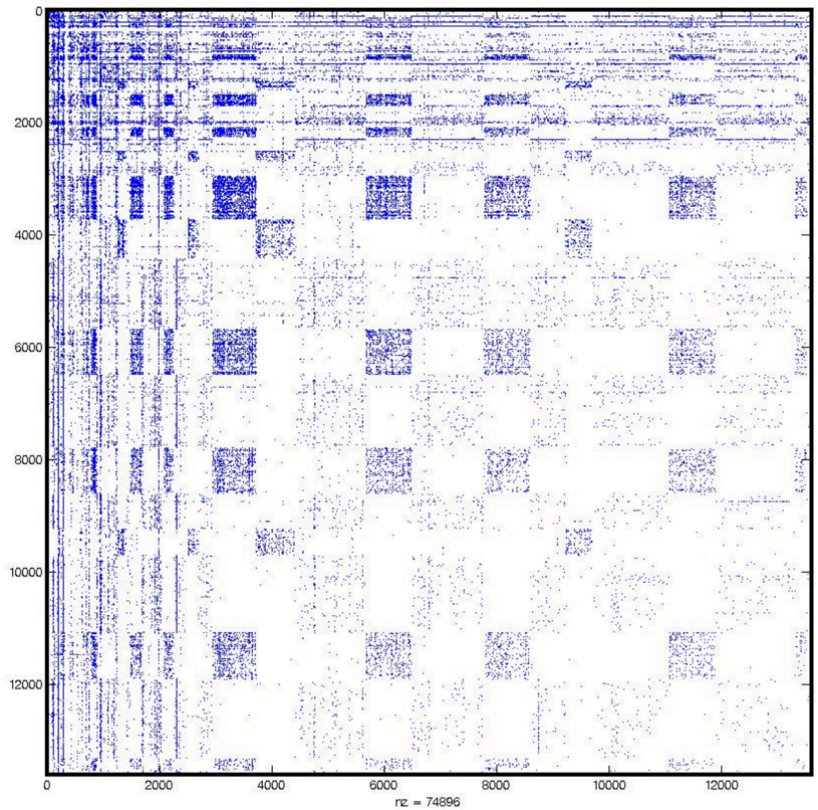
Neighbor Approximation - Definition

- The neighbor approximation
  - Limit computation to r_{family}
  - Estimate $r_{neighbor}$ by scaling r_{family} as follows:

$$\tilde{\mathbf{r}}_{\text{neighbor}} = \frac{\|\mathbf{r}_{\text{neighbor}}\|_1}{\|\mathbf{r}_{\text{family}}\|_1}\mathbf{r}_{\text{family}} = \frac{(1-c)^S - (1-c)^T}{1-(1-c)^S}\mathbf{r}_{\text{family}}$$

L1 length of r_{family} and $r_{neighbor}$ is proved in **Lemma2**

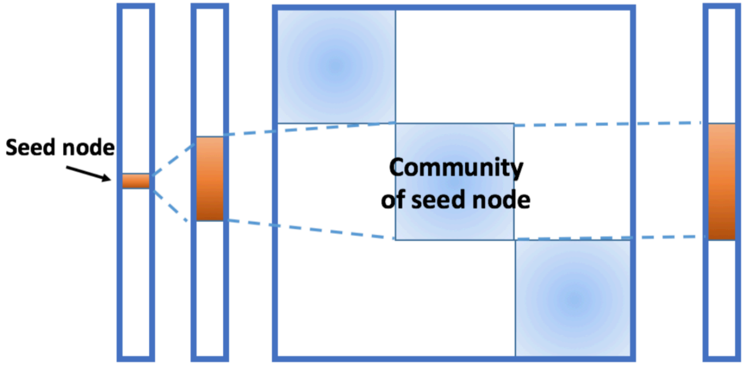
Neighbor Approximation - Intuition



Block-wise,
Community-like
structure
of real-world graphs[1]

[1] U. Kang and C. Faloutsos. Beyond 'caveman communities': Hubs and spokes for graph compression and mining. In *ICDM*, 2011

Neighbor Approximation - Intuition



- Nodes which receive scores in the early iterations (family part)
  - Would receive scores again in the following iterations (neighbor part)
- Nodes which have more in-edges thus receive more scores in the early iterations
  - Would receive more scores than other nodes in the following iterations.

Neighbor Approximation - Intuition

- Maintain ratios of scores among nodes
- Scale the scores in r_{family} to reflect smaller amount of scores in $r_{neighbor}$
  - Scores in the following iterations would be smaller than previous scores
  - Decaying coefficient (1 − c)

$$\tilde{\mathbf{r}}_{\text{neighbor}} = \frac{||\mathbf{r}_{\text{neighbor}}||_1}{||\mathbf{r}_{\text{family}}||_1}\mathbf{r}_{\text{family}}$$

TPA: Two Phase Approximation

Exact RWR: $\mathbf{r}_{\text{CPI}} = \mathbf{r}_{\text{family}} + \mathbf{r}_{\text{neighbor}} + \mathbf{r}_{\text{stranger}}$

Preprocessing phase

$\mathbf{p}_{\text{CPI}} = \mathbf{p}_{\text{family}} + \mathbf{p}_{\text{neighbor}} + \mathbf{p}_{\text{stranger}}$

$\tilde{\mathbf{r}}_{\text{stranger}} \leftarrow \mathbf{p}_{\text{stranger}}$ **: Stranger approximation**

Online phase

Compute $\mathbf{r}_{\text{family}}$ **using CPI**

$\tilde{\mathbf{r}}_{\text{neighbor}} \leftarrow \frac{\|\mathbf{r}_{\text{neighbor}}\|_1}{\|\mathbf{r}_{\text{family}}\|_1} \mathbf{r}_{\text{family}}$ **: Neighbor approximation**

Approximate RWR: $\mathbf{r}_{\text{TPA}} = \mathbf{r}_{\text{family}} + \tilde{\mathbf{r}}_{\text{neighbor}} + \tilde{\mathbf{r}}_{\text{stranger}}$

Selecting S and T

$$\begin{aligned}&\mathbf{r}_{\text{CPI}}\\&=\mathbf{r}_{\text{family}}+\mathbf{r}_{\text{neighbor}}+\mathbf{r}_{\text{stranger}}\\&=\underbrace{\mathbf{x}^{(0)}+\cdots+\mathbf{x}^{(S-1)}}_{\text{family part}}+\underbrace{\mathbf{x}^{(S)}+\cdots+\mathbf{x}^{(T-1)}}_{\text{neighbor part}}+\underbrace{\mathbf{x}^{(T)}+\cdots}_{\text{stranger part}}\end{aligned}$$

- The starting iteration S of the neighbor approximation
  - Accuracy and Speed
  - with large S, running time increases: computation for r_{family}
  - with small S, error increases : a portion of exact computation decreases

Selecting S and T

$$
\begin{aligned}
&\mathbf{r}_{\text{CPI}} \\
&= \mathbf{r}_{\text{family}} + \mathbf{r}_{\text{neighbor}} + \mathbf{r}_{\text{stranger}} \\
&= \underbrace{\mathbf{x}^{(0)} + \cdots + \mathbf{x}^{(S-1)}}_{\text{family part}} + \underbrace{\mathbf{x}^{(S)} + \cdots + \boxed{\mathbf{x}^{(T-1)}}}_{\text{neighbor part}} + \underbrace{\mathbf{x}^{(T)} + \cdots}_{\text{stranger part}}
\end{aligned}
$$

- The starting iteration T of the stranger approximation
  - Accuracy
  - with small T, error(Stranger Approximation) increase
    - Effect of PageRank > Effect of seed node.
  - with large T, error(Neighbor Approximation) increase
    - Assumption of Neighbor Approximation

Outline

- Problem Definition
- Proposed Method
- **Experiments**
- Conclusion

Experimental Questions

- Q1. Performance
  - How much does TPA enhance the computational efficiency compared with its competitors?
- Q2. Accuracy
  - How much does TPA sacrifice accuracy?
- Q3. Effects of parameters
  - How does S affect the accuracy and speed of TPA?
  - How does T affect the accuracy of TPA?

Experimental Settings

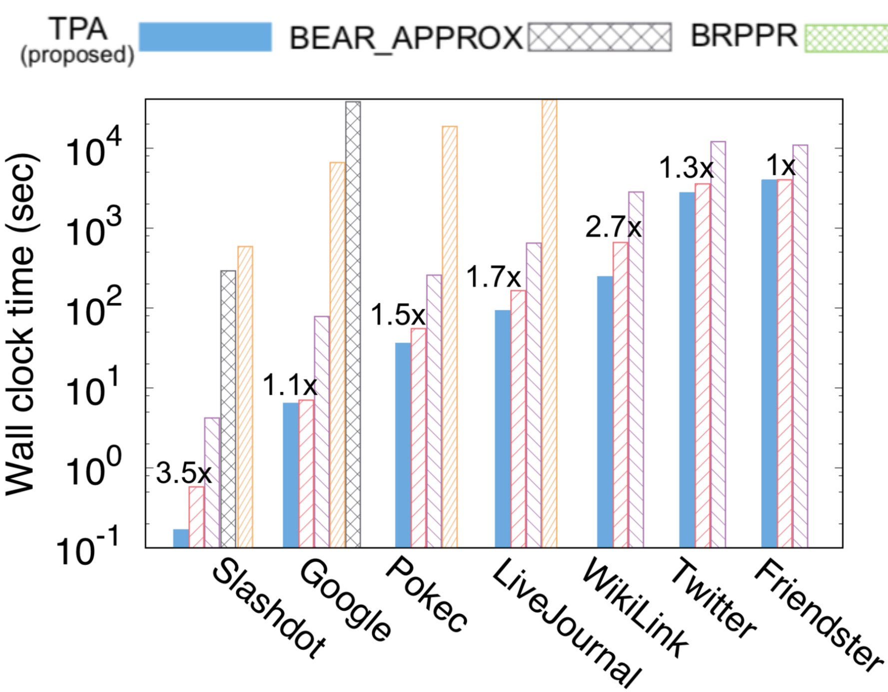
- Machine: single workstation with 200GB memory
- Datasets: large-scale real-world graph data

Dataset	Nodes	Edges	S	T
Friendster[1]	68,349,466	2,586,147,869	4	20
Twitter[1]	41,652,230	1,468,365,182	4	6
WikiLink[1]	12,150,976	378,142,420	5	6
LiveJournal[1]	4,847,571	68,475,391	5	10
Pokec[1]	1,632,803	30,622,564	5	10
Google[1]	875,713	5,105,039	5	20
Slashdot[1]	82,144	549,202	5	15

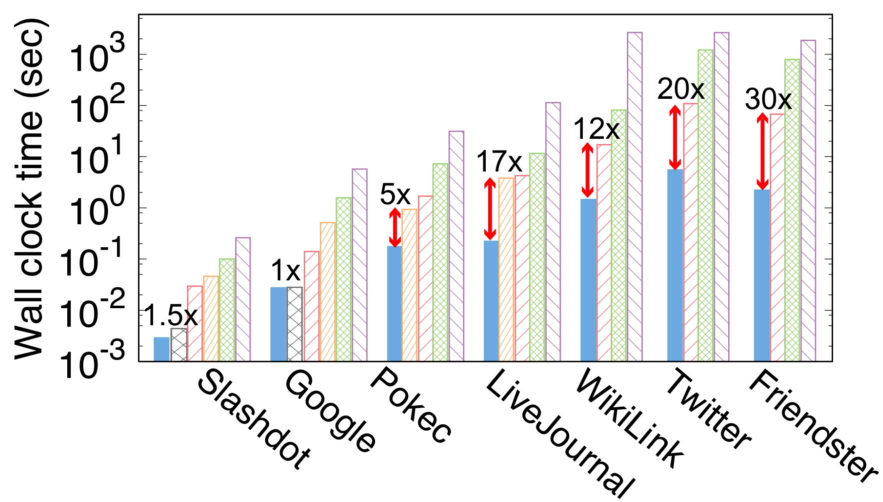
[1] http://konect.uni-koblenz.de/

Q1. Performance of TPA: Speed

- How long does TPA take for preprocessing phase and online phase, respectively?



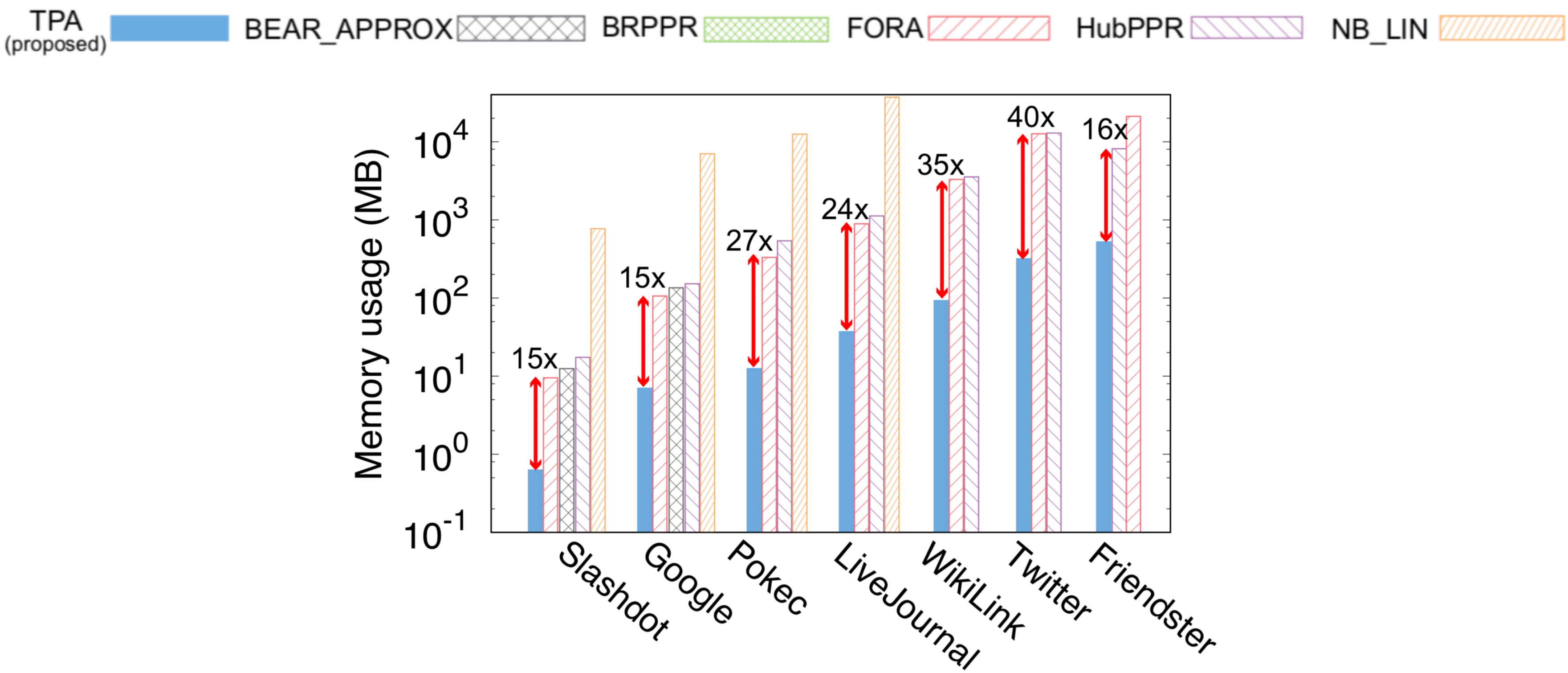
(a) Preprocessing Time



(b) Online Time

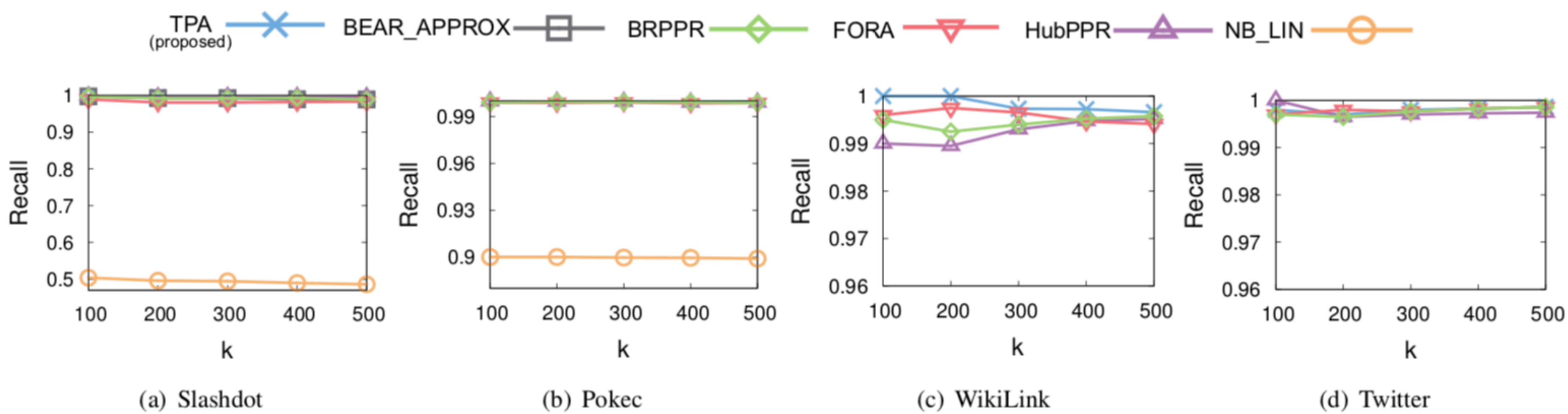
Q1. Performance of TPA: Memory Usage

- How much memory space does TPA requires for preprocessed results?



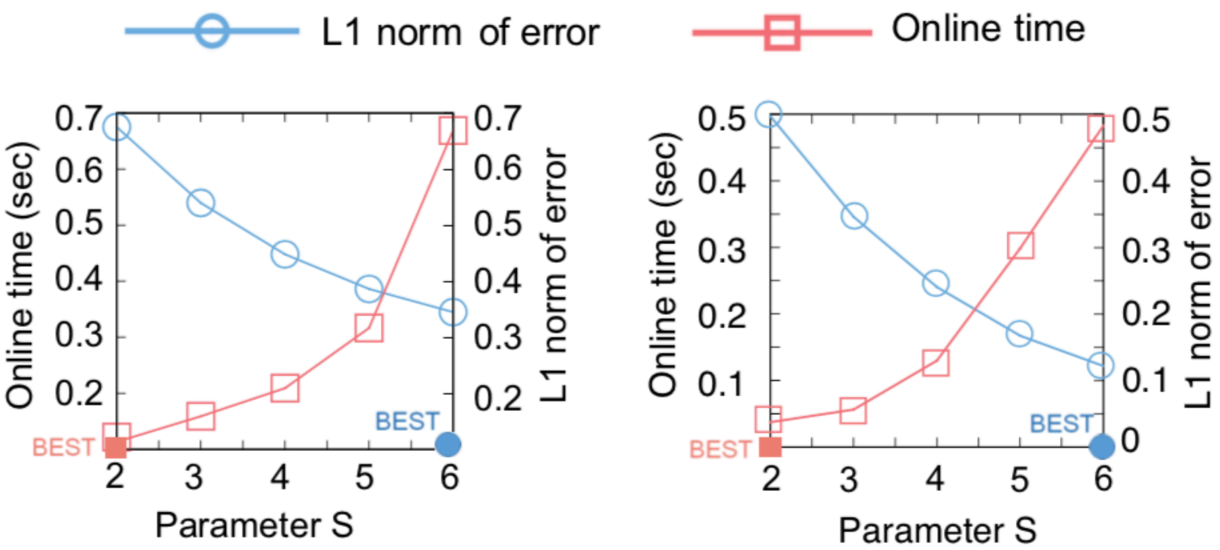
Q2. Accuracy of TPA

- Recall of Top-k RWR nodes
  - Twitter's "Who to Follow": top-500 ranked users
- How much does TPA sacrifice its accuracy?



Q3. Effects of Parameters - S

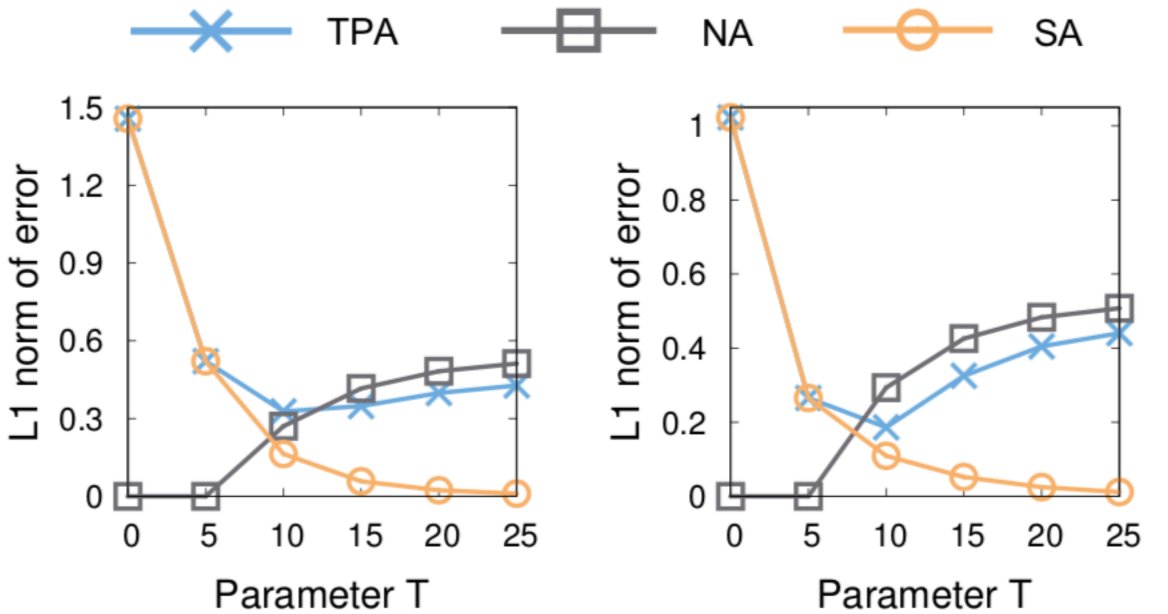
- How does the starting iteration S of Neighbor Approximation affect the accuracy and speed of TPA?



(a) Online time vs. L1 norm of error on the LiveJournal dataset (b) Online time vs. L1 norm of error on the Pokec dataset

Q3. Effects of Parameters - T

- How does the starting iteration T of Stranger Approximation affect the accuracy of TPA?



(a) L1 norm of error on the LiveJournal dataset (b) L1 norm of error on the Pokec dataset

Outline

- Problem Definition
- Proposed Method
- Experiments
- **Conclusion**

Conclusion

- TPA (Two Phase Approximation)
  - Neighbor Approximation
    - block-wise structure of real-world graphs
  - Stranger Approximation
    - PageRank

- Main Results
  - Requires 40x less memory space & preprocesses 3.5x faster than other preprocessing methods
  - Computes RWR 30x faster than other existing methods in online phase
  - Maintaining high accuracy

Thank you !

Codes & datasets
http://datalab.snu.ac.kr/tpa