IRWR: Incremental Random Walk with Restart

THE UNIVERSITY OF NEW SOUTH WALES

Weiren Yu, Xuemin Lin
School of Computer Science & Engineering, University of New South Wales, Sydney, Australia

RWR Overview

- Random Walk with Restart (RWR)
  - An appealing link-based node-proximity measure (ICDM '06)
  - Basic philosophy

    The proximity of u w.r.t. query q is defined as the limiting probability that a random surfer, starting from q, and then either moving to one of its out-neighbors, or restarting from q, will eventually arrive at u.
- Matrix Formula

restarting prob.

$$[\mathbf{P}]_{\star,q} = (1-c)\cdot \mathbf{A}\cdot [\mathbf{P}]_{\star,q} + c\cdot \mathbf{e}_q$$

proximities of node u w.r.t. query q

column normalized adjacency matrix

Motivations

- Prior Work
  - Existing computational methods are in a batch style, with the aim to accelerate the matrix inversion:

$$[\mathbf{P}]_{\star,q} = c(\mathbf{I}-(1-c)\cdot\mathbf{A})^{-1}\cdot \mathbf{e}_q$$

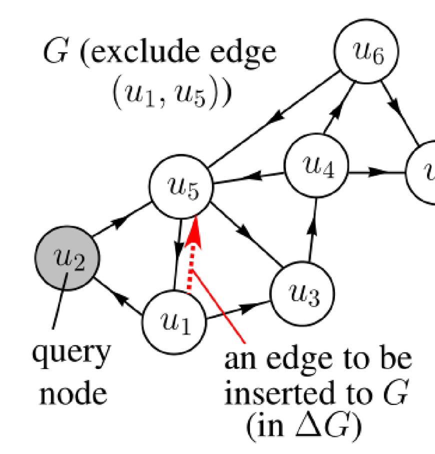
  - B_LIN and NB_LIN need O(|V|^2) time and space for computing all node proximities via SVD. [ICDM '06]
  - k-dash uses LU decomposition, yielding O(|E| + |V|), to find top-k highest proximity. [PVLDB '12]
- Our Contributions
  - devise an exact and fast incremental algorithm (IRWR) computing any node proximity in O(1) time for every link update

Problem Statement

- Problem (Incremental Update for RWR)

Given: a graph G, proximities P for G, changes ΔG to G, a query node q, and a restarting probability c ∈ (0,1).

Compute: changes to the proximities w.r.t. q exactly.



$$0.25\times[\mathbf{P}]_{\star,u_1} - 0.32\times[\mathbf{P}]_{\star,u_5} + 0.06\times\mathbf{e}_5 = [\mathbf{\Delta P}]_{\star,u_2}$$

$$\mathbf{P} = \begin{bmatrix} .31 & .25 & .18 & .14 & .13 & .11 & .10 \\ .08 & .26 & .07 & .06 & .10 & .04 & .04 \\ .11 & .09 & .30 & .24 & .14 & .19 & .17 \\ .14 & .11 & .13 & .30 & .17 & .24 & .22 \\ .29 & .23 & .26 & .21 & .36 & .17 & .15 \\ .08 & .06 & .07 & .06 & .10 & .24 & .12 \\ 0 & 0 & 0 & 0 & 0 & 0 & .20 \end{bmatrix} \rightarrow \begin{bmatrix} .036 \\ -.011 \\ -.016 \\ -.020 \\ .022 \\ -.011 \\ 0 \end{bmatrix}$$

adding edge (u_1,u_5) incurs change $[\mathbf{\Delta P}]_{\star,u_2}$

Challenges

- Observations
  - The update proximity of RWR can be expressed as the linear combination of the old proximities.
  - Convert matrix-vector multiplications into vector scaling and additions.
- Challenge
  - It seems hard to determine the scalars α, β, λ for

$$[\mathbf{\Delta P}]_{\star,u_2} = \alpha\cdot[\mathbf{P}]_{\star,u_1} + \beta\cdot[\mathbf{P}]_{\star,u_5} + \lambda\cdot\mathbf{e}_5$$

Our Results on IRWR

- For edge update (i, j), the changes to P can be computed as

$$[\mathbf{\Delta P}]_{\star,q} = \frac{(1-c)[\mathbf{P}]_{j,q}}{1-(1-c)[\mathbf{y}]_j}\cdot \mathbf{y}, \text{ where}$$

For edge insertion,

$$\mathbf{y} = \begin{cases} \frac{1}{c}[\mathbf{P}]_{\star,i} & (d_j=0) \\ \frac{1}{c(d_j+1)}\left([\mathbf{P}]_{\star,i} - \frac{1}{1-c}[\mathbf{P}]_{\star,j}\right) + \frac{1}{(1-c)(d_j+1)}\mathbf{e}_j & (d_j>0) \end{cases}$$

For edge deletion,

$$\mathbf{y} = \begin{cases} -\frac{1}{c}[\mathbf{P}]_{\star,i} & (d_j=1) \\ \frac{1}{c(d_j-1)}\left(\frac{1}{1-c}[\mathbf{P}]_{\star,j} - [\mathbf{P}]_{\star,i}\right) - \frac{1}{(1-c)(d_j-1)}\mathbf{e}_j & (d_j>1) \end{cases}$$

Key Rationale

$$\tilde{\mathbf{A}} = \mathbf{A} + \mathbf{a}\mathbf{e}_j^T \;\; with\ \mathbf{a} = \begin{cases} \mathbf{e}_i & (d_j=0) \\ \frac{1}{d_j+1}(\mathbf{e}_i - [\mathbf{A}]_{\star,j}) & (d_j>0) \end{cases}$$

Rank-one update

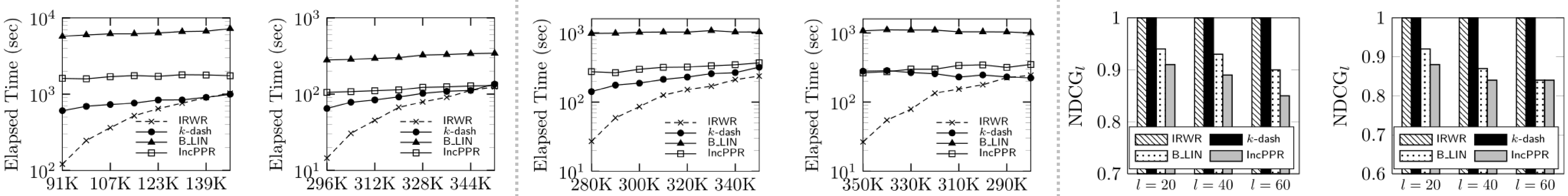
$$[\tilde{\mathbf{P}}]_{\star,q} = [\mathbf{P}]_{\star,q} + (1-c)\gamma\cdot\mathbf{z}$$

$$with\ \gamma = \frac{[\mathbf{P}]_{j,q}}{1-(1-c)\cdot[\mathbf{z}]_j} \quad and\ \ \mathbf{z} = (\mathbf{I}-(1-c)\mathbf{A})^{-1}\mathbf{a}$$

$$(\mathbf{I}-(1-c)\mathbf{A})^{-1}\mathbf{A} = \sum_{k=0}^{\infty}(1-c)^k\mathbf{A}^{k+1}$$
$$= \frac{1}{1-c}((\mathbf{I}-(1-c)\mathbf{A})^{-1}-\mathbf{I}) = \frac{1}{1-c}(\tfrac{1}{c}\mathbf{P}-\mathbf{I})$$

Avoid matrix inversion

Experimental Evaluations



(a) p2p-Gnutella (b) cit-HepPh (c) edge insertions (d) edge deletions (e) p2p-Gnutella (f) cit-HepPh

Speedup on Real Datasets | Speedup on Synthetic Datasets | Exactness on Real Datasets