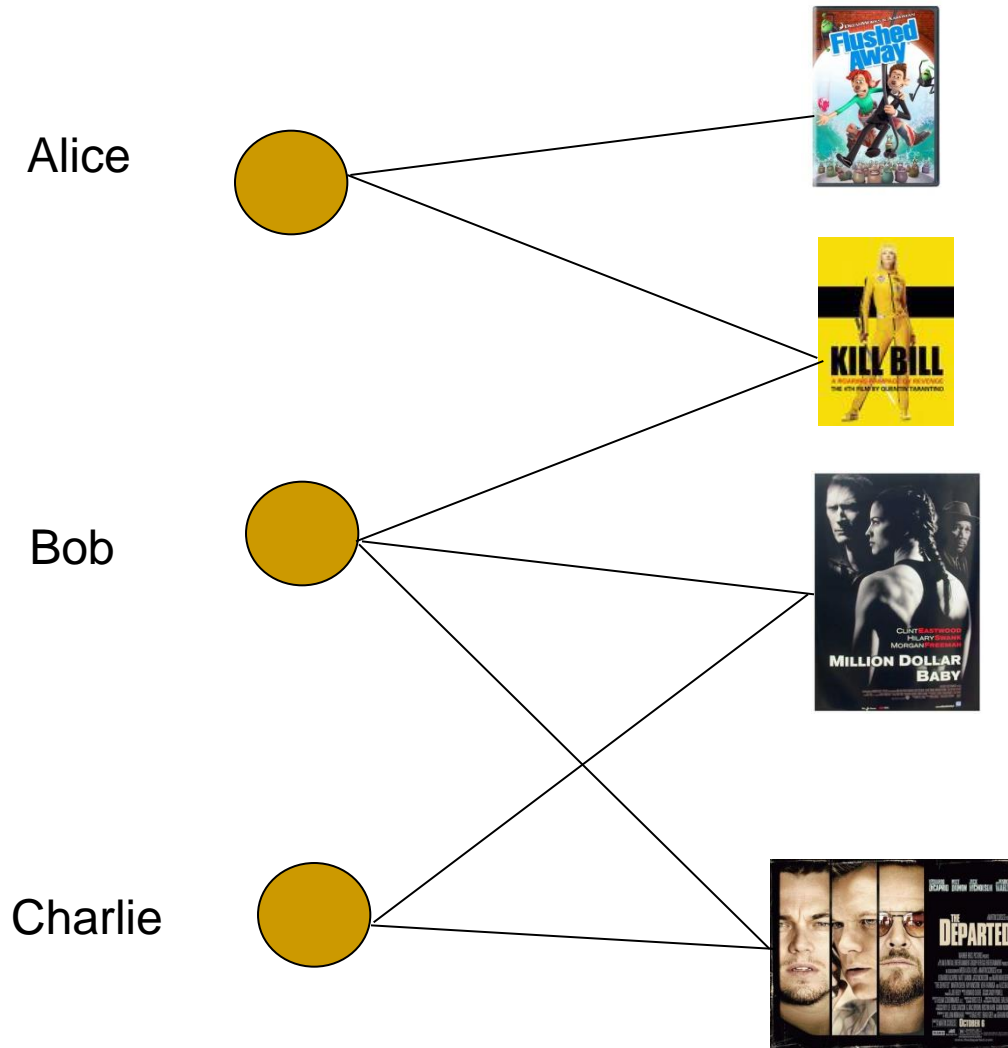


Random Walk based Proximity Measures in Directed Graphs

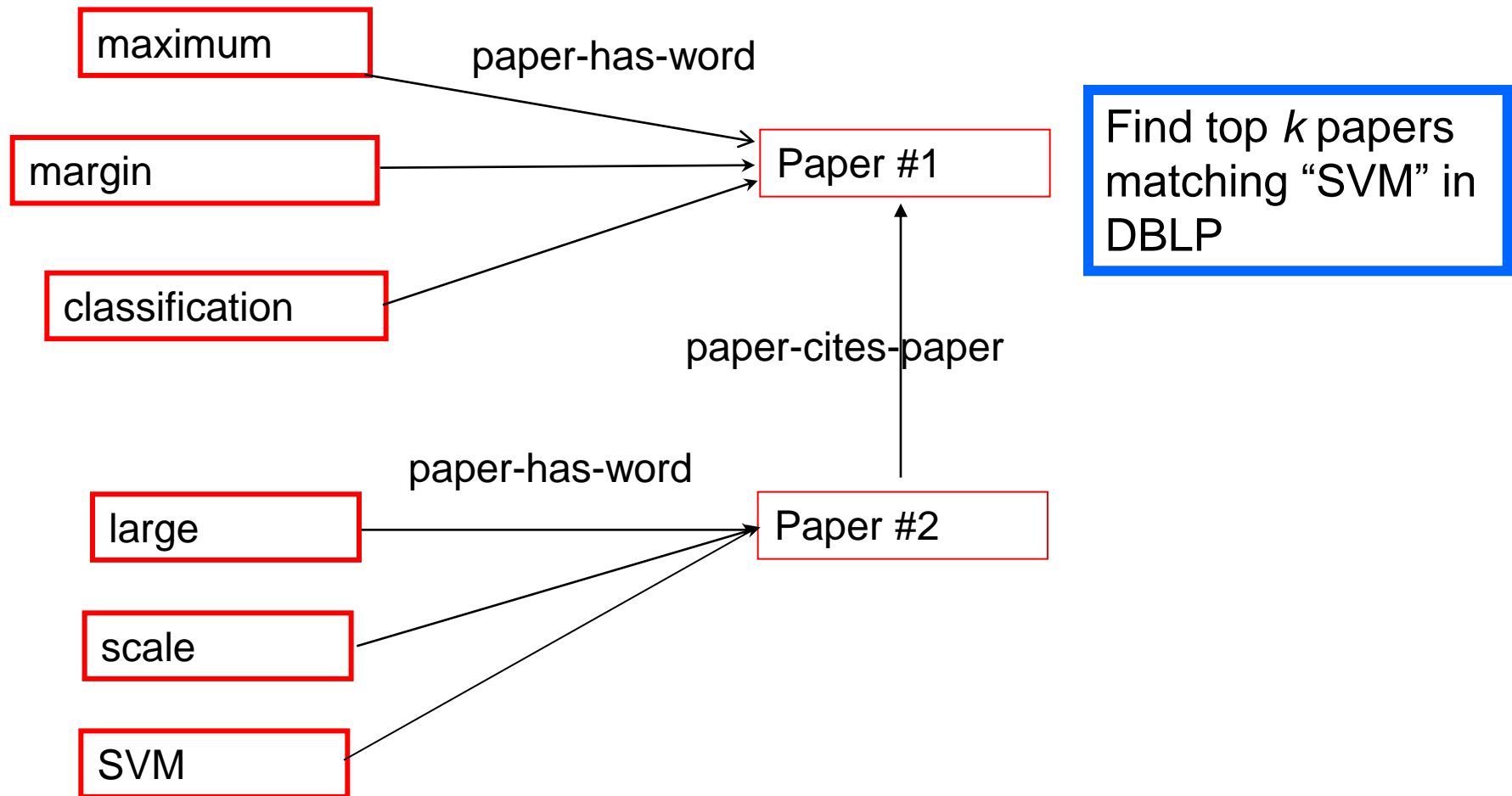
Speaker: 李 寰

Recommender systems¹



What are the top k movie recommendations for Alice in IMDB?

Content-based search in databases^{1, 2}



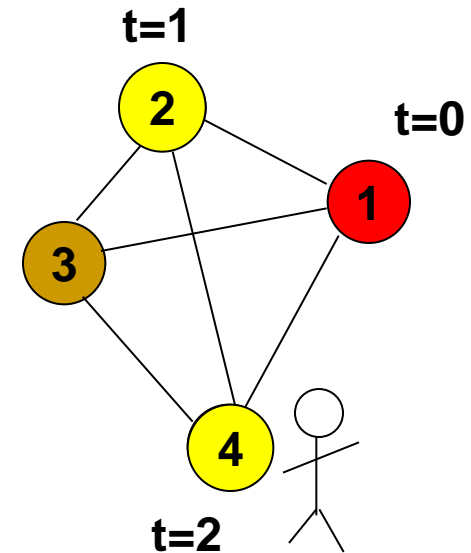
1. S. Chakrabarti. Dynamic personalized pagerank in entity-relation graphs. *WWW '07*.
2. A. Balmin, V. Hristidis, & Y. Papakonstantinou. ObjectRank: Authority-based keyword search in databases. *VLDB '04*.

Random walk based proximity measures in directed graphs

- Personalized pagerank
 - G. Jeh & J. Widom (*WWW '03*)
 - Truncated hitting and commute times
 - P. Sarkar, A. Moore, & A. Prakash (*ICML '08*)
 - Escape probability
 - H. Tong, Y. Koren, & C. Faloutsos (*KDD '07*)
-

Random walks

- Start at 1
- Move to a neighbor j randomly
- Continue



- Transition matrix¹ $P = [p(i, j)]$
 - $p(i, j) \triangleq \Pr[i \text{ moves to } j]$
 - $p_t \triangleq$ probability vector at time t
 - $p_{t+1} = p_t P$

$$P = \begin{pmatrix} 0 & \frac{1}{3} & \frac{1}{3} & \frac{1}{3} \\ \frac{1}{3} & 0 & \frac{1}{3} & \frac{1}{3} \\ \frac{1}{3} & \frac{1}{3} & 0 & \frac{1}{3} \\ \frac{1}{3} & \frac{1}{3} & \frac{1}{3} & 0 \end{pmatrix}$$

Personalized pagerank

- Stationary distribution $\pi = \pi P$

- Pagerank¹

- Rank web-pages by distribution satisfying

$$\mathbf{v} = (1 - \alpha)\mathbf{v}P + \frac{\alpha}{n}\mathbf{1}$$

- Personalized pagerank²

- Using a non-uniform restart distribution

$$\mathbf{v} = (1 - \alpha)\mathbf{v}P + \alpha\mathbf{r}$$

- e.g. $\mathbf{r} = \mathbf{e}_i$ when computing proximities from node i

1. S. Brin & L. Page. The anatomy of a large-scale hypertextual web search engine. *WWW* '98.

2. G. Jeh & J. Widom. Scaling personalized web search. *WWW* '03.

Hitting and commute times

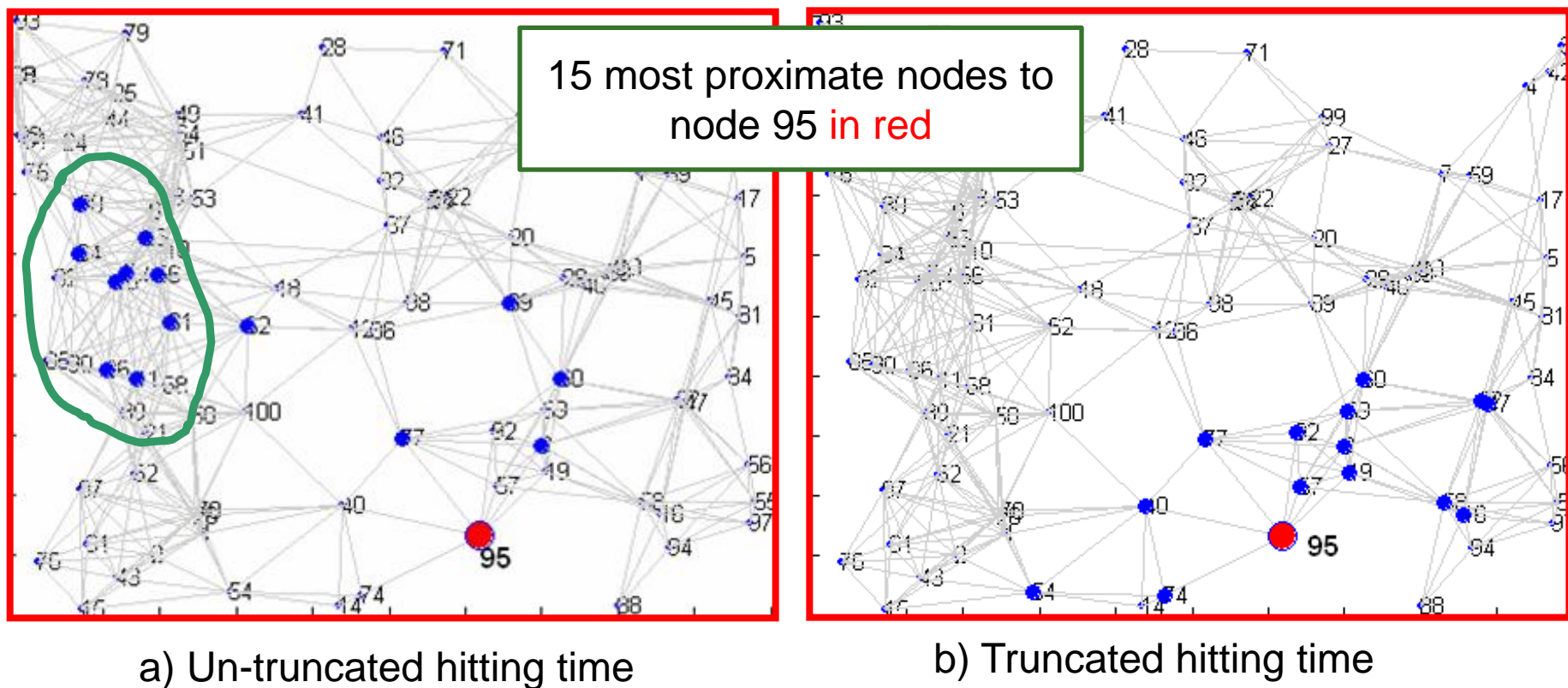
- Hitting time $h(i, j)$
 - Expected length of the path $i \longrightarrow j$
- Commute time $c(i, j) = h(i, j) + h(j, i)$
 - Expected length of the path $i \longrightarrow j \longrightarrow i$
- Drawbacks^{1, 2}
 - Take into account very long paths
 - $h(i, j)$ is small whenever j has a large stationary probability π_j
 - Alice likes cartoons, so her top 10 recommendations should not be the 10 most popular movies

1. D. Liben-Nowell & J. Kleinberg. The link predication problem for social networks. *CIKM* '03.

2. M. Brand. A random walks perspective on maximizing satisfaction and profit. *SIAM* '05.

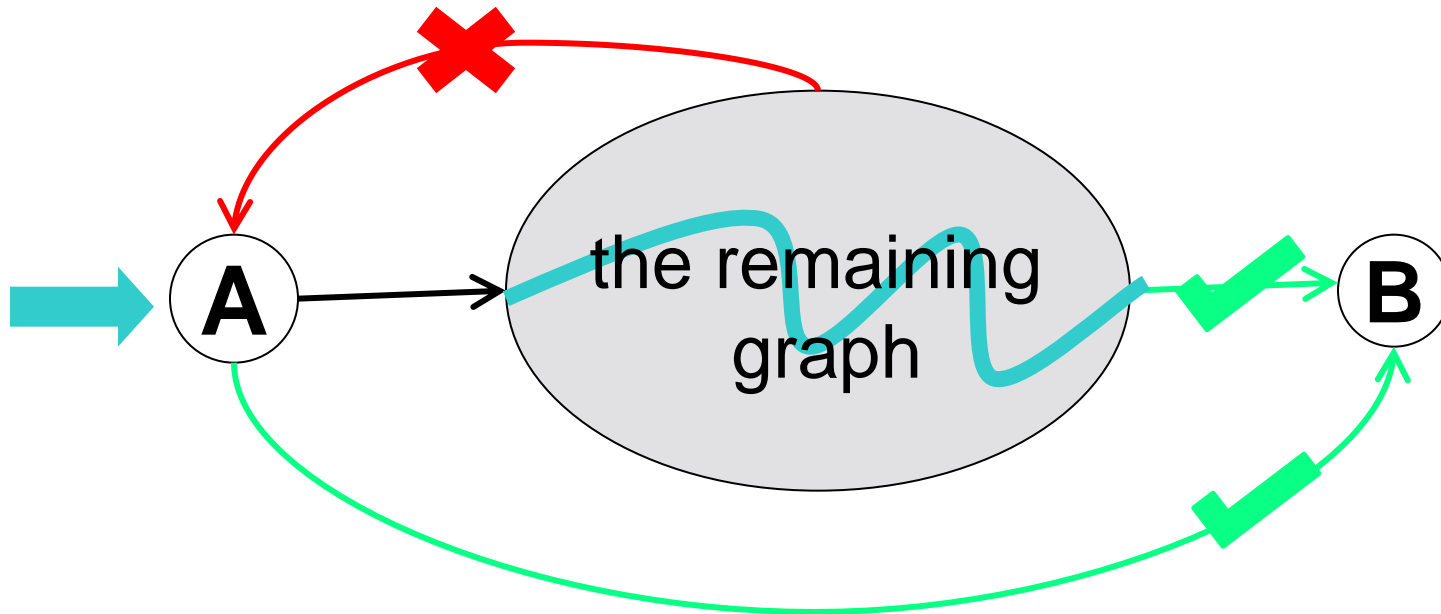
Truncated hitting and commute times¹

- Truncated version of hitting time and commute times
 - Only considers paths of length at most T



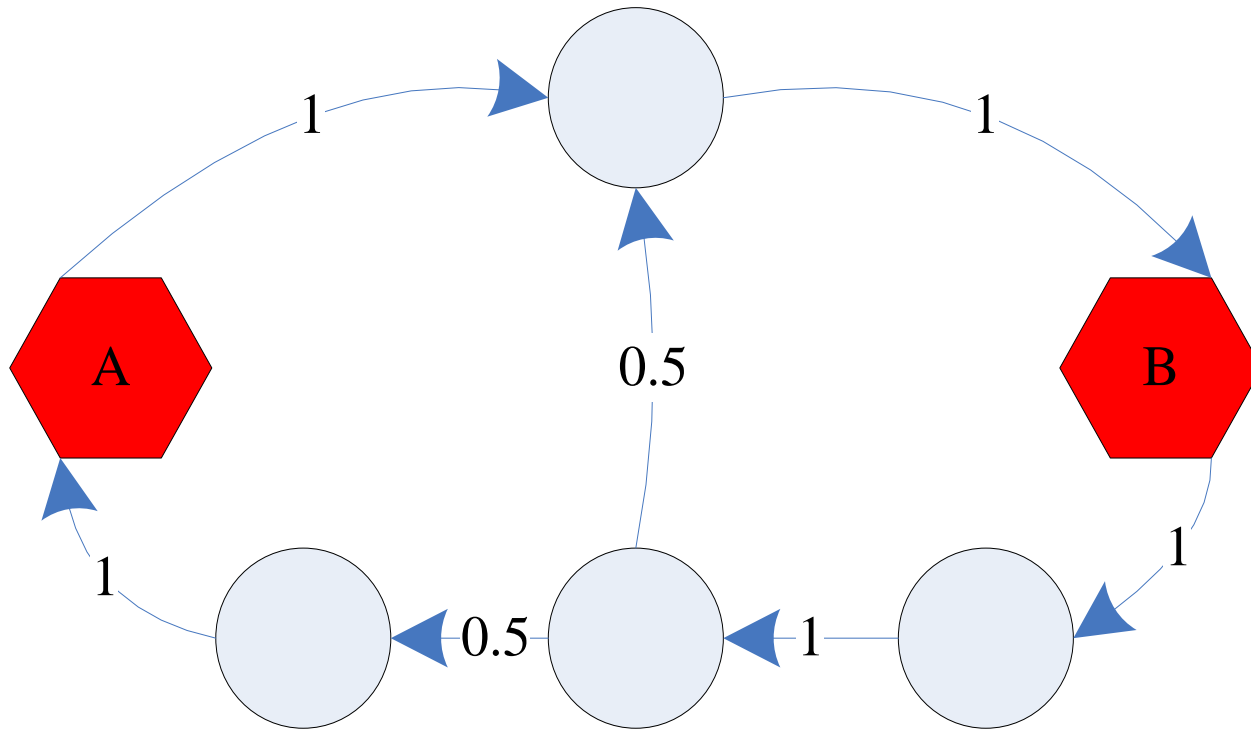
Escape probability¹

- The escape probability from node A to node B
 - Denoted as $\text{ep}(A \rightarrow B)$
 - $\text{Pr} [\text{starting at } A, \text{ reaches } B \text{ before returning to } A]$



$$\text{ep}(A \rightarrow B) = \text{Pr} \left[\text{✓ comes before ✗} \right]$$

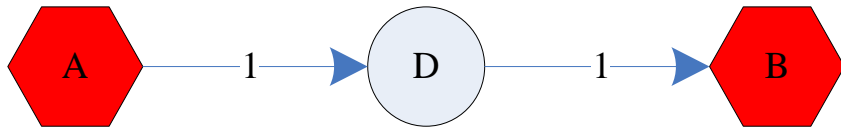
Asymmetry of escape probability



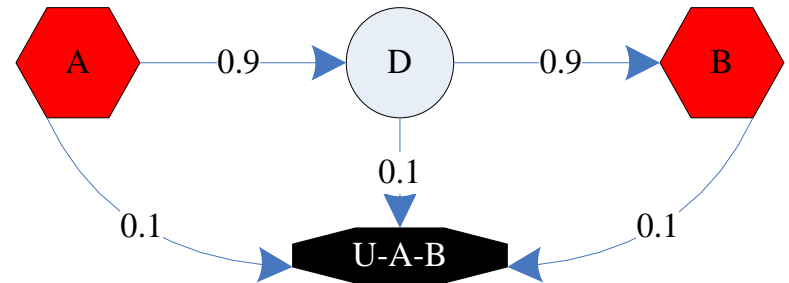
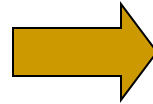
$$\text{ep}(A \rightarrow B) = 1 > \text{ep}(B \rightarrow A) = 0.5$$

Issue 1: “Degree-1 node” effect

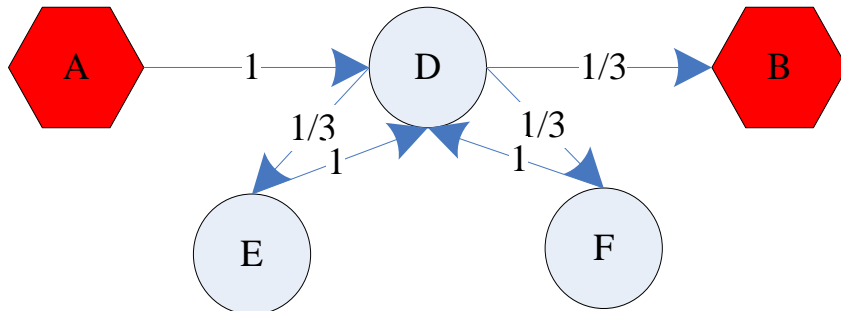
- Adding an absorbing node



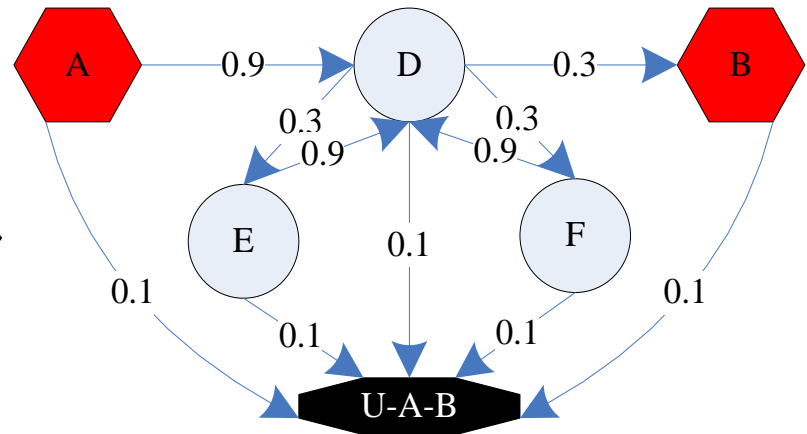
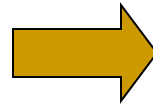
$$\text{ep}(A \rightarrow B) = 1$$



$$\text{ep}(A \rightarrow B) = 0.91$$

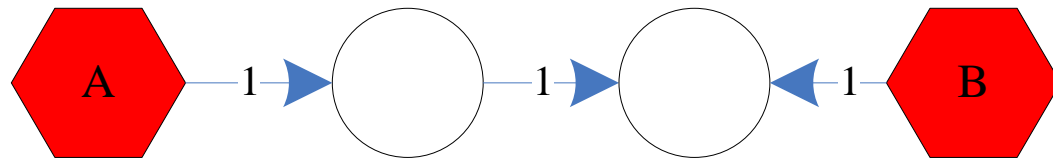


$$\text{ep}(A \rightarrow B) = 1$$



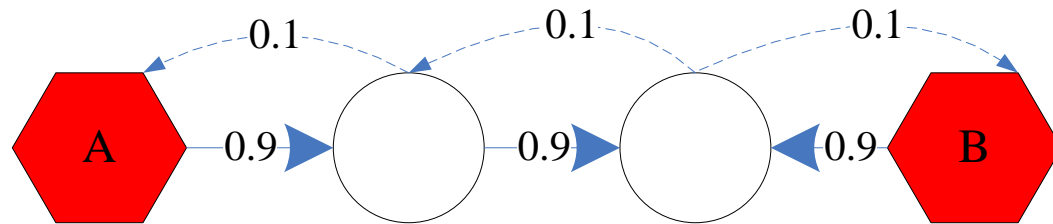
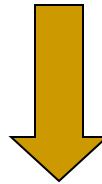
$$\text{ep}(A \rightarrow B) = 0.74$$

Issue 2: Weakly connected pair



$$\text{ep}(A \rightarrow B) = \text{ep}(B \rightarrow A) = 0$$

■ Partial symmetry



$$\text{ep}(A \rightarrow B) = 0.081 \quad > \quad \text{ep}(B \rightarrow A) = 0.009$$

Solving $ep(i \rightarrow j)$

- The generalized voltage

- $v_k \triangleq \Pr[\text{A random walk starting at } k \text{ visits } j \text{ before } i]$

- Calculating $\mathbf{v} \triangleq (v_1 \ v_2 \ \cdots \ v_n)^\top$

- $v_i = 0$, $v_j = 1$, $\forall k \neq i, j$, $v_k = \sum_l p_{kl} \cdot v_l$

- Split $\mathbf{P} = \begin{pmatrix} \hat{\mathbf{P}} & \mathbf{c}_i & \mathbf{c}_j \\ \mathbf{r}_i^\top & 0 & p(i, j) \\ \mathbf{r}_j^\top & p(j, i) & 0 \end{pmatrix}$ and $\mathbf{v} = (\hat{\mathbf{v}} \ 0 \ 1)^\top$

- Then $\hat{\mathbf{v}} = \hat{\mathbf{P}}\hat{\mathbf{v}} + \mathbf{c}_j \Rightarrow \hat{\mathbf{v}} = (\mathbf{I} - \hat{\mathbf{P}})^{-1}\mathbf{c}_j$

- $ep(i \rightarrow j) = \sum_k p_{ik} \cdot v_k = \mathbf{r}_i^\top (\mathbf{I} - \hat{\mathbf{P}})^{-1} \mathbf{c}_j + p(i, j)$

Solving $\text{ep}(i \rightarrow j)$

- The generalized voltage

- $v_k \triangleq \Pr[\text{A random walk starting at } k \text{ visits } j \text{ before } i]$

- Calculating $\mathbf{v} \triangleq (v_1 \ v_2 \ \cdots \ v_n)^\top$

- $v_i = 0$, $v_j = 1$, $\forall k \neq i, j$, $v_k = \sum_l p_{kl} \cdot v_l$

- Split $\mathbf{P} = \begin{pmatrix} \hat{\mathbf{P}} & \mathbf{c}_i & \mathbf{c}_j \\ \mathbf{r}_i^\top & 0 & p(i, j) \\ \mathbf{r}_j^\top & p(j, i) & 0 \end{pmatrix}$ and $\mathbf{v} = (\hat{\mathbf{v}} \ 0 \ 1)^\top$

Computing all $\text{ep}(i \rightarrow j)$ requires $\Theta(n^2)$ matrix inversions

- Then $\hat{\mathbf{v}} = \hat{\mathbf{P}}\hat{\mathbf{v}} + \mathbf{c}_j \Rightarrow \hat{\mathbf{v}} = (\mathbf{I} - \hat{\mathbf{P}})^{-1}\mathbf{c}_j$

- $\text{ep}(i \rightarrow j) = \sum_k p_{ik} \cdot v_k = \mathbf{r}_i^\top (\mathbf{I} - \hat{\mathbf{P}})^{-1} \mathbf{c}_j + p(i, j)$

Fast solution for all-pair proximities

Theorem. Let $\mathbf{Q} = [q(i, j)] \triangleq (\mathbf{I} - c\mathbf{P})^{-1}$. $\forall i \neq j$, there is

$$\text{ep}(i \rightarrow j) = \frac{q(i, j)}{q(i, i)q(j, j) - q(i, j)q(j, i)}.$$

- Proved by Block Matrix Inversion Lemma
- Fast solution to all-pair proximities
 - Compute $\mathbf{Q} = (\mathbf{I} - c\mathbf{P})^{-1}$
 - For all pair of nodes, compute $\text{Prox}(i, j) = \frac{q(i, j)}{q(i, i)q(j, j) - q(i, j)q(j, i)}$
- Time complexity $\Theta(1 \text{ matrix inversion}) + \Theta(n^2)$

Fast solution for one-pair proximity

Theorem. Let $\mathbf{Q} = [q(i, j)] \triangleq (\mathbf{I} - c\mathbf{P})^{-1}$. $\forall i \neq j$, there is

$$\text{ep}(i \rightarrow j) = \frac{q(i, j)}{q(i, i)q(j, j) - q(i, j)q(j, i)}.$$

■ Fast solution to one-pair proximity

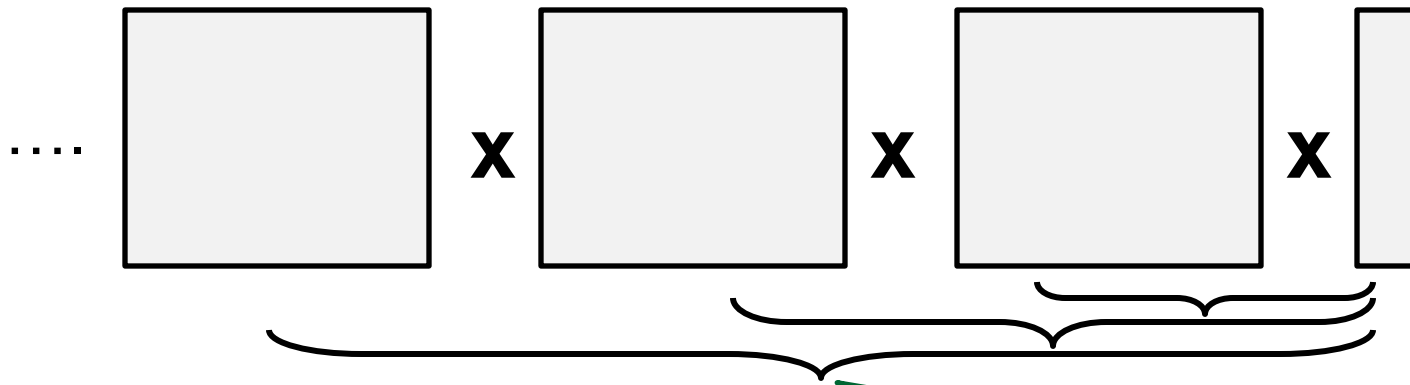
- Only need two columns of \mathbf{Q}
- Taylor expansion

$$(\mathbf{I} - c\mathbf{P})^{-1} = \mathbf{I} + c\mathbf{P} + (c\mathbf{P})^2 + \dots$$

- Computing i^{th} column of \mathbf{Q}

$$\mathbf{Q}\mathbf{e}_i = (\mathbf{I} - c\mathbf{P})^{-1}\mathbf{e}_i = \mathbf{e}_i + c\mathbf{P}\mathbf{e}_i + (c\mathbf{P})^2\mathbf{e}_i + \dots$$

Fast solution for one-pair proximity



■ Fast solution to one-pair proximity

- Only need two columns of \mathbf{Q}
- Taylor expansion

$$(\mathbf{I} - c\mathbf{P})^{-1} = \mathbf{I} + c\mathbf{P} + (c\mathbf{P})^2 + \dots$$

- Computing i^{th} column of \mathbf{Q}

$$\mathbf{Q}e_i = (\mathbf{I} - c\mathbf{P})^{-1}e_i = e_i + c\mathbf{P}e_i + (c\mathbf{P})^2e_i + \dots$$

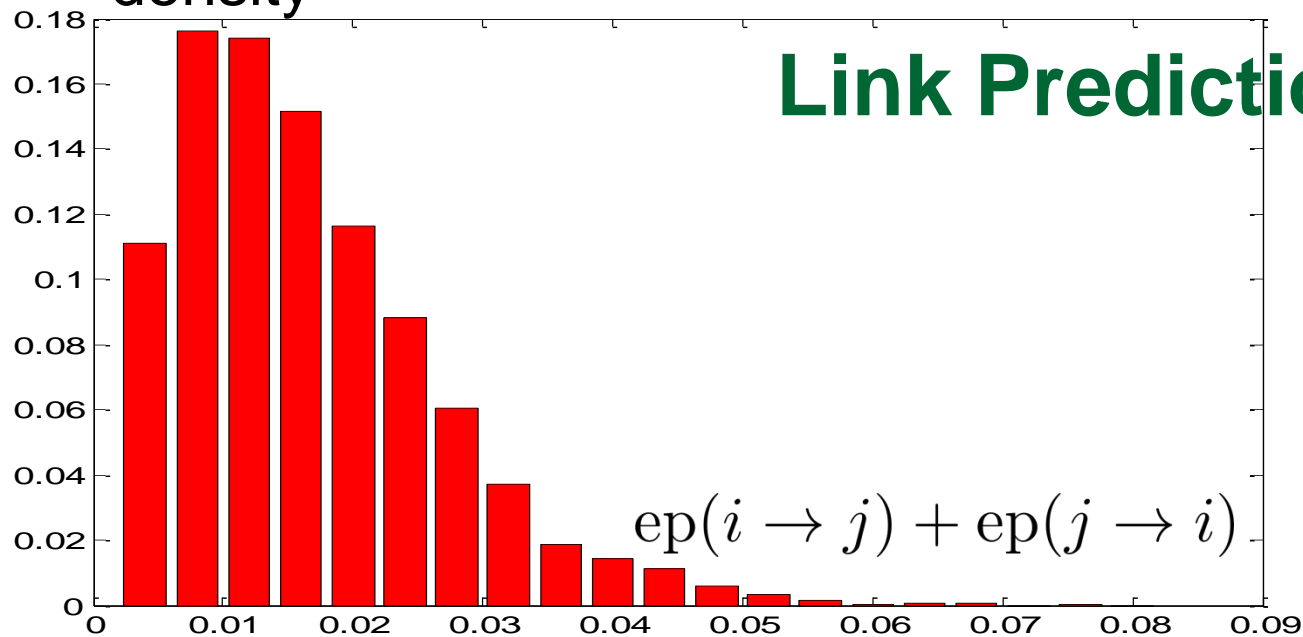
Experimental results

- Effectiveness
 - Link Prediction
 - Existence
 - Direction
 - Efficiency
 - Fast all-pair proximities
 - Fast one-pair proximity
-

Datasets (all real)

Name	Node #	Edge #	Directionality
WL	4k	10k	A-links to-B
PC	36k	64k	Who-contact-whom
EP	76k	509k	Who-trust-whom
CN	28k	353k	A-cites-B
AE	38k	115k	Who-email to-whom

density

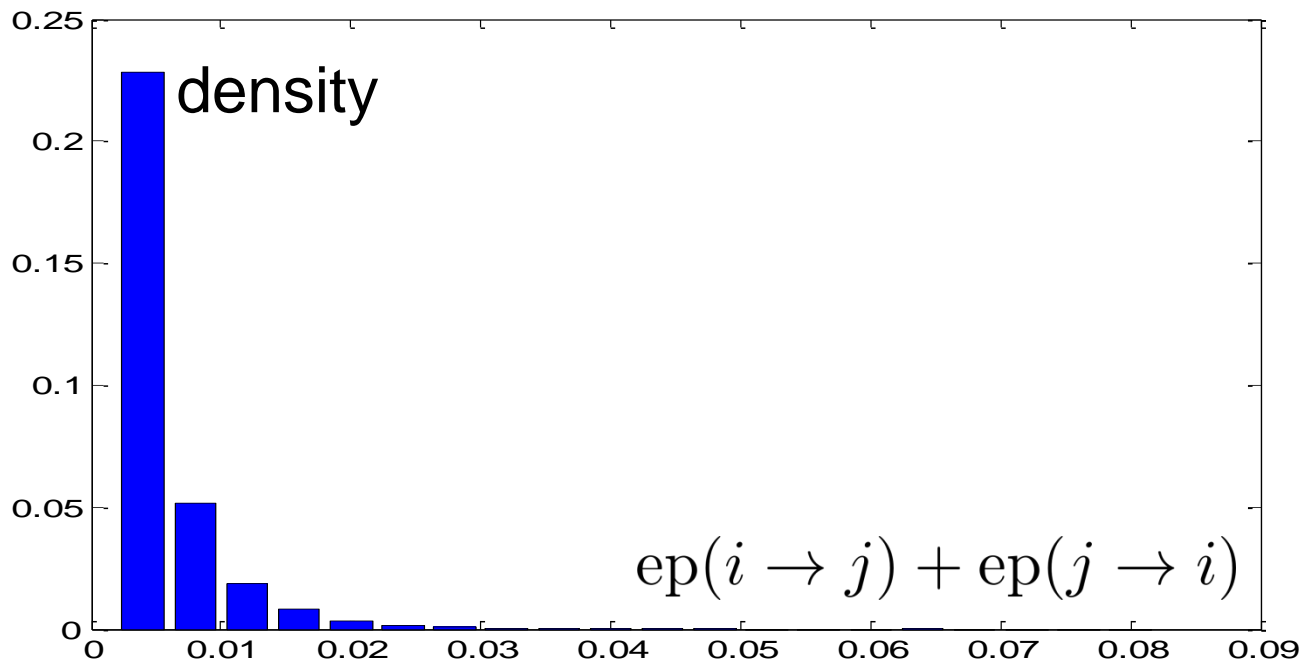


Link Prediction: existence

with link

$$ep(i \rightarrow j) + ep(j \rightarrow i)$$

density



no link

$$ep(i \rightarrow j) + ep(j \rightarrow i)$$

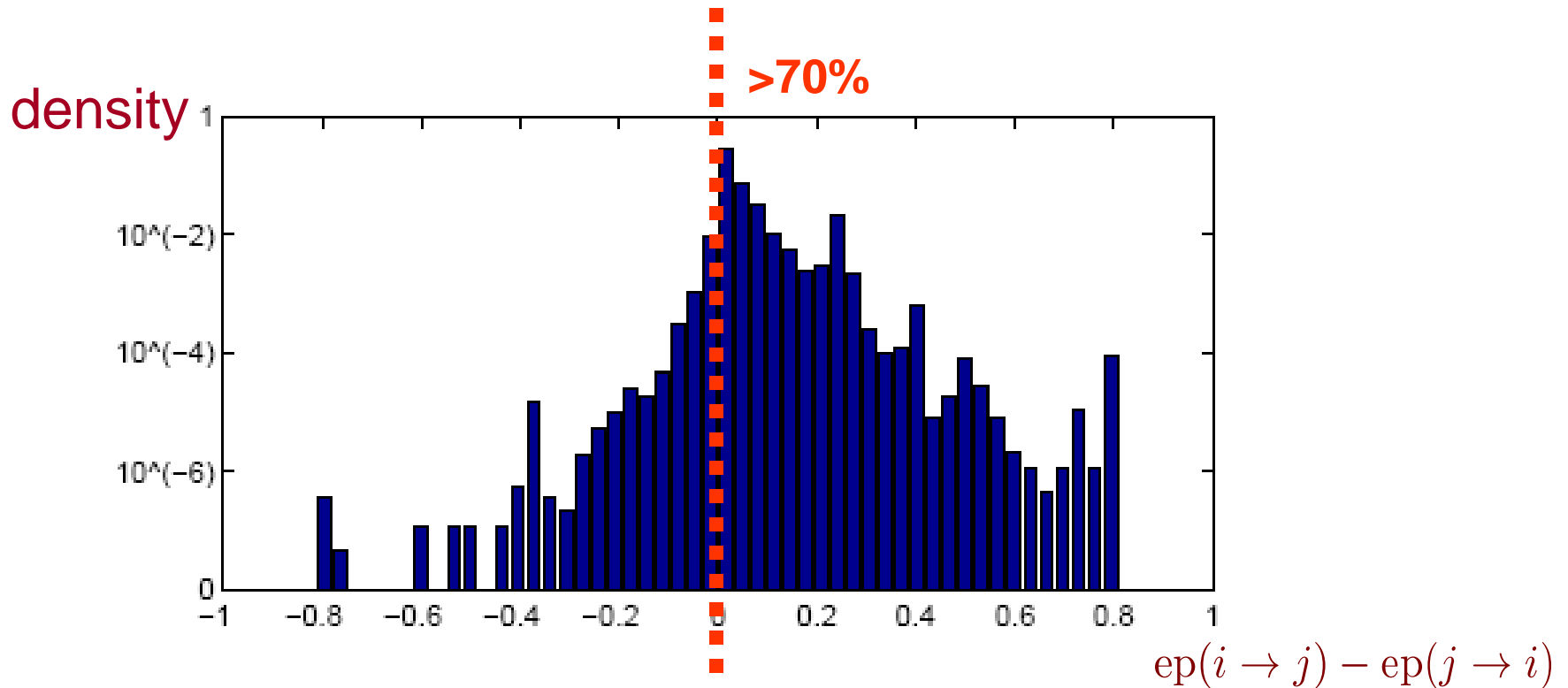
Link Prediction: existence

- Q: Given a pair of nodes i and j , is there a link between them?
- A: Yes iff $\text{ep}(i \rightarrow j) + \text{ep}(j \rightarrow i)$ reaches a given threshold

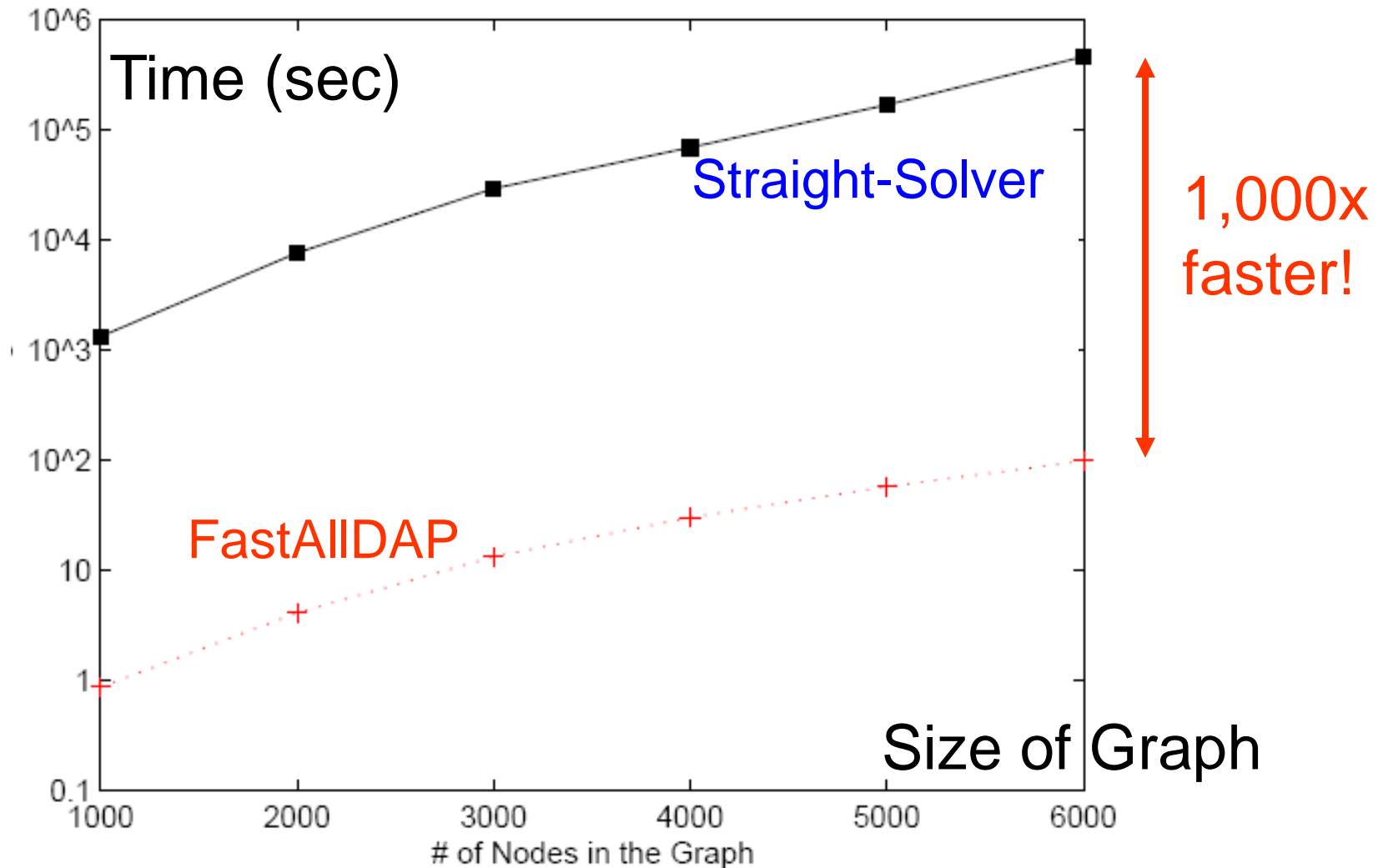
Dataset	Accuracy
WL	65.40%
PC	79.60%
AE	81.51%
CN	86.71%
EP	92.21%

Link Prediction: direction

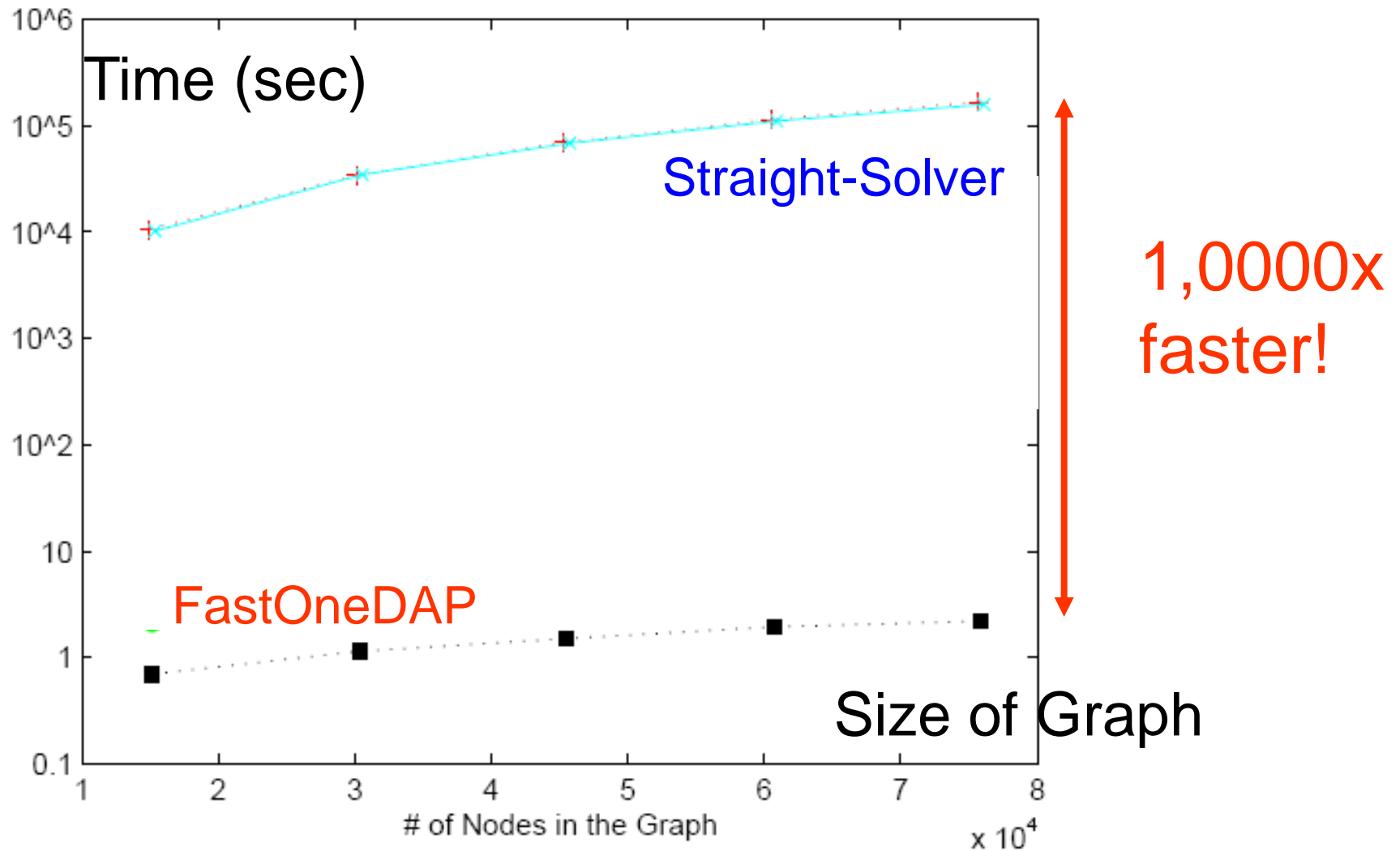
- Q: Given the existence of the link between i and j , what is the direction of it?
- A: Compare $\text{ep}(i \rightarrow j)$ and $\text{ep}(j \rightarrow i)$, pick the greater one



Efficiency: Fast all-pair proximities



Efficiency: Fast one-pair proximity



Relation to commute times

Lemma. *The expected time r_i for a random walk starting at node i to return to i is the reciprocal of the stationary probability of i . That is*

$$r_i = \frac{1}{\pi_i}.$$

■ Intuitively¹

- A long walk always ends up in stationary distribution π
- Suppose the walk length is T , then the expected number it visits i is $\pi_i T$
- The average time between two visits is $\frac{T}{\pi_i \cdot T} = \frac{1}{\pi_i}$

■ Rigorously proved by the Strong Law of Large Numbers²

1. L. Lovász. Random walks on graphs: A survey. 1993.

2. A. Blum, J. Hopcroft, & R. Kannan. Foundations of Data Science. 2016.

Relation to commute times

Theorem. *The probability that a random walk starting at node i visits j before returning to i , which equals $\text{ep}(i \rightarrow j)$, satisfies*

$$\text{ep}(i \rightarrow j)c(i, j) = \frac{1}{\pi_i},$$

where $c(i, j)$ is the commute time between i and j .

■ Proof¹

- Consider a random walk w starting at i , and random variables
 - X = the first time w returns to i
 - Y = the first time w returns to i after visiting j
- By definition $E(X) = \frac{1}{\pi_i}$ and $E(Y) = c(i, j)$
- Clearly $X \leq Y$, and $\Pr[X = Y] = p \triangleq \text{ep}(i \rightarrow j)$
 - $E(Y - X) = p \cdot 0 + (1 - p) \cdot E(Y) = (1 - p)c(i, j)$
- Also $E(Y - X) = E(Y) - E(X) = c(i, j) - \frac{1}{\pi_i}$

Relation to commute times

Theorem. *The probability that a random walk starting at node i visits j before returning to i , which equals $\text{ep}(i \rightarrow j)$, satisfies*

$$\text{ep}(i \rightarrow j)c(i, j) = \frac{1}{\pi_i},$$

where $c(i, j)$ is the commute time between i and j .

- $\text{ep}(i \rightarrow j) + \text{ep}(j \rightarrow i) = \frac{1}{c(i, j)} \left(\frac{1}{\pi_i} + \frac{1}{\pi_j} \right)$
- Recall that $h(i, j)$ is small whenever π_j is large
 - Bad for personalization
- To alleviate this
 - Sarkar et al. restrict the length of random walk ¹
 - Tong et al. reduce the dependence on stationary distribution ²

1. P. Sarkar, A. Moore, & A. Prakash. Fast Incremental Proximity Search in Large Graphs. *ICML '08*.

2. H. Tong, Y. Koren, & C. Faloutsos. Fast direction-aware proximity for graph mining. *KDD '07*.

The End
