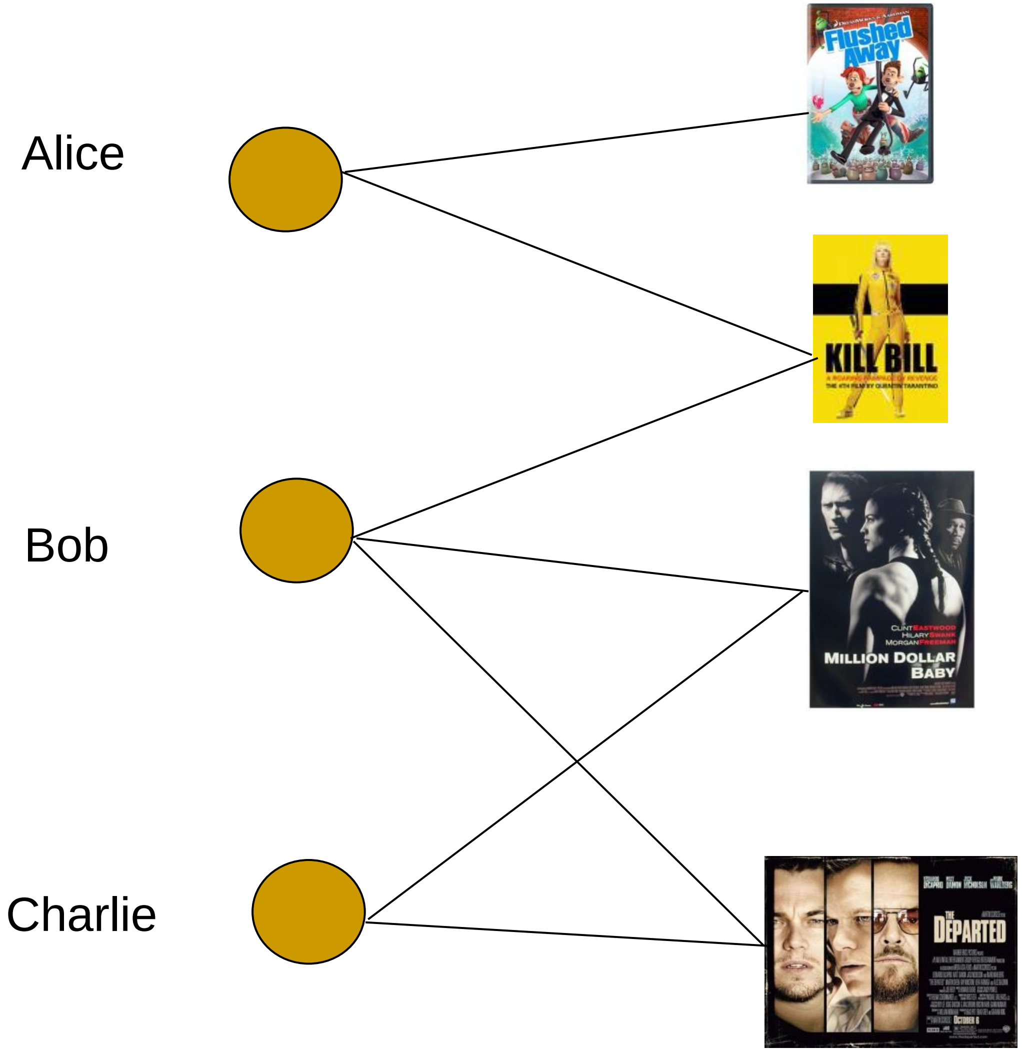
Random Walk based Proximity Measures in Directed Graphs

Speaker: 李 寰

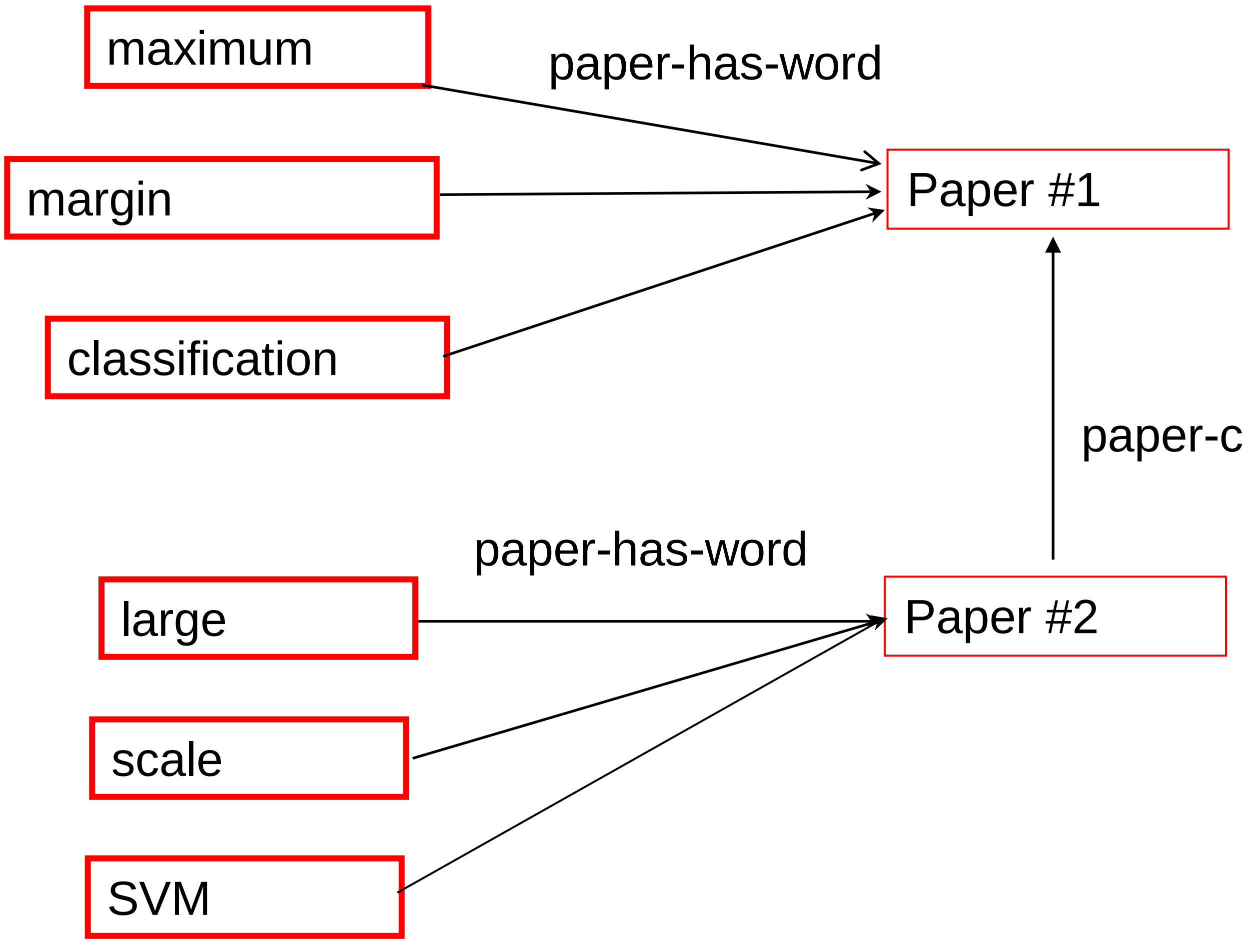
Recommender systems[1]



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

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

Content-based search in databases[1, 2]



Find top *k* papers matching "SVM" in DBLP

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)***

Personalized pagerank

- Stationary distribution $\boldsymbol{\pi} = \boldsymbol{\pi} \boldsymbol{P}$

- Pagerank[1]
  - Rank web-pages by distribution satisfying

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

- Personalized pagerank[2]
  - Using a non-uniform restart distribution

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

  - e.g. $\boldsymbol{r} = \boldsymbol{e_i}$ 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

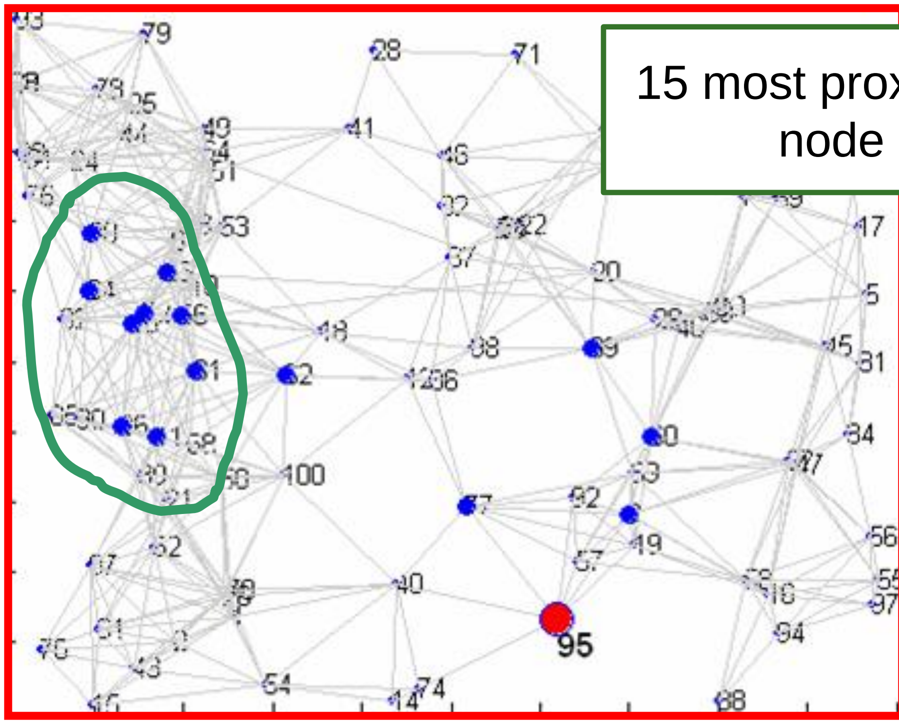
- Hitting time $h(i,j)$
  - Expected length of the path $i \longrightarrow j$
- Commute time $c(i,j) = h(i,j) + h(j,i)$
  - Expected length of the path $i \longrightarrow j \longrightarrow i$
- Drawbacks[1, 2]
  - Take into account very long paths
  - $h(i,j)$ is small whenever *j* has a large stationary 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.*

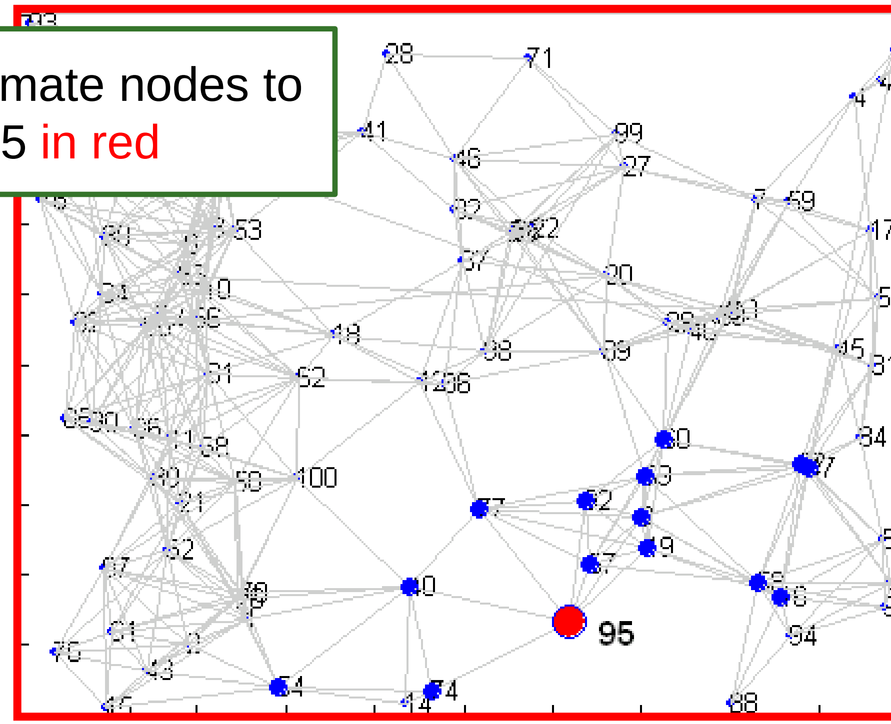
Truncated hitting and commute times[1]

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

15 most proximate nodes to node 95 in red



a) Un-truncated hitting time

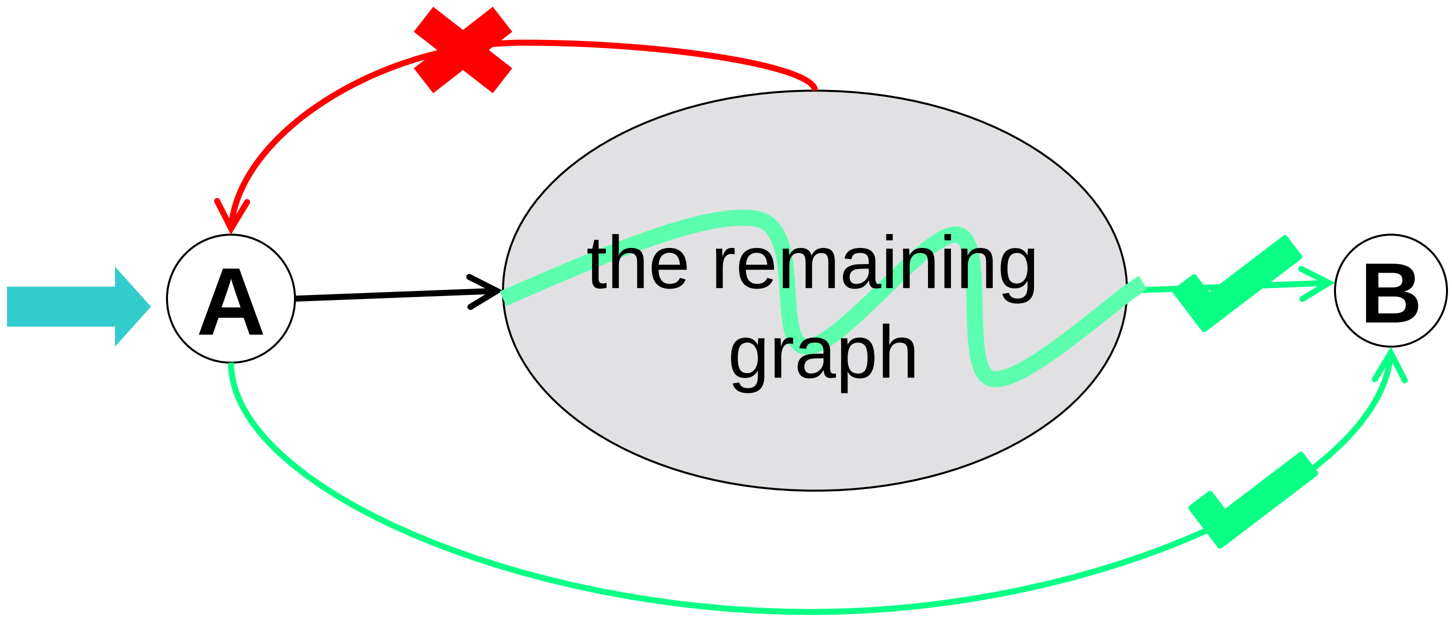


b) Truncated hitting time

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

Escape probability[1]

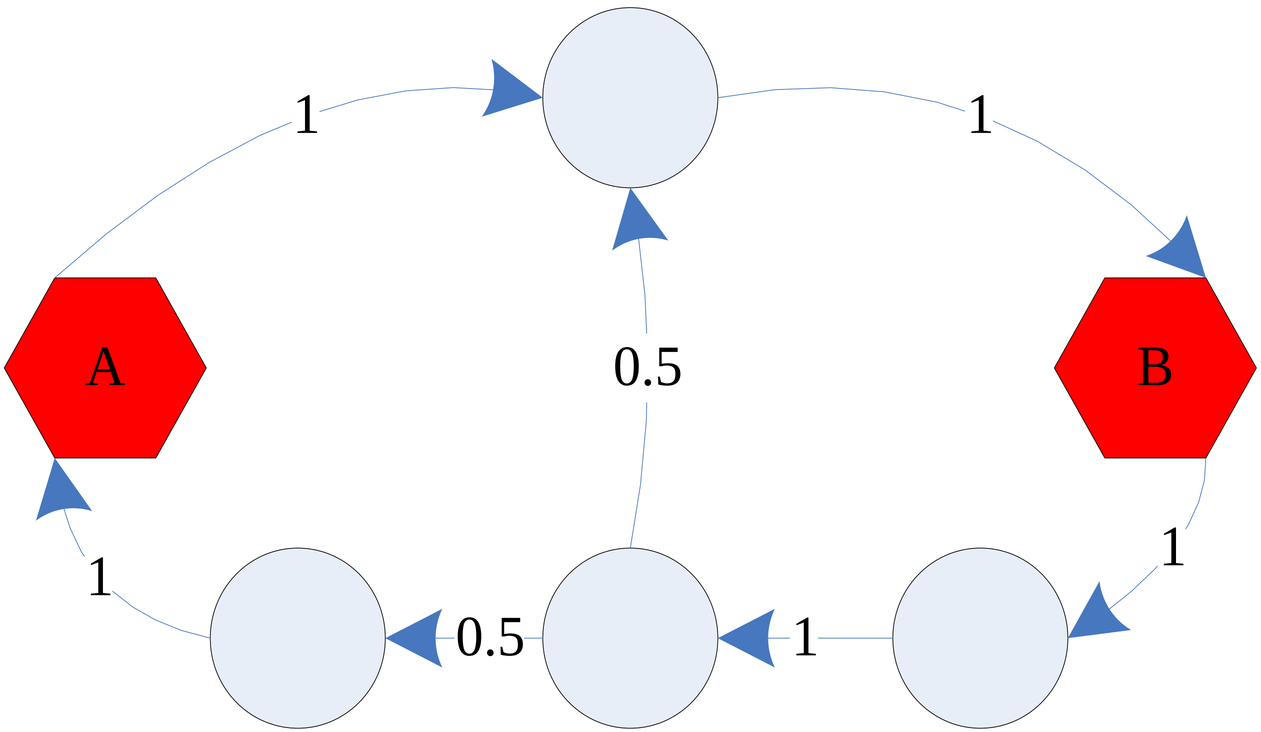
- The escape probability from node *A* to node *B*
  - Denoted as $\mathrm{ep}(A \to B)$
  - Pr [starting at *A*, reaches *B* before returning to *A*]



$$\mathrm{ep}(A \to B) = \Pr\left[\checkmark \text{ comes before } \times \right]$$

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

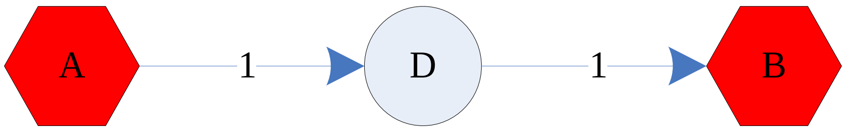
Asymmetry of escape probability



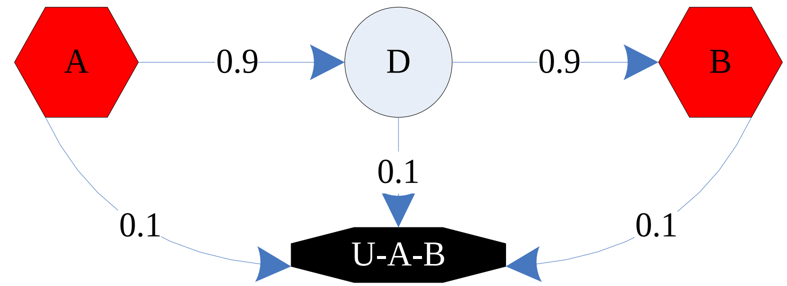
$$\mathrm{ep}(A \to B) = 1 \quad > \quad \mathrm{ep}(B \to A) = 0.5$$

Issue 1: "Degree-1 node" effect

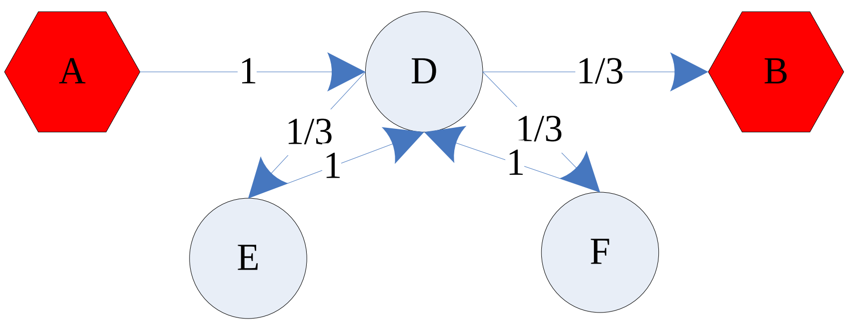
- Adding an absorbing node



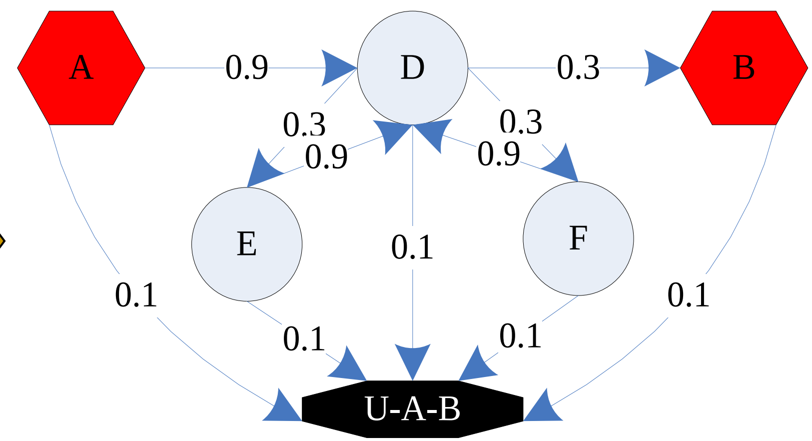
$$\mathrm{ep}(A \rightarrow B) = 1$$



$$\mathrm{ep}(A \rightarrow B) = 0.81$$

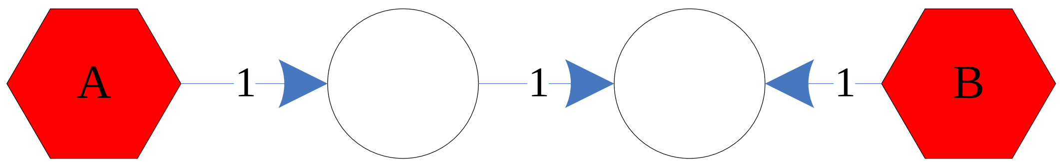


$$\mathrm{ep}(A \rightarrow B) = 1$$



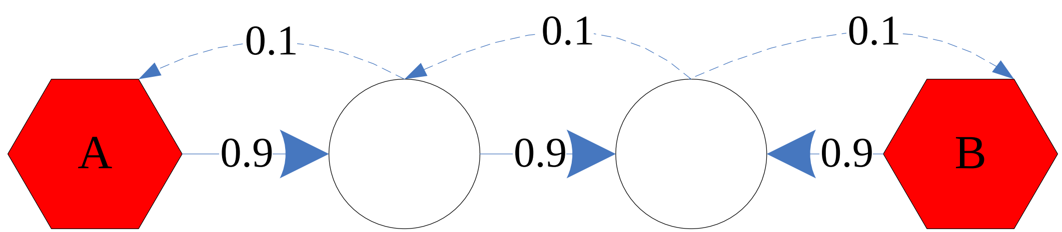
$$\mathrm{ep}(A \rightarrow B) = 0.74$$

Issue 2: Weakly connected pair



$$\mathrm{ep}(A \to B) = \mathrm{ep}(B \to A) = 0$$

- Partial symmetry



$$\mathrm{ep}(A \to B) = 0.081 \quad > \quad \mathrm{ep}(B \to A) = 0.009$$

Solving ep(i -> j)

- $\mathrm{ep}(i \to j) = \boldsymbol{u(i)}^\top (\boldsymbol{I} - c\hat{\boldsymbol{P}})^{-1} \boldsymbol{v(j)} + p(i,j)$
  - $p(i,j)$: entry (i,j) of **P**
  - $\boldsymbol{u(i)}$: the ith row of **P** with ith and jth elements removed
  - $\boldsymbol{v(j)}$: the jth column of **P** with ith and jth elements removed
  - $\hat{\boldsymbol{P}}$: **P** with the ith and jth rows and columns removed

- Computing all ep(i -> j) requires $\Theta(n^2)$ matrix inversions
  - Each time need to invert a submatrix of $(\boldsymbol{I} - c\boldsymbol{P})$
  - There is a lot of redundancy
  - Use relation between inverses of matrix and its submatrices to accelerate

Fast all-pair proximities

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

$$\mathrm{ep}(i \to j) = \frac{q(i,j)}{q(i,i)q(j,j) - q(i,j)q(j,i)}.$$

- Proved by Block Matrix Inversion Lemma

- Fast all-pair proximities
  - Compute $\boldsymbol{Q} = (\boldsymbol{I} - c\boldsymbol{P})^{-1}$
  - For all pairs of nodes, compute $\mathrm{Prox}(i,j) = \frac{q(i,j)}{q(i,i)q(j,j) - q(i,j)q(j,i)}$

- Time complexity $\Theta(1 \text{ matrix inversion}) + \Theta(n^2)$

Fast one-pair proximity

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

$$\mathrm{ep}(i \to j) = \frac{q(i,j)}{q(i,i)q(j,j) - q(i,j)q(j,i)}.$$

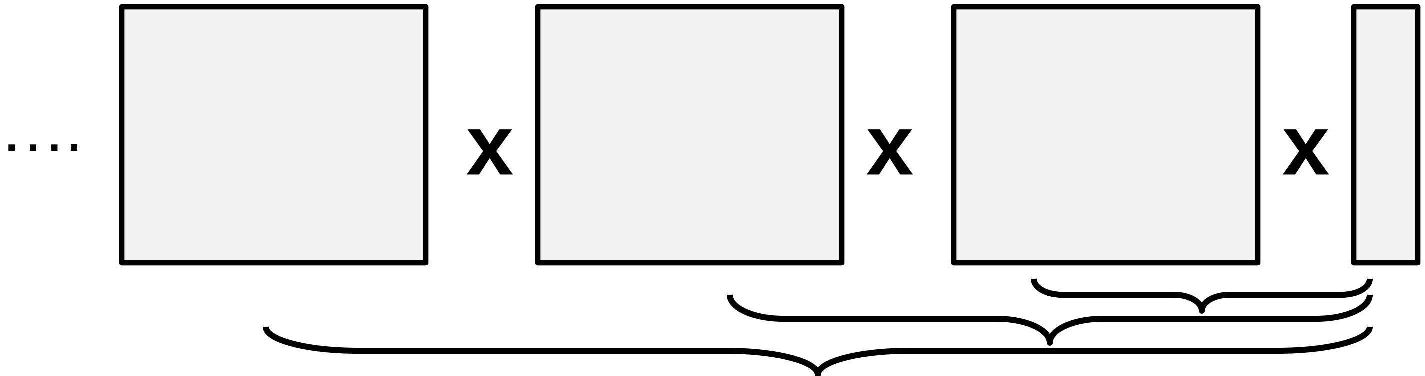
- Fast one-pair proximity
  - Only need two columns of **Q**
  - Taylor expansion $(\text{as } \rho(c\boldsymbol{P}) < 1 \text{ holds})$

$$(\boldsymbol{I} - c\boldsymbol{P})^{-1} = \boldsymbol{I} + c\boldsymbol{P} + (c\boldsymbol{P})^2 + \cdots$$

  - Computing the i-th column of **Q**

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

Fast one-pair proximity



Time complexity
$\Theta\left(t(n+m)\right)$

- Fast one-pair proximity
  - Only need two columns of **Q**
  - Taylor expansion $(\text{as } \rho(c\boldsymbol{P}) < 1 \text{ holds})$

$$(\boldsymbol{I} - c\boldsymbol{P})^{-1} = \boldsymbol{I} + c\boldsymbol{P} + (c\boldsymbol{P})^2 + \cdots$$

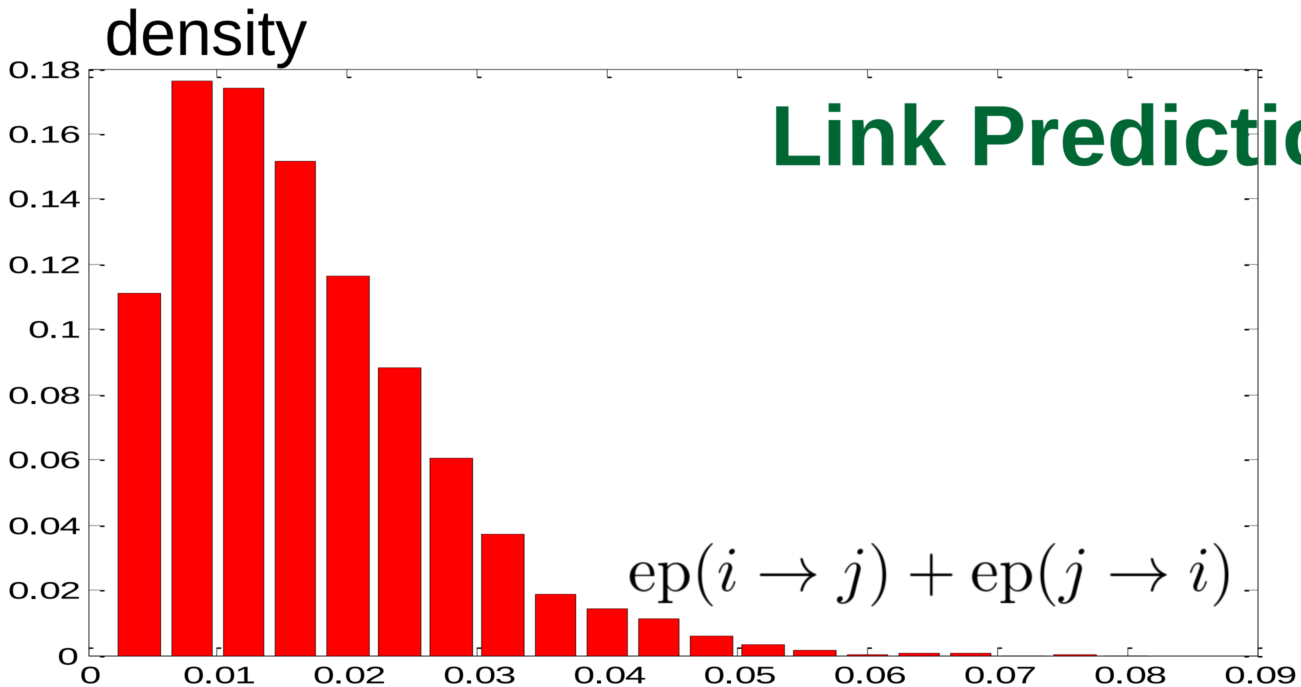
  - Computing the i^th column of **Q**

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

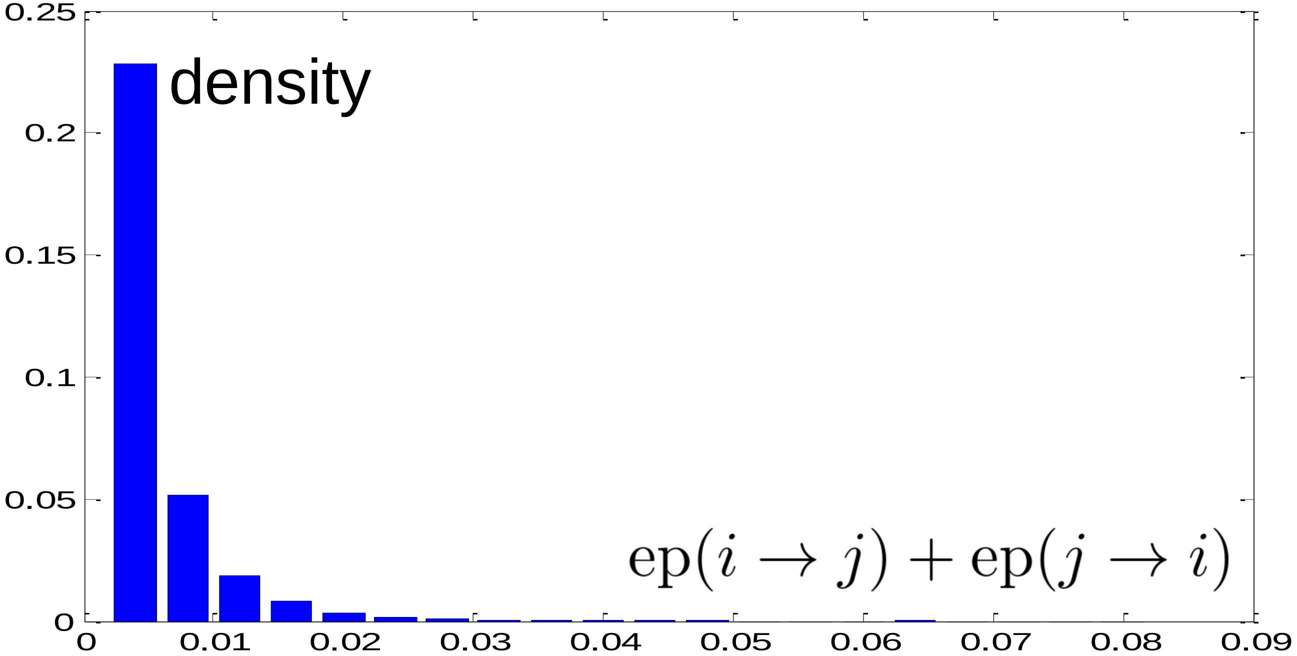
Datasets (all real)

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

Link Prediction: existence



with link



no link

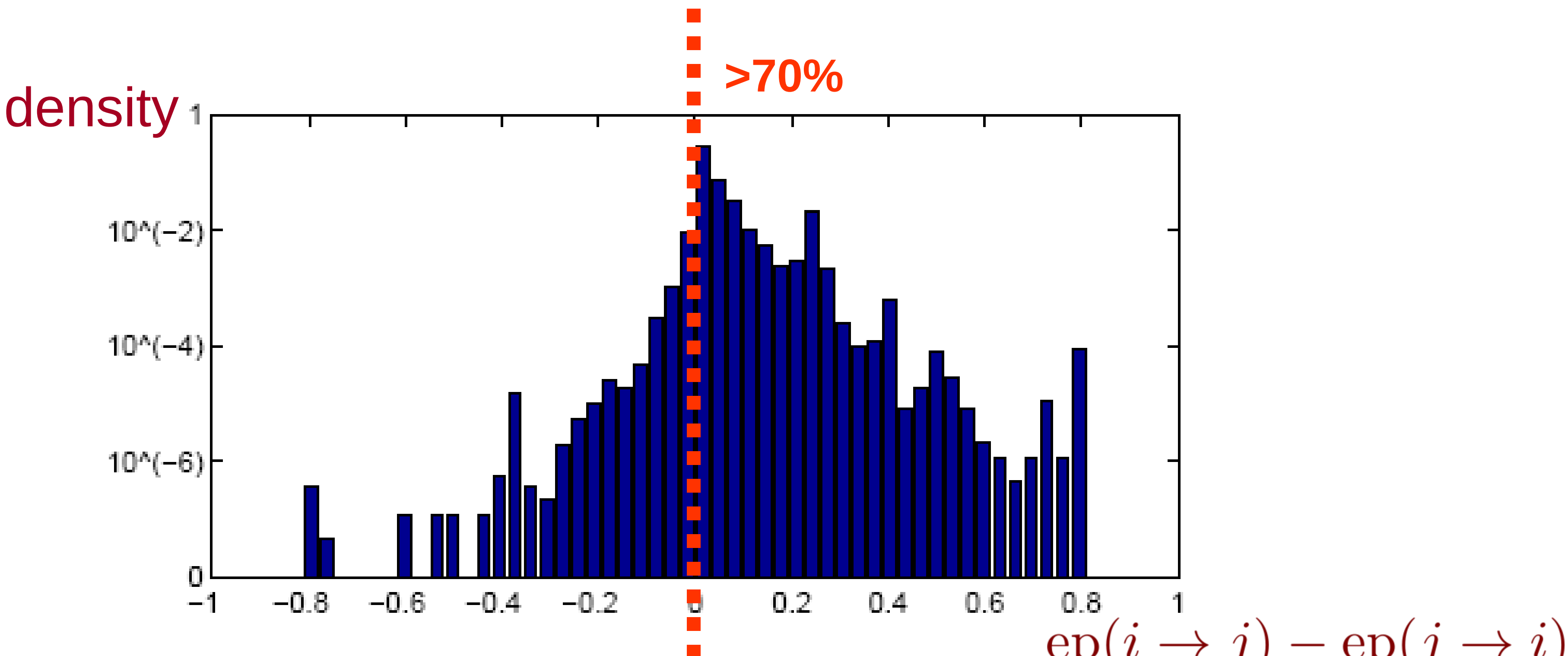
Link Prediction: existence

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

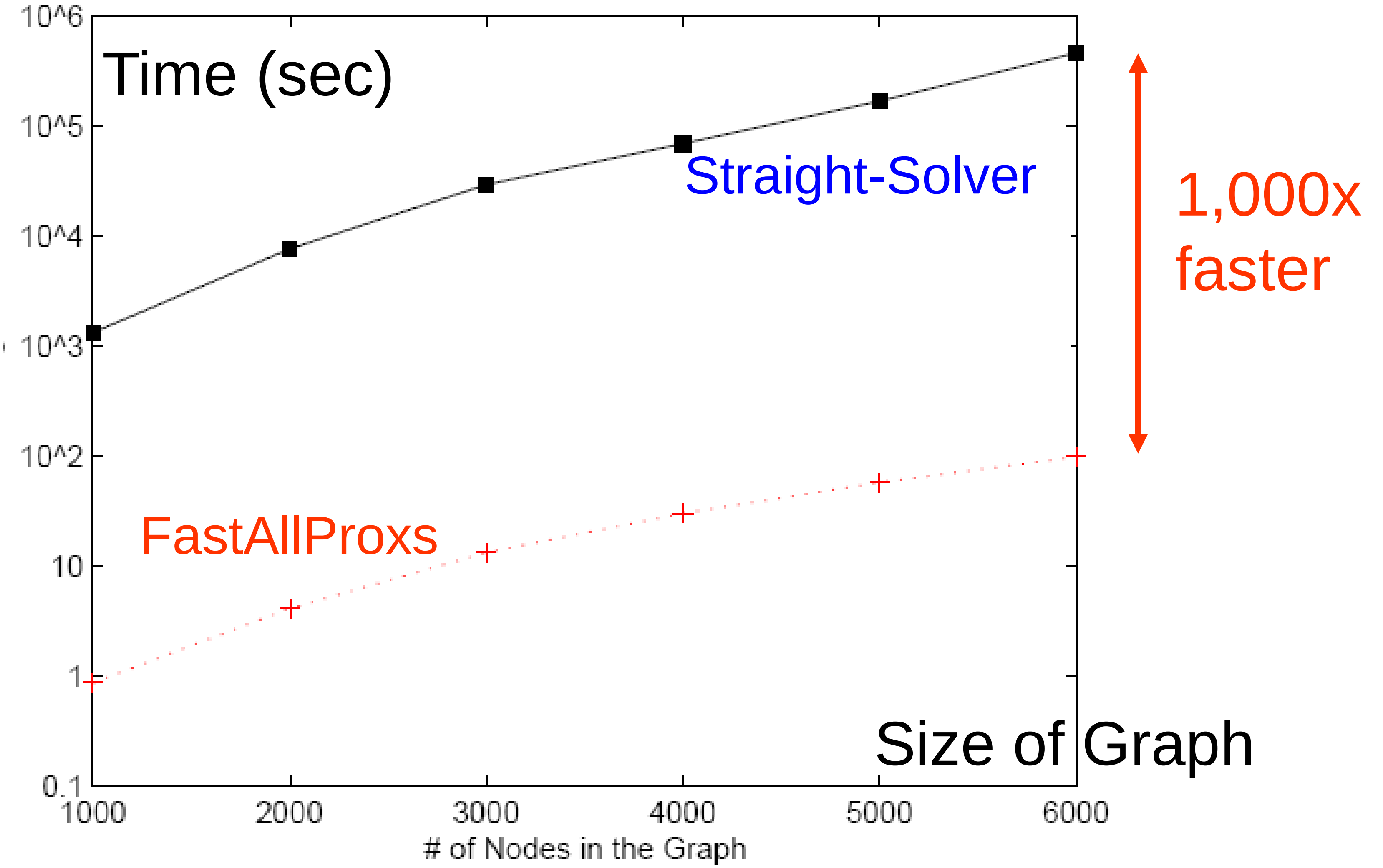
Dataset	Accuracy
WL	**65.40%**
PC	**79.60%**
AE	**81.51%**
CN	**86.71%**
EP	**92.21%**

Link Prediction: direction

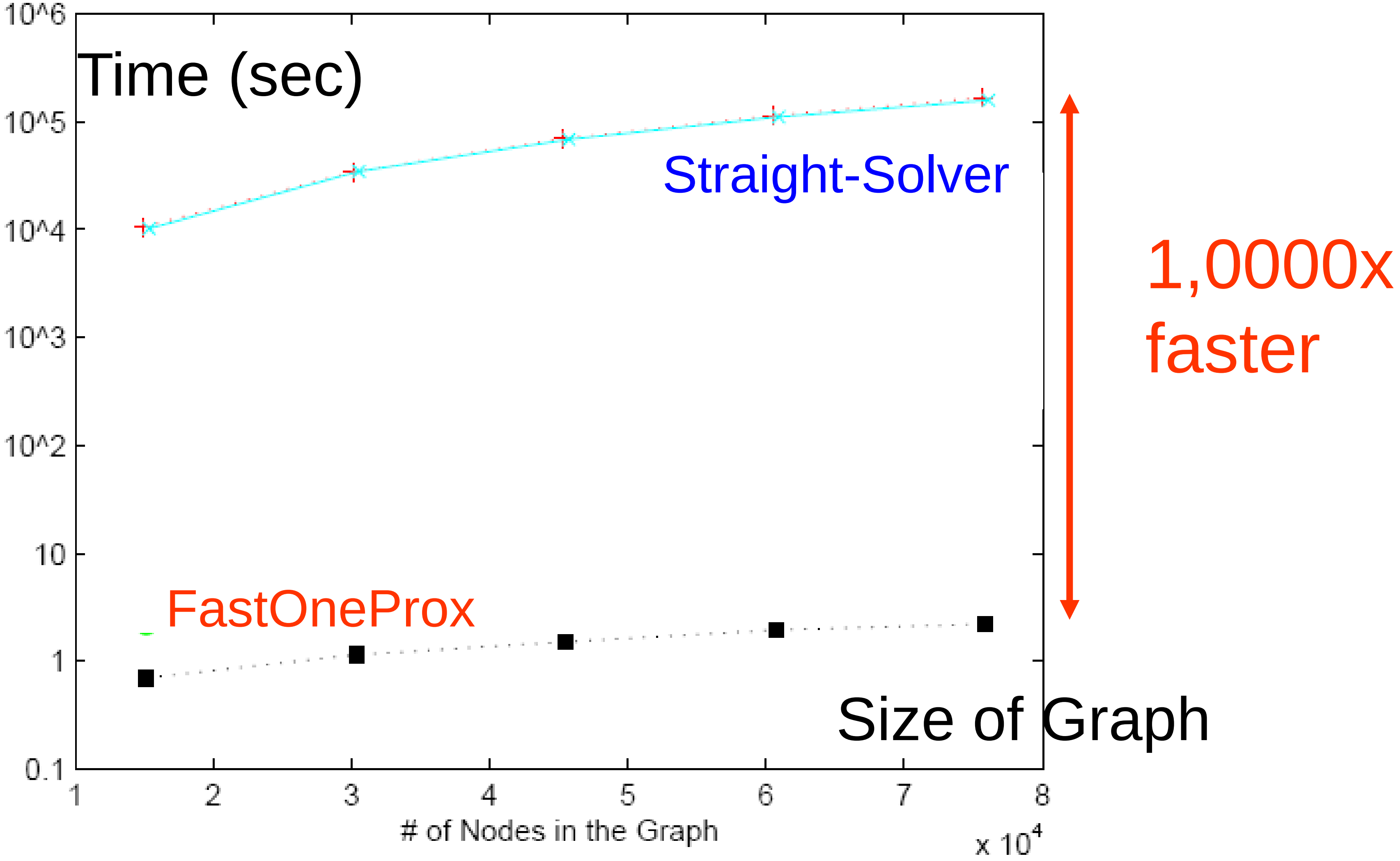
- Q: Given the existence of the link between *i* and *j*, what is the direction of it?
- A: Compare $\mathrm{ep}(i \to j)$ and $\mathrm{ep}(j \to i)$, pick the greater one



Efficiency: Fast all-pair proximities



Efficiency: Fast one-pair proximity



Relation to commute times

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

$$\mathrm{ep}(i \to j) = \frac{1}{c(i,j)} \cdot \frac{1}{\pi_i},$$

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

- $\mathrm{ep}(i \to j) + \mathrm{ep}(j \to i) = \frac{1}{c(i,j)}\left(\frac{1}{\pi_i} + \frac{1}{\pi_j}\right)$
- 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