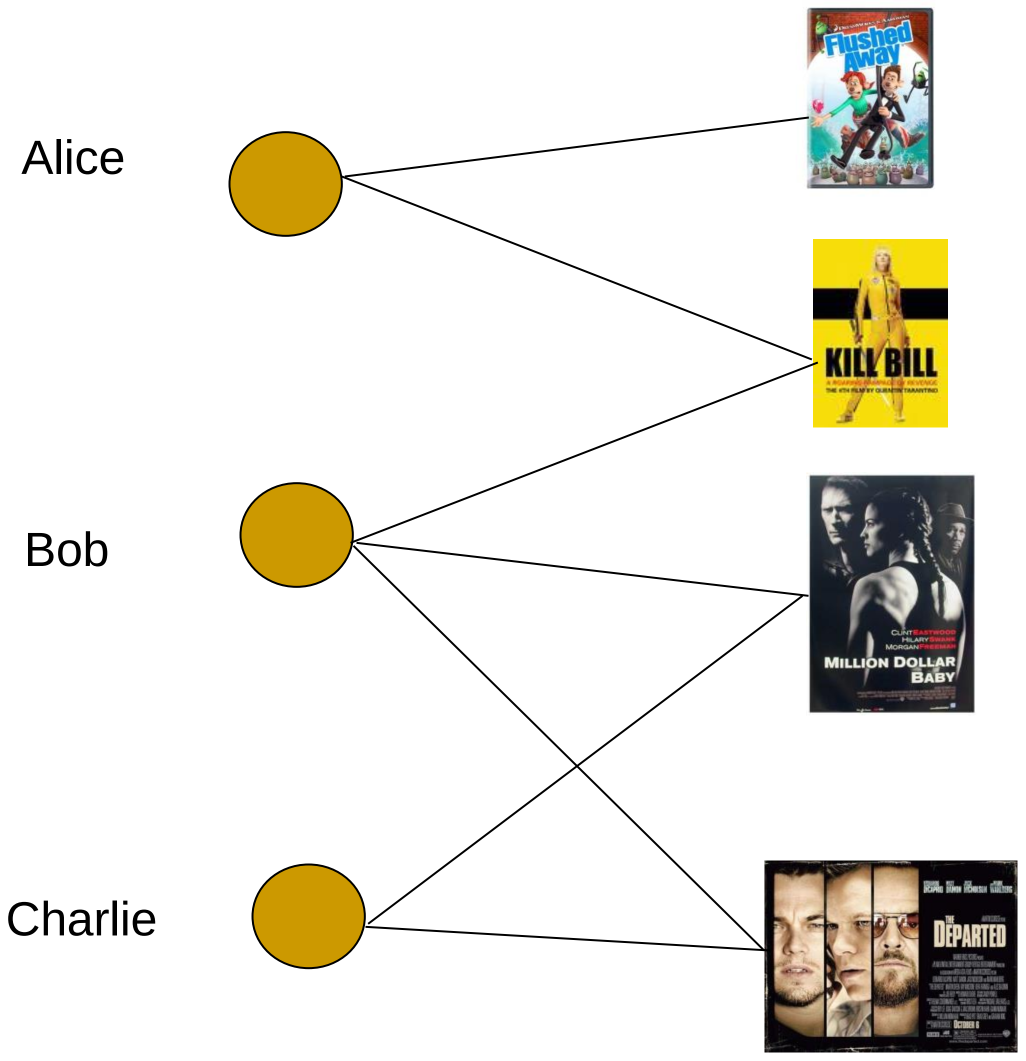
Random Walk based Proximity Measures in Directed Graphs

Speaker: 李 寰

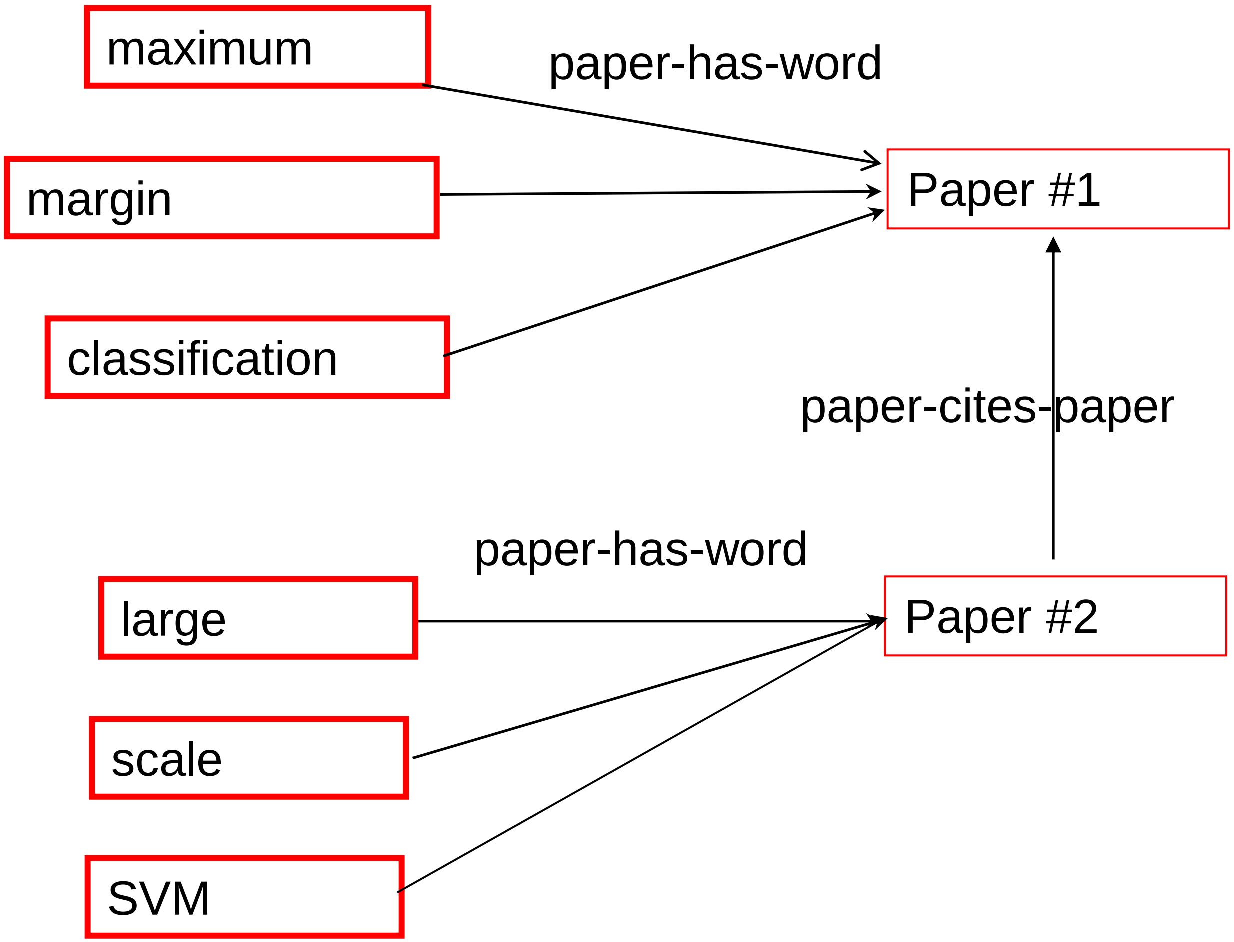
Recommender systems[1]



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.*

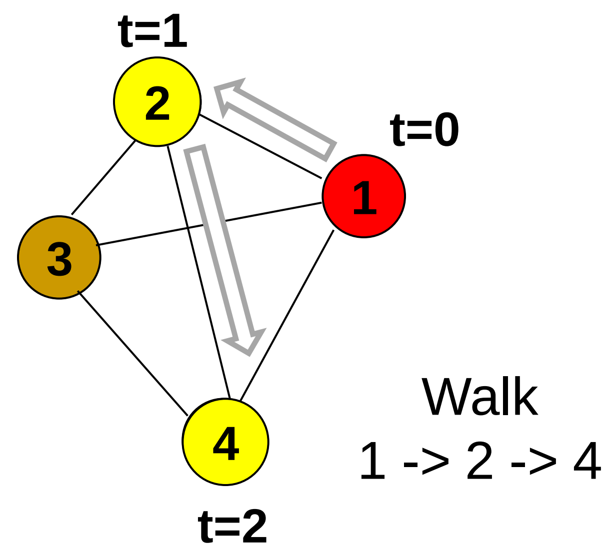
2. A. Balmin, V. Hristidis, & Y. Papakonstantinou. ObjectRank: Authority-based keyword search in databases. *VLDB ‘04.*

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)***

Random walks

- Starts at *i*
- Moves to a neighbor *j* randomly
- Continues



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

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

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

Personalized pagerank

- Stationary distribution $\boldsymbol{\pi} = \boldsymbol{\pi P}$
- Pagerank[1]
  - Rank web-pages by distribution satisfying

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

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

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

  - 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.*

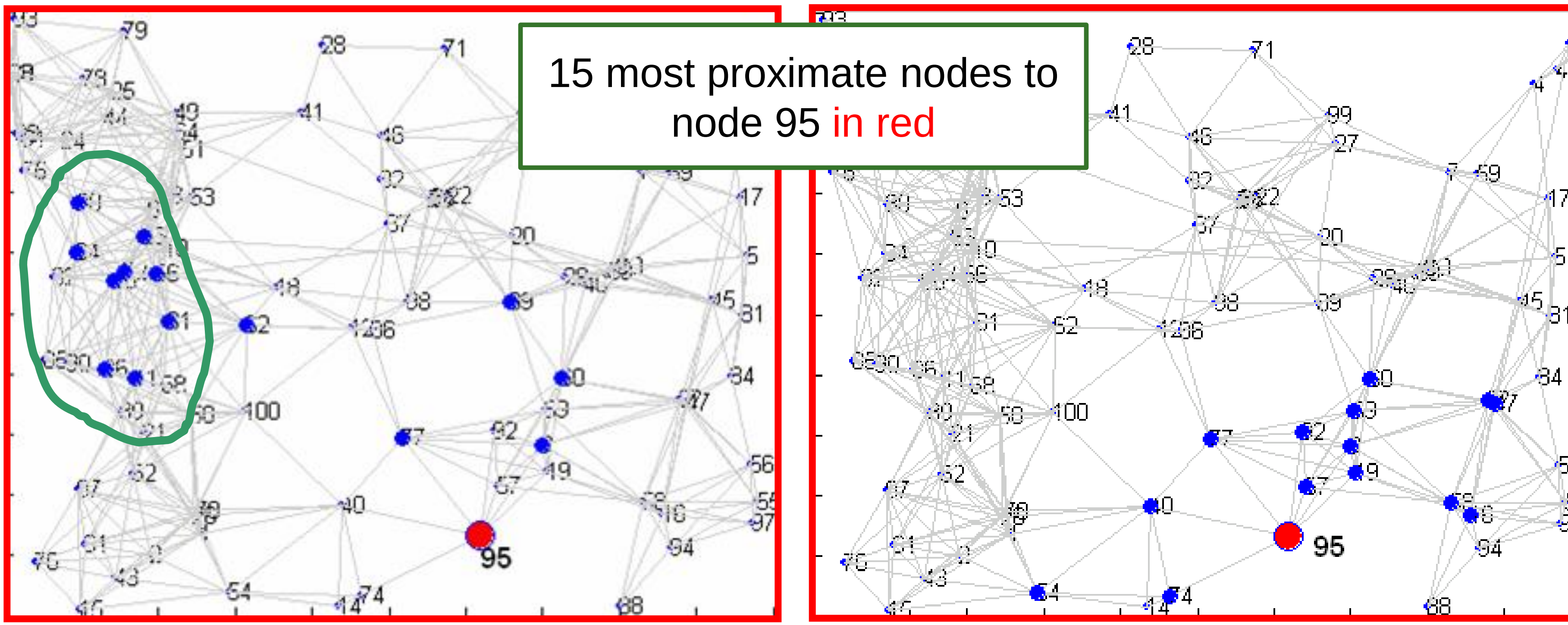
Hitting and commute times

- 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*



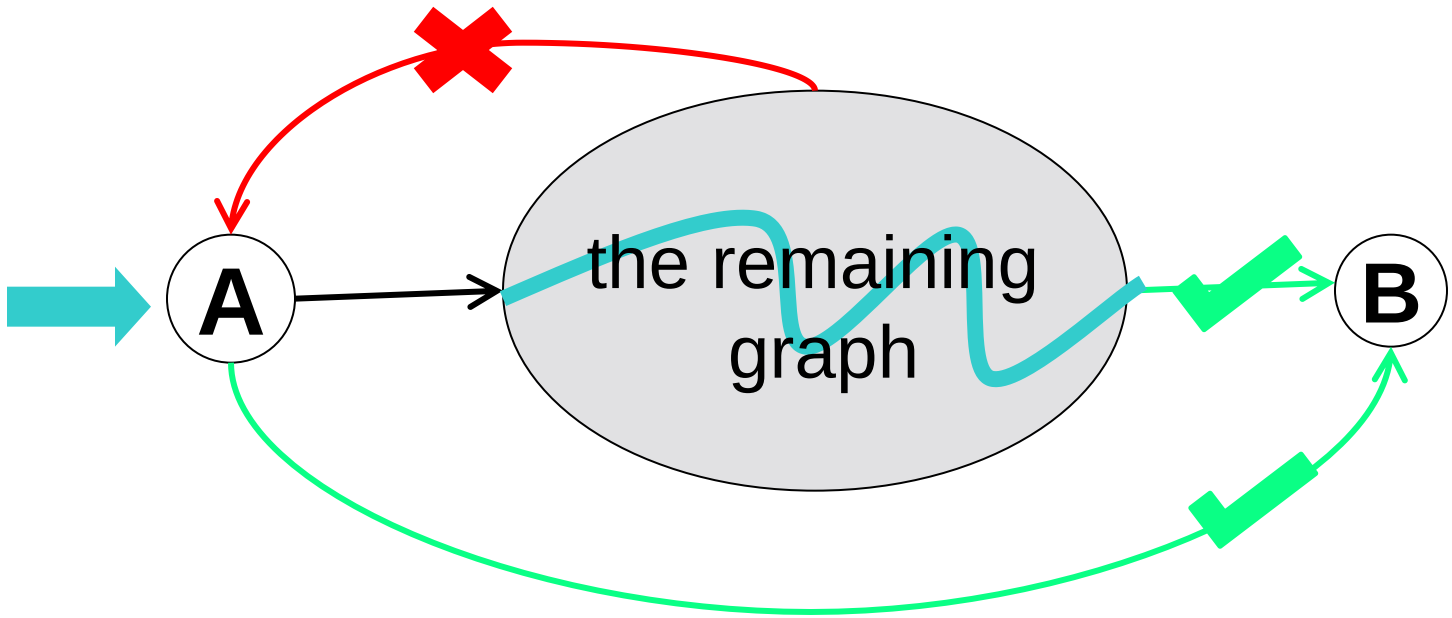
a) Un-truncated hitting time

b) Truncated hitting time

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

Escape probability[1]

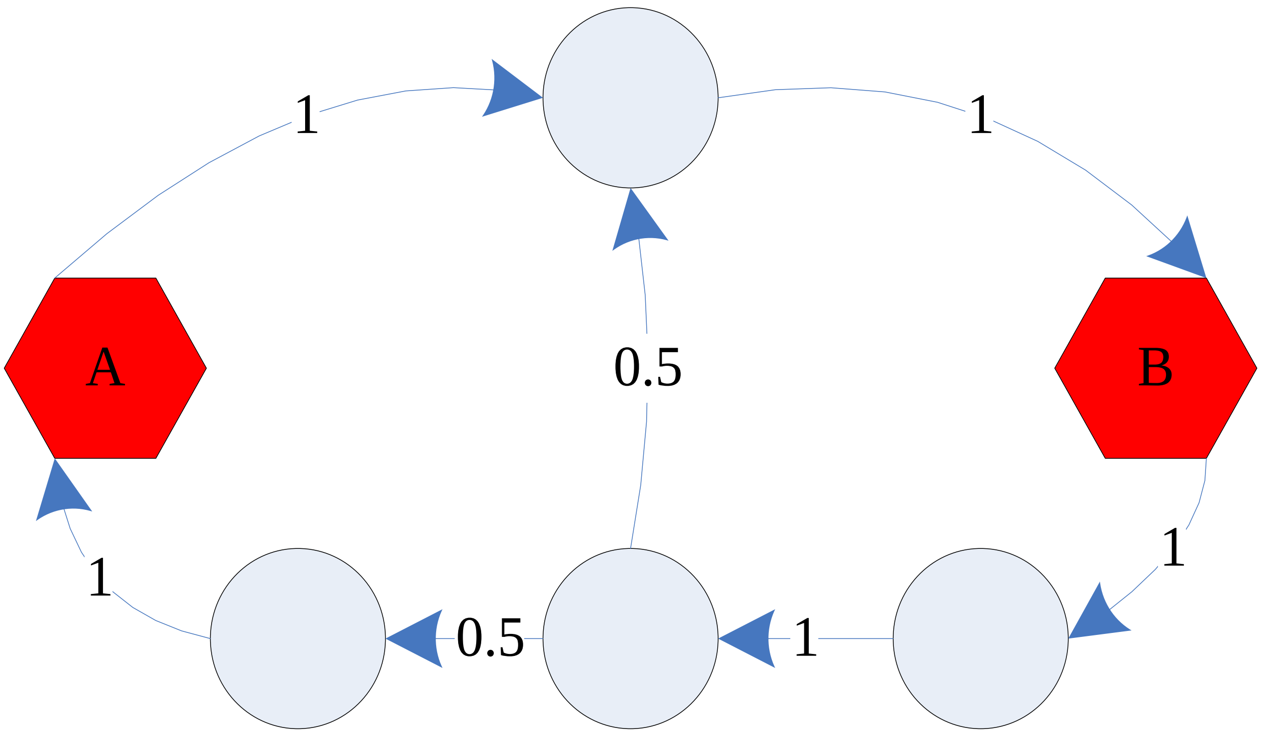
- The escape probability from node *A* to node *B*
  - Denoted as $\mathrm{ep}(A \to B)$
  - Pr [starting at *A*, reaches *B* before returning to *A*]



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

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

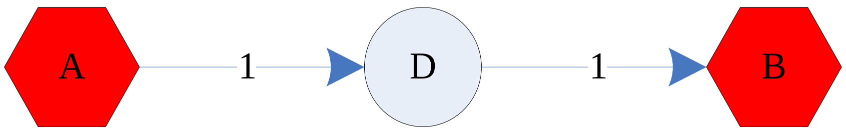
Asymmetry of escape probability



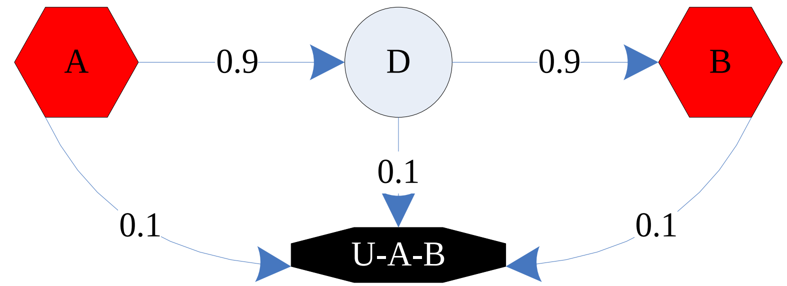
$$\mathrm{ep}(A \to B) = 1 \quad > \quad \mathrm{ep}(B \to A) = 0.5$$

Issue 1: “Degree-1 node” effect

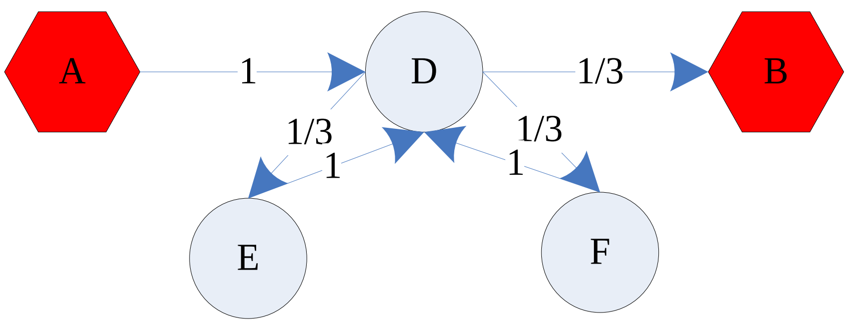
- Adding an absorbing node



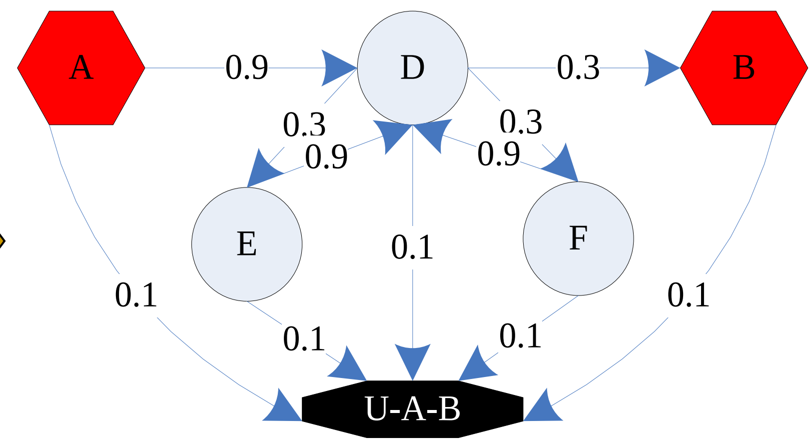
$$\mathrm{ep}(A \to B) = 1$$



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

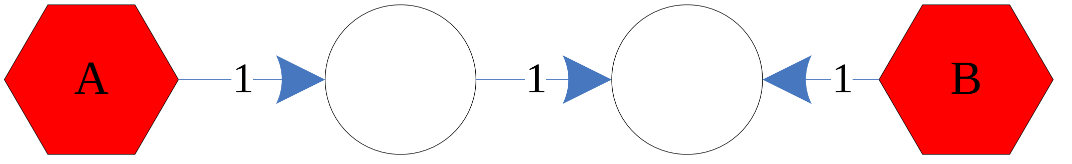


$$\mathrm{ep}(A \to B) = 1$$



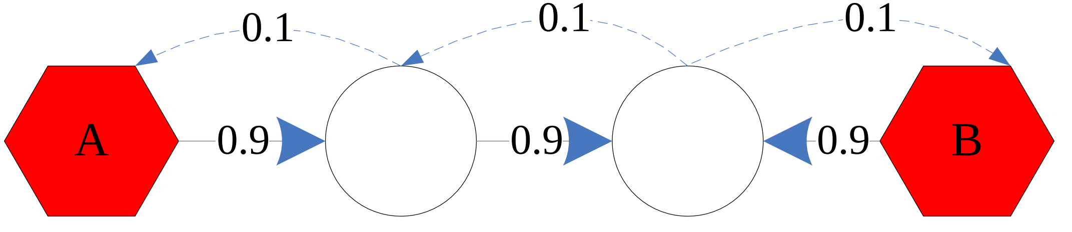
$$\mathrm{ep}(A \to B) = 0.74$$

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)

- The generalized voltage
  - $v(k) \triangleq \Pr[\text{A random walk starting at } k \text{ visits } j \text{ before } i]$
- Calculating $\boldsymbol{v} \triangleq \begin{pmatrix} v(1) & v(2) & \cdots & v(n) \end{pmatrix}^\top$
  - $v(i) = 0$, $v(j) = 1$, $\forall k \neq i, j,\ v(k) = \sum_l p(k,l) \cdot v(l)$
  - Split $\boldsymbol{P} = \begin{pmatrix} \hat{\boldsymbol{P}} & \boldsymbol{c(i)} & \boldsymbol{c(j)} \\ \boldsymbol{r(i)}^\top & 0 & p(i,j) \\ \boldsymbol{r(j)}^\top & p(j,i) & 0 \end{pmatrix}$, $\boldsymbol{v} = \begin{pmatrix} \hat{\boldsymbol{v}} & 0 & 1 \end{pmatrix}^\top$
  - Then $\hat{\boldsymbol{v}} = \hat{\boldsymbol{P}}\hat{\boldsymbol{v}} + \boldsymbol{c(j)} \Rightarrow \hat{\boldsymbol{v}} = (\boldsymbol{I} - \hat{\boldsymbol{P}})^{-1}\boldsymbol{c(j)}$
- $\mathrm{ep}(i \to j) = \sum_k p(i,k) \cdot v(k) = \boldsymbol{r(i)}^\top (\boldsymbol{I} - \hat{\boldsymbol{P}})^{-1}\boldsymbol{c(j)} + p(i,j)$

Solving ep(i -> j)

- The generalized voltage
  - $v(k) \triangleq \Pr[\text{A random walk starting at } k \text{ visits } j \text{ before } i]$
- Calculating $\boldsymbol{v} \triangleq \begin{pmatrix} v(1) & v(2) & \cdots & v(n) \end{pmatrix}^\top$
  - $v(i) = 0$, $v(j) = 1$, $\forall k \neq i, j,\ v(k) = \sum_l p(k,l) \cdot v(l)$
  - Split $\boldsymbol{P} = \begin{pmatrix} \hat{\boldsymbol{P}} & \boldsymbol{c(i)} & \boldsymbol{c(j)} \\ \boldsymbol{r(i)}^\top & 0 & p(i,j) \\ \boldsymbol{r(j)}^\top & p(j,i) & 0 \end{pmatrix}$, $\boldsymbol{v} = \begin{pmatrix} \hat{\boldsymbol{v}} & 0 & 1 \end{pmatrix}^\top$
  - Then $\hat{\boldsymbol{v}} = \hat{\boldsymbol{P}}\hat{\boldsymbol{v}} + \boldsymbol{c(j)} \Rightarrow \hat{\boldsymbol{v}} = (\boldsymbol{I} - \hat{\boldsymbol{P}})$

Computing all ep(i -> j) requires $\Theta(n^2)$ matrix inversions

- $\mathrm{ep}(i \to j) = \sum_k p(i,k) \cdot v(k) = \boldsymbol{r(i)}^\top \boxed{(\boldsymbol{I} - \hat{\boldsymbol{P}})^{-1}} \boldsymbol{c(j)} + p(i,j)$

Fast solution for 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 solution to all-pair proximilities
  - Compute $\boldsymbol{Q} = (\boldsymbol{I} - c\boldsymbol{P})^{-1}$
  - For all pair 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 solution for one-pair proximity

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

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

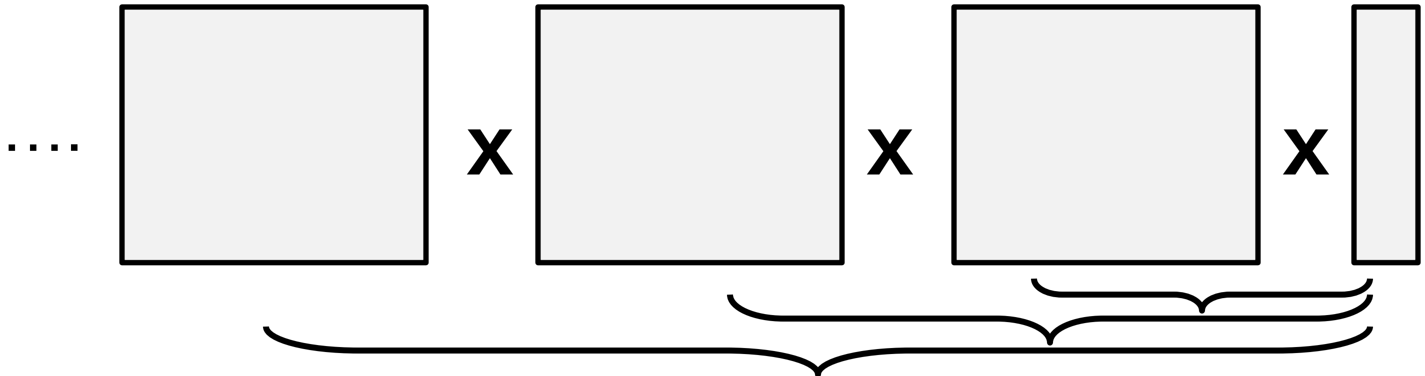
- Fast solution to one-pair proximity
  - Only need two columns of **Q**
  - Taylor expansion (as $\rho(c\boldsymbol{P}) < 1$ holds)

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

  - Computing the ith 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 solution for one-pair proximity



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

- Fast solution to 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**

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

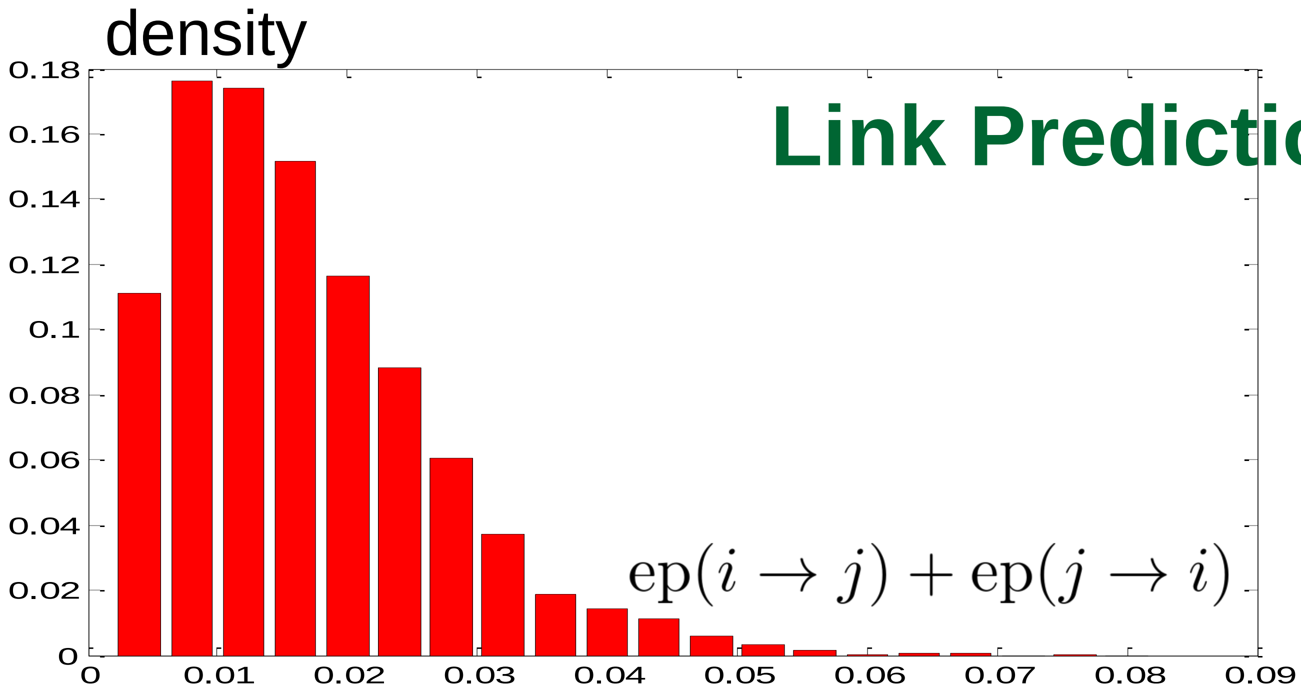
Experimental results

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

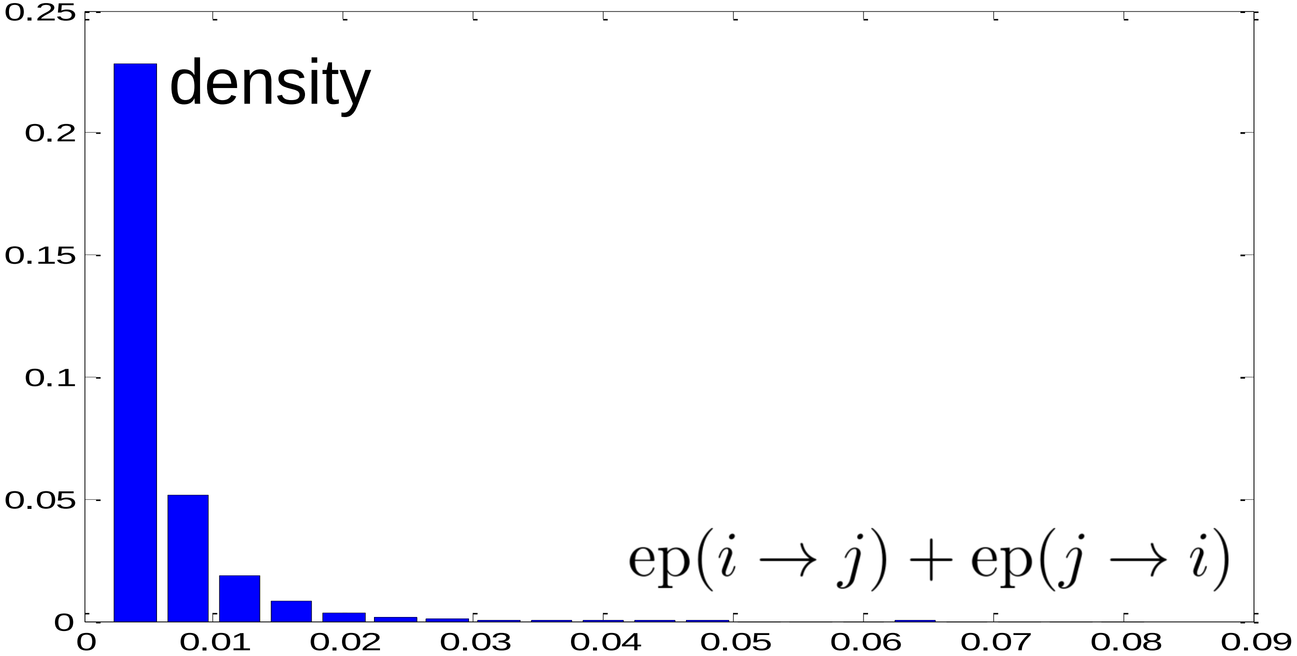
Datasets (all real)

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

Link Prediction: existence



with link



no link

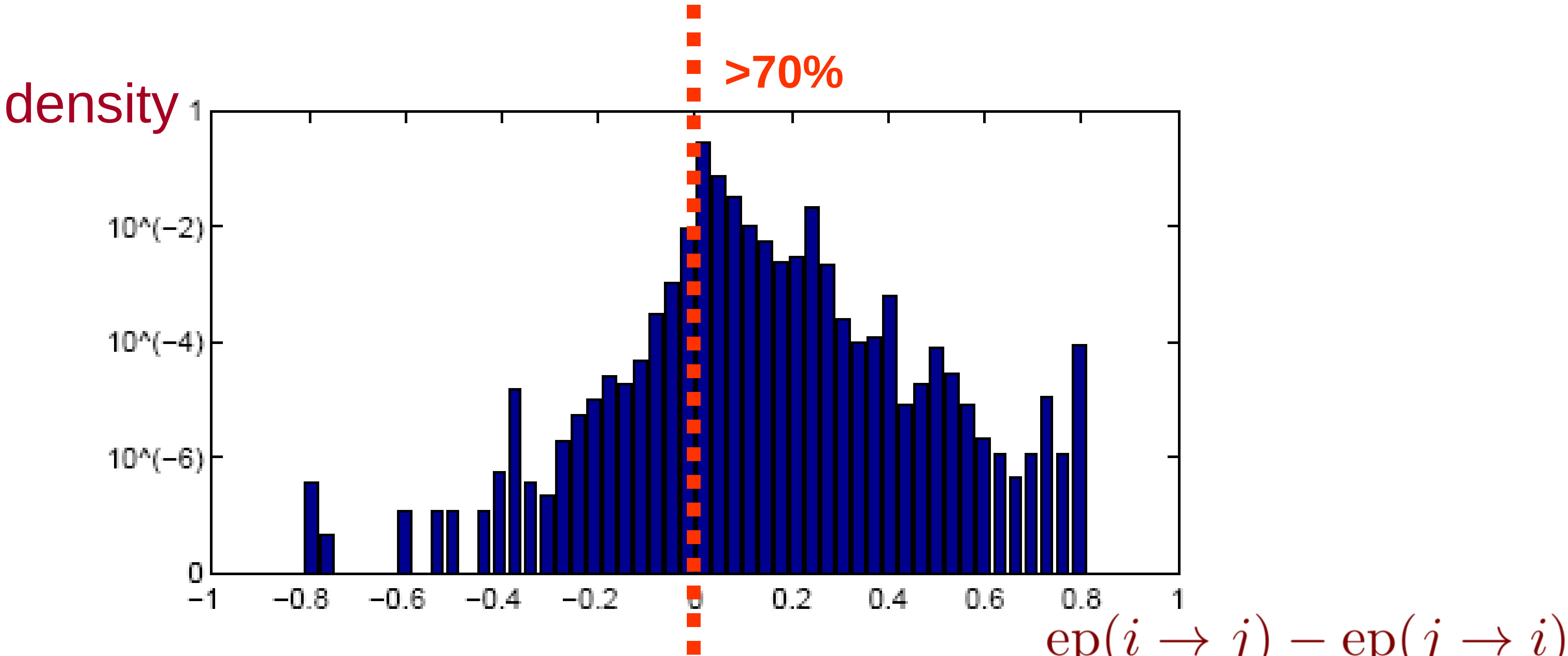
Link Prediction: existence

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

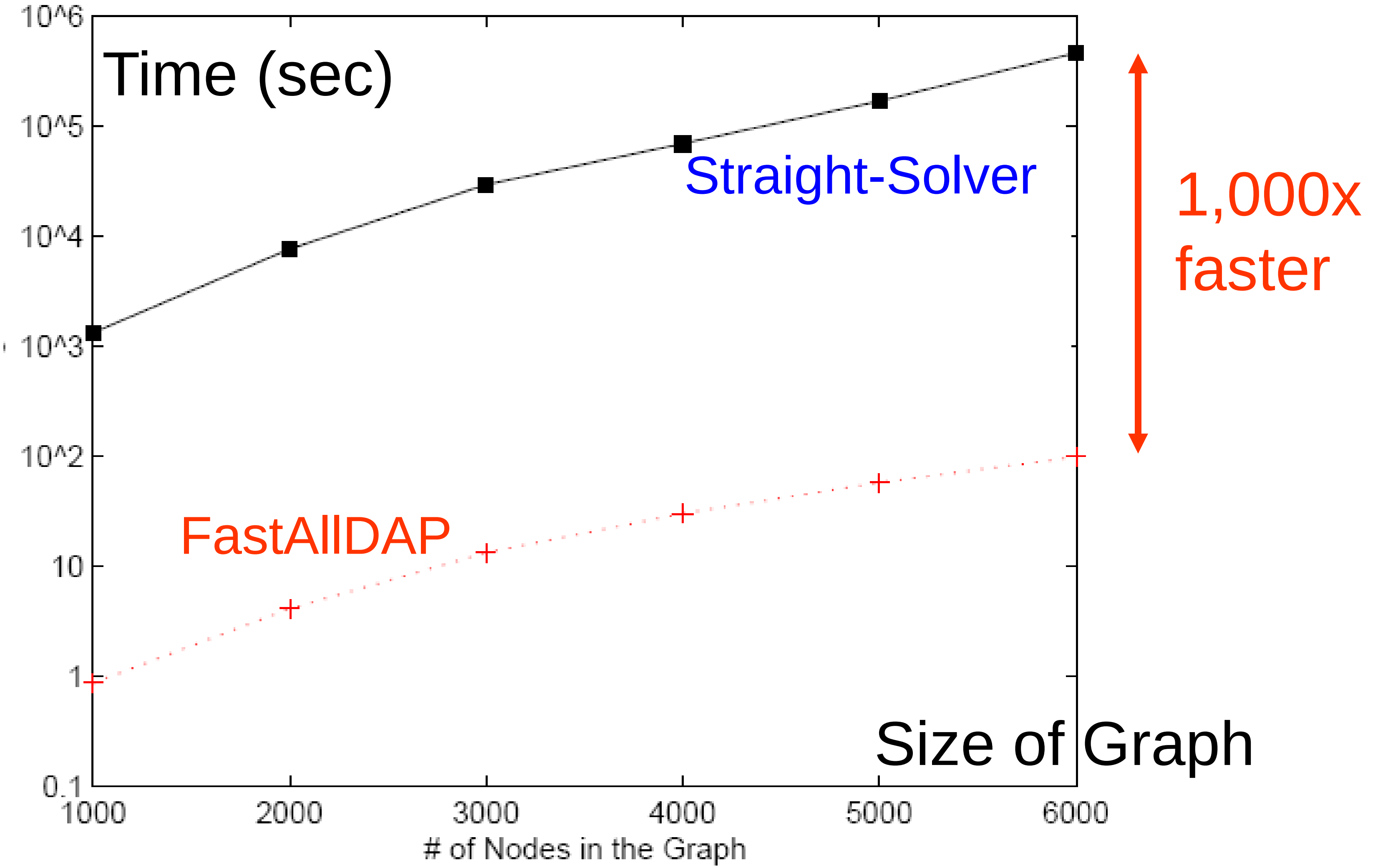
Dataset	Accuracy
WL	**65.40%**
PC	**79.60%**
AE	**81.51%**
CN	**86.71%**
EP	**92.21%**

Link Prediction: direction

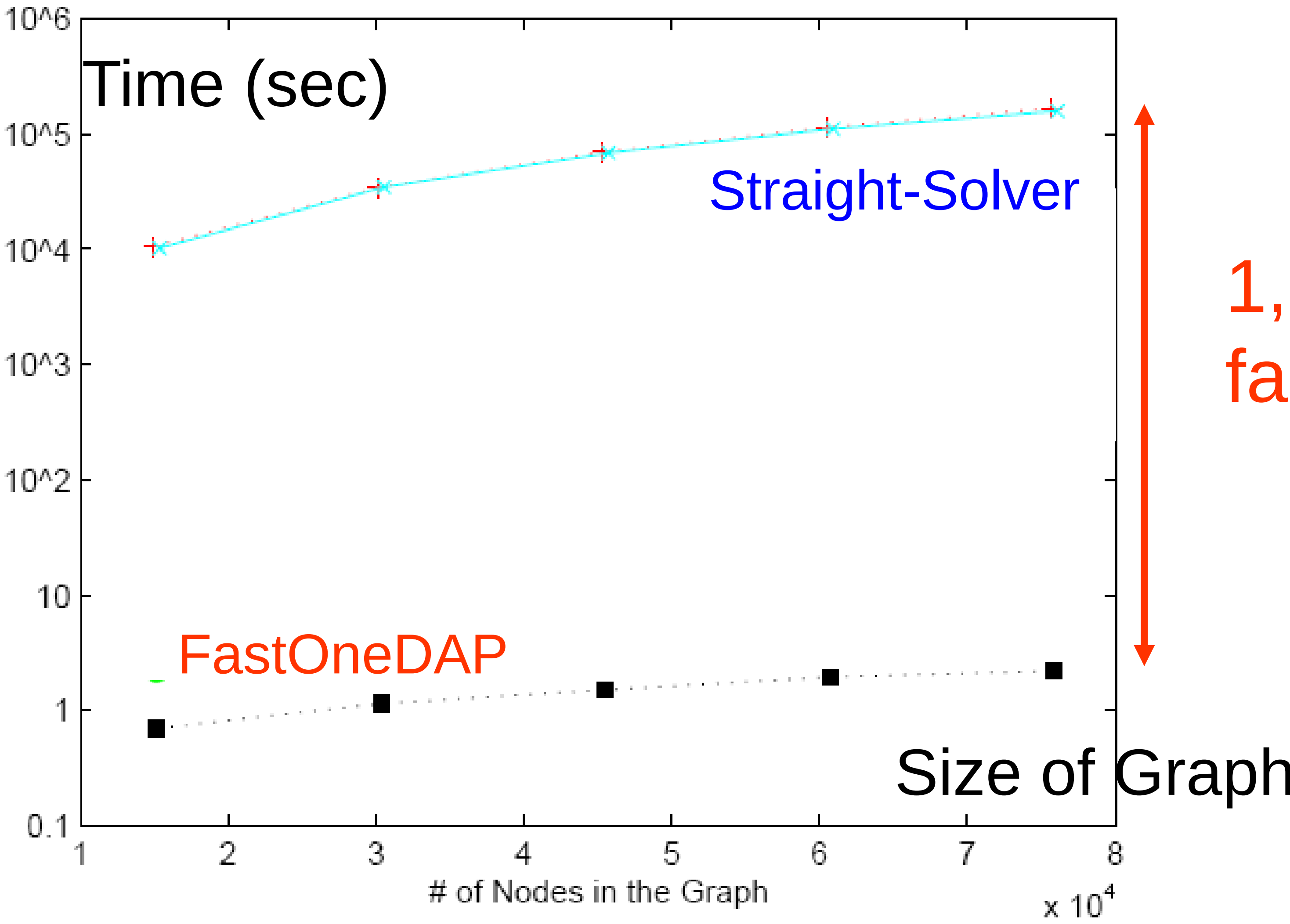
- Q: Given the existence of the link between *i* and *j*, what is the direction of it?
- A: Compare $\mathrm{ep}(i \to j)$ and $\mathrm{ep}(j \to i)$, pick the greater one



Efficiency: Fast all-pair proximities



Efficiency: Fast one-pair proximity



1,0000x faster

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*

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

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*

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

- Intuitively[1]
  - 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 $\frac{T}{\pi_i \cdot T} = \frac{1}{\pi_i}$

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

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*

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

- Intuitively[1]
  - 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 Numbers[2]

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 is precisely* $\mathrm{ep}(i \to j)$, *satisfies*

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

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

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*

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

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

- Proof[1]
  - 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 $E(Y) = c(i,j)$

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

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*

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

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

- Proof[1]
  - 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 $E(Y) = c(i,j)$
  - Cleary *X* ≤ *Y*, and $\Pr\left[X = Y\right] = p \triangleq \mathrm{ep}(i \to j)$

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

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*

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

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

- Proof[1]
  - 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 $E(Y) = c(i,j)$
  - Cleary *X* ≤ *Y*, and $\Pr\left[X = Y\right] = p \triangleq \mathrm{ep}(i \to j)$
    - $E(Y - X) = p \cdot 0 + (1-p) \cdot E(Y) = (1-p)c(i,j)$

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

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*

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

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

- Proof[1]
  - 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 $E(Y) = c(i,j)$
  - Cleary *X* ≤ *Y*, and $\Pr[X = Y] = p \triangleq \mathrm{ep}(i \to j)$
    - $E(Y - X) = p \cdot 0 + (1-p) \cdot E(Y) = (1-p)c(i,j)$
  - Also $E(Y - X) = E(Y) - E(X) = c(i,j) - \frac{1}{\pi_i}$

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

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*

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

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