k-Nearest Neighbors in Uncertain Graphs

Student : Mengzi ZHAO
Professor : Mr. Fabian Suchanek, Mr. Silviu Maniu

Summary

- Context
- Problem
- Motivation
- Approach
- Conclusion

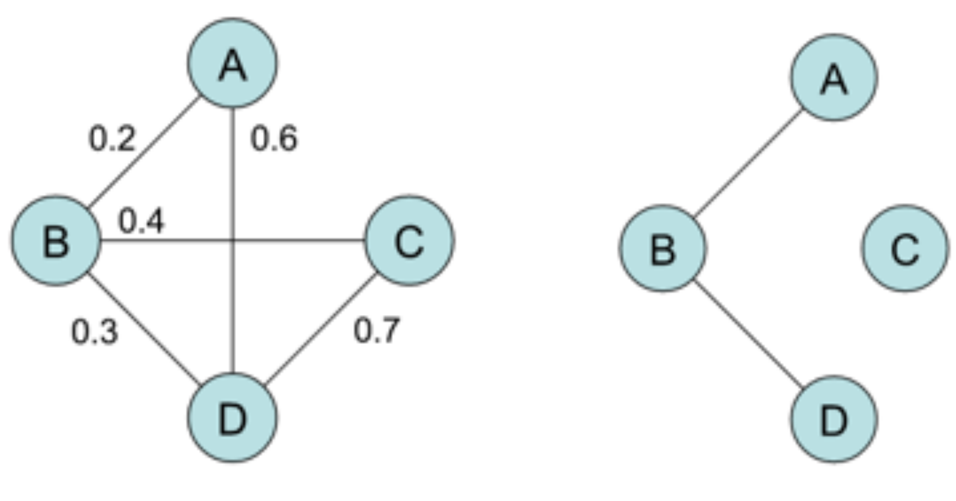
Context

- Complex networks : uncertainty => probabilistic graphs

Problem

- k-nearest neighbor queries (k-NN) : compute the k closest nodes to some specific node in a probabilistic graphs

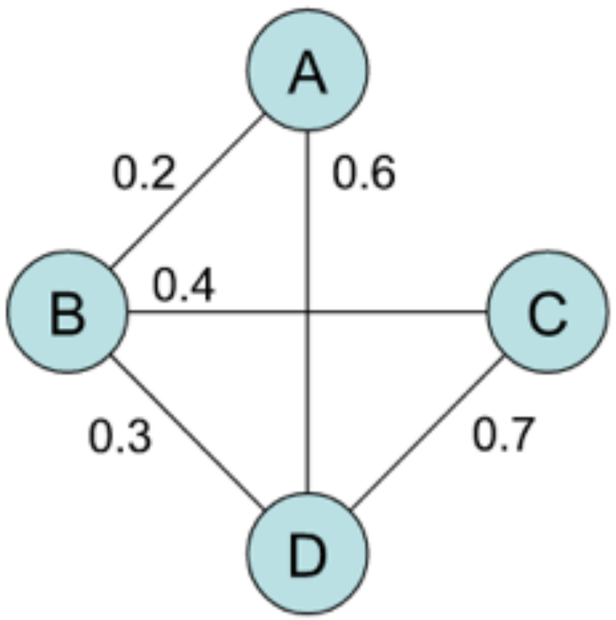
Probabilistic graph model



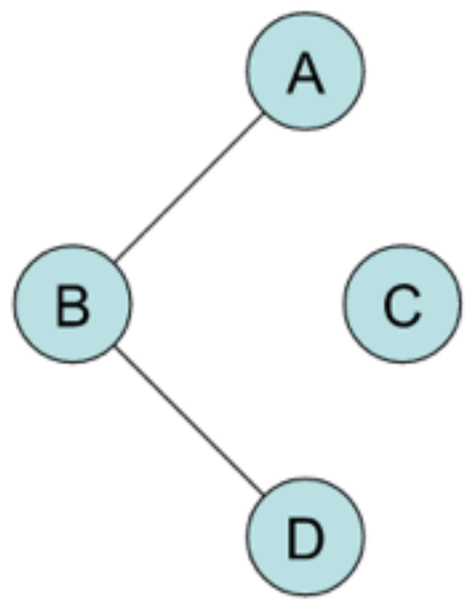
(a) (b)

- a) : probabilistic graph
- nb of edges : 5
- nb of possible worlds : 2^5
- b) : one of possible world sampled from graph a

Probabilistic graph model



(a)

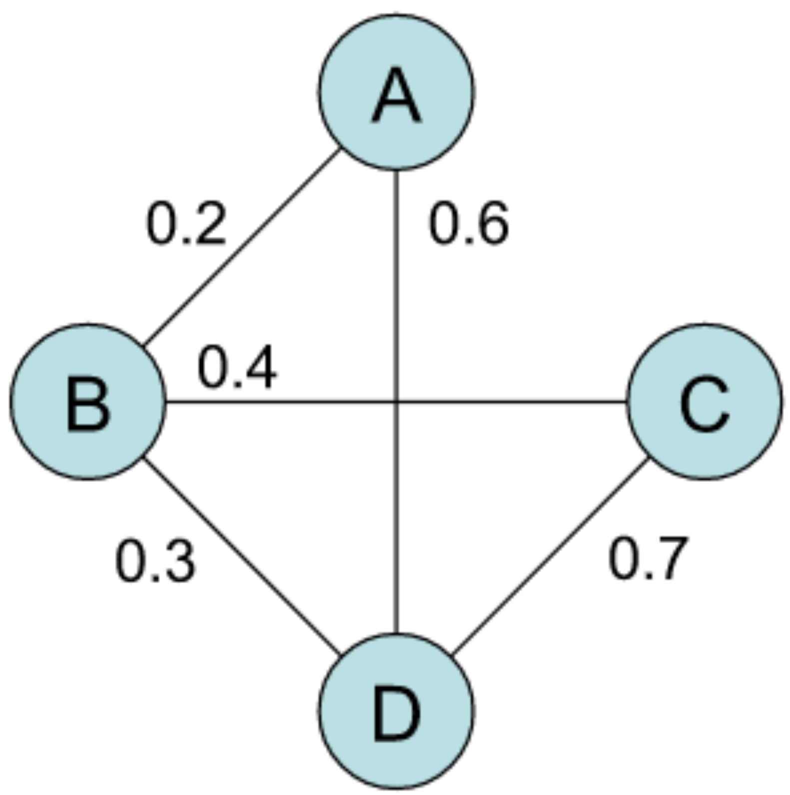


(b)

$$\Pr[G] = \prod_{e \in E_G} p(e) \prod_{e \in E \setminus E_G} (1 - p(e)).$$

- Probability of G :
- Pr[G] = $\underbrace{\overset{0.2}{p(AB)} * \overset{0.3}{p(BD)}}_{\text{probability of edges exist in (b)}} * \underbrace{(1\text{-}\overset{1-0.6}{p(AD)}) * (1 - \overset{1-0.4}{p(BC)} \ * (1 - \overset{1-0.7}{p(CD)})}_{\text{probability of edges exist in (a) but not in (b)}}$

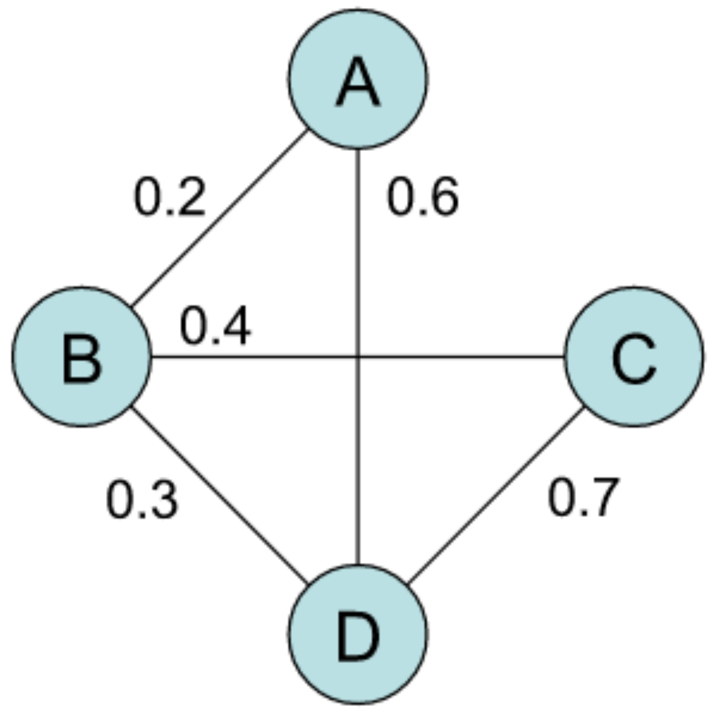
Probabilistic graph model



(a)

- from B to D
    - possible paths : BD, BCD, BAD

Probabilistic graph model

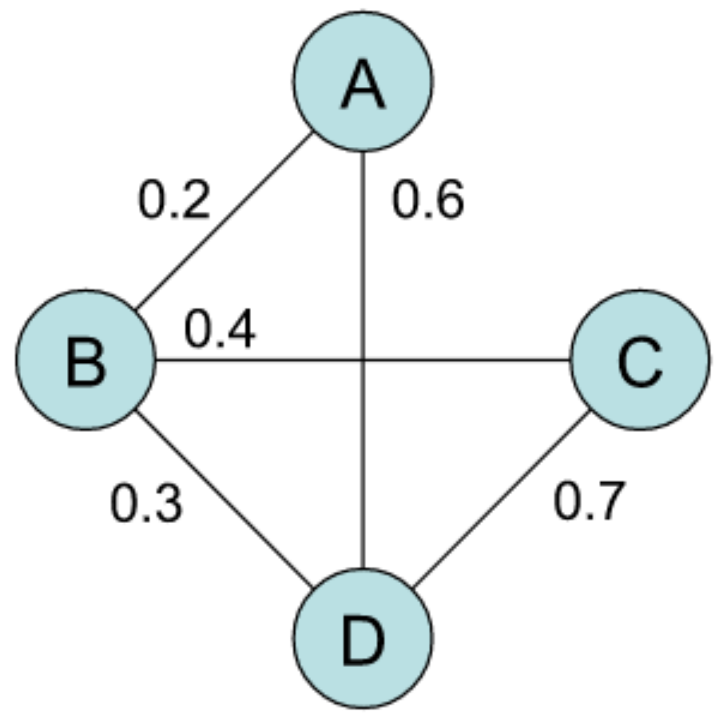


(a)

Reliability : at least one of possible paths between B and D exist

R = 1 - (1-p(BD))*(1-p(BAD))*(1-p(BCD)) = 0.56

Probabilistic graph model



(a)

MostProbPath distance : the length of the most probable path

Distribution of the shortest path distance between B and D :

(distance, probability)

(1,0.3),(2,0.26) ,(∞,0.44)

P(BD)=0.3 1-0.44-0.3=0.26 1-P=1-0.56=0.44

Probabilistic graph distance

- Limitations of MostProbPath function
  - The shortest path probability : may be arbitrarily small
    - ex : (1,0.02), (2,0.34), (∞,0.64)
  - Probability that it is indeed the shortest path : arbitrarily small
    - ex : (1,0.2),(2,0.1),(3,0.2),(4,0.2),(5,0.1),(∞,0.2)

Probabilistic graph distance

$$\mathbf{p}_{s,t}(d) = \sum_{G \mid d_G(s,t)=d} \Pr[G].$$

- G : possible world sampled from a graph
- $p_{s,t}(d)$: sum of the probabilities of all the worlds in which the shortest path distance between s and t is exactly d

Probabilistic graph distance

$$d_{\mathrm{M}}(s,t) = \arg\max_{D} \left\{ \sum_{d=0}^{D} \mathbf{p}_{s,t}(d) \leq \frac{1}{2} \right\}.$$

- Median-Distance $d_M(s,t)$
  - median shortest-path distance among all possible worlds

Probabilistic graph distance

$$d_{\mathrm{J}}(s,t) = \arg\max_{d} \mathbf{p}_{s,t}(d).$$

- Majority-Distance : the shortest-path distance that is most likely to be observed
- ex : (distance, $p_{s,t}(d)$)
  - (1,0.4), (2,0.25), (∞,0.35)
  - $d_J(s,t)$ = 1

Probabilistic graph distance

$$d_{\mathrm{ER}}(s,t) = \sum_{d \mid d<\infty} d \cdot \frac{\mathbf{p}_{s,t}(d)}{1-\mathbf{p}_{s,t}(\infty)}$$

- Expected-Reliable Distance :
  - distance in all worlds in which there exists a path between s and t

KNN problem

- find set of k nodes $T_k = \{ t_1 \ . . . , t_k \}$
  - distance between source and t_i =< distance between source and any other node in the graph

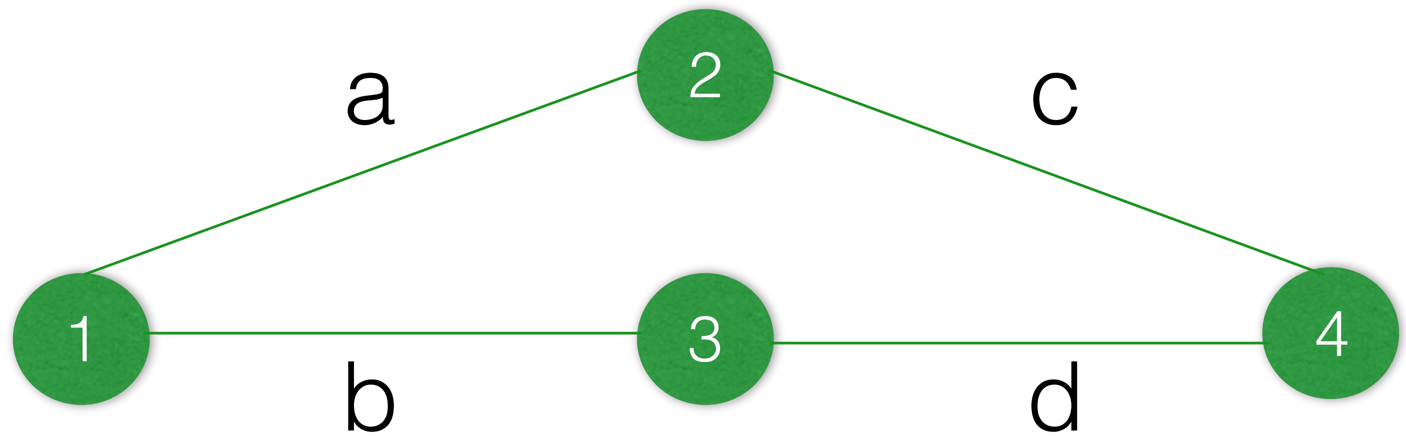
Computing distance functions

- Difficulty of Median-Distance :
  - execute algorithm in every world and taking the median
- Way to overcome : approximate Median-Distance using sampling
  - (i) sample r possible graphs according to probability of edges
  - (ii) compute the median distances for the sample graphs

Median-distance k-NN pruning

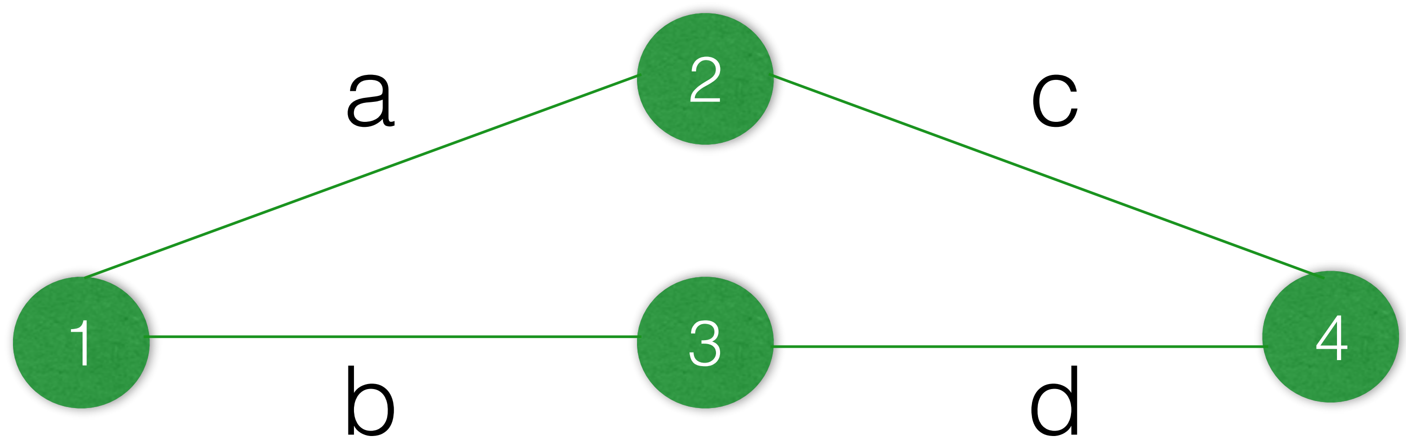
- input :
  - probabilistic graph G, node s, number of samples r, number of nearest-neighbor k, distance increment x
- output : T_k
  - result of KNN query
- Dijkstra : visite nodes in sampled graph

Median-distance k-NN pruning

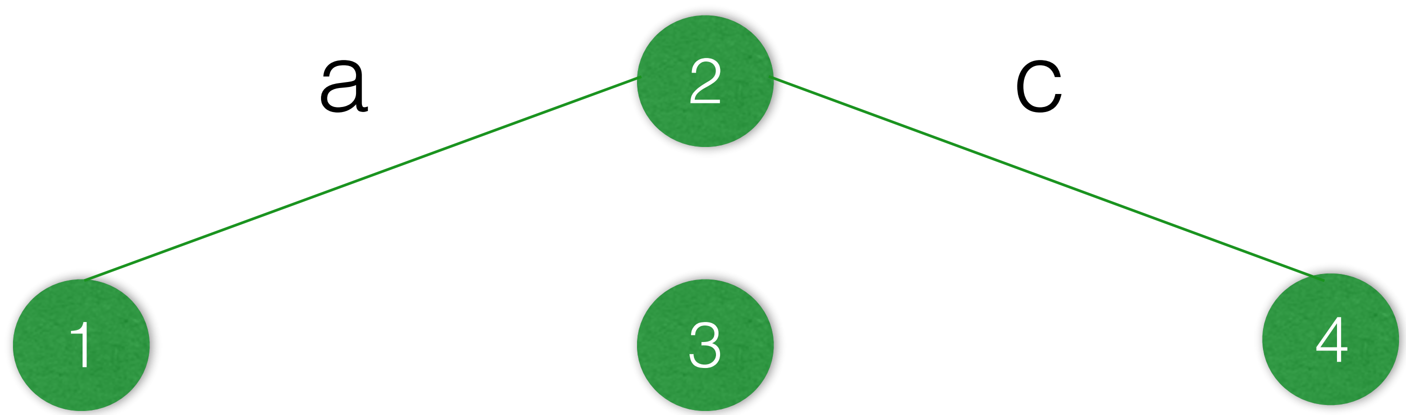


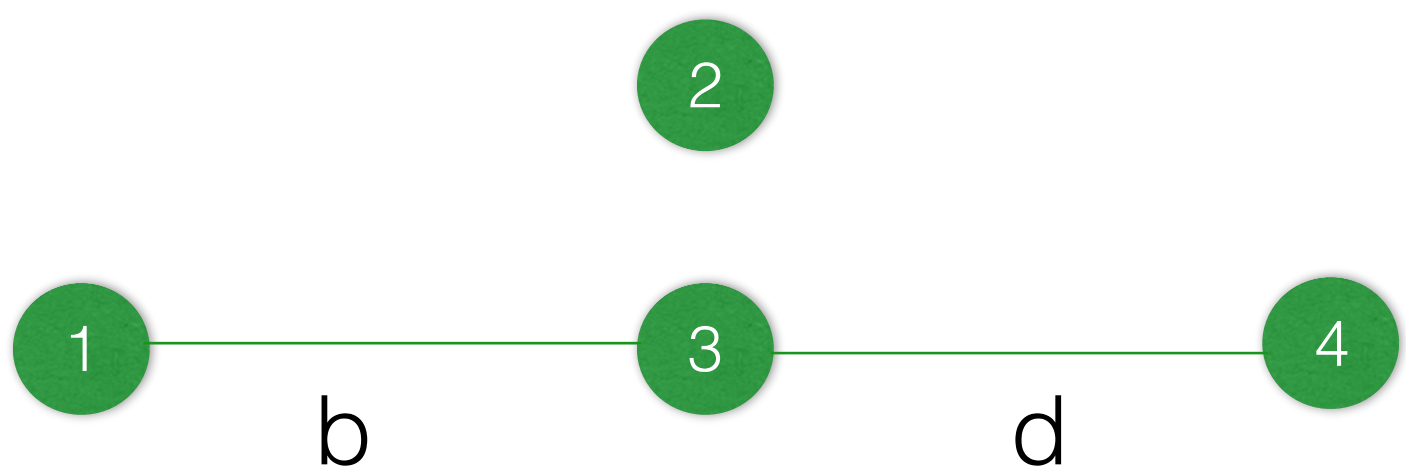
- $T_k \leftarrow \varnothing$, D←0
- suppose k = 2, x = 2, r = 2, s is node 1
- r : initiate r executions of Dijkstra from node 1

Median-distance k-NN pruning



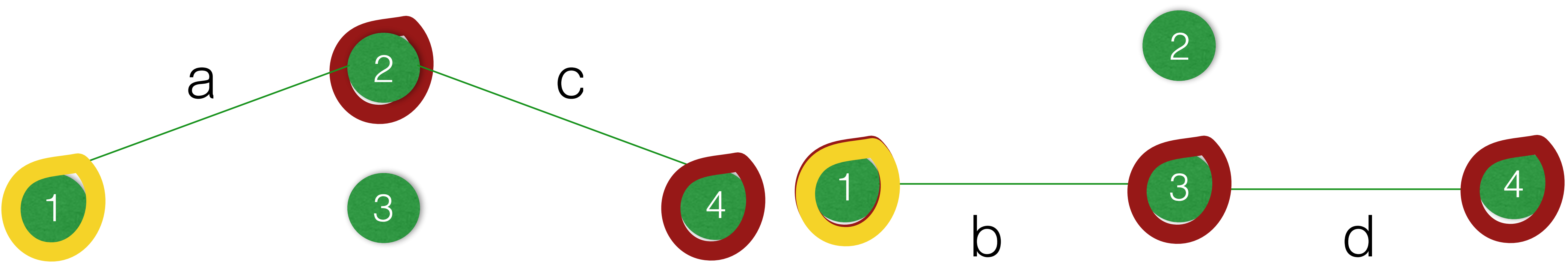
- 2 sampled graph from the main graph :





Median-distance k-NN pruning

- $|T_k| < k$, D = D + x = 0 + 2 = 2
- for 2 sampled graphs:
    - continue executing Dijkstra until reaching the distance D

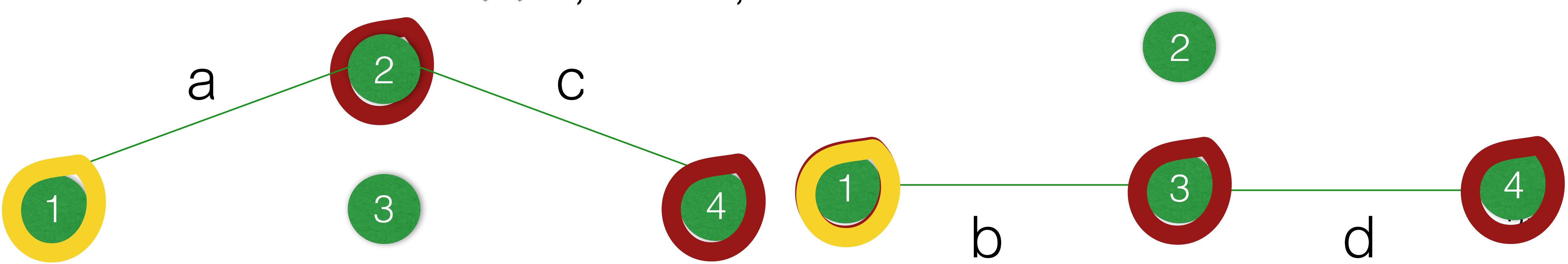


Median-distance k-NN pruning

- every node visited : update or instantiate $p_{D,s,t}$

$$\mathbf{p}_{D,s,t}(d) = \begin{cases} \mathbf{p}_{s,t}(d) & \text{if } d < D \\ \sum_{x=D}^{\infty} \mathbf{p}_{s,t}(x) & \text{if } d = D \\ 0 & \text{if } d > D \end{cases}$$

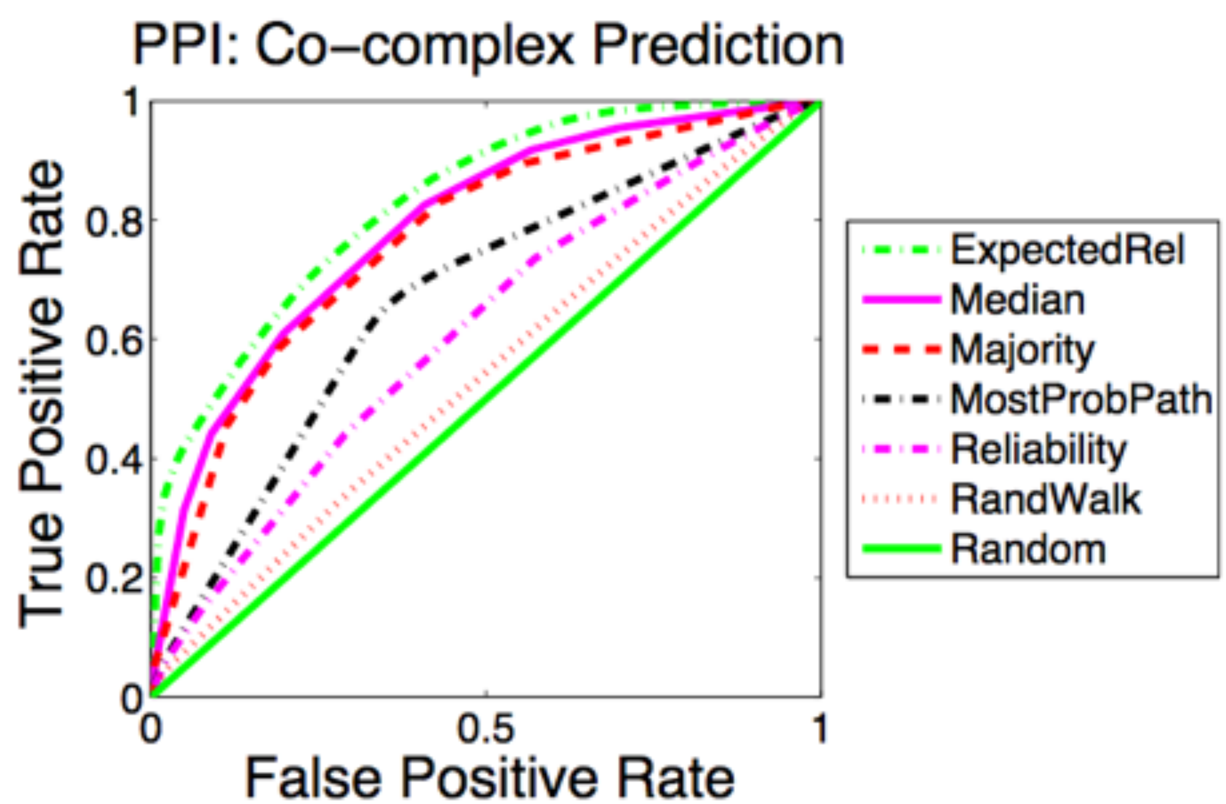
  - in this case
  - we have : $p_{2,1,2}$; $p_{2,1,3}$; $p_{2,1,4}$



Median-distance k-NN pruning

- for all nodes t who has pD,s,t and who doesn't exist in Tk :
    - If d of median(pD,s,t) < D, add the node in Tk
- |Tk| < k ?, if NO => algorithm is finished

Qualitative analysis



- ROC : Receiver Operating Characteristic
- x : false positive rate
- y : true positive rate

TPR = nb true positives detected / nb all true positive
FPR = nb false positives detected / nb all true negative

Efficiency analysis

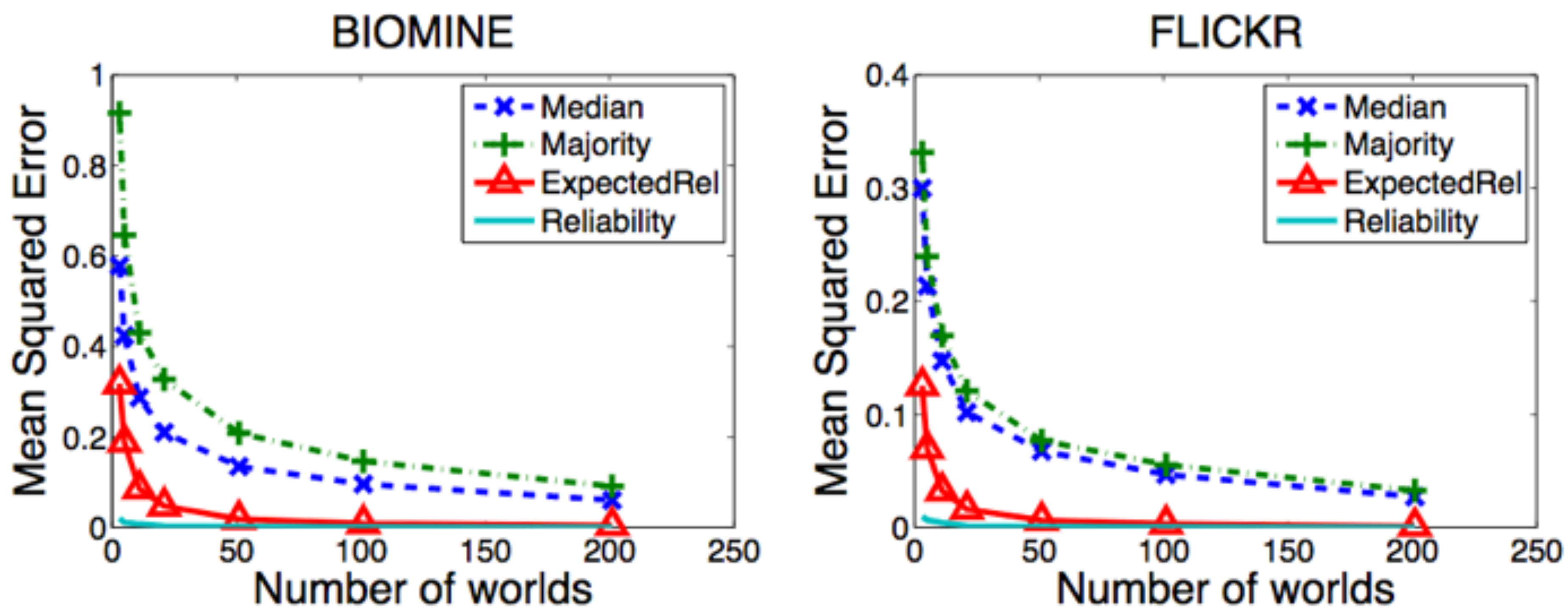


Figure 5: MSE vs. worlds. 200 worlds are enough.

- BIOMINE, FKICKR:datasets with a large amount of data
- MSE : converge to 0
- 200 worlds : enough to do answer KNN queries in the datasets with tens of millions of edges

KNN pruning

Table 1: Pruning Speedup.

MEDIAN, *200 Worlds*				
k	5	10	20	50
DBLP	269	267	208	185
BIOMINE	247	183	121	95
FLICKR	111	102	81	66
MAJORITY, *10-NN*				
Worlds	20	50	100	200
DBLP	18	22	22	23
BIOMINE	55	59	59	65
FLICKR	3.6	3.6	3.8	4.0

KNN pruning

Table 1: Pruning Speedup.

MEDIAN, *200 Worlds*				
k	5	10	20	50
DBLP	269	267	208	185
BIOMINE	247	183	121	95
FLICKR	111	102	81	66
MAJORITY, *10-NN*				
Worlds	20	50	100	200
DBLP	18	22	22	23
BIOMINE	55	59	59	65
FLICKR	3.6	3.6	3.8	4.0

Conclusion

- problem : KNN queries in large probabilistic graphs
- distance functions
- approximation algorithms based on sampling
- algorithms KNN pruning
- result is better than their competitors

Thanks for your attention !