Uncertain Data Management
Non-relational models: Graphs

Antoine Amarilli[1], **Silviu Maniu**[2]

[1]Télécom ParisTech

[2]Université Paris-Sud

January 16th, 2017

Credits

- M. Potamias, F. Bonchi, A. Gionis, G. Kolios. **k-Nearest Neighbors in Uncertain Graphs**. PVLDB 3(1), 2010. (number of samples, median measure, figure in slide 17, algorithm in slide 20)
- M. Ball. Computational Complexity of Network Reliability Analysis: An Overview. IEEE Trans. Reliab. R-35(3), 1986.
- L. Valiant. **The Complexity of Enumeration And Reliability Problems**. SIAM J. Comput. 8(3), 1979. (complexity of reliability/reachability)

PDFs of the slides available at http://silviu.maniu.info/teaching/

Uncertain Graphs

Graphs: a natural way to represent data in various domains

- transport data: road, air links between locations
- social networks: relationships between humans, citation networks
- interactions between proteins: contacts due to biochemical processes

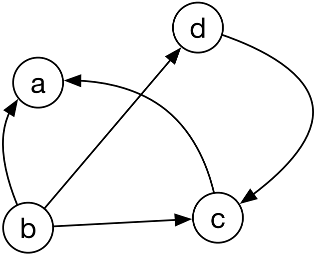
Uncertain Graphs

Graphs: a natural way to represent data in various domains

- transport data: road, air links between locations
- social networks: relationships between humans, citation networks
- interactions between proteins: contacts due to biochemical processes

For all the above examples, the links are not exact. (*Why?*)

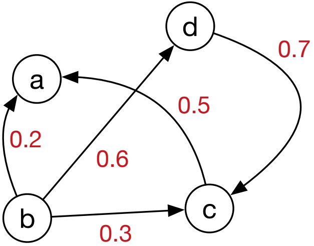
(Deterministic) Graphs



A graph $G = (V, E)$ is formed of

- a set V of vertices (nodes)
- a set $E \subseteq V \times V$, of edges

Uncertain Graphs



An uncertain graph $\mathcal{G} = (V, E, p)$ is formed of

- a set V of vertices (nodes)
- a set $E \subseteq V \times V$, of edges
- a function $p : E \to [0, 1]$, representing the probability p_e that the edge $e \in E$ exists or not

What are the possible worlds and their probability for this model?

Uncertain Graphs: Possible Worlds

A possible world of $\mathcal{G}$, denoted $G \sqsubseteq \mathcal{G}$ is a *deterministic* graph $G = (V, E_G)$ where each $e \in E_G$ is chosen from E

Uncertain Graphs: Possible Worlds

A possible world of $\mathcal{G}$, denoted $G \sqsubseteq \mathcal{G}$ is a *deterministic* graph $G = (V, E_G)$ where each $e \in E_G$ is chosen from E

The probability of G is:

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

How many possible worlds are there?

Uncertain Graphs: Other models

Other models are possible:

- each edge is replaced by a distribution of weights – instead of choosing if the edge exists or not, a possible world is an instantiation of weights
- each edge has a formula of events, capturing correlations
- probabilities can be on nodes also – equivalent to the edge model (*Why?*)

Queries on Uncertain Graphs

Generally, the queries we want to answer are distance queries:

- the reachability or reliability query – get the probability that two nodes s and t are connected

Queries on Uncertain Graphs

Generally, the queries we want to answer are distance queries:

- the reachability or reliability query – get the probability that two nodes s and t are connected
- queries on the distance distribution:

$$p_{s,t}(d) = \sum_{G|d_G(s,t)=d} \Pr[G]$$

Queries on Uncertain Graphs

Generally, the queries we want to answer are distance queries:

- the reachability or reliability query – get the probability that two nodes s and t are connected
- queries on the distance distribution:

$$p_{s,t}(d) = \sum_{G \mid d_G(s,t)=d} \Pr[G]$$

Multiple uses of distance queries:

- link prediction, social search, travel estimation

Queries on Uncertain Graphs

Generally, the queries we want to answer are distance queries:

- the reachability or reliability query – get the probability that two nodes s and t are connected

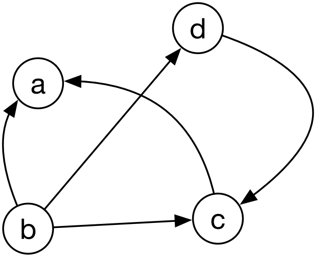
Queries on Uncertain Graphs

Generally, the queries we want to answer are distance queries:

- the reachability or reliability query – get the probability that two nodes s and t are connected
- queries on the distance distribution:

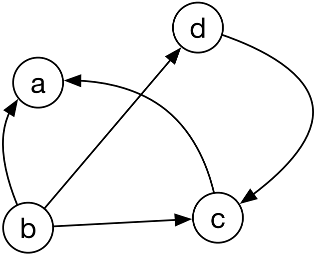
$$p_{s,t}(d) = \sum_{G \mid d_G(s,t)=d} \Pr[G]$$

Queries on Uncertain Graphs



What is the distance (in hops) between *b* and *a* ?

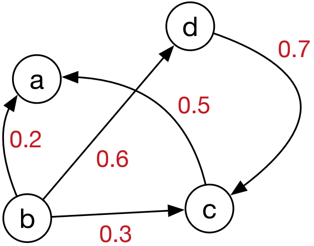
Queries on Uncertain Graphs



What is the distance (in hops) between b and a ?

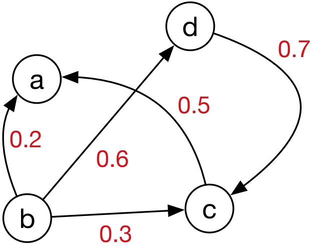
- BFS search (or Dijkstra's algorithms) finds the edge $b \rightarrow a$
- the cost is $\mathrm{O}(E)$ (linear in the size of the graph)

Queries on Uncertain Graphs



What is the distance (in hops) between b and a ?

Queries on Uncertain Graphs

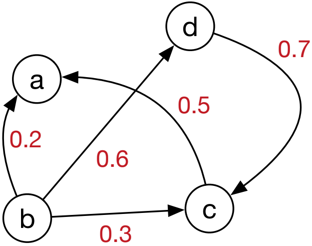


What is the distance (in hops) between b and a ?

- the edge $b \to a$ does not appear in all possible worlds:

$$p_{b,a}(1) = p(b \to a)$$

Queries on Uncertain Graphs



What is the distance (in hops) between b and a ?

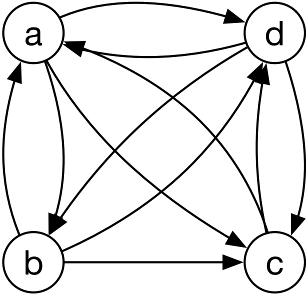
- the edge $b \rightarrow a$ does not appear in all possible worlds:

$$p_{b,a}(1) = p(b \rightarrow a)$$

- there are two possible paths of distance 2 ($b \rightarrow c \rightarrow a$) and 3 ($b \rightarrow d \rightarrow c \rightarrow a$)

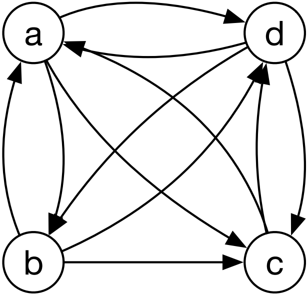
$$p_{b,a}(1) = (1 - p_{b,a}(1)) \times p(b \rightarrow c \rightarrow a)$$

Queries on Uncertain Graphs



What is the distance (in hops) between b and a ?

Queries on Uncertain Graphs



What is the distance (in hops) between *b* and *a* ?

- the number of paths is exponential in the size of the graph
- specifically, there are $3!$ paths

Queries on Uncertain Graphs

Distance query answering in uncertain graphs is at least as hard as in relational databases (*logical formulas* of paths; the number of which can be exponential)

Queries on Uncertain Graphs

Distance query answering in uncertain graphs is at least as hard as in relational databases (*logical formulas* of paths; the number of which can be exponential)

Computing the reachability probability (i.e, computing the probability of there being a path between a source and a target) is known to be #P hard [Valiant, SIAM J. Comp, 1979]

Computing Answers to Distance Queries on Probabilistic Graphs

Distance estimations in uncertain graphs can be approximated via Monte Carlo sampling

Computing Answers to Distance Queries on Probabilistic Graphs

Distance estimations in uncertain graphs can be approximated via Monte Carlo sampling

1. generate sampled graphs for r rounds (is this the optimal way for an s, t distance estimation?)
2. compute the desired measure (reachability probability, distance distributions) by averaging results

Computing Answers to Distance Queries on Probabilistic Graphs

Distance estimations in uncertain graphs can be approximated via Monte Carlo sampling

1. generate sampled graphs for r rounds (is this the optimal way for an s, t distance estimation?)
2. compute the desired measure (reachability probability, distance distributions) by averaging results

Same issue: *how many rounds?*

Number of Samples: Median Distance

Median distance:

$$d_M(s, t) = \arg\max_D \left\{ \sum_{d=0}^{D} p_{s,t}(d) \leqslant \frac{1}{2} \right\}$$

Number of Samples: Median Distance

Median distance:

$$d_M(s,t) = \arg\max_D \left\{ \sum_{d=0}^{D} p_{s,t}(d) \leqslant \frac{1}{2} \right\}$$

Let μ be the real median, and α and β values $\pm\epsilon N$ away from μ. Then for:

$$r > \frac{c}{\epsilon^2} \log(\frac{2}{\delta})$$

and a good choice of c:

$$\Pr(\hat{\mu} \in [\alpha, \beta]) > 1 - \delta$$

Number of Samples: Expected Distance

Expected reliable distance (generalization of reliability):

$$d_{\mathsf{ER}}(s,t) = \sum_{d \mid d<\infty} d \cdot \frac{p_{s,t}(d)}{1 - p_{s,t}(\infty)}$$

Number of Samples: Expected Distance

Expected reliable distance (generalization of reliability):

$$d_{\mathsf{ER}}(s,t) = \sum_{d \mid d < \infty} d \cdot \frac{p_{s,t}(d)}{1 - p_{s,t}(\infty)}$$

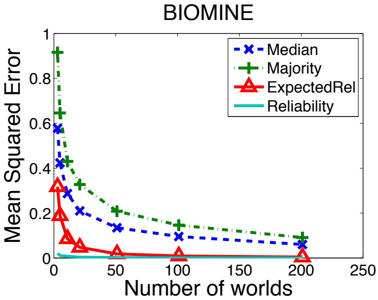
By estimating the connectivity ρ, we need to sample at least:

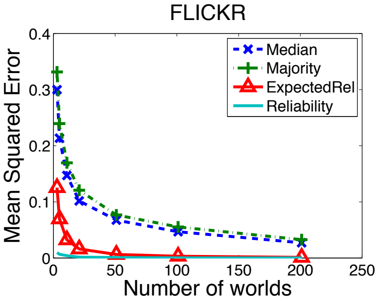
$$r \geqslant \max\left\{\frac{3}{\epsilon^2\rho}, \frac{(n-1)^2}{2\epsilon^2}\right\} \cdot \log\left(\frac{2}{\delta}\right)$$

for an (ϵ, δ) approximation.

Number of Samples In Reality

The number of needed samples can be surprisingly low (but it depends on the actual probabilities)





Sampling Graphs

Generating the entirety of the graph G_i for each round $i < r$ is not optimal

Sampling Graphs

Generating the entirety of the graph G_i for each round $i < r$ is not optimal

- we do not need to estimate the entire graph G_i
- we can start from s and do a BFS or Dijkstra search by sampling only the outgoing edges
- based on the generated outgoing edges, we re-do the computation for each generated outgoing node, until we find t

Example: Median Distance k-NN

k-NN (k nearest neighbours) – finding the k nodes from s the "closest" by some measure

- let us consider the median distance (reminder: it is the highest distance in the distribution that has mass less or equal to 0.5)

Example: Median Distance k-NN

k-NN (k nearest neighbours) – finding the k nodes from s the "closest" by some measure

- let us consider the median distance (reminder: it is the highest distance in the distribution that has mass less or equal to 0.5)

We only care about the top-k nodes, and not their values, and we do not want to evaluate all the graph if possible

- we can evaluate a truncated distribution up to a distance D

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

- for any two nodes t_1, t_2, $d_{D,M}(s, t_1) < d_{D,M}(s, t_2)$ implies $d_M(s, t_1) < d_M(d, t_2)$

Example: Median Distance k-NN

Input: Probabilistic graph $\mathcal{G} = (V, E, P, W)$, node $s \in V$, number of samples r, number k, distance increment γ
Ouput: T_k, a result set of k nodes for the k-NN query

$T_k \leftarrow \emptyset$; $D \leftarrow 0$
Initiate r executions of Dijkstra from s
while $|T_k| < k$ **do**
    $D \leftarrow D + \gamma$
    for $i \leftarrow 1 : r$ **do**
        Continue visiting nodes in the i-th execution of Dijkstra until reaching distance D
        For each node $t \in V$ visited update the distribution $\tilde{\mathbf{p}}_{D,s,t}$ {Create the distribution $\tilde{\mathbf{p}}_{D,s,t}$ if t has never been visited before}
    end for
    for all nodes $t \notin T_k$ for which $\tilde{\mathbf{p}}_{D,s,t}$ exists **do**
        if median$(\tilde{\mathbf{p}}_{D,s,t}) < D$ **then**
            $T_k \leftarrow T_k \cup \{t\}$
        end if
    end for
end while

- start from a small distance D
- decide whether there are nodes to add to the k-NN set
- increase the distance, and "re-start" each sampled graph from the new distance

Example: Median Distance *k*-NN

The algorithm does not need to visit all nodes