Social and Graph Data Management
Probabilistic Graphs and Influence Algorithms

Benoît Groz (slides and content are mostly from Silviu Maniu)

December 1st, 2023

M2 Data Science

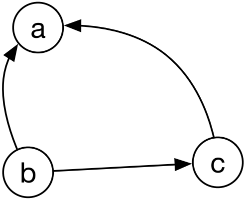
Uncertain Graphs

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?*)

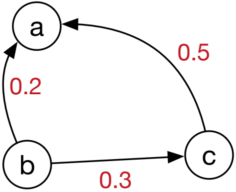
(Deterministic) Graphs



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

Uncertain Graphs: Possible Worlds

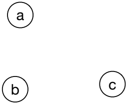
A possible world of $\mathcal{G}$, denoted $G \sqsubseteq \mathcal{G}$ is a *deterministic* graph $G = (V, E_G)$ where each $e \in E_G$ is chosen from E

The probability of G is:

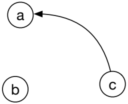
$$\Pr[G] = \prod_{e \in E_G} p_e \prod_{e \in E \setminus E_G} (1 - p_e)$$

How many possible worlds are there?

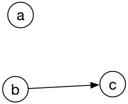
Example: Possible Worlds



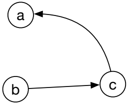
$\Pr[G_1] = 0.8 \times 0.7 \times 0.5 = 0.28$



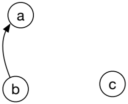
$\Pr[G_2] = 0.8 \times 0.7 \times 0.5 = 0.28$



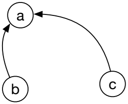
$\Pr[G_3] = 0.8 \times 0.3 \times 0.5 = 0.12$



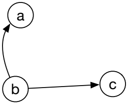
$\Pr[G_4] = 0.8 \times 0.3 \times 0.5 = 0.12$



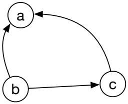
$\Pr[G_5] = 0.2 \times 0.7 \times 0.5 = 0.07$



$\Pr[G_6] = 0.2 \times 0.7 \times 0.5 = 0.07$



$\Pr[G_7] = 0.2 \times 0.3 \times 0.5 = 0.03$



$\Pr[G_8] = 0.2 \times 0.3 \times 0.5 = 0.03$

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

Queries on Uncertain Graphs

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

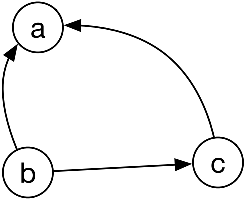
- the **reachability** or **reliability** query – get the probability that two nodes s and t are connected
- queries on the **distance distribution**:

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

Multiple uses of distance queries:

- link prediction, social search, travel estimation

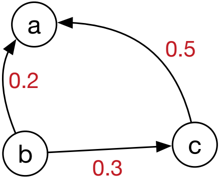
Queries on Deterministic Graphs



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

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

Queries on Uncertain Graphs: Reachability



What is the probability that we can reach a from b?

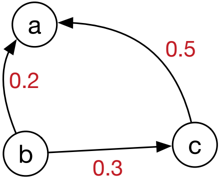
- **logical formula**: going through the edge (b, a) or the path $b \to c \to a$:

$$\begin{aligned}\Pr[b \to a] &= p(b,a) + \\ &\quad + (1 - p(b,a))p(b,c)p(c,a) \\ &= 0.2 + 0.8 \times 0.3 \times 0.5 = 0.32\end{aligned}$$

- or by **counting** the number of possible worlds in which there is a path from b to a

$$\begin{aligned}\Pr[b \to a] &= \Pr[G_3] + \Pr[G_4] + \Pr[G_5] + \\ &\quad + \Pr[G_6] + \Pr[G_7] = 0.32\end{aligned}$$

Queries on Uncertain Graphs: Distance Distribution



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

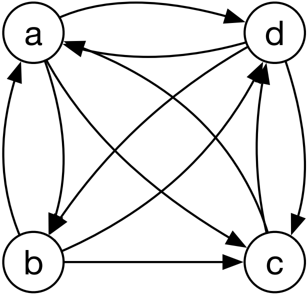
- the edge $b \to a$ does not appear in all possible worlds:

$$\Pr[d(b,a) = 1] = p(b,a) = 0.2$$

- there is one possible path of distance 2 ($b \to c \to a$)

$$\begin{aligned}\Pr[d(b,a) = 2] &= (1 - \Pr[d(b,a) = 1]) \\ &\quad \times p(b,c)p(c,a) \\ &= 0.8 \times 0.3 \times 0.5 = 0.12 \\ \Pr[d(b,a) = \infty] &= 1 - \Pr[d(b,a) = 1] \\ &\quad - \Pr[d(b,a) = 2] = 0.68\end{aligned}$$

Queries on Uncertain Graphs



What is the distance (in hops) between ***b*** and ***a***, or what is the reachability probability? We have to write a formula over all paths.

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

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 – as hard as model counting

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

Main issue: *how many rounds are needed?*

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

Table of contents

Social Influence

Social Influence: important problem in social network, with applications in marketing, computational advertising

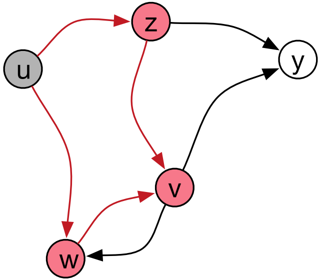
Objective: given a promotion budget of k social network users, maximize the expected influence spread given some influence or propagation model

Social Influence

Data Model: an uncertain graph $G(V, E, p)$

- V and E are the social network
- p is, on each edge, the influence probability

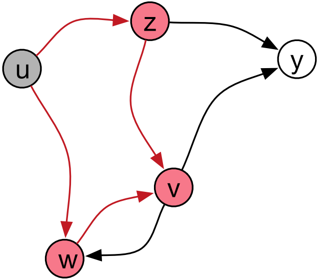
Influence Spread via Cascades



Independent Cascade Model: discrete time model of propagation

1. at time 0, activate seed u
2. for a node i activated at time t: activate at time $t+1$ each neighbour v with probability p_{iv}
3. once a node is activated, it cannot be activated again or de-activated

Influence Spread via Cascades



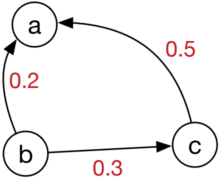
We wish to compute the **expected spread** from a seed seed set S, $\sigma(S)$

By linearity of expectation:

$$\sigma(u) = \sum_{v \in V} \Pr(u \to v)$$

- for a seed set S, more complicated
- same hardness as **reachability**

Influence Spread



Influence spread of each node in V:

$$\begin{aligned}\sigma(a) &= \Pr[a \to a] + \Pr[a \to b] + \Pr[a \to c] \\ &= 1 + 0 + 0 = 1\end{aligned}$$

$$\begin{aligned}\sigma(b) &= \Pr[b \to a] + \Pr[b \to b] + \Pr[b \to c] \\ &= 0.32 + 1 + 0.3 = 1.62\end{aligned}$$

$$\begin{aligned}\sigma(c) &= \Pr[c \to a] + \Pr[c \to b] + \Pr[c \to c] \\ &= 0.5 + 0 + 1 = 1.5\end{aligned}$$

Maximizing the Influence

Influence maximization is **computationally hard**

Two **sources of hardness**:

1. computing $\sigma(S)$ is #P-hard (as we seen before, it is equivalent to **reachability**) – Monte Carlo with additive approximations
2. computing the selection of k seeds in S is NP-hard – maximization of a **submodular** function

Submodular function: the influence spread is submodular:

$$\sigma(S \cup \{u\}) - \sigma(S) \geqslant \sigma(T \cup \{u\}) - \sigma(T) \quad \text{if} \quad S \subseteq T$$

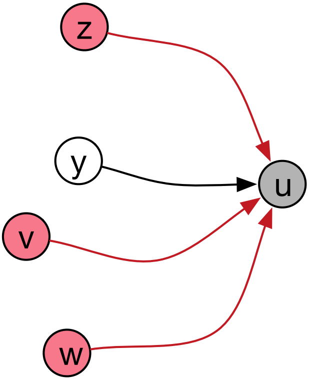
Influence Maximization: Greedy Algorithm

We can obtain a $(1 - \frac{1}{\epsilon})$-approximation factor for influence maximization by using the **greedy algorithm**

Steps:

1. initialize $S = \emptyset$
2. choose the user u that maximizes $\sigma(S \cup \{u\}) - \sigma(S)$
3. $S = S \cup u$
4. repeat steps 2 and 3 k times
5. **return** S

Learning Propagation Probabilities



The probability that v is influenced by its neighbours

$$\Pr(v) = 1 - \prod_{u}(1 - p_{uv})$$

Given a log of actions $A = \{(\text{act}, u, v), \dots\}$:

1. maximum likelihood: $p_{vu} = \frac{A_{vu}}{A_v}$
2. Jaccard similarity: $p_{vu} = \frac{A_{vu}}{A_{u|v}}$

References i