# Algorithms for Large Uncertain Graphs

Samira Daruki

Database Seminar

# Motivation

- Network data is the core of many scientific fields such as social, biological and mobile ad-hoc networks
- Such real network data is associated with *uncertainty*:

  - data collection process

  - machine-learning methods employed at preprocessing

  - privacy-preserving reasons

# Problems on Uncertain Graphs

- Clustering
- Finding k-Nearest Neighbors
- Efficient Subgraph Search
- Distant-Constraint Reachability Computation
- Query Answering on Uncertain RDF Graphs
- ….

# Clustering Uncertain Graphs

- Partitioning graphs into clusters is a fundamental problem for uncertain graphs just as for deterministic graphs

- Many application such as finding complexes in protein-protein interaction networks, communities of users in social networks

# Uncertain Graph Model

- Represent uncertain graph Ĝ= <V, P, W>
- V : set of nodes
- P: maps every pair of nodes to a real number in [0, 1]
- P_uv : represents the probability that edge {u, v} exists
- For weighted graphs: W: V x V → R denotes the weights associated with every edge

# Uncertain Graph as a generative model

- Uncertain Graph is a generative model for deterministic graphs
- A deterministic graph G is generated by Ĝ by connecting two nodes u, v via an edge with probability P_uv
- The probability that G=(V, E) sampled from Ĝ = (V, P) ) is: $\Pr[G] = \prod_{\{u,v\} \in E_G} P_{uv} \prod_{\{u,v\} \in (V \times V) \setminus E_G} (1 - P_{uv}).$

# Naive approaches for clustering

- Considering the edge probabilities as *weights*

  - no meaningful way to perform such a casting

  - no easy way to additionally encode normal weights on the edges

- Setting a *threshold* value to the edge probabilities and ignore any edge below that

  - no principled way of deciding what the right value of the threshold

# Uncertain Graph Clustering Formulation

- First define the *edit distance* between two graphs
- Generalize the definition for uncertain graphs
- Set the objective for clustering as *a cluster graph* (a special deterministic graph consists of vertex-disjoint disconnected cliques)
- Use this definition to formulate the uncertain graph clustering as an *optimization problem*

# Edit Distance on Deterministic Graphs

- Edit Distance for two deterministic graphs G, Q:

$$D(G,Q) = |E_G \setminus E_Q| + |E_Q \setminus E_G|.$$

- Using adjacency matrix notation:

$$D(G,Q) = \sum_{\substack{u=1, \\ v<u}}^{n} |\mathbf{G}(u,v) - \mathbf{Q}(u,v)|\,.$$

# Edit Distance on Uncertain Graphs

- Edit distance between an uncertain graph Ĝ and a deterministic graph Q:

  \- defined as the *expected* edit distance between every possible world G in uncertain graph Ĝ and Q → *Compute by generating all exponential possible worlds is <u>inefficient</u>!*

$$D(\mathcal{G}, Q) = \mathop{\mathbb{E}}_{G \sqsubseteq \mathcal{G}} [D(G, Q)] = \sum_{G \sqsubseteq \mathcal{G}} \Pr[G]\, D(G, Q).$$

# Polynomial Time! ... How?!

- Using adjacency matrices of G and Q:

$$\begin{aligned} D(\mathcal{G}, Q) &= \mathop{\mathbb{E}}_{G \sqsubseteq \mathcal{G}} \left[ \sum_{\substack{u=1 \\ v<u}}^{n} |\mathbf{G}(u,v) - \mathbf{Q}(u,v)| \right] \\ &= \mathop{\mathbb{E}}_{G \sqsubseteq \mathcal{G}} \left[ \sum_{\substack{u=1 \\ v<u}}^{n} X_{uv} \right] \\ &= \sum_{u<v} \left( \mathop{\mathbb{E}}_{G \sqsubseteq \mathcal{G}} X_{uv} \right) \\ &= \sum_{\{u,v\} \in E_Q} (1 - P_{uv}) + \sum_{\{u,v\} \notin E_Q} P_{uv}. \end{aligned}$$

# Clustering as optimization problem

- Given an uncertain graph Ĝ = (V, P), find the cluster graph C=(V, E) such that D(Ĝ, C) is *minimized*.

- The *cost function* is the same as weighted *correlation clustering* with probability constraints :

$$\mathrm{Cc}(\mathcal{P}) = \sum_{\substack{(u,v) \\ \mathcal{P}(u)=\mathcal{P}(v)}} W_{uv}^{-} + \sum_{\substack{(u,v) \\ \mathcal{P}(u)\neq\mathcal{P}(v)}} \left(1 - W_{uv}^{-}\right).$$

# Deviation from Expectation

- We focus on finding the cluster graph C that minimize the *expected* edit distance from the input uncertain graph!
- How large are the observed differences in D(G, C) across different worlds?

**Chernoff Bounds**

# Chernoff Bounds

- The mass of distribution over all possible worlds is concentrated around its mean:

$$\Pr\left[D\left(G,C\right) > (1+\delta)D\left(\mathcal{G},C\right)\right] < \left[\frac{e^{\delta}}{(1+\delta)^{1+\delta}}\right]^{D(\mathcal{G},C)},$$

$$\Pr\left[D\left(G,C\right) < (1-\delta)D\left(\mathcal{G},C\right)\right] < \left[\frac{e^{-\delta}}{(1-\delta)^{1-\delta}}\right]^{D(\mathcal{G},C)}.$$

- We can also easily show that the *variance* is bounded and independent of clustering C:

$$\sum_{u<v} \mathrm{Var}\left[X_{uv}\right] = \sum_{u<v} P_{uv}\left(1-P_{uv}\right).$$

# Open Problems

- Think about alternative clustering definitions:

  - find the cluster graph with the maximum probability

- Extend the proposed framework:

  - to support probabilistic assignments of nodes to clusters

  - to Identify overlapping clusters

# K-Nearest Neighbors in Uncertain Graphs

- A fundamental problem for uncertain graphs is to compute the k closest nodes to some specific node.
- Biological networks (PPI):

  - predicting possible memberships /new interaction
- Social Networks:

  - link prediction, influence of a person to another in viral marketing
- Mobile ad-hoc Networks:

  - addressing the probabilistic-routing problem

# Uncertain Graph Distance

- Most-Probable-Path:

  - computed easily by running Dijkstra algorithm on a deterministic weighted instance of graph with edge weights $-\log(p(e))$

- Limitations:

  - probability of the path may be arbitrary small
  - even if the probability of the path itself is large, the probability that it is indeed the shortest path can be arbitrary small

# Shortest Path Distribution

- Defined as the sum of the probabilities of all the possible worlds in which the shortest path distance between two node is exactly *d*.

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

# Distance Definitions

- ***MEDIAN-DISTANCE:*** the median shortest-path distance among all possible worlds

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

- ***MAJORITY-DISTANCE:*** the most probable shortest-path distance among all the possible worlds

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

# Computing the Median Distance

- Instead of executing a point-to-point shortest path algorithm in every possible world and taking the median, approximate it using ***Sampling:***

- *Sample r possible graphs according to P*
- *Compute the median of the shortest-path distances in the sample graphs*
- *Guarantee the bounds using Chernoff Bounds*

# Median-distance k-NN pruning

- Algorithm is based on exploring the **local neighborhood** around the source node s and computing the distribution p_(s , t), truncated to smaller distances.

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

# Approximating distribution

- Start from s, Perform a computation of *Dijkstra* algorithm:

  *when it is required to explore one node we generate (sample) the outgoing edges from that node. Stop when visit a node whose distance exceed* D.

- For all nodes t that were visited, either update or instantiate their distribution.

# Open Problems

- Enrich the proposed framework with more powerful models that can handle:

- *node failures* in computing shortest path

- arbitrary probability distribution

# Conclusion

- **Uncertain Graphs and Data** is an interesting research area that opens many research problems in different domains:

    Algorithms,

    Data Mining,

    Machine Learning,

    Database

    …

# Thanks!