DP Range Query on Shortest Paths

Discrete Math Seminar

Jesse Campbell

Duke Kunshan University

November 5, 2024

Motivation

You are a doctor, and you have a patient who needs to be transferred to another facility.

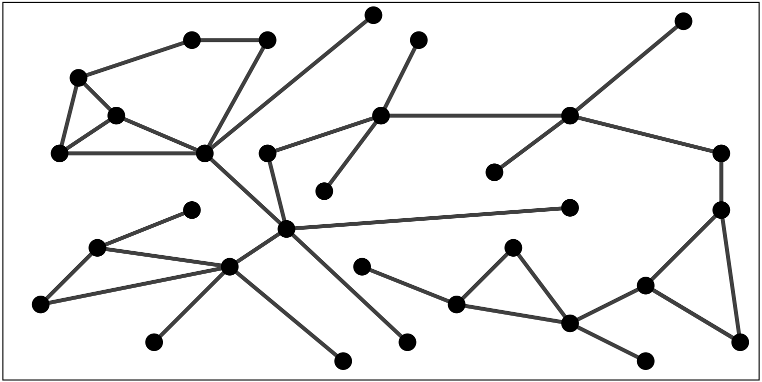
You are a doctor, and you have a patient who needs to be transferred to another facility.

You know the shortest paths between hospitals, but you *don't know* how many patients are *en route* along each path.

You are a doctor, and you have a patient who needs to be transferred to another facility.

You know the shortest paths between hospitals, but you *don't know* how many patients are *en route* along each path.

How can we release an *private, accurate* estimate of the number of patients *en route* on the shortest paths to have a successful transfer?



$G = (V, E, \omega)$ with private edge attribute $f : E \to \mathbb{R}^+$.

Background

Two isomorphic graphs $G_1, G_2 = (V, E, \omega)$ with edge attribute functions $f_1, f_2 : E \to \mathbb{R}^+$ are said to be neighboring if

$$\sum_{e \in E} |f_1(e) - f_2(e)| \leq 1$$

The l_1 sensitivity of $\mathcal{A} : \mathcal{X} \to \mathbb{R}^D$ is defined as

$$\Delta_1(\mathcal{A}) \coloneqq \max_{X,X'} \|\mathcal{A}(X) - \mathcal{A}(X')\|_1$$

where X, X' are neighboring datasets.

An algorithm $\mathcal{M} : \mathcal{X} \to \mathbb{R}^D$ is said to be (ε, δ)-differentially private if, for all outcomes $S \subseteq \mathbb{R}^D$ and neighboring datasets X, X',

$$\mathbb{P}[\mathcal{M}(X) \in S] \leq e^{\varepsilon} \cdot \mathbb{P}[\mathcal{M}(X') \in S] + \delta$$

An algorithm $\mathcal{M} : \mathcal{X} \to \mathbb{R}^D$ is said to be (ε, δ)-differentially private if, for all outcomes $S \subseteq \mathbb{R}^D$ and neighboring datasets X, X',

$$\mathbb{P}[\mathcal{M}(X) \in S] \leq e^{\varepsilon} \cdot \mathbb{P}[\mathcal{M}(X') \in S] + \delta$$

We call the case where $\delta = 0$ *pure* differential privacy and the case where $\delta > 0$ *approximate* differential privacy.

(Basic composition) Let $\varepsilon, \delta \in [0, 1]$ and $k \in \mathbb{N}$. If we run k mechanisms where each mechanism is $(\varepsilon/k, \delta/k)$-DP, then the entire algorithm is (ε, δ)-DP.

Given any function $f : \mathcal{X} \to \mathbb{R}^k$, the **Laplace mechanism** on input $X \in \mathcal{X}$ independently samples $Y_1, ..., Y_k$ according to Lap$(\Delta_1(f)/\varepsilon)$ and outputs,

$$\mathcal{M}_{f,\varepsilon}(X) = f(X) + (Y_1, ..., Y_k)$$

The Laplace mechanism is ε-differentially private.

Concrete Example

Let $\mathcal{X} \subset \mathbb{N}$. We say that $\mathcal{X}' \sim \mathcal{X}$ (neighboring) if $|\text{avg}(\mathcal{X}) - \text{avg}(\mathcal{X}')| \leq 1$ and $|\mathcal{X}| = |\mathcal{X}'| = n$.

Let $\mathcal{X} \subset \mathbb{N}$. We say that $\mathcal{X}' \sim \mathcal{X}$ (neighboring) if $|\text{avg}(\mathcal{X}) - \text{avg}(\mathcal{X}')| \leq 1$ and $|\mathcal{X}| = |\mathcal{X}'| = n$.

Let $\mathcal{A} : 2^{\mathbb{N}} \to \mathbb{R}$ be an algorithm given by $\mathcal{A}(\mathcal{X}) = \text{avg}(\mathcal{X})$ for any $\mathcal{X} \in 2^{\mathbb{N}}$.

Let $\mathcal{X} \subset \mathbb{N}$. We say that $\mathcal{X}' \sim \mathcal{X}$ (neighboring) if $|\text{avg}(\mathcal{X}) - \text{avg}(\mathcal{X}')| \leq 1$ and $|\mathcal{X}| = |\mathcal{X}'| = n$.

Let $\mathcal{A} : 2^{\mathbb{N}} \to \mathbb{R}$ be an algorithm given by $\mathcal{A}(\mathcal{X}) = \text{avg}(\mathcal{X})$ for any $\mathcal{X} \in 2^{\mathbb{N}}$.

$\mathcal{A}$ has sensitivity 1.

Let $Y_1, Y_2, ..., Y_n \sim \mathsf{Lap}(1/\varepsilon)$. Suppose that $\mathcal{X} = \{x_1, x_2, ..., x_n\}$.

Let $Y_1, Y_2, ..., Y_n \sim \mathsf{Lap}(1/\varepsilon)$. Suppose that $\mathcal{X} = \{x_1, x_2, ..., x_n\}$.

We define $\widetilde{\mathcal{X}} = \{x_1 + Y_1, x_2 + Y_2, ..., x_n + Y_n\}$.

Let $Y_1, Y_2, ..., Y_n \sim \mathsf{Lap}(1/\varepsilon)$. Suppose that $\mathcal{X} = \{x_1, x_2, ..., x_n\}$.

We define $\widetilde{\mathcal{X}} = \{x_1 + Y_1, x_2 + Y_2, ..., x_n + Y_n\}$.

$$\mathcal{A}(\widetilde{\mathcal{X}}) = \frac{x_1 + x_2 + ... + x_n}{n} + \frac{Y_1 + ... + Y_n}{n}$$

By a concentration inequality for i.i.d. Laplace random variables, with probability at least $1 - \gamma$, we have

$$|Y_1 + ... + Y_n| < O(\sqrt{n} \log(1/\gamma)/\varepsilon)$$

By a concentration inequality for i.i.d. Laplace random variables, with probability at least $1 - \gamma$, we have

$$|Y_1 + ... + Y_n| < O(\sqrt{n} \log(1/\gamma)/\varepsilon)$$

Hence, our ε-DP algorithm $\mathcal{A}$ is

$$O(\log(1/\gamma)/(\varepsilon \cdot \sqrt{n}))\text{-accurate}$$

with probability $1 - \gamma$.

Main Algorithm

Lemma 5. Let $T = (V, E, \omega)$ be a rooted tree with root z and private edge attribute $\phi : E \to \mathbb{R}^+$.

Lemma 5. Let $T = (V, E, \omega)$ be a rooted tree with root z and private edge attribute $\phi : E \to \mathbb{R}^+$.

We can release an ε-DP estimate of the counting queries from z to every other vertex in T with $O(\log^{1.5}(n) \cdot \log(n/\gamma)/\varepsilon)$ error w.p. $1 - \gamma$.

Let $T = (V, E, \omega)$ be a rooted tree with root z.

(1) Locate the unique centroid z^* that is closest to z

Let $T = (V, E, \omega)$ be a rooted tree with root z.

(1) Locate the unique centroid z^* that is closest to z

(2) Let z_i be the children of z^*, and $\mathcal{T}_i = (V_i, E_i)$ their corresponding subtrees, $i \in \{1, 2, ..., t\}$

Let $T = (V, E, \omega)$ be a rooted tree with root z.

(1) Locate the unique centroid z^* that is closest to z

(2) Let z_i be the children of z^*, and $\mathcal{T}_i = (V_i, E_i)$ their corresponding subtrees, $i \in \{1, 2, ..., t\}$

(3) Release the counting queries between z and z^*, as well as between z^* and its children z_i by adding Laplace noise from Lap$(\log(n)/\varepsilon)$.

Let $T = (V, E, \omega)$ be a rooted tree with root z.

(1) Locate the unique centroid z^* that is closest to z

(2) Let z_i be the children of z^*, and $\mathcal{T}_i = (V_i, E_i)$ their corresponding subtrees, $i \in \{1, 2, ..., t\}$

(3) Release the counting queries between z and z^*, as well as between z^* and its children z_i by adding Laplace noise from Lap$(\log(n)/\varepsilon)$.

(4) Recursively repeat on each subtree $\mathcal{T}_i$.

Key points

(1) Since each subtree $\mathcal{T}_i$ contains at most $n/2$ vertices, the recusion depth is bounded by $\log_2(n)$. By **basic composition** of DP algorithms, the composition of $\log(n)$, $(\varepsilon/\log(n))$-DP mechanisms is ε-DP.

Key points

(1) Since each subtree $\mathcal{T}_i$ contains at most $n/2$ vertices, the recusion depth is bounded by $\log_2(n)$. By **basic composition** of DP algorithms, the composition of $\log(n)$, $(\varepsilon/\log(n))$-DP mechanisms is ε-DP.

(2) Let $u \in V$, the number of estimates used to calculate $\omega(z, u)$ is bounded above by $2\log(n)$.

Lemma 2. Let $X_1, ..., X_t$ be independent random variables distributed according to Lap(b), and let $X = X_1 + ... + X_t$. Then for all $\gamma \in (0, 1)$, with probability at least $1-\gamma$ we have,

$$|X| < O(b\sqrt{t}\log(1/\gamma))$$

Lemma 2. Let $X_1, ..., X_t$ be independent random variables distributed according to Lap(b), and let $X = X_1 + ... + X_t$. Then for all $\gamma \in (0, 1)$, with probability at least $1-\gamma$ we have,

$$|X| < O(b\sqrt{t}\log(1/\gamma))$$

By a union bound, with probability at least $1 - \gamma$, the error for the estimate from z to all vertices $u \in V$ is bounded above by $O(\log^{1.5}(n) \cdot \log(n/\gamma)/\varepsilon)$.

Lemma 6. We can release an ε-DP approximation of the counting queries between all pairs of vertices in T with $O(\log^{1.5}(n) \cdot \log(n/\gamma)/\varepsilon)$.

(1) Randomly sample a hitting set $S \subset V$ of vertices with size $n^{1/3}$.

(1) Randomly sample a hitting set $S \subset V$ of vertices with size $n^{1/3}$.

(2) Release $\varepsilon/(2 \cdot n^{1/3})$ counting queries along each shortest path tree rooted at a vertex $v \in S$ (**Lemma 6**).

(1) Randomly sample a hitting set $S \subset V$ of vertices with size $n^{1/3}$.

(2) Release $\varepsilon/(2 \cdot n^{1/3})$ counting queries along each shortest path tree rooted at a vertex $v \in S$ (**Lemma 6**).

(3) Release the t-hop shortest paths from each vertex where $t := \lceil 10 \cdot n^{2/3} \log(n) \rceil$ by adding noise from Lap$(2/\varepsilon)$ to each edge.

(1) Randomly sample a hitting set $S \subset V$ of vertices with size $n^{1/3}$.

(2) Release $\varepsilon/(2 \cdot n^{1/3})$ counting queries along each shortest path tree rooted at a vertex $v \in S$ (**Lemma 6**).

(3) Release the t-hop shortest paths from each vertex where $t := \lceil 10 \cdot n^{2/3} \log(n) \rceil$ by adding noise from Lap$(2/\varepsilon)$ to each edge.

(4) Let $\widetilde{\omega}(u, v)$ be the minimum of the estimates from (2) and (3).

For shortest paths that have fewer than t edges, we can directly release the approximation with $O(n^{1/3} \cdot \log^{1.5}(n) \cdot \log(1/\gamma)/\varepsilon)$ error.

For shortest paths that have fewer than t edges, we can directly release the approximation with $O(n^{1/3} \cdot \log^{1.5}(n) \cdot \log(1/\gamma)/\varepsilon)$ error.

For shortest paths that have more than t edges, we can release their approximation via a shortest path tree that contains them with
$O(n^{1/3} \log^{2.5}(n) \cdot \log(1/\gamma)/\varepsilon)$

Question: How do we know that *long* shortest paths are contained in a shortest path tree?

Question: How do we know that *long* shortest paths are contained in a shortest path tree?

Answer: Let $u = p_0, p_1, ..., p_\ell = v$ be the shortest path between u and v and assume that $\ell \geq t$.

Question: How do we know that *long* shortest paths are contained in a shortest path tree?

Answer: Let $u = p_0, p_1, ..., p_\ell = v$ be the shortest path between u and v and assume that $\ell \geq t$.

Each vertex has probability $n^{-2/3}$ of being included in S. Hence, the probability that none of p_i are included is $(1 - n^{-2/3})^t \leq 1/n^4$.

Question: How do we know that *long* shortest paths are contained in a shortest path tree?

Answer: Let $u = p_0, p_1, ..., p_\ell = v$ be the shortest path between u and v and assume that $\ell \geq t$.

Each vertex has probability $n^{-2/3}$ of being included in S. Hence, the probability that none of p_i are included is $(1 - n^{-2/3})^t \leq 1/n^4$.

We condition on this event for all n^2 shortest paths with probability at least $1 - 1/n^2$, by a union bound.

Question: Why can't we use the same algorithm to release shortest distances, where edge weights are private?

Question: Why can't we use the same algorithm to release shortest distances, where edge weights are private?

Answer: Because calculation the shortest path trees relies on the private edge weights.

Collective Tree Spanners

Given a graph $G = (V, E, \omega)$, a graph t-spanner $H = (V, E_H, \omega_H)$ is subgraph such that for any $u, v \in V$,

$$d_H(u, v) \leq t \cdot d_G(u, v)$$

Given a graph $G = (V, E, \omega)$, a graph t-spanner $H = (V, E_H, \omega_H)$ is subgraph such that for any $u, v \in V$,

$$d_H(u, v) \leq t \cdot d_G(u, v)$$

That is, H approximately preserves pairwise distances.

A *collective tree t-spanner* is a collection of tree $\{\mathcal{T}_i\}$ such that $\mathcal{T}_i$ is a spanning subtree of G and, if $\mathbf{T} = \cup_i \mathcal{T}_i$, then,

$$d_{\mathbf{T}}(u, v) \leq t \cdot d_G(u, v)$$

By first constructing a collective tree spanner of G, we can run **Lemma 6** on each tree to release a private estimate of counting queries over t-approximate shortest paths in G.

By first constructing a collective tree spanner of G, we can run **Lemma 6** on each tree to release a private estimate of counting queries over t-approximate shortest paths in G.

This allows us to achieve a tradeoff between accuracy and distance.

Lemma 1. Let $\mathbf{T}$ be a t-collective tree spanner of G such that $|\mathbf{T}| = \eta_t$. There is an ε-DP algorithm for releasing the counting query between $u, v \in V$ on a t-approximate shortest path in G that is $O(\eta_t \cdot \log^{2.5}(n) \cdot \log(1/\gamma)/\varepsilon)$-accurate w.p. $1 - \gamma$.

Lemma 1. Let $\mathbf{T}$ be a t-collective tree spanner of G such that $|\mathbf{T}| = \eta_t$. There is an ε-DP algorithm for releasing the counting query between $u, v \in V$ on a t-approximate shortest path in G that is $O(\eta_t \cdot \log^{2.5}(n) \cdot \log(1/\gamma)/\varepsilon)$-accurate w.p. $1 - \gamma$.

$\widetilde{O}(\sqrt{\eta_t}/\varepsilon)$ in the (ε, δ)-DP case.

Lower bounds for graph spanners: Consider an undirected, unweighted graph $G = (V, E)$ whose shortest cycle has more than $t + 1$ edges (*girth* $> t + 1$).

Lower bounds for graph spanners: Consider an undirected, unweighted graph $G = (V, E)$ whose shortest cycle has more than $t + 1$ edges (*girth* $> t + 1$).

Then G has no proper subgraph that is a t-spanner.

Moore bounds. Let $\gamma(n,k)$ denote the maximum number of edges in an n-vertex graph with girth $> k$, then,

$$\gamma(n,k) = O\left(n^{1+\frac{1}{\lfloor k/2 \rfloor}}\right)$$

Erdös girth conjecture (open): The Moore bounds are tight.

In the context of multiplicative spanners, any $(2k-1)$-spanner must have at least $\Omega(n^{1+(1/k)})$ edges.

In the context of multiplicative spanners, any $(2k-1)$-spanner must have at least $\Omega(n^{1+(1/k)})$ edges.

We give a framework to construct a collective tree $(2k-1)$-spanner with $O(kn^{1/k})$ trees, which is optimal up to a factor of k.

References

Sealfon, Adam. (2016). Shortest Paths and Distances with Differential Privacy. 29-41. 10.1145/2902251.2902291.

Chen, Justin et al. (2023). Differentially Private All-Pairs Shortest Path Distances: Improved Algorithms and Lower Bounds, Proceedings of the 2023 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA), 5040-5067, 10.1137/1.9781611977554.ch184.

Chengyuan, Deng et al. (2022). Differentially Private Range Query on Shortest Paths. arXiv preprint. https://arxiv.org/abs/2212.07997