Section 5: Different Types of Sensitivity and Graph Privacy

CS 208 Applied Privacy for Data Science, Spring 2022

March 8, 2022

1 Agenda

- Discuss any questions about problem sets.
- Go through differences between and examples of global sensitivity, local sensitivity, and restricted sensitivity.
- Graph Privacy.
- Lipschitz transformations.

2 Beyond Global Sensitivity

Let $\mathcal{X}$ be a data universe (e.g., $\{0,1\}$), and $\mathcal{X}^n$ (e.g., $\{0,1\}^n$) be a space of datasets, where n, the length of the dataset, is public information.

For $x, x' \in \mathcal{X}^n$, we write $x \sim x'$ to denote the neighboring relation. For tabular data, $x \sim x'$ means that x, x' differ on one row.

Then, we define global sensitivity of a query q as follows:

Definition 2.1 (Global Sensitivity)**.** For a query $q : \mathcal{X}^n \to \mathbb{R}$, the global sensitivity is:

$$GS_q = \max_{x \sim x'} |q(x) - q(x')|$$

Global sensitivity measures the maximum impact one individual's data can have on the result of a specific query or function. Note that global sensitivity does not depend on the specific database; only on the query q, the data universe $\mathcal{X}$, and (sometimes) the size of the database n. It gives an upper bound on how much we must perturb the output to preserve individual privacy.

The local sensitivity, on the other hand, *does depend on the specific database!* It measures the maximum impact one individual's data can have on the result of a query applied to that database.

Definition 2.2 (Local Sensitivity)**.** For a query $q : \mathcal{X}^n \to \mathbb{R}$, the local sensitivity is:

$$\mathrm{LS}_q(x) = \max_{x' : x' \sim x} |q(x) - q(x')|.$$

Notice that unlike the global sensitivity, which takes the max over all pairs of neighboring databases (x, x'), the local sensitivity takes the max over all databases x' adjacent to a *fixed* database x.

Exercise 2.3. Why does adding noise scaled to the local sensitivity violate privacy?

Solution. The local sensitivity may itself be sensitive. This fact, combined with some auxiliary information about the database, can lead to privacy problems: the adversary may know that the database is one of x, which has very low local sensitivity for the query in question, or a neighboring x', for which the query has very high local sensitivity. In this case the adversary may be able to guess rather accurately which of x and x' is the true database. For example, if $q(x) = q(x') = 0$ and the response is far from 0, then the adversary would guess x' and be correct with high probability.

Exercise 2.4. Show that adding noise proportional to the local sensitivity (of the median, for example) is not differentially-private.

Solution. Let our dataset universe $\mathcal{X} = [0, 10^6]$ and neighboring relation be the change model ($n = 5$ is fixed). We provide an example of two databases $x, x' \in \mathcal{X}^n$ with the same median but with drastically different local sensitivities.

$$x = (0, 0, 0, 0, 10^6)$$
$$x' = (0, 0, 0, 10^6, 10^6).$$

If a mechanism outputs an answer is far from 0, the adversary would guess x' and be correct with high probability.

Definition 2.5 (Restricted Sensitivity)**.** Let $\mathcal{G}$ be the set of all datasets, with neighbor relation $\sim$, and let $d(x, x')$ be the distance between datasets $x, x' \in \mathcal{G}$. Let $\mathcal{H} \subseteq \mathcal{G}$ be a "hypothesis" set of datasets in which we hope our dataset lies. Then, for a query $q : \mathcal{G} \to \mathbb{R}$, the *restricted sensitivity* of q with respect to $\mathcal{H}$ is

$$\mathrm{RS}_q^{\sim,\mathcal{H}} = \max_{x,x' \in \mathcal{H}} \frac{|q(x) - q(x')|}{d(x, x')},$$

where x, x' are not necessarily neighboring datasets.

For each of the following sets $\mathcal{G}$ of datasets and neighbor relations $\sim$, hypotheses $\mathcal{H} \subseteq \mathcal{G}$, for functions $f : \mathcal{G} \to \mathbb{R}$, calculate (i) the global sensitivity of f (denoted GS_f or ∂f), (ii) the minimum local sensitivity of f for $x \in \mathcal{G}$, i.e., $\min_{x \in \mathcal{G}} \mathrm{LS}_f(x)$, (iii), the maximum local sensitivity of f for $x \in \mathcal{H}$, i.e., $\max_{x \in \mathcal{H}} \mathrm{LS}_f(x)$ and (iv) the restricted sensitivity of f (denoted $\partial_{\mathcal{H}} f$ or $\mathrm{RS}_f^{\mathcal{H}}$).

Exercise 2.6. $\mathcal{G} = \mathbb{R}^n, \mathcal{H} = [a, b]^n$ where $x \sim x'$ if x and x' differ on one row, for real numbers $a \leq b$, and $f(x) = (1/n) \sum_{i=1}^n x_i$.

Solution. The global sensitivity is ∞, the minimum local sensitivity on $x \in \mathbb{R}$ is ∞, the maximum local sensitivity on $x \in [a, b]^n$ is ∞, and the restricted sensitivity is $(b - a)/n$.

Exercise 2.7. $\mathcal{G} = \mathbb{R}^n, \mathcal{H} = [a, b]^n$ where $x \sim x'$ if x and x' differ on one row, for real numbers $a \leq b$, and $f(x) = \mathrm{median}(x_1, \ldots, x_n)$.

Solution. The global sensitivity is $\mathbb{R}$, the minimum local sensitivity on $x \in \mathbb{R}$ is 0, the maximum local sensivity on $x \in [a, b]$ is ∞, and the restricted sensitivity is $(b - a)$.

3 Graph Privacy

For a graph G, possibly with labels on nodes and edges, we have the following notions of privacy.

- **Edge-level privacy**: Graphs G, G' are neighbors if they differ on at most one edge.
- **Node-level privacy**: Graphs G, G' are neighbors if they differ on at most one node and incident edges.

Note that edge-level privacy only protects particular relationships within a social network graph, while node-level privacy protects any individual's information.

Consider the graph $G = (V, E)$ where V is the vertex set and E is the edge set. That is, for all $i \neq j, i, j \in V$, $\{i, j\} \in E$ if nodes i, j are connected.

Exercise 3.1. Given the (unweighted, undirected) graph $G = (V, E)$, how can you reduce its representation to a tabular form?

Can you relate the adjacency matrix format to the tabular form we have been using to think about adjacency of databases?

Solution. Use an adjacency matrix representation. That is, you can form a matrix $M \in \mathbb{Z}^{|V|\times|V|}$ such that for any $i, j \in V$, $M(i, j) = 1$, $M(j, i) = 1$ if $\{i, j\} \in E$. Otherwise, $M(i, j) = 0$.

You can also think of M as a vector of length $|V|^2$.

Exercise 3.2. For a given dataset x and query q, when will the restricted sensitivity $\mathrm{RS}_q^{\sim,\mathcal{H}}$ be smaller than local sensitivity $\mathrm{LS}_q(x)$?

Solution. $\mathrm{RS}_q^{\sim,\mathcal{H}}$ may be smaller than $\mathrm{LS}_q(x)$ for some x if for some $x \in \mathcal{H}$, x has a neighbor $x' \notin \mathcal{H}$.

query	range	$\mathrm{LS}_q^{\mathrm{edge}}(G)$	$\mathrm{GS}_q^{\mathrm{edge}}$	$\mathrm{LS}_q^{\mathrm{node}}(G)$	$\mathrm{GS}_q^{\mathrm{node}}$
# edges	$[0, dn/2]$	1	1	$\Theta(n)$	$n-1$
max-deg	$[0, d]$	1	1	$\Theta(n)$	$n-1$
# triangles	$[0, \Theta(nd^2)]$	$\leq d-1$	$n-2$	$\leq$ #edges	$\Theta(n^2)$

$d = \mathrm{maxdeg}(G)$, typically $d \ll n$

The next few exercises are to fill in the table above we went over in lecture. For these exercises, let $G = (V, E)$ be an undirected graph where V, E are the vertex and edge sets, respectively.

Exercise 3.3. Let $q : G \to \mathbb{Z}$ be the query that computes the number of edges in the graph. Then show that $\mathrm{LS}_{\mathrm{q}}^{\mathrm{edge}}(\mathrm{G}) = 1$ and $\mathrm{LS}_{\mathrm{q}}^{\mathrm{node}}(\mathrm{G}) = n - 1$.

Solution. Using the definition of edge privacy, the number of edges between any graph x and any neighboring graph x' can differ by at most 1. Under the definition of node privacy, any node has at most $n - 1$ edges, so the number of edges in neighboring graphs can differ by at most $n - 1$.

Exercise 3.4. Let $q : G \to \mathbb{Z}$ be the query that computes the maximum degree in the graph. Then show that $\mathrm{GS}_q^{edge} = \mathrm{LS}_{\mathrm{q}}^{\mathrm{edge}}(\mathrm{G}) = 1$ and $\mathrm{GS}_q^{node} = \mathrm{LS}_{\mathrm{q}}^{\mathrm{node}}(\mathrm{G}) = n - 1$.

Solution. The argument is similar to the one above.

Exercise 3.5. Let $q : G \to \mathbb{Z}$ be the query that computes the # triangles in the graph. Then show that $\mathrm{LS}_{\mathrm{q}}^{\mathrm{edge}}(\mathrm{G}) \leq d-1$, $\mathrm{GS}_q^{edge} = n-2$, $\mathrm{LS}_q^{node}(G) \leq$ # edges, and $\mathrm{GS}_{\mathrm{q}}^{\mathrm{node}} = \Theta(n^2)$.

Solution. For a given node of degree d, any edge incident to that node can be a part of at most $d-1$ triangles, so the local sensitivity under edge privacy is at most $d-1$. In the worst case, $d = n-1$, so the global sensitivity is $n-2$. Under node privacy, adding a node that is connected to all other nodes in the graph creates # edges new triangles. In the worst case, the initial graph is fully created, so the # edges is $\Theta(n^2)$.

Exercise 3.6. Let $\mathcal{H}$ be the set of graphs with max-degree $d \in [1, n-1]$. What is the restricted sensitivity (using node privacy) for each of the three queries?

Solution. The restricted sensitivity of # edges is d or $d-1$ depending on how the graph is defined, max-deg is d, and # triangles is d^2.

4 Lipschitz Dataset Transformations

Let $d(x, x')$ denote the distance between datasets x, x'. In the known n model, this is equivalent to the number of rows on which the datasets differ, and in the unknown model, this is equivalent to the number of removals or additions that need to be made to go from x to x'. Essentially, d is the same as whichever notion of adjacency we are using for privacy.

Definition 4.1. A mapping T from datasets to datasets is c-Lipschitz, also called c-stable or c-sensitive, iff

$$\forall x, x', \quad d(T(x), T(x')) \leq c \cdot d(x, x')$$

Lemma 4.2. *If M is ϵ-DP and T is c-Lipschitz, then $M \circ T$ is $c\epsilon$-DP.*

Proof. This follows by the definition of ϵ-DP and Lipschitzness property on adjacent databases. Let x, x' be any neighboring datasets (so $d(x, x') = 1$). By the Lipschitz property, there exists a series of adjacent datasets $T(x) = z_0 \sim z_1 \sim \cdots \sim z_c = T(x')$ of length c. Applying the ϵ-DP property of M consecutively on this series, we have that for all sets Y,

$$\Pr[M(z_0) \in Y] \leq e^{\epsilon} \cdot \Pr[M(z_1) \in Y] \leq \cdots \leq e^{c \cdot \epsilon} \cdot \Pr[M(z_c) \in Y]$$

As $z_0 = T(x)$ and $z_c = T(x')$, this gives the desired result. □

Lemma 4.3. *If T_1 is c_1-Lipschitz and T_2 is c_2-Lipschitz, then $T_2 \circ T_1$ is $c_1 \cdot c_2$-Lipschitz.*

If we think of DP as itself a Lipschitz property ("close databases yield close distributions"), then the first lemma is a special case of the second.

In the following exercises, we will calculate the Lipschitz constants for the given transformations. Note that it suffices to think of adjacent datasets, since we can get to $d(x, x') = k$ using k hops.

Exercise 4.4. Per-row transformations: $T(x_1, \ldots, x_n) = (f(x_1), \ldots, f(x_n))$
Solution. $c = 1$. If we change, add, or remove one row of the original dataset, we change, add or remove at most one row on the transformed dataset as well.

Exercise 4.5. Removing: $T(x_1, \ldots, x_n) = (x_{11}, \ldots, x_{n-11})$
Solution. $c = 1$. If we change one row in either the tails or the center of the original dataset, we change at most one row of the transformed dataset. The constant is the same in the add/remove model as well, because adding/removing in the tails causes an add/removal rather than a change in the center (note that the size of the center is variable).

Exercise 4.6. Group-by: For database of students, return counts of student by state.
Solution. $c = 2$ in the change model, $c = 1$ in the add/remove model. Changing one row causes one count to decrease and another to increase. Adding/removing one row causes one count to change.