CS839: Differential Privacy and Learning February 12, 2026

Lecture 8: Beating Global Sensitivity

Instructor: Gavin Brown *Scribe:* Sungjun (June) Cho

Disclaimer: This document is intended as an informal supplement to in-class note-taking. It has not been given the level of scrutiny expected in polished lecture notes, let alone that reserved for peer-reviewed publications.

1 Local Notions of Sensitivity

Recall the Laplace Mechanism adds noise proportional to the *global sensitivity* of a query function f. While this yields a valid DP guarantee, the resulting noise can be far larger than necessary: it is calibrated to the worst-case pair of neighboring datasets, even if such pairs are atypical in practice. This lecture explores how to do better by leveraging the structure of the data we actually observe.

We begin by stating the definition of global sensitivity, then examining two motivating examples.

Definition 1.1. The *global sensitivity* of f is

$$\Delta \triangleq \max_{x \sim x'} \left| f(x) - f(x') \right|.$$

Remember that privacy guarantees must hold in the worst case, but the *utility* of a mechanism can depend on what the data actually looks like.

Example 1.2 (Median)**.** Let $x_1, \ldots, x_n \in [0, R]$. The global sensitivity of the median function is

$$\max_{x \sim x'} \left| \text{med}(x) - \text{med}(x') \right| = R.$$

To see why, consider a dataset of seven points with $x_1 = x_2 = x_3 = x_4 = 0$ and $x_5 = x_6 = x_7 = R$. The median is currently 0, but if we change a single point from 0 to R, the median jumps to R, so the global sensitivity equals R. Using the Laplace Mechanism with noise scaled to R would give extremely poor estimates of the median, even for "typical" data where no such balanced split exists.

Example 1.3 (Mean)**.** Let $x_1, \ldots, x_n \in \mathbb{R}$ (unbounded). The global sensitivity of the mean function is

$$\max_{x \sim x'} \left| \text{mean}(x) - \text{mean}(x') \right| = +\infty.$$

Since data points are unbounded, an adversary can always construct a neighboring dataset that shifts the mean by an arbitrary amount. The Laplace Mechanism cannot even be directly applied in this setting.

1.1 Why Global Sensitivity is Often Pessimistic

In many cases, global sensitivity is large because of pathological or atypical data instances such as outliers, adversarial constructions, or unbounded domains. For many "natural" datasets, the function's output is far more stable than the worst case suggests.

For instance, suppose $x_1, \ldots, x_n, x_{n+1} \overset{\text{i.i.d.}}{\sim} \mathcal{N}(\mu, \sigma^2)$. Consider how much the mean changes when we swap out one element:

$$\left(\frac{1}{n}\sum_{i=1}^{n} x_i\right) - \left(\frac{1}{n}\sum_{i=2}^{n+1} x_i\right) = \frac{1}{n}(x_1 - x_{n+1}) \sim \mathcal{N}\left(0, \frac{2\sigma^2}{n^2}\right).$$

With high probability the mean changes by only $O(\sigma/n)$, which is small with large n, yet the global sensitivity is infinite. To beat global sensitivity, we can aim to exploit this gap between typical-case and worst-case behavior.

1.2 Local Sensitivity

The first step toward exploiting this gap is to define a notion of sensitivity that depends on a particular dataset, rather than the worst-case one.

Definition 1.4. The *local sensitivity* of f at x is

$$\mathrm{LS}_f(x) = \max_{\substack{x' \\ \text{s.t. } x' \sim x}} \left|f(x) - f(x')\right|.$$

Intuitively, local sensitivity measures the maximum change in f when we modify a single record in the specific dataset x. Note that global sensitivity is simply the maximum of local sensitivity taken over all possible datasets: $\Delta = \max_x \mathrm{LS}_f(x)$.

Example 1.5 (Local sensitivity of the median)**.** Suppose our dataset is $x = (0, 0, 0, 0)$. Then $\mathrm{LS}_{\mathrm{med}}(x) = 0$: no matter which point we change or what we change it to, the median remains 0 since three of the four points are still at 0. Locally, the function is completely stable.

1.3 Naïve Use of Local Sensitivity Fails

A natural idea is to simply replace global sensitivity with local sensitivity in the Laplace Mechanism:

$$f(x) + \mathrm{Lap}\left(\frac{\mathrm{LS}_f(x)}{\varepsilon}\right).$$

Unfortunately, this is **not** differentially private for general f. The fundamental problem is that $\mathrm{LS}_f(x)$ itself depends on the data, so the amount of noise we add leaks information.

Consider the following attack scenario on estimating the median. Let $x = \{0, 0, 0, 0, R, R\}$. Assuming that ties are broken by taking the smaller value, the median is 0 and the local sensitivity is $\mathrm{LS}_{\mathrm{med}}(x) = 0$. Now consider the neighboring dataset $x' = \{0, 0, 0, R, R, R\}$ with one point changed from 0 to R. The median of x' is 0 (by the tie-breaking rule), but now $\mathrm{LS}_{\mathrm{med}}(x') = R$, since changing a single 0 to R would shift the median from 0 to R.

Under this naïve scheme, the mechanism adds zero noise on input x (returning exactly 0) but adds $\mathrm{Lap}(R/\varepsilon)$ noise on input x'. An adversary seeing the output can easily distinguish these two neighboring datasets, violating differential privacy.

Remark 1.6. The local sensitivity of the mean with unbounded data is $\mathrm{LS}_{\mathrm{mean}}(x) = \infty$ for *all* x, so local sensitivity does not help in this case either.

1.4 Down-Sensitivity

The failure of the naïve approach motivates us to look for another notion of data-dependent sensitivity. One variant restricts the neighboring dataset to be a *subset* of the original:

Definition 1.7. The *down-sensitivity* of f at x is

$$\mathrm{DS}_f(x) = \max_{\substack{x' \\ \text{s.t. } x' \subseteq x \\ |x'|=|x|-1}} \left| f(x) - f(x') \right|.$$

Down-sensitivity only considers removing a single record from the dataset, whereas local sensitivity considers both removing *and* adding a record. Therefore, we have $\mathrm{DS}_f(x) \leq \mathrm{LS}_f(x)$ (*i.e.*, down-sensitivity is a lower bound on local sensitivity).

Claim 1.8 (Informal)**.** *For Gaussian data,* $\mathrm{DS}_{mean}(x) \overset{w.h.p}{\approx} \frac{\sigma}{n}$.

This is encouraging: while the global sensitivity (and even the local sensitivity) of the mean is infinite for unbounded data, the down-sensitivity is small with high probability when the data is well-structured.

There are a number of exciting recent developments in algorithms that make use of down-sensitivity. We'll return to these later in the course.

1.5 Roadmap

Before discussing how to actually use local or down-sensitivity in a private mechanism, let us clarify a key definition and briefly note what approaches are available.

Definition 1.9. A is *differentially private* if $\forall x, x'$ such that x and x' are adjacent, $A(x) \approx A(x')$.

Keep in mind that later in the course, we may change what it means (1) for datasets to be *adjacent* and (2) for outputs to be *indistinguishable* in $A(x) \approx A(x')$.

The upshot of our discussion so far is: when we can add noise scaled to LS or DS instead of Δ, we often get much better accuracy while still satisfying privacy constraints. The challenge is *how* to do this without leaking information through the noise scale itself. We now have four frameworks for beating global sensitivity:

1. Privately bound $\mathrm{LS}_f(x)$. See Vadhan [2017], Section 7.3.4 for an illustration of this.
2. Inverse sensitivity mechanism
3. Smooth sensitivity
4. Propose-Test-Release

We focus on frameworks 3 and 4 in this lecture.

1.6 Smooth Sensitivity

The core insight of smooth sensitivity is to find an upper bound on local sensitivity that is close to $\mathrm{LS}_f(x)$ when f is stable near x, but changes slowly between neighboring datasets, so that the calibrated noise does not leak information.

Definition 1.10. The *smooth sensitivity* of f at x (with parameter ε) is

$$\mathrm{SS}_f^\varepsilon(x) \triangleq \max_{x'} e^{-\varepsilon\, d(x,x')} \cdot \mathrm{LS}_f(x'),$$

where $d(x, x') = \sum_{i=1}^n \mathbb{1}\{x_i \neq x'_i\}$ denotes the Hamming distance.

The exponential decay $e^{-\varepsilon\, d(x,x')}$ is what makes this work: datasets far from x are discounted heavily, so a large local sensitivity at some distant dataset x' contributes very little to $\mathrm{SS}_f^\varepsilon(x)$. At the same time, the smooth sensitivity always upper-bounds the local sensitivity (take $x' = x$ in the max to see that $\mathrm{SS}_f^\varepsilon(x) \geq \mathrm{LS}_f(x)$).

Crucially, smooth sensitivity is itself a "smooth" function of the dataset: for neighboring $x \sim x''$,

$$\mathrm{SS}_f^\varepsilon(x) \leq e^\varepsilon \cdot \mathrm{SS}_f^\varepsilon(x'').$$

This bounded multiplicative change ensures that calibrating noise to $\mathrm{SS}_f^\varepsilon(x)$ does not violate privacy. There exist noise distributions Z (*e.g.*, the standard Cauchy distribution) for which the mechanism

$$f(x) + Z \cdot \frac{\mathrm{SS}_f^\varepsilon(x)}{\varepsilon}$$

is differentially private.

What benefits does this have over global sensitivity? The following example makes the gain concrete.

Example 1.11 (Smooth sensitivity of the median)**.** Consider a dataset x consisted of ten copies of 0, with each data point bounded in $[0, R]$. We have $\mathrm{LS}_{\mathrm{med}}(x) = 0$. To compute the smooth sensitivity, we need to understand how the local sensitivity grows as we move away from x. Suppose we change points from 0 to R one at a time:

- For x' with $d(x, x') = 1$: $\mathrm{LS}_{\mathrm{med}}(x') = 0$
- For x' with $d(x, x') = 2$: $\mathrm{LS}_{\mathrm{med}}(x') = 0$
- For x' with $d(x, x') = 3$: $\mathrm{LS}_{\mathrm{med}}(x') = 0$
- For x' with $d(x, x') = 4$: $\mathrm{LS}_{\mathrm{med}}(x') = 0$
- For x' with $d(x, x') \geq 5$: $\mathrm{LS}_{\mathrm{med}}(x') \leq R$

The local sensitivity remains zero until we have changed at least half of the ten points because the median is controlled by the majority value. The smooth sensitivity is therefore bounded by

$$\mathrm{SS}_{\mathrm{med}}^\varepsilon(x) \leq e^{-4\varepsilon} \cdot R.$$

Compare this to the global sensitivity of R: the smooth sensitivity reduces the noise by a factor of $e^{-4\varepsilon}$, which is significant. When we have more than roughly $\frac{\log R}{\varepsilon}$ points on zero, the smooth sensitivity becomes $O(1)$, meaning we can add noise of constant scale for estimating the median.

2 Propose-Test-Release

Another framework for beating global sensitivity is *Propose-Test-Release* (PTR). The high-level idea is somewhat simple: rather than constructing a smooth upper bound on sensitivity, we guess that the sensitivity is small and then privately test whether the guess was correct.

2.1 Informal Description

In its simplest form, PTR proceeds as follows:

1. **Propose:** Before looking at the data, guess an upper bound B on $\mathrm{LS}_f(x)$.
2. **Test:** Privately check if the guess is good (*i.e.*, $\mathrm{LS}_f(x') \le B$ for all x' close to x).
3. **Release:** If the guess is good, release $f(x) + \mathrm{Lap}(B/\varepsilon)$. Otherwise, return $\perp$ (failure).

The key insight is that the test step can be made private using the distance to instability: if the dataset is far from any dataset where the guess fails, then we can privately certify the guess is good.

2.2 PTR as a General Wrapper

More broadly, PTR can be seen as a useful "wrapper" around any algorithm A that would be private, if we could restrict our attention to some subset of typical or well-behaved datasets.

1. **Propose:** Come up with an algorithm A that
   (a) on typical data has high accuracy and
   (b) on two adjacent & typical datasets, has indistinguishable outputs.
2. **Test:** Privately check if the guess is good (*i.e.*, the current dataset is in the safe region where A looks private).
3. **Release:** If the guess is good, release $A(x)$. Otherwise, return $\perp$.

One advantage over smooth sensitivity is its simplicity: PTR does not require computing a smooth upper bound, which can be difficult for complex queries. It also allows us to make full use of the power of approximate DP in statistical inference; we'll discuss this more when we get to lower bounds.

2.3 Formal Algorithm

We now state the PTR algorithm formally.

Algorithm 1 Propose-Test-Release

Input: Privacy parameters $\varepsilon, \delta \in (0,1)$, dataset $x \in \mathcal{X}^n$, algorithm $A : \mathcal{X}^n \to \mathcal{U}$
Returns: $\tilde{y} \in \mathcal{Y}$ or $\perp$

Define SAFE $= \{x' \in \mathcal{X}^n : \forall x'' \sim x',\ A(x') \approx_{\varepsilon,\delta} A(x'')\}$ ▷ $\approx_{\varepsilon,\delta}$ denotes indistinguishability
UNSAFE $= \mathcal{X}^n \setminus$ SAFE
$k \leftarrow \min\{d(x, x') : x' \in \text{UNSAFE}\}$ ▷ Distance to nearest unsafe dataset
if $\tilde{k} = k + \mathrm{Lap}(1/\varepsilon) \ge \dfrac{2\log(1/\delta)}{\varepsilon} + 1$ **then**
    Release $\tilde{y} = A(x)$
else
    Return $\perp$
end if

The quantity k is the *distance to instability*: how many records must change before the algorithm A ceases to be private. When k is large, the dataset is deep inside the safe region, and the noisy

check $\tilde{k}$ will pass with high probability. Note this algorithm appeared in the infinite-number-of-bins histogram problem from Homework 1.

References

Salil Vadhan. The complexity of differential privacy. In *Tutorials on the Foundations of Cryptography*, pages 347–450. Springer, 2017.