Section 9: Local Differential Privacy

CS 208 Applied Privacy for Data Science, Spring 2022

April 5, 2022

1 Agenda

- Local differential privacy.
- Randomized response.
- Group privacy in the local model.

2 Local Differential Privacy (LDP)

Recall that in the central model, the algorithm is run by a (trusted) central authority (e.g., the U.S. Census Bureau). This authority holds **all** the data that we will use to release a statistic or ML model. In the local model, we no longer have a centralized, trusted party. Instead, data is distributed amongst several parties (e.g., individuals). A protocol specifies the interaction between the parties to perform some task.

Let $\mathcal{A}$ be an algorithm that we wish to run to compute a task, statistic, or model. Use $\mathcal{T}_{\mathcal{A}}$ to denote the transcript of the protocol between the interacting parties to perform the computation. [1] We say that an algorithm $\mathcal{A}$ is (ϵ, δ)**-differentially private (DP)** in the local model if for all neighboring sets S, S' that can be input to $\mathcal{A}$ and for all outputs $O \subseteq \text{Range}(\mathcal{A})$, we have

$$\Pr[\mathcal{T}_{\mathcal{A}}(S) \in O] \leq e^{\epsilon} \Pr[\mathcal{T}_{\mathcal{A}}(S') \in O] + \delta, \tag{1}$$

for all $\epsilon \geq 0, \delta \in [0, 1]$.

Equation (1) encapsulates the interactive [Kasiviswanathan et al., 2011] and non-interactive models of local DP. See [Duchi, Rogers, 2019] for more discussion on interactivity in the local model. For the remaining parts of this section, we will focus on the non-interactive local model of DP where each user sends their (perturbed) data to the algorithm just once.

Note that in the local (non-interactive) DP setting, every randomizer only works on 1-row databases (each individual), so all databases are neighboring databases. In the non-interactive model, every individual randomizes their own data and sends it to an untrusted aggregator.

Now, we can move on to defining ϵ-local DP.

Definition 2.1 (ϵ-Local (non-interactive) Differential Privacy)**.** A protocol is ϵ-local DP iff each party's local randomizer Q_i is an ϵ-DP mechanism for 1-row databases.

The benefits of the local model are that there is no trusted curator, no single point of failure, and the system is highly distributed. The main drawback is that the local model cannot reach the same level of accuracy as the central model.

[1] See Salil's survey on "The Complexity of Differential Privacy" to gain more clarity on communication protocols for differential privacy.

Theorem 2.2 (Utility in the Local Model)**.** *If f is the target function and M is (ϵ, δ)-LDP, then for all adversaries A,*

$$\mathbb{E}\left[\left|A(M(x)) - \frac{1}{n}\sum_i f(x_i)\right|\right] = \Omega\big(\frac{1}{\epsilon\sqrt{n}}\big).$$

As we will see through the example of randomized response, this lower bound on expected error is tight.

3 Randomized Response

Randomized response [Warner, 1965], which has existed long before the notion of differential privacy, is the canonical example of a mechanism in the local model. In randomized response, each person has data $x_i \in \mathcal{X}$, and the analyst wants to compute an average of a boolean predicate $f : \mathcal{X} \to \{-1, 1\}$ over x. For example, the analyst might want to know the fraction of diabetics in the dataset.

Each person applies f to x_i (themselves) to get the sensitive bit $y_i = f(x_i)$, which in our example, would indicate whether or not they are diabetic. Then, they apply the following local randomizer Q to y_i:

$$Q = \begin{cases} +y_i & \text{w.p. } \frac{e^\epsilon}{e^\epsilon+1} \\ -y_i & \text{w.p. } \frac{1}{e^\epsilon+1} \end{cases}.$$

First, note that the ratio between probability of $+y_i$ and $-y_i$ is e^ϵ, so the randomizer is ϵ-DP. Next, let us solve for the scaling we need to apply so that the expectation of the randomizer is the true value.

Exercise 3.1. Solve for the scaling factor c_ϵ such that $E[c_\epsilon \cdot Q(y_i)] = y_i$, and use this to construct an unbiased estimator M that the analyst can use to estimate the average of f over x.

Solution.

$$\begin{aligned}\mathbb{E}[c_\epsilon \cdot Q(y_i)] &= c_\epsilon\left(y_i\frac{e^\epsilon}{e^\epsilon+1} - y_i\frac{1}{e^\epsilon+1}\right) \\ &= c_\epsilon \cdot y_i \cdot \frac{e^\epsilon-1}{e^\epsilon+1}.\end{aligned}$$

We see that $\mathbb{E}[c_\epsilon \cdot Q(y_i)] = y_i$ when $c_\epsilon = \frac{e^\epsilon+1}{e^\epsilon-1}$. Thus, the analyst can estimate the average of f over x by computing

$$M(x) = \frac{1}{n}\sum_i c_\epsilon Q(y_i).$$

The expected absolute error is on the order of $1/\epsilon\sqrt{n}$, in contrast with the smaller error of $1/\epsilon n$ in the central model (using, for example, Laplace or Geometric noise). Note that the expected error for randomized response matches the optimal utility for an ϵ-local DP algorithm.

$$\mathbb{E}\left[\left|A(M(x)) - \frac{1}{n}\sum_i y_i\right|\right] = O\left(\frac{1}{\epsilon\sqrt{n}}\right).$$

Where does the $1/\epsilon\sqrt{n}$ come from? The $1/\epsilon$ is the minimal noise we must apply when using privacy loss of ϵ to hide the effect of each individual (similar to adding Laplace scale of GS/ϵ, where $GS = 1$ here). Then, notice that we have n users each with independent randomness. When averaging n independent random variables, we get standard deviation of $1/\sqrt{n}$.

Exercise 3.2. Using the fact that the $Q(f(x_i))$'s are pairwise independent, and the max variance of a ± 1 Bernoulli is $1/2$, show that the standard deviation of $\frac{1}{n}\sum_{i=1}^{n} c_\epsilon \cdot Q(y_i) = O(1/\sqrt{n})$.

Solution. We show the derivation below:

$$\begin{aligned}
\mathrm{Var}\Big(\frac{1}{n}\sum_{i=1}^{n} c_\epsilon Q(y_i)\Big) &= \frac{c_\epsilon^2}{n^2}\mathrm{Var}\Big(\sum_{i=1}^{n} Q(y_i)\Big) \\
&= \frac{c_\epsilon^2}{n^2}\sum_{i=1}^{n}\mathrm{Var}(Q(y_i)) && (Q(f(x_i))\text{'s are pairwise independent}) \\
&= O\Big(\frac{c_\epsilon^2}{n^2}\sum_{i=1}^{n}\frac{1}{2}\Big) && (\text{By max variance of a } \pm 1 \text{ Bernoulli}) \\
&= O\Big(\frac{c_\epsilon^2}{n^2}\cdot\frac{n}{2}\Big) \\
&= O\Big(\frac{c_\epsilon^2}{2n}\Big).
\end{aligned}$$

Taking the square root, we have that standard deviation is $O(c_\epsilon/\sqrt{2n}) \approx O(1/(\epsilon\sqrt{n}))$.

Exercise 3.3. Let $z = (z_1 \quad z_0)^T$ be the true fraction of 1s and -1s in the sensitive bits and $\hat{z} = (\hat{z}_1 \quad \hat{z}_0)^T$ be the fraction of 1s and -1s from the randomized responses. Let $\tilde{z} = (\tilde{z}_1 \quad \tilde{z}_0)^T$ be the (unbiased) estimate of the fraction of 1s and -1s.

1. Define a map H such that $\hat{z} = Hz$.
2. Is H always invertible? When invertible, what is H^{-1}? How can you use it to obtain $\tilde{z}$?

3.1 Randomized Response for Bounded Continuous Functions

To modify the randomized rounding protocol for bounded continuous functions (say, $f(x_i) = y_i \in [-1, 1]$), we linearly interpolate between $\frac{e^\epsilon}{e^\epsilon+1}$ and $\frac{1}{e^\epsilon+1}$. For example, $y_i = 1$ should induce the same randomizer as above, $y_i = -1$ should make the probabilities flip, and $y_i = 0$ should yield equal probabilities of outputting 1 and -1.

Let the function be $f : \mathcal{X} \to [-1, 1]$, and let $c = (1 + y_i)/2$, and $d = (1 - y_i)/2$. Then, we can define the local randomizer as

$$Q(y_i) = \begin{cases} 1 & \text{w.p. } \frac{e^\epsilon c + d}{e^\epsilon+1} \\ -1 & \text{w.p. } \frac{e^\epsilon d + c}{e^\epsilon+1} \end{cases}.$$

Exercise 3.4. Prove that Q is ϵ-DP.

Solution. The worst case change in probability of $Q(y_i)$ for any given output occurs when y_i changes from 1 to -1. In this case, we can that $\Pr(Q(1) = 1) = e^\epsilon \Pr(Q(-1) = 1)$, and vice versa.

Next, let $c_\epsilon = \frac{e^\epsilon+1}{e^\epsilon-1}$, and note that

$$\begin{aligned}\mathbb{E}[c_\epsilon \cdot Q(y_i)] &= c_\epsilon\left(1\frac{e^\epsilon c + d}{e^\epsilon+1} - 1\frac{e^\epsilon d + c}{e^\epsilon+1}\right)\\ &= c_\epsilon \cdot \frac{(e^\epsilon-1)(c-d)}{1+e^\epsilon}\\ &= c_\epsilon \cdot y_i \cdot \frac{e^\epsilon-1}{e^\epsilon+1}\\ &= y_i.\end{aligned}$$

The expected error is the same as in the discrete case.

4 Applying Randomized Response to Histograms

Let every participant i have a bit $x_i \in \{1,\dots,D\}$. We want to estimate the histogram $f(x) = (n_1,\dots,n_D)$, where $n_j = \#\{i : x_i = j\}$. We will map each x_i to a "1-hot" indicator vector of length D and feed this into a local randomizer $Q(x_i)$, which works as follows.

1. Constructs "1-hot" vector $e_{x_i} = (0,\dots,0,1,0,\dots,0) \in \{0,1\}^D$.
2. Applies $(\epsilon/2)$-DP randomized response to each coordinate to get $y_i \in \{0,1\}^D$:

$$y_i = \begin{cases} e_{x_i}[j] & \text{w.p. } \frac{e^{\epsilon/2}}{1+e^{\epsilon/2}} \\ 1 - e_{x_i}[j] & \text{o.w.}\end{cases}.$$

3. Sends y_i to server.

Without privacy, the server would just sum up these indicator vectors to estimate the histogram. Since the sums are noisy, however, the server needs to add a multiplicative scaling factor c_ϵ (and, if we were using ± 1, an additive term) to correct the expectations of the sums. Thus, the server estimates $\hat{f}$ of the histogram f as follows:

$$\hat{f}(x) = c_\epsilon \sum_{i=1}^{n} y_i.$$

The expected error per bin is $\pm O(\sqrt{n}/\epsilon)$, using a similar analysis as we did when looking at basic randomized response, except now with counts instead of means. The expected max error over all D bins can be shown to be $\pm O(\sqrt{n \log D}/\epsilon)$.

The problem with this approach for estimating histograms is that the vectors can get quite long for the aggregator to store, which is why companies have used hashing and other compression techniques in their deployments.

5 Group Privacy in the Local Model

Next, we will see that the local model yields stronger guarantees of group privacy than does the centralized model. This result is due to a recent work by Bun, Nelson and Stemmer.

Recall that group privacy ensures that the effect that a group of individuals has on the outcome of the computation degrades gracefully with the size of the group. In the central model, an ϵ-DP algorithm satisfies $k\epsilon$-DP for every group of k individuals. In the local model, group privacy degrades the privacy parameter only by a factor of $\sqrt{k}$.

Theorem 5.1 (Advanced Grouposition for pure LDP)**.** *Let $x, x' \in \mathcal{X}^n$ differ in at most k entries, where $k \leq n$. Let $M = (Q_1, \ldots, Q_n) : \mathcal{X}^n \to \mathcal{Y}$ be an ϵ-DP mechanism. Then, for every $\delta > 0$ and $\epsilon' = O(k\epsilon^2/2 + \epsilon\sqrt{2k\ln(1/\delta)})$, we have that*

$$\Pr[M(x) \in T] \leq e^{\epsilon'} \Pr[M(x') \in T] + \delta.$$

Proof. Let us denote privacy loss between two random variables A, B as

$$L_{A,B} = \ln\Big(\frac{\Pr[A = y]}{\Pr[B = y]}\Big).$$

Without loss of generality, assume that x, x' differ in the first k coordinates. Since each randomizer is applied independently, we can express the privacy loss between $M(x)$ and $M(x')$ as

$$L_{M(x),M(x')} = \sum_{i=1}^{k} L_{Q(x_i),Q(x_i')}.$$

Recall Hoefdding's inequality. For independent random variables $X_1, \ldots, X_n$, such that $a_i \leq X_i \leq b_i$, $S_n = \sum_{i=1}^n X_i$, and $t > 0$,

$$\Pr[S_n > E[S_n] + t] \leq e^{-2t^2/\sum_{i=1}^n (b_i - a_i)^2}$$

We will use as a fact (see Salil's Complexity of DP survey for derivation) that the expected privacy loss of any individual local randomizer is $L_{Q_i(x_i),Q_i(x_i')} = \frac{1}{2}\epsilon^2$, which is substantially smaller than the worst-case privacy loss of ϵ. Then, noting that each $L_{Q_i(x_i),Q_i(x_i')} \in [-\epsilon, \epsilon]$, and applying Hoeffding's inequality, we have that for every $t > 0$,

$$\Pr\left[L_{M(x),M(x')} > k\epsilon^2/2 + t\right] \leq e^{-t^2/2k\epsilon^2}.$$

Finally, we can set $t = \epsilon\sqrt{2k\ln(1/\delta)}$ to yield

$$\Pr\left[L_{M(x),M(x')} > k\epsilon^2/2 + \epsilon\sqrt{2k\ln(1/\delta)}\right] \leq \delta.$$

□

This proof is very similar to the proof of advanced composition, and in fact, the result can be seen as an application of the advanced composition theorem. We leverage the fact that the expected privacy loss is substantially smaller than the worst case privacy loss of ϵ. Since each local randomizer is applied independently (analogous to independent releases in composition), the cumulative privacy loss concentrates to within $O(\sqrt{k\epsilon})$ of its expectation $O(k\epsilon^2)$.

6 Other Models of Differential Privacy

Local and central are just two models; we can think of many more! One active area of research is the multiparty model. Multiparty DP is an intermediate model where there are k curators (denoted C_j), each with a dataset $D^{(j)}$ on n_j subjects. Each curator is allowed to access the raw data of its own subjects, but not of the subjects of other curators. The k curators interact to carry out a joint analysis.

Definition 6.1 ((ϵ, δ)-Multiparty Differential Privacy)**.** For all $D^{(j)}, D^{(j)'}$ differing on one row, and for all polynomial-time adversaries A,

$$\Pr[A(C_j(D^{(j)})) = \text{ YES }] \leq e^{\epsilon} \Pr[A(C_j(D^{(j)'})) = \text{ YES }] + \delta.$$

We can also quantify over only "honest-but-curious" or "semi-honest" adversaries that follow the protocol but may try to glean sensitive information or do extra computations afterwards. This may be a reasonable model for aggregators such as Apple or Google, who want to protect themselves from subpeonas but who still have incentives to learn sensitive data. Other approaches are to use "threshold" adversaries or to anonymize the participants using a mixnet and verifiable shuffle for a boost in privacy. See Salil's survey on "The Complexity of Differential Privacy" for more discussion on this topic.