CS325: Private Data Analysis **Fall 2023**

Lecture 2: Definition of Differential Privacy

Lecturer: Di Wang *Scribes: Di Wang*

Note: *LaTeX template courtesy of UC Berkeley EECS dept.*

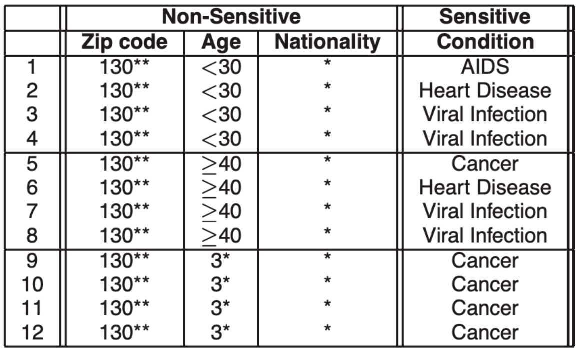
Disclaimer: *These notes have not been subjected to the usual scrutiny reserved for formal publications. They may be distributed outside this class only with the permission of the Instructor.*

2.1 Definition of "Privacy"

In the previous lecture, we have seen that it is possible to perform the reconstruction attack based on the statistics it released, we now want to get a handle on what it means that some set of statistics are actually "ok" to releasethat they dont expose individuals data (too much?) to attacks like the ones of the last lecture.

The question isn't new. Researchers in statistics, computer science, and information theory have been tackling variations on it since the 1960s [3], and a many algorithms and techniques were developed that resist specific suites of attacks. Generally speaking, there are two approaches: The first one formulates suite of attack algorithms, look at mechanism that empirically resist those attacks, such as K-anonymity. And the second one formulates general criteria by proving that algorithms which satisfy the criteria resist all attacks in a class. Actually, Differential Privacy, the core definition in this class, belongs to the latter approach of defining DP. However, before that, we first see why the first approach does not satisfy our desiderata.

k-Anonymity [2] is one popular approach to reasoning about the privacy implications of publishing statistical tables. It applies only to specific kinds of information, called generalized micro-data. This means a table of individual records, where each entry is either the original records entry (a specific persons real age, for example) or a set of possible values for that entry (often an interval, like 30-34). Figure 1 gives an example of such a table with age and zip code data. The basic idea of k-anonymity is to divide attributes into non-sensitive attributesassumed to be available to an attackerand sensitive onesassumed to be unknownand to ensure that every record matches at least $k-1$ others in the nonsensitive attributes. That is, given an integer k, a table is k-anonymous if, when we delete the sensitive attributes and leave only the non-sensitive ones, each row appears at least k times. The table of Figure 2.1 is 4-anonymous.



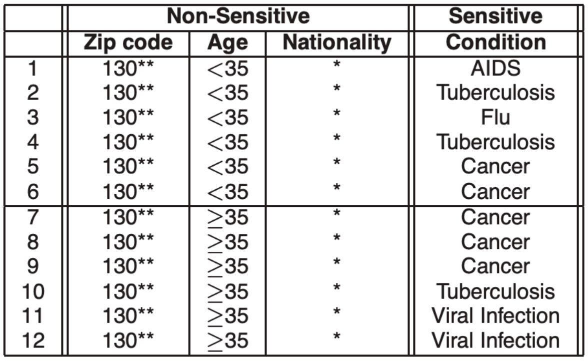
	Non-Sensitive			Sensitive
	Zip code	Age	Nationality	Condition
1	130**	<30	*	AIDS
2	130**	<30	*	Heart Disease
3	130**	<30	*	Viral Infection
4	130**	<30	*	Viral Infection
5	130**	≥40	*	Cancer
6	130**	≥40	*	Heart Disease
7	130**	≥40	*	Viral Infection
8	130**	≥40	*	Viral Infection
9	130**	3*	*	Cancer
10	130**	3*	*	Cancer
11	130**	3*	*	Cancer
12	130**	3*	*	Cancer

Figure 2.1: A 4-anonymous table

The idea behind this criterion is that it makes linkage attacks harder to carry outif an attacker has access to another table that contains some of the non-sensitive attributes, there are at least k samples in the anonymous table will match each record in other table.

While k-anonymity is likely to make those specific attacks harder than they would be with raw data, a k-anonymous table can still leak lots of individual-level information. We can glean lots of information from the table in Figure 2.1: everyone in their 30's has cancer; our friend Alice, who's data we happen to know is in the table, cannot have visited the hospital because of a broken leg; etc. Of course, the example is simplistic (real hospital records don't look like the example in the table...) but it illustrates two important points: 1. Defending against one type of attack isnt sufficient, and 2. Criteria that limit the form of the output (in this case, the number of occurrences of each vector of non-sensitive attributes) do not necessarily constrain the information that is revealed. Besides that, in the following we show that k-anonymous table does not have the property of composition.

Composition: k-anonymity illustrates another important point, namely that when the same record is included in two (or more) data sets that are anonymized separately, the combination of the two might reveal far more than the two do individually [1]. For example, consider the table of Figure 2.1. Suppose we know that Alices record appears in both tables, and that she is 28 years old and lives in zip code 13012. Neither table on its own pins down her condition exactly (each one narrows it down to a few possibilities), but taken together they pin it down exactly. This problem is known as compositionwhat happens when many different pieces of information are revealed about me? We return to this question in the next lecture.



	Non-Sensitive			Sensitive
	Zip code	Age	Nationality	Condition
1	130**	<35	*	AIDS
2	130**	<35	*	Tuberculosis
3	130**	<35	*	Flu
4	130**	<35	*	Tuberculosis
5	130**	<35	*	Cancer
6	130**	<35	*	Cancer
7	130**	≥35	*	Cancer
8	130**	≥35	*	Cancer
9	130**	≥35	*	Cancer
10	130**	≥35	*	Tuberculosis
11	130**	≥35	*	Viral Infection
12	130**	≥35	*	Viral Infection

Figure 2.2: A 6-anonymous table

2.2 Differential Privacy

In this section we introduce the definition of differential privacy. First we provide some notations. Let $\mathcal{X}$ be the set of all possible records for each individual. A dataset D is thus a multiset of values on $\mathcal{X}$. When the size n is fixed we may think of it as a list $D = \{x_1, \cdots, x_n\}$.

The main idea of DP is to consider a thought experiment in which we compare how an algorithm behaves on a dataset D with the way it behaves on a hypothetical dataset D' in which one persons record has been replaced with some other values. We say two data sets are neighbors if they differ in one individuals record. A simple way to model this is to think of the size n of data sets as fixed , and to consider two data sets adjacent if one record has been replaced with a different value. For example, if they differ in index i, we would have

$$D = \{x_1, x_2, \cdots, x_{i-1}, x_i, x_{i+1}, \cdots, x_n\}$$
$$D' = \{x_1, x_2, \cdots, x_{i-1}, x_i', x_{i+1}, \cdots, x_n\}.$$

Now consider a **randomized algorithm** $\mathcal{A}$. For each possible input data set D, its output is a random variable $\mathcal{A}(D)$. We say an algorithm is differentially private if running the algorithm on two neighboring data sets yields roughly the same distribution on outcomes. Specifically, well ask that for every set E of possible outcomesfor example, those outputs from a healthcare study that lead to individual i being denied health insurancethe probability of an outcome in E should be the same under $\mathcal{A}(D)$ and $\mathcal{A}(D')$, up to a small multiplicative factor. In other words, the algorithms outcomes should be about the same whether or not individual is real data was used.

Definition 2.1 *A randomized algorithm $\mathcal{A} : \mathcal{X}^n \mapsto \mathcal{Y}$ taking inputs in $\mathcal{X}^n$ is ϵ-differentially private (DP) if, for every pair of neighboring data set D, D' (we denote it as $D \sim D'$, for all events $E \subseteq \mathcal{Y}$ we have*

$$\mathbb{P}(\mathcal{A}(D) \in E) \le e^{\epsilon}\mathbb{P}(\mathcal{A}(D') \in E). \tag{2.1}$$

Here are some notations on the definition:

1. DP is a property of an algorithm. Specifically, it is only for randomized algorithms.
2. Differential privacy is quantitative in nature. A small ϵ corresponds to strong privacy, degrading as ϵ increases. This is due to that when ϵ is small, then the two distributions will be close, thus they are difficult to be distinguished.
3. The parameter ϵ should be thought of as a small-ish constant. Anything between (say) 0.1 and 5 might be a reasonable level privacy guarantee (smaller corresponds to stronger privacy), and you should be slightly skeptical of claims significantly outside this range. However as we will mention in the next, ϵ cannot be too small.
4. In the definition of DP, we require the ratio between the two probabilities is upper bounded by e^{ϵ}. The question is that why we use e^{ϵ}? There are two reasons: 1) when ϵ is small enough then we have $e^{\epsilon} \approx 1 + \epsilon$, i.e., when ϵ is small, then the two distributions will be almost the same.
5. The given definition is convenient because of the fact that $e^{\epsilon_1+\epsilon_2} = e^{\epsilon_1}e^{\epsilon_2}$. This will be helpful for the composition and group privacy in the next lecture.
6. Note that the definition of DP is a worst case guarantee, that is we have the inequality for any pair of neighboring data $D \sim D'$. While there do exist some notions of average-case privacy, these should be approached with caution Steinke and Ullman write a series of posts which warn about the pitfalls of average-case notions of differential privacy.
7. In the original definition of DP, we assume the data is tabular data so we define neighboring data to be the data will one row replaced. However, based on different types of data we may define other neighboring data. We will illustrate it in the later lectures.
8. Note that in the original definitions of DP, we use the **ratio** between two probabilities as the measurement of closeness. Actually, we can define new definitions of DP be changing the measurement of closeness, such as KL-divergence, Wasserstein distance between two probabilities... we will show some of them in the later lectures. However, as we will see in the next lecture, some measurement may cannot preserve the privacy!

In the above part we introduced DP. In the following two sections we will see how can we design mechanisms that satisfy ϵ-DP.

2.3 Random Response

Suppose we have an n-size dataset $D = \{x_1, x_2, \cdots, x_n\}$ where for each individual $x_i \in \{0, 1\}$. For each person, we will generate a biased random bit Y_i as follows: roll a fair die. If the die comes up $1, 2, 3$ or 4 we will set $Y_i = x_i$. Otherwise we set Y_i to be the opposite value of x_i (that is $Y_i = 1 - x_i$). The algorithm's output is the list of values $(Y_1, \cdots, Y_n)$. If we call the algorithm as RR_{basic}, in the following we will show it is $\ln 2$-DP.

Consider a neighboring dataset of D, D' which differs with D in the i-th entry, *i.e.*, $D' = \{x_1, x_2, \cdots, x'_i, \cdots, x_n\}$. For a fixed output $y = (y_1, \cdots, y_n)$ we have

$$\frac{\Pr(RR_{\text{basic}}(D) = y)}{\Pr(RR_{\text{basic}}(D') = y)} = \frac{\Pi_{j=1}^{n} \Pr(Y_j = y_j | x_j)}{\Pi_{j=1, j\neq i}^{n} \Pr(Y_j = y_j | x_j) \cdot \Pr(Y_i = y_i | x'_i)} = \frac{\Pr(Y_i = y_i | x_i)}{\Pr(Y_i = y_i | x'_i)} = \{\frac{1}{2}, 2\} \leq 2, \tag{2.2}$$

where the last inequality is due to the distribution of Y_i. Thus for any set E we have

$$\frac{\Pr(RR_{\text{basic}}(D) \in E)}{\Pr(RR_{\text{basic}}(D') \in E)} = \frac{\sum_{y\in E} \Pr(RR_{\text{basic}}(D) = y)}{\sum_{y\in E} \Pr(RR_{\text{basic}}(D') = y)} \leq e^{\ln 2}. \tag{2.3}$$

Thus, it is $\ln 2$-DP.

The proof that randomized response is differentially private uses a useful trick that is true quite general

Theorem 2.2 (Equivalent form of ϵ-DP) *If the output space $\mathcal{Y}$ is discrete, then an algorithm $\mathcal{A} : \mathcal{X}^n \mapsto \mathcal{Y}$ is ϵ-DP if and only if for every $y \in \mathcal{Y}$ we have*

$$\mathbb{P}(\mathcal{A}(D) = y) \leq e^{\epsilon} \mathbb{P}(\mathcal{A}(D') = y).$$

Similarly, if the distributions of $\mathcal{A}(D)$ and $\mathcal{A}(D')$ both have probability densities, then $\mathcal{A}$ is ϵ-DP if $f_{A(D)}(y) \leq f_{A(D')}(y)$ for all possible output $y \in \mathcal{Y}$, where $f_{A(D)}$ and $f_{A(D')}$ is the density function of $A(D)$ and $A(D')$ respectively.

The factor of 2 maybe not be quite satisfactory. But we can get it to be arbitrarily close to 1 by changing the mechanism a bit. Now we consider a general case, where each $x_i \in \mathcal{X}$ and $\mathcal{X}$ is the data universe. We also have a predict ϕ that maps each record to a bot. For example, ϕ could be "whether an individual has diabetics?". And we want to know the statistics $\Phi(D) = \sum_{i=1}^{n} \phi(x_i)$. The generalized Random Response is the following

Algorithm 1 Generalized Random Response

```
Require: Dataset $D = \{x_1, \cdots, x_n\}$, predicate $\phi : \mathcal{X} \mapsto \{0, 1\}$ and parameter $\epsilon > 0$
Ensure: Bits $(Y_1, \cdots, Y_n)$
for $i = 1, \cdots, n$ do
    Let $Y_i = \phi(x_i)$ w.p $\frac{e^\epsilon}{e^\epsilon + 1}$ and $Y_i = 1 - \phi(x_i)$ otherwise.
end for
```

Theorem 2.3 *Algorithm 1 is ϵ-DP.*

The above theorem tells us the random response is private. But how about the accuracy? First we show that Y_i will be a biased estimator of $\phi(x_i)$ but there is some constants a, b such that $aY_i + b$ is an unbiased estimator.

Theorem 2.4

$$\mathbb{E}[Y_i] = \frac{e^\epsilon - 1}{e^\epsilon + 1}\phi(x_i) + \frac{1}{e^\epsilon + 1} \tag{2.4}$$

Thus we have

$$\mathbb{E}[\frac{e^\epsilon + 1}{e^\epsilon - 1}Y_i - \frac{1}{e^\epsilon - 1}] = \phi(x_i). \tag{2.5}$$

Proof: This is just followed by the definition of Y_i. ■

By the previous theorem, we can see that if we want to get the statistics, then it is preferable to output $\{\frac{e^\epsilon+1}{e^\epsilon-1}Y_i - \frac{1}{e^\epsilon-1}\}_{i=1}^n$. The question is, what is the error if we release $\sum_{i=1}^n(\frac{e^\epsilon+1}{e^\epsilon-1}Y_i - \frac{1}{e^\epsilon-1})$? In the following theorem we will show it is $O(\frac{\sqrt{n}}{\epsilon})$

Theorem 2.5 *If we denote $Z_i = \frac{e^\epsilon+1}{e^\epsilon-1}Y_i - \frac{1}{e^\epsilon-1}$ then when ϵ is small enough we have*

$$\sqrt{\mathbb{E}[(\sum_{i=1}^n Z_i - \sum_{i=1}^n \phi(x_i))^2]} \leq O(\frac{\sqrt{n}}{\epsilon}). \tag{2.6}$$

Proof: By (2.5) we have

$$\mathbb{E}[(\sum_{i=1}^n Z_i - \sum_{i=1}^n \phi(x_i))^2] = (\frac{e^\epsilon + 1}{e^\epsilon - 1})^2\mathbb{E}[(\sum_{i=1}^n(Y_i - \mathbb{E}(Y_i)))^2]. \tag{2.7}$$

Since each Y_i is independent on others, we have $\mathbb{E}[(Y_i - \mathbb{E}(Y_i))\cdot(Y_j - \mathbb{E}(Y_j))] = \mathbb{E}(Y_i - \mathbb{E}(Y_i))\cdot\mathbb{E}(Y_j - \mathbb{E}(Y_j)) = 0$. Thus we have

$$\mathbb{E}[(\sum_{i=1}^n(Y_i - \mathbb{E}(Y_i)))^2] = \sum_{i=1}^n \mathbb{E}[(Y_i - \mathbb{E}(Y_i))^2] \leq n, \tag{2.8}$$

where the last inequality is due to $Y_i \in \{0, 1\}$. In total we have

$$\mathbb{E}[(\sum_{i=1}^n Z_i - \sum_{i=1}^n \phi(x_i))^2] \leq n(1 + \frac{2}{e^\epsilon - 1})^2 = O(\frac{n}{\epsilon^2}), \tag{2.9}$$

where the last inequality is due to that $e^\epsilon - 1 \geq \epsilon$ and $\epsilon \leq 1$. ■

Thus if we want to estimate the average, *i.e.,* $\frac{1}{n}\sum_{i=1}^n \phi(x_i)$ via using $\frac{1}{n}\sum_{i=1}^n Z_i$, we can see that the error will be $O(\frac{1}{\sqrt{n}\epsilon})$. The question is, can we do better? In the following section, we will show a mechanism which has improved estimation error.

2.4 Laplacian Mechanism

Another natural way to add randomness to a computation is to simply add noise to the output of some function f evaluated on the data. This function could just return a single real number (like a proportion or a sum), or it could be something more complex that returns a vector in $\mathbb{R}^d$, (such as the roughly 3 billion statistics produced by the US Census Bureau from its decennial census). When does adding noise satisfy differential privacy? How does the choice of the function f we evaluate affet the amount of noise we must add? One basic idea is to look at how sensitive a function is to a change in one of its input records. We measure this via the global sensitivity of f.

Definition 2.6 *Given a function $f : \mathcal{X} \mapsto \mathbb{R}^d$, we define the global sensitivity of f in the ℓ_1-norm to be*

$$GS_{f,\ell_1} = \sup_{D \sim D'} \|f(D) - f(D')\|_1, \tag{2.10}$$

where $D \sim D'$ means that D and D' are neighbors.

If f has global sensitivity Δ, we can satisfy ϵ-DP by adding noise from a Laplace distribution with scale $\frac{\Delta}{\epsilon}$, independently to each entry of the output. The Laplace distribution, also called the double exponential distribution, is sort of a pointy Gaussian. The mean-0, scale-1 Laplace has density $h(y) = \frac{1}{2}\exp(-|y|)$ for $y \in \mathbb{R}$. We can scale the distribution by a positive number $\lambda > 0$, to get the general form $\mathrm{Lap}(\lambda)$ with density $p_\lambda(y) = \frac{1}{2\lambda}\exp(-\frac{|y|}{\lambda})$. The resulting algorithm (Algorithm 3) is called the Laplace mechanism4 , and is a basic building block for the design of many other differentially private algorithms.

Algorithm 2 Laplacian Mechanism

```
Require: Dataset $D = \{x_1, \cdots, x_n\}$, query function $f : \mathcal{X}^n \mapsto \mathbb{R}^d$ and parameter $\epsilon > 0$
  Calculate the $\ell_1$-norm global sensitivity of $f$ and denote it as $\mathrm{GS}_f$.
  Return $M(D) = f(D) + (Z_1, \cdots, Z_d)$ where $Z \sim \mathrm{Lap}(\frac{\mathrm{GS}_f}{\epsilon})$ i.i.d. Here the a random variable $Z \sim \mathrm{Lap}(\lambda)$
  if it has a density function of $h_\lambda(y) = \frac{1}{2\lambda}\exp(-\frac{|y|}{\lambda})$.
```

Theorem 2.7 *The Laplacian mechanism (Algorithm 2) is ϵ-DP.*

Proof: Fix two neighboring datasets D and D'. Let $\Delta = \mathrm{GS}_f$ be the ℓ_1-norm global sensitivity of f. Let $\mu = f(D)$ and $\mu' = f(D')$. Because we add noise independently to each entry of the output, the density of the output at vector y on input x can be written as

$$h_D(y) = \frac{\epsilon}{2\Delta}\exp(-\frac{\epsilon}{\Delta}|y_1 - \mu_1|) \times \cdots \times \frac{\epsilon}{2\Delta}\exp(-\frac{\epsilon}{\Delta}|y_d - \mu_d|) \tag{2.11}$$

The same for D'

$$h_{D'}(y) = \frac{\epsilon}{2\Delta}\exp(-\frac{\epsilon}{\Delta}|y_1 - \mu'_1|) \times \cdots \times \frac{\epsilon}{2\Delta}\exp(-\frac{\epsilon}{\Delta}|y_d - \mu'_d|) \tag{2.12}$$

Thus we have

$$\begin{aligned}
\frac{h_D(y)}{h_{D'}(y)} &= \frac{\frac{\epsilon}{2\Delta}\exp(-\frac{\epsilon}{\Delta}|y_1 - \mu_1|) \times \cdots \times \frac{\epsilon}{2\Delta}\exp(-\frac{\epsilon}{\Delta}|y_d - \mu_d|)}{\frac{\epsilon}{2\Delta}\exp(-\frac{\epsilon}{\Delta}|y_1 - \mu'_1|) \times \cdots \times \frac{\epsilon}{2\Delta}\exp(-\frac{\epsilon}{\Delta}|y_d - \mu'_d|)} \\
&= \exp(\frac{\epsilon}{\Delta}(|y_1 - \mu'_1| - |y_1 - \mu_1|)) \times \cdots \times \exp(\frac{\epsilon}{\Delta}(|y_d - \mu'_d| - |y_1 - \mu_d|)) \\
&\leq \Pi_{j=1}^d \exp(\frac{\epsilon}{\Delta}|\mu'_j - \mu_j|) = \exp(\frac{\epsilon}{\Delta}\|u' - u\|_1) \leq \exp(\frac{\epsilon}{\Delta}\Delta) = e^\epsilon.
\end{aligned}$$

■

The following lemmas provides useful bounds of the Laplacian mechanisms error.

Lemma 2.8

1. *If $Z \sim Lap(\lambda)$ us a Laplace-distributed random variable, we have*
   - *(a)* $\mathbb{E}(|Z|) = \lambda$
   - *(b)* $\sqrt{\mathbb{E}(Z^2)} = \sqrt{2}\lambda$
   - *(c) For every $t > 0$:* $\mathbb{P}(z > \lambda t) \leq \exp(-t)$.
2. *Let $Z_1, \cdots, Z_d$ are i.i.d sampled from $Lap(\lambda)$, and let $M = \max\{|Z_1|, \cdots |Z_d|\}$ then*

(a) $\mathbb{E}(\|(Z_1, \cdots, Z_d)\|_1) = d\lambda$

(b) For every $t > 0$*:* $\mathbb{P}(M > \lambda(\ln d + t)) \leq \exp(-t)$.

(c) $\mathbb{E}(M) \leq \lambda(\ln d + t)$

Proof: We first proof 1.(a). We have

$$\begin{aligned}\mathbb{E}(|Z|) &= \int_{-\infty}^{\infty} \frac{|y|}{2\lambda} \exp(-\frac{|y|}{\lambda})dy \\ &= \int_0^{\infty} \frac{|y|}{\lambda} \exp(-\frac{|y|}{\lambda})dy = \int_0^{\infty} yd(-\exp(-\frac{y}{\lambda})) \\ &= -\exp(-\frac{y}{\lambda})y|_0^{\infty} + \int_0^{\infty} \exp(-\frac{y}{\lambda})dy = \lambda\end{aligned}$$

The 1.(b) and 1.(c) just follows the density function of the Laplacian distribution.

Next we consider 2. For 2.(a) it is just followed by 1.(a) and each Z_i is i.i,d. sampled. For 2.(b) we have

$$\begin{aligned}&\mathbb{P}(M > \lambda(\ln d + t)) \\ &\leq \mathbb{P}\left(|Z_1| > \lambda(\ln d + t) \text{ or } |Z_2| > \lambda(\ln d + t) \cdots, \text{ or } |Z_d| > \lambda(\ln d + t)\right) \\ &\leq \sum_{i=1}^{d} \mathbb{P}\left(|Z_1| > \lambda(\ln d + t)\right) \\ &\leq d \times \mathbb{P}(Z > \lambda(\ln d + t)) \leq d\exp(-(\ln d + t)) = \exp(-t).\end{aligned}$$

For the term of $\mathbb{E}(M)$, recall that for any non-negative random variable M, $\mathbb{E}(M) = \int_{x=0}^{\infty} \mathbb{P}(M > x)dx$. Thus, we have

$$\begin{aligned}\mathbb{E}(M) = \int_0^{\infty} \mathbb{P}(M > t)dt &= \int_0^{\lambda \ln d} \mathbb{P}(M > t)dt + \int_{\lambda \ln d}^{\infty} \mathbb{P}(M > t)dt \\ &\leq \lambda \ln d + \int_0^{\infty} \mathbb{P}(M > \lambda(x + \ln d))d(\lambda(x + \ln d)) \\ &\leq \lambda \ln d + \lambda \int_0^{\infty} e^{-x}dx \\ &\leq \lambda(\ln d + 1).\end{aligned}$$

■

In the following we will use the Laplacian mechanism to the average query and the histogram:

Average Query: Now we focus on the average query, $\Phi(D) = \frac{1}{n}\sum_{i=1}^{n} \phi(x_i)$ where $\phi(x) \in \{0, 1\}$. To use the Laplacian mechanism, it is sufficient to calculate the global sensitivity. To compute it, consider any neighboring data pair D, D'. For simplicity we assume they differ in the first term which is x_1 and x'_1 in D and D' respectively. Then we have $|\Phi(D) - \Phi(D')| \leq \frac{1}{n}$. Thus the global sensitivity is $\frac{1}{n}$. Thus, the algorithm $M(D) = \Phi(D) + \text{Lap}(\frac{1}{n\epsilon})$ is ϵ-DP. By the previous Lemma we have with probability at least $1 - \zeta$,

$$|M(D) - \Phi(D)| \leq O(\frac{\ln \frac{1}{\zeta}}{n\epsilon}). \tag{2.13}$$

For the Random Response we have the following by the Chebyshev inequality

$$\mathbb{P}(|M(D) - \Phi(D)| \geq O(\frac{1}{t\sqrt{n}\epsilon})) \leq \frac{\mathbb{E}|M(D) - \Phi(D)|^2}{O(\frac{1}{n\epsilon^2 t^2})} \leq t^2. \tag{2.14}$$

Thus, with probability at least $1-\zeta$, we have

$$|M(D)-\Phi(D)|\leq O(\frac{1}{\sqrt{\zeta}\sqrt{n}\epsilon}) \tag{2.15}$$

Compared with (2.15) we can see the error of Laplacian mechanism is smaller and is in the high probability form.

Histogram: Given a data set $D\in\mathcal{X}^n$, and a partition of $\mathcal{X}$ into d disjoint sets $B_1,\cdots,B_d$ (think of these as bins or types of items in $\mathcal{X}$), we count how many records there are of each type. In this case the histogram query $f(D)=(n_1,n_2,\cdots,n_d)$ where $n_i=\{\#x\in B_i|x\in D\}$. So, for example, if we wanted to compute the number of residents of each of the 50 US states from a census of the US population, we would be asking a histogram query. We now consider the global sensitivity of f. Consider two neighboring data D,D' and they differ in the first item x_1,x_1'. For simplicity we assume $x_1\in B_1$ and $x_1'\in B_2$. Then we can see that $f(D)-f(D')=(1,-1,0,\cdots,0)$. Thus, the global sensitivity of a histogram query is at most 2, regardless of how many bins there are. Thus, the mechanism $M(D)=f(D)+\text{Lap}(\frac{2}{\epsilon})^d$ is ϵ-DP. Moreover by the previous lemma we know the utility is with probability at least $1-\exp(-t)$ we have

$$\|M(D)-f(D)\|_\infty\leq O(\frac{\log d+t}{\epsilon}).$$

References

[1] Srivatsava Ranjit Ganta, Shiva Prasad Kasiviswanathan, and Adam Smith. Composition attacks and auxiliary information in data privacy. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 265–273, 2008.

[2] Latanya Sweeney. k-anonymity: A model for protecting privacy. *International journal of uncertainty, fuzziness and knowledge-based systems*, 10(05):557–570, 2002.

[3] Stanley L Warner. Randomized response: A survey technique for eliminating evasive answer bias. *Journal of the American Statistical Association*, 60(309):63–69, 1965.