Introduction to Differential Privacy

10/19/2023

Outline

1. DP: formalization and basic notions
2. DP immunity to post-processing
3. Composition theory
4. DP mechanisms: the Laplace mechanism
5. DP and Deep Learning

1. “The Algorithmic Foundations of Differential Privacy”, Dwork, C., Roth, A.
2. “Differential Privacy”, Dwork, C.
3. “Deep Learning with Differential Privacy”, Abadi, M., Chu, A., Goodfellow, I., McMahan, H., Mironov, I., Talwar, K., Zhang, L.

Differential Privacy

- Differential privacy (DP) promises that the *data subject* will not be adversely affected by allowing her/his data to be used in any study or analysis, no matter what other studies, data sets, or information sources, are available.

Differential Privacy

- Differential privacy (DP) promises that the *data subject* will not be adversely affected by allowing her/his data to be used in any study or analysis, no matter what other studies, data sets, or information sources, are available.
- DP addresses the paradox of learning nothing about an individual while learning useful information about a population.

Differential Privacy

- Differential privacy (DP) promises that the *data subject* will not be adversely affected by allowing her/his data to be used in any study or analysis, no matter what other studies, data sets, or information sources, are available.
- DP addresses the paradox of learning nothing about an individual while learning useful information about a population.
- DP is a definition, not an algorithm.

Differential Privacy

- Differential privacy (DP) promises that the *data subject* will not be adversely affected by allowing her/his data to be used in any study or analysis, no matter what other studies, data sets, or information sources, are available.
- DP addresses the paradox of learning nothing about an individual while learning useful information about a population.
- DP is a definition, not an algorithm.
- For a given computational task T and a given value of ε there will be many differentially private algorithms for achieving T in an ε-differentially private manner. Some will have better accuracy than others.

Differential Privacy

- Data Cannot be Fully Anonymized and Remain Useful.

Differential Privacy

- Data Cannot be Fully Anonymized and Remain Useful.
- Re-Identification of “Anonymized” Records is Not the Only Risk.

Differential Privacy

- Data Cannot be Fully Anonymized and Remain Useful.
- Re-Identification of “Anonymized” Records is Not the Only Risk.
- Queries Over Large Sets are Not Protective.

Differential Privacy

- Data Cannot be Fully Anonymized and Remain Useful.
- Re-Identification of “Anonymized” Records is Not the Only Risk.
- Queries Over Large Sets are Not Protective.
- Query Auditing Is Problematic.

Differential Privacy

- Data Cannot be Fully Anonymized and Remain Useful.
- Re-Identification of “Anonymized” Records is Not the Only Risk.
- Queries Over Large Sets are Not Protective.
- Query Auditing Is Problematic.
- Summary Statistics are Not “Safe.”

Differential Privacy

- Privacy mechanism: algorithm that takes as input a database, random bits, and, optionally, a set of queries, and produces an output string.

Differential Privacy

- Privacy mechanism: algorithm that takes as input a database, random bits, and, optionally, a set of queries, and produces an output string.
- The hope is that the output string can be decoded to produce relatively accurate answers to the queries, if the latter are present.

Differential Privacy

- Privacy mechanism: algorithm that takes as input a database, random bits, and, optionally, a set of queries, and produces an output string.
- The hope is that the output string can be decoded to produce relatively accurate answers to the queries, if the latter are present.
- If no queries are presented then we are in the non-interactive case, and the hope is that the output string can be interpreted to provide answers to future queries.

Differential Privacy

A randomized algorithm with domain A and (discrete) range B will be associated with a mapping from A to the probability simplex over B, denoted $\Delta(B)$.

Differential Privacy

A randomized algorithm with domain A and (discrete) range B will be associated with a mapping from A to the probability simplex over B, denoted $\Delta(B)$.

Probability Simplex. Given a discrete set B, the probability simplex over B, denoted $\Delta(B)$ is defined to be:

Differential Privacy

A randomized algorithm with domain A and (discrete) range B will be associated with a mapping from A to the probability simplex over B, denoted $\Delta(B)$.

Probability Simplex. Given a discrete set B, the probability simplex over B, denoted $\Delta(B)$ is defined to be:

$$\Delta(B) = \left\{ x \in \mathbb{R}^{|B|} : x_i \geq 0 \text{ for all } i \text{ and } \sum_{i=1}^{|B|} x_i = 1 \right\}$$

Differential Privacy

Distance Between Databases (histogram representation)

Differential Privacy

Distance Between Databases (histogram representation)

Database x: collection of records from a universe $\mathcal{X}$.

Differential Privacy

Distance Between Databases (histogram representation)

Database x: collection of records from a universe $\mathcal{X}$.

Histogram representation: $x \in \mathbb{N}^{|\mathcal{X}|}$, in which each entry x_i represents the number of elements in the database x of type $i \in \mathcal{X}$.

Differential Privacy

Distance Between Databases (histogram representation)

Database x: collection of records from a universe $\mathcal{X}$.

Histogram representation: $x \in \mathbb{N}^{|\mathcal{X}|}$, in which each entry x_i represents the number of elements in the database x of type $i \in \mathcal{X}$.

The ℓ_1 norm of a database x is denoted $\|x\|_1$ and is defined to be:

$$\|x\|_1 = \sum_{i=1}^{|\mathcal{X}|} |x_i|$$

Differential Privacy

Distance Between Databases (histogram representation)

Database x: collection of records from a universe $\mathcal{X}$.

Histogram representation: $x \in \mathbb{N}^{|\mathcal{X}|}$, in which each entry x_i represents the number of elements in the database x of type $i \in \mathcal{X}$.

The ℓ_1 norm of a database x is denoted $\|x\|_1$ and is defined to be:

$$\|x\|_1 = \sum_{i=1}^{|\mathcal{X}|} |x_i|$$

$\|x\|_1$: size of a database x, in terms of number of records it contains

Differential Privacy

Distance Between Databases (histogram representation)

Database x: collection of records from a universe $\mathcal{X}$.

Histogram representation: $x \in \mathbb{N}^{|\mathcal{X}|}$, in which each entry x_i represents the number of elements in the database x of type $i \in \mathcal{X}$.

The ℓ_1 norm of a database x is denoted $\|x\|_1$ and is defined to be:

$$\|x\|_1 = \sum_{i=1}^{|\mathcal{X}|} |x_i|$$

$\|x\|_1$: size of a database x, in terms of number of records it contains

$\|x - y\|_1$: measure of how many records differ between x and y.

Differential Privacy

Differential Privacy A randomized algorithm $\mathcal{M}$ with domain $\mathbb{N}^{|\mathcal{X}|}$ is (ε, δ)-differentially private if for all $\mathcal{S} \subseteq \text{Range}(\mathcal{M})$ and for all $x, y \in \mathbb{N}^{|\mathcal{X}|}$ such that $\|x - y\|_1 \leq 1$:

Differential Privacy

Differential Privacy A randomized algorithm $\mathcal{M}$ with domain $\mathbb{N}^{|\mathcal{X}|}$ is (ε, δ)-differentially private if for all $\mathcal{S} \subseteq \text{Range}(\mathcal{M})$ and for all $x, y \in \mathbb{N}^{|\mathcal{X}|}$ such that $\|x - y\|_1 \leq 1$:

Differential Privacy

$$\Pr[\mathcal{M}(x) \in \mathcal{S}] \leq \exp(\varepsilon) \Pr[\mathcal{M}(y) \in \mathcal{S}] + \delta$$

Differential Privacy

Differential Privacy A randomized algorithm $\mathcal{M}$ with domain $\mathbb{N}^{|\mathcal{X}|}$ is (ε, δ)-differentially private if for all $\mathcal{S} \subseteq \text{Range}(\mathcal{M})$ and for all $x, y \in \mathbb{N}^{|\mathcal{X}|}$ such that $\|x - y\|_1 \leq 1$:

Differential Privacy

$$\Pr[\mathcal{M}(x) \in \mathcal{S}] \leq \exp(\varepsilon) \Pr[\mathcal{M}(y) \in \mathcal{S}] + \delta$$

where the probability space is over the coin flips of the mechanism $\mathcal{M}$. If $\delta = 0$, we say that $\mathcal{M}$ is ε-differentially private.

Differential Privacy

Differential Privacy A randomized algorithm $\mathcal{M}$ with domain $\mathbb{N}^{|\mathcal{X}|}$ is (ε, δ)-differentially private if for all $\mathcal{S} \subseteq \text{Range}(\mathcal{M})$ and for all $x, y \in \mathbb{N}^{|\mathcal{X}|}$ such that $\|x - y\|_1 \leq 1$:

Differential Privacy

$$\Pr[\mathcal{M}(x) \in \mathcal{S}] \leq \exp(\varepsilon) \Pr[\mathcal{M}(y) \in \mathcal{S}] + \delta$$

where the probability space is over the coin flips of the mechanism $\mathcal{M}$. If $\delta = 0$, we say that $\mathcal{M}$ is ε-differentially private.

(ε, δ)-differential privacy ensures that for all adjacent x, y, the absolute value of the *loss in terms of privacy* will be bounded by ε with probability at least $1 - \delta$.

Differential Privacy

δ should be less than the inverse of any polynomial in the size of the database. In particular, values of δ on the order of $1/\|x\|_1$ are to be avoided becuase they allow privacy by publishing the complete information of small groups of records.

Differential Privacy

δ should be less than the inverse of any polynomial in the size of the database. In particular, values of δ on the order of $1/\|x\|_1$ are to be avoided becuase they allow privacy by publishing the complete information of small groups of records.

$(\varepsilon, 0)$-differential privacy ensures that, for every run of the mechanism $\mathcal{M}(x)$, the output observed is (almost) equally likely to be observed on every neighboring database, simultaneously.

Differential Privacy

δ should be less than the inverse of any polynomial in the size of the database. In particular, values of δ on the order of $1/\|x\|_1$ are to be avoided becuase they allow privacy by publishing the complete information of small groups of records.

$(\varepsilon, 0)$-differential privacy ensures that, for every run of the mechanism $\mathcal{M}(x)$, the output observed is (almost) equally likely to be observed on every neighboring database, simultaneously.

In contrast (ε, δ)-differential privacy allows that, given an output $\xi \sim \mathcal{M}(x)$, it may be possible to find a database y such that ξ is much more likely to be produced on y than it is when the database is x: the mass of ξ in the distribution $\mathcal{M}(y)$ may be substantially larger than its mass in the distribution $\mathcal{M}(x)$.

Differential Privacy

Privacy loss

Differential Privacy

Privacy loss

$$\mathcal{L}^{(\xi)}_{\mathcal{M}(x)\|\mathcal{M}(y)} = \ln\left(\frac{\Pr[\mathcal{M}(x)=\xi]}{\Pr[\mathcal{M}(y)=\xi]}\right)$$

Differential Privacy

Privacy loss

$$\mathcal{L}^{(\xi)}_{\mathcal{M}(x)\|\mathcal{M}(y)} = \ln\left(\frac{\Pr[\mathcal{M}(x)=\xi]}{\Pr[\mathcal{M}(y)=\xi]}\right)$$

This loss might be positive (when an event is more likely under x than under y) or it might be negative (when an event is more likely under y than under x).

DP immunity to post-processing

Differential privacy is immune to post-processing.

DP immunity to post-processing

Differential privacy is immune to post-processing.

Without additional knowledge about the private database, no one can compute a function of the output of a private algorithm $\mathcal{M}$ and make it less differentially private.

DP immunity to post-processing

Differential privacy is immune to post-processing.

Without additional knowledge about the private database, no one can compute a function of the output of a private algorithm $\mathcal{M}$ and make it less differentially private.

Formally, the composition of a data-independent mapping f with an (ε, δ) differentially private algorithm $\mathcal{M}$ is also (ε, δ) differentially private.

DP immunity to post-processing

Differential privacy is immune to post-processing.

Without additional knowledge about the private database, no one can compute a function of the output of a private algorithm $\mathcal{M}$ and make it less differentially private.

Formally, the composition of a data-independent mapping f with an (ε, δ) differentially private algorithm $\mathcal{M}$ is also (ε, δ) differentially private.

Post-Processing

Let $\mathcal{M} : \mathbb{N}^{|\mathcal{X}|} \rightarrow R$ be a randomized algorithm that is (ε, δ)-differentially private. Let $f : R \rightarrow R'$ be an arbitrary randomized mapping. Then $f \circ \mathcal{M} : \mathbb{N}^{|\mathcal{X}|} \rightarrow R'$ is (ε, δ) differentially private.

DP immunity to post-processing

Let us consider a deterministic function $f : R \to R'$.

DP immunity to post-processing

Let us consider a deterministic function $f : R \to R'$.

Fix any pair of neighboring databases x, y with $\|x - y\|_1 \leq 1$, and fix any event $S \subseteq R'$.

DP immunity to post-processing

Let us consider a deterministic function $f : R \to R'$.

Fix any pair of neighboring databases x, y with $\|x - y\|_1 \leq 1$, and fix any event $S \subseteq R'$.

Let $T = \{r \in R : f(r) \in S\}$. We then have:

$$\Pr[f(\mathcal{M}(x)) \in S] = \Pr[\mathcal{M}(x) \in T]$$

DP immunity to post-processing

Let us consider a deterministic function $f : R \to R'$.

Fix any pair of neighboring databases x, y with $\|x - y\|_1 \leq 1$, and fix any event $S \subseteq R'$.

Let $T = \{r \in R : f(r) \in S\}$. We then have:

$$\begin{aligned} \Pr[f(\mathcal{M}(x)) \in S] &= \Pr[\mathcal{M}(x) \in T] \\ &\leq \exp(\epsilon) \Pr[\mathcal{M}(y) \in T] + \delta \end{aligned}$$

DP immunity to post-processing

Let us consider a deterministic function $f : R \to R'$.

Fix any pair of neighboring databases x, y with $\|x - y\|_1 \leq 1$, and fix any event $S \subseteq R'$.

Let $T = \{r \in R : f(r) \in S\}$. We then have:

$$\begin{aligned} \Pr[f(\mathcal{M}(x)) \in S] &= \Pr[\mathcal{M}(x) \in T] \\ &\leq \exp(\epsilon) \Pr[\mathcal{M}(y) \in T] + \delta \\ &= \exp(\epsilon) \Pr[f(\mathcal{M}(y)) \in S] + \delta \end{aligned}$$

which was what we wanted.

DP composition

DP composition

$(\varepsilon, 0)$-differential privacy composes in a straightforward way: the composition of two $(\varepsilon, 0)$ differentially private mechanisms is $(2\varepsilon, 0)$-differentially private.

DP composition

DP composition

$(\varepsilon, 0)$-differential privacy composes in a straightforward way: the composition of two $(\varepsilon, 0)$ differentially private mechanisms is $(2\varepsilon, 0)$-differentially private.
More generally, "the epsilons and the deltas add up": the composition of k differentially private mechanisms, where the i th mechanism is $(\varepsilon_i, \delta_i)$-differentially private, for $1 \leq i \leq k$, is $(\sum_i \varepsilon_i, \sum_i \delta_i)$ differentially private.

Composition theorems

DP composition

Let $\mathcal{M}_1 : \mathbb{N}^{|\mathcal{X}|} \to \mathcal{R}_1$ be an ε_1-differentially private algorithm, and let $\mathcal{M}_2 : \mathbb{N}^{|\mathcal{X}|} \to \mathcal{R}_2$ be an ε_2-differentially private algorithm. Then their combination, defined to be $\mathcal{M}_{1,2} : \mathbb{N}^{|\mathcal{X}|} \to \mathcal{R}_1 \times \mathcal{R}_2$ by the mapping: $\mathcal{M}_{1,2}(x) = (\mathcal{M}_1(x), \mathcal{M}_2(x))$ is $\varepsilon_1 + \varepsilon_2$-differentially private.

Composition theorems

Let $x, y \in \mathbb{N}^{|\mathcal{X}|}$ be such that $\|x - y\|_1 \leq 1$. Fix any $(r_1, r_2) \in \mathcal{R}_1 \times \mathcal{R}_2$. Then:

$$\frac{\Pr\left[\mathcal{M}_{1,2}(x) = (r_1, r_2)\right]}{\Pr\left[\mathcal{M}_{1,2}(y) = (r_1, r_2)\right]}$$

Composition theorems

Let $x, y \in \mathbb{N}^{|\mathcal{X}|}$ be such that $\|x - y\|_1 \leq 1$. Fix any $(r_1, r_2) \in \mathcal{R}_1 \times \mathcal{R}_2$. Then:

$$\frac{\Pr\left[\mathcal{M}_{1,2}(x) = (r_1, r_2)\right]}{\Pr\left[\mathcal{M}_{1,2}(y) = (r_1, r_2)\right]} = \frac{\Pr\left[\mathcal{M}_1(x) = r_1\right] \Pr\left[\mathcal{M}_2(x) = r_2\right]}{\Pr\left[\mathcal{M}_1(y) = r_1\right] \Pr\left[\mathcal{M}_2(y) = r_2\right]}$$

Composition theorems

Let $x, y \in \mathbb{N}^{|\mathcal{X}|}$ be such that $\|x - y\|_1 \leq 1$. Fix any $(r_1, r_2) \in \mathcal{R}_1 \times \mathcal{R}_2$. Then:

$$\frac{\Pr\left[\mathcal{M}_{1,2}(x) = (r_1, r_2)\right]}{\Pr\left[\mathcal{M}_{1,2}(y) = (r_1, r_2)\right]} = \frac{\Pr\left[\mathcal{M}_1(x) = r_1\right]\Pr\left[\mathcal{M}_2(x) = r_2\right]}{\Pr\left[\mathcal{M}_1(y) = r_1\right]\Pr\left[\mathcal{M}_2(y) = r_2\right]}$$

$$= \left(\frac{\Pr\left[\mathcal{M}_1(x) = r_1\right]}{\Pr\left[\mathcal{M}_1(y) = r_1\right]}\right)\left(\frac{\Pr\left[\mathcal{M}_2(x) = r_1\right]}{\Pr\left[\mathcal{M}_2(y) = r_1\right]}\right)$$

Composition theorems

Let $x, y \in \mathbb{N}^{|\mathcal{X}|}$ be such that $\|x - y\|_1 \leq 1$. Fix any $(r_1, r_2) \in \mathcal{R}_1 \times \mathcal{R}_2$. Then:

$$\begin{aligned}
\frac{\Pr\left[\mathcal{M}_{1,2}(x) = (r_1, r_2)\right]}{\Pr\left[\mathcal{M}_{1,2}(y) = (r_1, r_2)\right]} &= \frac{\Pr\left[\mathcal{M}_1(x) = r_1\right]\Pr\left[\mathcal{M}_2(x) = r_2\right]}{\Pr\left[\mathcal{M}_1(y) = r_1\right]\Pr\left[\mathcal{M}_2(y) = r_2\right]} \\
&= \left(\frac{\Pr\left[\mathcal{M}_1(x) = r_1\right]}{\Pr\left[\mathcal{M}_1(y) = r_1\right]}\right)\left(\frac{\Pr\left[\mathcal{M}_2(x) = r_1\right]}{\Pr\left[\mathcal{M}_2(y) = r_1\right]}\right) \\
&\leq \exp\left(\varepsilon_1\right)\exp\left(\varepsilon_2\right)
\end{aligned}$$

Composition theorems

Let $x, y \in \mathbb{N}^{|\mathcal{X}|}$ be such that $\|x - y\|_1 \leq 1$. Fix any $(r_1, r_2) \in \mathcal{R}_1 \times \mathcal{R}_2$. Then:

$$\begin{aligned}
\frac{\Pr\left[\mathcal{M}_{1,2}(x) = (r_1, r_2)\right]}{\Pr\left[\mathcal{M}_{1,2}(y) = (r_1, r_2)\right]} &= \frac{\Pr\left[\mathcal{M}_1(x) = r_1\right]\Pr\left[\mathcal{M}_2(x) = r_2\right]}{\Pr\left[\mathcal{M}_1(y) = r_1\right]\Pr\left[\mathcal{M}_2(y) = r_2\right]} \\
&= \left(\frac{\Pr\left[\mathcal{M}_1(x) = r_1\right]}{\Pr\left[\mathcal{M}_1(y) = r_1\right]}\right)\left(\frac{\Pr\left[\mathcal{M}_2(x) = r_1\right]}{\Pr\left[\mathcal{M}_2(y) = r_1\right]}\right) \\
&\leq \exp\left(\varepsilon_1\right)\exp\left(\varepsilon_2\right) \\
&= \exp\left(\varepsilon_1 + \varepsilon_2\right)
\end{aligned}$$

Group privacy

$(\varepsilon, 0)$-differentially private mechanism $\mathcal{M}$ is $(k\varepsilon, 0)$ differentially private for groups of size k. That is, for all $\|x - y\|_1 \leq k$ and all $\mathcal{S} \subseteq \text{Range}(\mathcal{M})$

Group privacy

$$\Pr[\mathcal{M}(x) \in \mathcal{S}] \leq \exp(k\varepsilon) \Pr[\mathcal{M}(y) \in \mathcal{S}]$$

where the probability space is over the coin flips of the mechanism $\mathcal{M}$.

Group privacy

As the group gets larger, the privacy guarantee deteriorates, and this is what we want.

Group privacy

As the group gets larger, the privacy guarantee deteriorates, and this is what we want.

E.g.: if replace an entire surveyed population, say, of patients suffering from a disease, with a completely different group of respondents, say, healthy teenagers, we should get different answers to queries about the fraction of respondents who regularly run three miles each day.

Group privacy

As the group gets larger, the privacy guarantee deteriorates, and this is what we want.

E.g.: if replace an entire surveyed population, say, of patients suffering from a disease, with a completely different group of respondents, say, healthy teenagers, we should get different answers to queries about the fraction of respondents who regularly run three miles each day.

Although something similar holds for (ε, δ) differential privacy, the approximation term δ takes a big hit, and we only obtain $\left(k\varepsilon, ke^{(k-1)\varepsilon}\delta\right)$-differential privacy for groups of size k.

Counting queries

Counting queries are queries of the form "How many elements in the database satisfy Property P?"

Counting queries

Counting queries are queries of the form "How many elements in the database satisfy Property P?"

Counting is an extremely powerful primitive. It captures everything learnable in the statistical queries learning model, as well as many standard data mining tasks and basic statistics. Since the sensitivity of a counting query is 1 (the addition or deletion of a single individual can change a count by at most 1).

Histogram Queries

In the special (but common) case in which the queries are structurally disjoint we can do much better we don't necessarily have to let the noise scale with the number of queries. n example is the histogram query. In this type of query the universe $\mathbb{N}^{|\mathcal{X}|}$ is partitioned into cells, and the query asks how many database elements lie in each of the cells.

Sensitivity

The ℓ_1-sensitivity of a function $f : \mathbb{N}^{|\mathcal{X}|} \rightarrow \mathbb{R}^k$ is:

$$\Delta f = \max_{\substack{x,y \in |\mathcal{X}| \\ \|x-y\|_1 = 1}} \|f(x) - f(y)\|_1$$

Sensitivity

The ℓ_1-sensitivity of a function $f : \mathbb{N}^{|\mathcal{X}|} \rightarrow \mathbb{R}^k$ is:

$$\Delta f = \max_{\substack{x,y \in |\mathcal{X}| \\ \|x-y\|_1 = 1}} \|f(x) - f(y)\|_1$$

The ℓ_1 sensitivity of a function f captures the magnitude by which a single individual's data can change the function f in the worst case, and therefore, intuitively, the uncertainty in the response that we must introduce in order to hide the participation of a single individual.

Sensitivity

The ℓ_1-sensitivity of a function $f : \mathbb{N}^{|\mathcal{X}|} \to \mathbb{R}^k$ is:

$$\Delta f = \max_{\substack{x,y \in |\mathcal{X}| \\ \|x-y\|_1 = 1}} \|f(x) - f(y)\|_1$$

The ℓ_1 sensitivity of a function f captures the magnitude by which a single individual's data can change the function f in the worst case, and therefore, intuitively, the uncertainty in the response that we must introduce in order to hide the participation of a single individual.

The sensitivity of a function gives an upper bound on how much we must perturb its output to preserve privacy. One noise distribution naturally lends itself to differential privacy.

Laplace Mechanism

The Laplace Distribution. The Laplace Distribution (centered at 0) with scale b is the distribution with probability density function:

$$\mathrm{Lap}^{b}(x) = \frac{1}{2b} \exp\left(\frac{|x|}{b}\right)$$

The variance of this distribution is $\sigma^2 = 2b^2$. The Laplace distribution is a symmetric version of the exponential distribution.

Laplace Mechanism

Laplace mechanism

The Laplace mechanism computes f, and perturb each coordinate with noise drawn from the Laplace distribution. The scale of the noise will be calibrated to the sensitivity of f (divided by ε)[a].
Given any function $f : \mathbb{N}^{|\mathcal{X}|} \to \mathbb{R}^k$, the Laplace mechanism is defined as:

$$\mathcal{M}_L(x, f(\cdot), \varepsilon) = f(x) + (Y_1, \ldots, Y_k)$$

where Y_i are i.i.d. random variables drawn from $\text{Lap}(\Delta f/\varepsilon)$.

[a] Alternately, using Gaussian noise with variance calibrated to $\Delta f \ln(1/\delta)/\varepsilon$, one can achieve (ε, δ)-differential privacy. Use of the Laplace mechanism is cleaner.

Laplace Mechanism and its properties

Theorem 3.6. The Laplace mechanism preserves $(\varepsilon, 0)$-differential privacy.

Let $x \in \mathbb{N}^{|\mathcal{X}|}$ and $y \in \mathbb{N}^{|\mathcal{X}|}$ be such that $\|x - y\|_1 \leq 1$, and let $f(\cdot)$ be some function $f : \mathbb{N}^{|\mathcal{X}|} \rightarrow \mathbb{R}^k$. Let p_x denote the probability density function of $\mathcal{M}_L(x, f, \varepsilon)$, and let p_y denote the probability density function of $\mathcal{M}_L(y, f, \varepsilon)$. We compare the two at some arbitrary point $z \in \mathbb{R}^k$.

Laplace Mechanism and its properties

$$\frac{p_x(z)}{p_y(z)} = \prod_{i=1}^{k} \left(\frac{\exp\left(-\frac{\varepsilon |f(x)_i - z_i|}{\Delta f}\right)}{\exp\left(-\frac{\varepsilon |f(y)_i - z_i|}{\Delta f}\right)} \right)$$

Laplace Mechanism and its properties

$$\begin{aligned}
\frac{p_x(z)}{p_y(z)} &= \prod_{i=1}^{k} \left(\frac{\exp\left(-\frac{\varepsilon|f(x)_i - z_i|}{\Delta f}\right)}{\exp\left(-\frac{\varepsilon|f(y)_i - z_i|}{\Delta f}\right)} \right) \\
&= \prod_{i=1}^{k} \exp\left(\frac{\varepsilon\left(|f(y)_i - z_i| - |f(x)_i - z_i|\right)}{\Delta f} \right)
\end{aligned}$$

Laplace Mechanism and its properties

$$
\begin{aligned}
\frac{p_x(z)}{p_y(z)} &= \prod_{i=1}^{k} \left(\frac{\exp\left(-\frac{\varepsilon|f(x)_i - z_i|}{\Delta f}\right)}{\exp\left(-\frac{\varepsilon|f(y)_i - z_i|}{\Delta f}\right)} \right) \\
&= \prod_{i=1}^{k} \exp\left(\frac{\varepsilon\left(|f(y)_i - z_i| - |f(x)_i - z_i|\right)}{\Delta f} \right) \\
&\leq \prod_{i=1}^{k} \exp\left(\frac{\varepsilon\left|f(x)_i - f(y)_i\right|}{\Delta f} \right)
\end{aligned}
$$

Laplace Mechanism and its properties

$$
\begin{aligned}
\frac{p_x(z)}{p_y(z)} &= \prod_{i=1}^{k} \left(\frac{\exp\left(-\frac{\varepsilon |f(x)_i - z_i|}{\Delta f}\right)}{\exp\left(-\frac{\varepsilon |f(y)_i - z_i|}{\Delta f}\right)} \right) \\
&= \prod_{i=1}^{k} \exp\left(\frac{\varepsilon \left(|f(y)_i - z_i| - |f(x)_i - z_i|\right)}{\Delta f} \right) \\
&\leq \prod_{i=1}^{k} \exp\left(\frac{\varepsilon \left|f(x)_i - f(y)_i\right|}{\Delta f} \right) \\
&= \exp\left(\frac{\varepsilon \cdot \|f(x) - f(y)\|_1}{\Delta f} \right)
\end{aligned}
$$

Laplace Mechanism and its properties

$$
\begin{aligned}
\frac{p_x(z)}{p_y(z)} &= \prod_{i=1}^{k} \left(\frac{\exp\left(-\frac{\varepsilon |f(x)_i - z_i|}{\Delta f}\right)}{\exp\left(-\frac{\varepsilon |f(y)_i - z_i|}{\Delta f}\right)} \right) \\
&= \prod_{i=1}^{k} \exp\left(\frac{\varepsilon \left(|f(y)_i - z_i| - |f(x)_i - z_i| \right)}{\Delta f} \right) \\
&\leq \prod_{i=1}^{k} \exp\left(\frac{\varepsilon \, |f(x)_i - f(y)_i|}{\Delta f} \right) \\
&= \exp\left(\frac{\varepsilon \cdot \|f(x) - f(y)\|_1}{\Delta f} \right) \\
&\leq \exp(\varepsilon)
\end{aligned}
$$

Laplace Mechanism and its properties

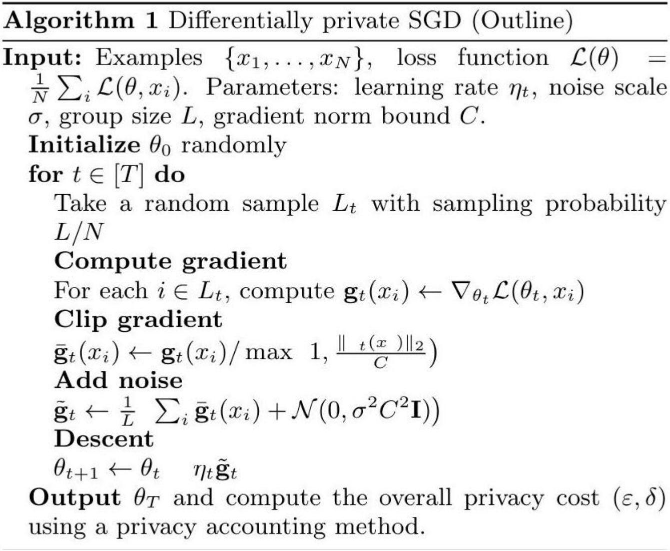
$(\varepsilon, 0)$ differential privacy can be achieved for counting queries by the addition of noise scaled to $1/\varepsilon$, that is, by adding noise drawn from $\mathrm{Lap}(1/\varepsilon)$. The expected distortion, or error, is $1/\varepsilon$, independent of the size of the database.

Laplace Mechanism and its properties

$(\varepsilon, 0)$ differential privacy can be achieved for counting queries by the addition of noise scaled to $1/\varepsilon$, that is, by adding noise drawn from $\mathrm{Lap}(1/\varepsilon)$. The expected distortion, or error, is $1/\varepsilon$, independent of the size of the database.

For histogram queries, since the cells are disjoint, the addition or removal of a single database element can affect the count in exactly one cell, and the difference to that cell is bounded by 1, so histogram queries have sensitivity 1 and can be answered by adding independent draws from $\mathrm{Lap}(1/\varepsilon)$ to the true count in each cell.

DP and Deep Learning



```
Algorithm 1 Differentially private SGD (Outline)
Input: Examples $\{x_1, \dots, x_N\}$, loss function $\mathcal{L}(\theta) = \frac{1}{N}\sum_i \mathcal{L}(\theta, x_i)$. Parameters: learning rate $\eta_t$, noise scale $\sigma$, group size $L$, gradient norm bound $C$.
  Initialize $\theta_0$ randomly
  for $t \in [T]$ do
    Take a random sample $L_t$ with sampling probability $L/N$
    Compute gradient
    For each $i \in L_t$, compute $\mathbf{g}_t(x_i) \leftarrow \nabla_{\theta_t}\mathcal{L}(\theta_t, x_i)$
    Clip gradient
    $\bar{\mathbf{g}}_t(x_i) \leftarrow \mathbf{g}_t(x_i) / \max\ 1, \frac{\| \ _t(x \ )\|_2}{C})$
    Add noise
    $\tilde{\mathbf{g}}_t \leftarrow \frac{1}{L}\ \sum_i \bar{\mathbf{g}}_t(x_i) + \mathcal{N}(0, \sigma^2 C^2 \mathbf{I}))$
    Descent
    $\theta_{t+1} \leftarrow \theta_t \quad \eta_t \tilde{\mathbf{g}}_t$
  Output $\theta_T$ and compute the overall privacy cost $(\varepsilon, \delta)$ using a privacy accounting method.
```

DP and Deep Learning

Norm clipping: clipping the gradient bounds the amount of information in a given update, which lets us reason about the maximum privacy loss.

DP and Deep Learning

Per-layer and time-dependent parameters: for multi-layer neural networks, we consider each layer separately, which allows setting different clipping thresholds C and noise scales σ for different layers. Additionally, the clipping and noise parameters may vary with the number of training steps t.

DP and Deep Learning

Lots: Like the ordinary SGD algorithm, Algorithm 1 estimates the gradient of $\mathcal{L}$ by computing the gradient of the loss on a group of examples and taking the average. This average provides an unbiased estimator, the variance of which decreases quickly with the size of the group. We call such a group a lot, to distinguish it from the computational grouping that is commonly called a batch. In order to limit memory consumption, we may set the batch size much smaller than the lot size L, which is a parameter of the algorithm. We perform the computation in batches, then group several batches into a lot for adding noise. In practice, for efficiency, the construction of batches and lots is done by randomly permuting the examples and then partitioning them into groups of the appropriate sizes. For ease of analysis, however, we assume that each lot is formed by independently picking each example with probability $q = L/N$, where N is the size of the input dataset.