Lecture Notes: Differential Privacy

1 Reconstruction Attack

Differential Privacy (DP) emerged from the observation that releasing overly precise answers to numerous queries about a dataset could allow adversaries to reconstruct large portions of the sensitive data. This phenomenon is known as a *reconstruction attack*. Formally:

Definition 1 (Reconstruction Attack)**.** *A reconstruction attack is a method where an attacker, given aggregate statistical query responses (possibly noisy), infers most or all individual entries of the private dataset.*

1.1 Motivational Example

Consider the following scenario. A company offers health insurance to employees, and periodically queries an insurer:

> *"How many employees born on [specific date], living in [specific ZIP], suffer from [specific stigmatizing condition]?"*

If the answer is "1," and the company knows that only one employee fits the birthdate and ZIP, the company immediately learns private health information about that employee.

One might try to prevent this by suppressing small counts, for example, answering "fewer than 5" instead of giving exact small numbers. However, consider these two queries:

1. How many employees joined before a certain date and have the condition?
2. How many employees joined before the next day and have the condition?

If answers differ by exactly 1, the company again reveals sensitive information. These simple *difference attacks* illustrate how aggregate statistics can leak information about individuals.

1.2 Formal Setup for Reconstruction

Let us formalize the reconstruction attack more rigorously. Consider a dataset represented as a binary vector $D \in \{0,1\}^n$, where $D = (d_1, d_2, \ldots, d_n)$ and each entry $d_i \in \{0,1\}$ denotes whether an individual has a sensitive attribute or not.

An attacker issues counting queries q_S defined on subsets $S \subseteq \{1, \ldots, n\}$:

$$q_S(D) = \sum_{i \in S} d_i.$$

Suppose the curator provides noisy answers $a_1, a_2, \ldots, a_m$ to queries $q_{S_1}, \ldots, q_{S_m}$ respectively. If each answer has an error bounded by E, meaning $|a_j - q_{S_j}(D)| \leq E$ for all j, then the attacker can reconstruct most of the dataset D under certain conditions. Note that the noisy answer can be a random answer, but also can be adversarially chosen (make the problem as hard as possible).

Given noisy answers to queries, a general reconstruction attack is the following:

$$\hat{D} = \arg\min_D \max_{1 \leq j \leq m} |q_{S_j}(D) - a_j| \tag{1}$$

which has an exponential complexity. Later, we will see more efficient attack.

Claim 1. *If every query is answered to within error $\leq \alpha n$ for some $\alpha < 1$, i.e.,*

$$\max_{S \subset \{1,\ldots,n\}} |q_S(D) - a_S| \leq \alpha n, \tag{2}$$

then the reconstruction attack returns $\hat{D}$ such that

$$\max_{S \subset \{1,\ldots,n\}} |q_S(\hat{D}) - a_S| \leq \alpha n. \tag{3}$$

This statement is trivial, since the reconstruction attack returns the best option among all possible $\hat{D}$ including the true D.

Our first nontrivial theorem is the following.

Theorem 1 (Dinur-Nissim Reconstruction Attack [4])**.** *If every queries are answered and all queries have error at most αn, then the reconstruction error is at most $4\alpha n$. I.e.,*

$$\sum_{i=1}^{n} |D_i - \hat{D}_i| \leq 4\alpha n. \tag{4}$$

Proof. For the output $\hat{D}$, define two sets

$$A_{01} =\{i : \hat{D}_i = 0, D_i = 1\} \tag{5}$$

$$A_{10} =\{i : \hat{D}_i = 1, D_i = 0\}. \tag{6}$$

If $\hat{D}$ and D disagree on more than $4\alpha n$ bits, then at least one of these two sets has size larger than $2\alpha n$. Without loss of generality, this set is A_{10}, i.e., $|A_{10}| > 2\alpha n$. Then, we have

$$|q_{A_{10}}(D) - q_{A_{10}}(\hat{D})| = |A_{10}| > 2\alpha n. \tag{7}$$

On the other hand,

$$\begin{aligned} |q_{A_{10}}(D) - q_{A_{10}}(\hat{D})| \leq& |q_{A_{10}}(D) - a_{A_{10}}| + |a_{A_{10}} - q_{A_{10}}(\hat{D})| && (8) \\ \leq& 2\alpha n, && (9) \end{aligned}$$

which is a contradiction. □

Note that the attacker does not need to know the set A_{10} since it has (noisy) answers for all possible queries. The following Corollary shows the impact of the above theorem.

Corollary 1. *If all queries are answered accuractely within an error* $\pm\frac{1}{100}n$*, then we can recover* D *correctly for at least* $\frac{96}{100}n$ *users.*

However, the assumption is unrealistic since the number of possible queries is exponentially large. Now, consider the more realistic setup. Suppose the attacker now choose $m = 20n$ queries $S_1, \dots, S_m$ uniform at random. We use the same reconstruction as before. The attacker will receive an answers $a_1, \dots, a_m$ with the guarantee

$$\max_j |q(S_j) - a_j| \leq \alpha n. \tag{10}$$

Note that even with smaller number of query, the reconstruction attack requires an exhaustive search which takes exponential complexity. Again, we will see more effective reconstruction method later.

The following theorem, shows that highly accurate answer can be achieved even with much polynomial number of queries, which is much smaller than expontially many queries.

Theorem 2 (Dinur-Nissim Reconstruction Attack [4])**.** *If all* m *queries have error at most* αn*, then with high probability, the reconstruction error is at most* $256\alpha^2 n^2$*. This bound is meaningful if* $\alpha \ll 1/\sqrt{n}$*.*

The choice of 20 and 256 is somewhat arbitrary.

Proof. Let the set

$$B = \{\hat{D} : \ \hat{D} \text{ and } D \text{ disagree on at least } 256\alpha^2 n^2 \text{ coordinates}\}. \tag{11}$$

For j, if $|q_{S_j}(D) - q_{S_j}(D')| \geq 4\alpha n$, then D' cannot be the output of the reconstruction attack because

$$|q_{S_j}(D') - a_j| \geq |q_{S_j}(D') - q_{S_j}(D)| - |q_{S_j}(D) - a_j| \geq 3\alpha n. \tag{12}$$

We will show that B cannot contain the output of the reconstruction attack $\hat{D}$.
Suppose $D_0 \in B$ and define $Z = D - D_0 \in \{-1, 0, 1\}^n$. Then, at least $256\alpha^2 n^2$ entries of Z are nonzero. Since the queries are chosen uniformly at random, we have

$$\Pr[|q_{S_j}(D) - q_{S_j}(D_0)| \leq 4\alpha n] \leq \frac{9}{10}. \tag{13}$$

Exercise 1. *Show the above inequality.*

Since the queries are chosen independently, we have

$$\Pr[\forall j, \quad |q_{S_j}(D) - q_{S_j}(D_0)| \leq 4\alpha n] \leq \left(\frac{9}{10}\right)^m \leq 2^{-2n}. \tag{14}$$

Finally, there are only 2^n possible length-n binary vectors,

$$\Pr[\exists D_0 \in B, \forall j, \quad |q_{S_j}(D) - q_{S_j}(D_0)| \leq 4\alpha n] \leq 2^{-2n} 2^n = 2^{-n}. \tag{15}$$

Thus, for probability of at least 2^{-n}, the set B is an empty set. □

Now, consider a more efficient attack, which can be obtained by relaxing the original reconstruction scheme to all vectors $D \in [0, 1]^n$ where $[0, 1] = \{x : 0 \leq x \leq 1\}$. I.e.,

$$\tilde{D} = \arg\min_{D \in [0,1]^n} \max_j |q_{S_j}(D) - a_j| \tag{16}$$

then round each entry to 0 or 1 to obtain a vector $\hat{D} \in \{0, 1\}^n$.

Remark 1. *A linear program with d variables an dm constraints asks us to minimize a linear objective function over $\mathbb{R}^d$ subject to m linear inequality constraints. Specifically, given an objective, represented by a vector $c \in \mathbb{R}^d$,* *and m constraints, each represented by a vector $a_i \in \mathbb{R}^d$ and a scalar $b_i \in \mathbb{R}$, a linear program can be written as*

$$\max_{x\in\mathbb{R}^d} \quad c \cdot x \tag{17}$$

$$s.t. \quad a_i \cdot x \leq b_i \quad \textit{for all } 1 \leq i \leq m. \tag{18}$$

Linear programs itself is a very interesting research topics in the field of optimization, but beyond the scope of this course. Here, we only use the fact that linear programs can be solved in polynomial time, and there are many efficient solvers in practice.

Exercise 2. *Show that the Equation (16) is indeed a linear program.*

1.3 Implications and Limitations

Reconstruction attacks demonstrate the fundamental limitation of releasing "too accurate" aggregate statistics. These attacks set inherent boundaries on the achievable privacy versus accuracy trade-off, motivating rigorous frameworks like differential privacy to formally quantify privacy loss.

The results in above theorems underline that careful noise calibration is essential; otherwise, even mildly inaccurate statistics can breach privacy dramatically.

2 Randomized Response

Consider an n users with a sensitive bit $D_i \in \{0, 1\}$. The individual reports a randomized response Y_i defined as

$$Y_i = \begin{cases} D_i & \text{with probability } \frac{1}{2} + \gamma \\ 1 - D_i & \text{with probability } \frac{1}{2} - \gamma, . \end{cases} \tag{19}$$

where is a parameter satisfying $0 < \gamma \leq \frac{1}{2}$. You can think Y_i is privatized version of D_i. If $\gamma = 1/2$, then there is no privacy since it is releasing the true sensitive bits. If $\gamma = 0$, then we have a perfect privacy, but the information is completely lost.

It is clear that

$$\mathbb{E}[Y_i] = D_i(1/2 + \gamma) + (1 - D_i)(1/2 - \gamma) \tag{20}$$

$$= 2\gamma D_i + 1/2 - \gamma. \tag{21}$$

Using the above relation, one can guess $\hat{D}_i$ based on the privatized bit Y_i:

$$\hat{D}_i = \frac{Y_i - 1/2 + \gamma}{2\gamma}. \tag{22}$$

Suppose the curator wants to release the fraction

$$p = \frac{1}{n}\sum_{i=1}^{n} D_i \tag{23}$$

to public in private manner. Then, instead of using the true private bits, it can use the estimated version of it ($\hat{D}_i$):

$$\hat{p} = \frac{1}{n}\sum_{i=1}^{n} D_i = \frac{1}{n}\sum_{i=1} \frac{Y_i - 1/2 + \gamma}{2\gamma}. \tag{24}$$

Note that $\hat{p}$ is unbiased estimate of p:

$$\mathbb{E}[\hat{p}] =\mathbb{E}\left[\frac{1}{n}\sum_{i=1} \frac{Y_i - 1/2 + \gamma}{2\gamma}\right] = \frac{1}{n}\sum_{i=1}^{n} D_i = p. \tag{25}$$

Thus, on average $\hat{p}$ is equal to the true fraction p. Now, let's focus on the error. First, the naive measure of error is a variance.

$$\begin{aligned} \mathrm{Var}[\hat{p}] =& \mathbb{E}[(\hat{p}-p)^2] && (26)\\ =& \frac{1}{n^2}\sum_{i=1}^{n} \mathrm{Var}\left[\frac{Y_i - 1/2 + \gamma}{2\gamma}\right] && (27)\\ =& \frac{1}{4n^2\gamma^2}\sum_{i=1}^{n} \mathrm{Var}\left[Y_i\right] && (28)\\ \leq& \frac{1}{16n\gamma^2}. && (29) \end{aligned}$$

Using Chebyshev's inequality, we can bound the error in probabilisitic sense.

Lemma 1 (Chebyshev's inequality). *Let X be a random variable with $\mathbb{E}[X] = \mu$ and $Var(X) = \sigma^2$, then*

$$Pr[|X-\mu| > k\sigma] \leq \frac{1}{k^2} \tag{30}$$

for all $k > 0$.

In our case, we have

$$\Pr[|\hat{p} - p| > k/4\gamma\sqrt{n}] \leq \frac{1}{k^2} \tag{31}$$

which implies that $\hat{p}$ is within $p \pm k/4\gamma\sqrt{n}$ with high probability.

If one wants a specific criteria to meet, such as

$$\Pr[|\hat{p} - p| > \alpha] \leq \beta \tag{32}$$

then, we can set $\alpha = k/4\gamma\sqrt{n}$ and $\beta = 1/k^2$. Thus,

$$n = \frac{1}{16\gamma^2\beta\alpha^2}. \tag{33}$$

This implies that

- If we want small α, which implies $\hat{p}$ is closer to p, then we have to pay with high n.
- If we want small β, which implies $\hat{p}$ is close to p with higher probability, then we have to pay with high n.
- If we want small γ, which implies more privacy, then we have to pay with high n.

3 Differential Privacy

Differential Privacy (DP) provides a formal privacy guarantee ensuring that the output of a computation does not significantly depend on any one individual's data. Informally, if an algorithm is differentially private, then whether or not a particular person's data is included in the input, the distribution of the algorithm's outputs remains nearly the same. Thus an adversary observing the output gains very little information about any specific individual's data.

We first define the notion of *neighboring datasets*. Typically, two datasets D and D' are said to be neighbors if they differ in the data of a single individual. For example, D and D' are the same except the i-th element: $D = (X_1, \dots, X_i, \dots, X_n)$ while $D = (X_1, \dots, X_i', \dots, X_n)$. In a database setting, one can think of D' as derived from D by changing one record's values (the specific definition can vary, but it should capture a minimal change in one person's data).

If the curator wants to release a statistics $f(D)$ in private manner. In other words, the curator wants to employ $\mathcal{M}$ so that $\mathcal{M}(D) \approx f(D)$ while $\mathcal{M}(D)$ is protecting individual private informations. More precisely, we will use the notion of differential privacy to define privacy mathematically and quantify the level of privacy.

Definition 2 (ϵ-Differential Privacy [5])**.** *A randomized mechanism $\mathcal{M} : \mathcal{X}^n \to \mathcal{Y}$ (algorithm) operating on datasets is ϵ-differentially private if for all pairs of neighboring datasets D and D', and for all subsets of outputs $O \subseteq Range(\mathcal{M})$, we have:*

$$\Pr[\mathcal{M}(D) \in O] \ \le \ e^{\epsilon} \ \Pr[\mathcal{M}(D') \in O].$$

One can think a small ϵ so that $e^{\epsilon} \approx 1$, which implies that two probabilities $\Pr[\mathcal{M}(D) \in O]$ $\Pr[\mathcal{M}(D') \in O]$ are similar. Since the above inequality holds for any D and D', it automatically implies that

$$e^{-\epsilon} \le \frac{\Pr[\mathcal{M}(D) \in O]}{\Pr[\mathcal{M}(D') \in O]} \le e^{\epsilon}.$$

Example 1. *Consider the case $f(D) = \frac{1}{n}\sum_{i=1}^{n} X_i$. If $\mathcal{M}(D) = f(D)$, then*

$$\mathcal{M}(D') = f(D') = \frac{X_1 + \cdots + X_i' + \cdots + X_n}{n} = f(X) + \frac{X_i' - X_i}{n}. \tag{34}$$

If we know both $f(D)$ and $f(D')$, then we will infer some meaningful information about X_i and X_i'. Furthermore, we will know whether $f(D)$ is from D or D'. Any deterministic $\mathcal{M}$ will have the same issue, and we need **randomization**.

Remark 2. *This definition is from Dwork, Mcsherry, Nissim and Smith (2006). They won 2016 Test-of-time award at TCC and 2017 Gödel Prize.*

Some important notes on differential privacy:

- $\epsilon = 0$ means perfect privacy, and smaller ϵ implies more privacy.
- We consider the worst-case guarantee (the inequality holds for "any" neighboring pair D and D')
- We consider the multiplicative factor. We will discuss later that the additive factor alone is not interesting.
- Using e^{ϵ} instead of ϵ is due to notational simplicity (see composition). Also, $e^{\epsilon} \approx 1 + \epsilon$

- $\mathcal{M}$ should be random (any deterministic mechanism cannot be differentially private.
- Differential privacy (DP) is simply the notion of similarity between distribution of $\mathcal{M}(D)$ and $\mathcal{M}(D')$. Thus, there are alternative definition of differential privacy using another measure distance between probability distributions.
- We define the neighboring dataset by "replacing" operation. One can define the neighboring dataset using "insertion" or "deletion". However, these two definition is somewhat equivalent since "replacing" is simply a composition of "deleting" and "inserting". We will see later this in composition.

Remark 3. *One way to interpret DP is that an adversary who sees the output of $\mathcal{M}$ cannot confidently tell whether any particular individual's data was used or not, up to an odds ratio of e^{ϵ} (and with probability δ of a larger deviation).*

Remark 4. *Another interpretation in terms of hypothesis testing: for any individual i, consider the hypothesis H_0 that i's data was included in D versus H_1 that i's data was replaced (or removed) to obtain D'. Differential privacy guarantees that any test performed on the output of $\mathcal{M}$ cannot distinguish H_0 vs H_1, the log-likelihood ratio is bounded by ϵ [10]. In simple terms, the presence or absence (or change) of a single individual's data has minimal effect on the distribution of outputs.*

For those of who are not familiar with hypothesis testing, here is an example.

Example 2. *Consider a simple Covid test which measure the body temperature. Suppose the body temperature of Covid patient has distribution of $Lap(39, b)$, where non Covid patient has distribution of $Lap(38, b)$. Given the body temperature of patient T, you have decide whether H_0 is true (the patient does not have Covid) or H_1 is true (the patient has Covid). The best test is check whether $T < 38.5$ or not. However, there are two types of error:*

- *False Positive (FP): H_0 is true (patient does not have Covid) but incorrectly indiciating H_1 is true.*
- *False Negative (FN): H_1 is true (patient has Covid) but incorrectly indiciating H_0 is true.*

The goal of designing a good test is to have small FP probability and small FN probability. However, there is a trade off (obviously).

Suppose one wants to guess whether dataset is D or D' based on observation of $\mathcal{M}(D)$. You can set a criteria O so that you want to declare H_0 is true if $\mathcal{M}(D) \in O$, and decalre H_1 is true if $\mathcal{M}(D) \in O$. Let the probability of false positive is q while the probability of false negative is p. Then, the differential privacy implies that for any test criteria O, we have

$$e^{-\epsilon} \leq \frac{p}{1-q} \tag{35}$$

$$e^{-\epsilon} \leq \frac{q}{1-p} \tag{36}$$

We can rewrite the above inequality

$$1 = q + 1 - q \leq q + pe^{\epsilon} \tag{37}$$

$$1 = p + 1 - p \leq p + qe^{\epsilon}. \tag{38}$$

It is clear that p and q cannot be small at the same time.

Our first differentially private mechanism is a randomized response. For $D = (D_1, \ldots, D_n)$, suppose $f(D) = (D_1, \ldots, D_n)$, and $\mathcal{M}(D) = (Y_1, \ldots, Y_n)$ where Y_i is flipped version of D_i with probability of $1/2 - \gamma$. For $D' = (D_1, \ldots, D'_i, \ldots, D_n)$, we have

$$\frac{\Pr\left[\mathcal{M}(D) = (y_1, \ldots, y_n)\right]}{\Pr\left[\mathcal{M}(D') = (y_1, \ldots, y_n)\right]} = \frac{\Pr\left[Y_i = y_i\right]}{\Pr\left[Y'_i = y_i\right]} \tag{39}$$

$$\leq \frac{1/2 + \gamma}{1/2 - \gamma} \leq e^{4\gamma}. \tag{40}$$

Thus, the randomized response is 4γ-DP.

Exercise 3. *Is the above argument enough? Where is O?*

Laplace Mechanism (for numeric queries). If $f(D)$ is a numeric query (or vector of queries) on dataset D, one can add Laplace noise proportional to the *sensitivity* of f. The sensitivity Δ_1 is the maximum change in f's output due to one individual's data change:

$$\Delta_1 = \max_{D, D' \text{ neighbors}} \left\| f(D) - f(D') \right\|_1 .$$

The Laplace mechanism [5] outputs

$$\mathcal{M}(D) = f(D) + (Y_1, \ldots, Y_k),$$

where Y_i are i.i.d. random variables drawn from Laplace$(0, \Delta_1/\epsilon)$ (the Laplace distribution with mean 0 and scale Δ_1/ϵ). Note that the Laplace distribution with parameter μ, b is

$$f_X(x) = \frac{1}{2b} \exp\left(-\frac{|x-\mu|}{b}\right) \tag{41}$$

This mechanism satisfies ϵ-DP. Intuitively, the noise magnitude $\sim \Delta_1/\epsilon$ is chosen such that a single individual's influence on $f(D)$ is hidden by the noise. Changing one record shifts $f(D)$ by at most Δ_1, and this change is overwhelmed by the Laplace noise in the sense required by Definition 2. The probability density functions for outputs under D vs D' are within a factor e^ϵ everywhere, ensuring the DP condition.

Proof. Let D and D' be neighboring datasets. For any output vector $z = (z_1, \ldots, z_k) \in \mathbb{R}^k$, the probability density functions of $M(D)$ and $M(D')$ are given by

$$f_{M(D)}(z^k) = \prod_{i=1}^{k} \frac{1}{2b} \exp\left(-\frac{|z_i - f(D)_i|}{b}\right) \tag{42}$$

$$f_{M(D')}(z^k) = \prod_{i=1}^{k} \frac{1}{2b} \exp\left(-\frac{|z_i - f(D')_i|}{b}\right). \tag{43}$$

Thus, the ratio of the densities is

$$\frac{f_{M(D)}(z^k)}{f_{M(D')}(z^k)} = \prod_{i=1}^{k} \exp\left(-\frac{|z_i - f(D)_i| - |z_i - f(D')_i|}{b}\right) \tag{44}$$

$$\leq \prod_{i=1}^{k} \exp\left(\frac{|f(D)_i - f(D')_i|}{b}\right) \tag{45}$$

$$= \exp\left(\sum_{i=1}^{k} \frac{|f(D)_i - f(D')_i|}{b}\right) \tag{46}$$

$$= \exp\left(\frac{\|f(D)_i - f(D')_i\|_1}{b}\right) \tag{47}$$

$$\leq \exp\left(\frac{\Delta_1}{b}\right) \tag{48}$$

$$= e^\epsilon. \tag{49}$$

Finally, For any $T \subset \mathbb{R}^k$, we have

$$\frac{\Pr\left[M(D) \in T\right]}{\Pr\left[M(D') \in T\right]} = \frac{\int_{z^k \in T} f_{M(D)}(z^k)\, dz^k}{\int_{z^k \in T} f_{M(D')}(z^k)\, dz^k} \tag{50}$$

$$\leq e^{\epsilon}. \tag{51}$$

□

Remark 5. *Note that the variance of Y_i is $2b^2 = 2\Delta_1^2/\epsilon^2$. Large Δ_1 implies more sensitive f (leaking more information), which requires larger noise variance to protect private information. Also, small ϵ implies more strict privacy constraint, which also requires larger noise variance to protect private information.*

Example 3. *Suppose $f(D) = \frac{1}{n}\sum_{i=1}^{n} X_i$ where $X_i \in [a, b]$ for all i. Then, the sensitivity is $\Delta_1 = \frac{b-a}{n}$.*

4 Properties of Differential Privacy

Differential privacy has several key properties that make it an appealing and robust privacy definition. We outline some fundamental properties:

4.1 Composition Theorems

When we run multiple differentially private mechanisms on the same dataset, the privacy losses accumulate. DP provides theoretical guarantees for how privacy degrades under composition of multiple analyses. There are basic composition results as well as more advanced analyses that give tighter bounds which we will see later in this course.

Suppose we have two algorithms $\mathcal{M}_1 : \mathcal{X}^n \to \mathcal{Y}_1$ and $\mathcal{M}_2 : \mathcal{X}^n \times \mathcal{Y}_1 \to \mathcal{Y}_2$. We allow the second mechanism be "adaptive" which means it can depends on the output of $\mathcal{M}_1$. The composition theorem guarantees that the $\mathcal{M}_2 \circ \mathcal{M}_1$ is $\epsilon_1 + \epsilon_2$-DP.

Theorem 3. *Suppose $\mathcal{M}_1 : \mathcal{X}^n \to \mathcal{Y}_1$ is ϵ_1-DP and $\mathcal{M}_2 : \mathcal{X}^n \times \mathcal{Y}_1 \to \mathcal{Y}_2$ is ϵ_2-DP. Then, releasing both $\mathcal{M}_1$ and $\mathcal{M}_2 \circ \mathcal{M}_1$ is at most $\epsilon_1 + \epsilon_2$-DP.*

Proof. For neighboring dataset D and D', we have

$$\frac{\Pr\left[(\mathcal{M}_1(D), \mathcal{M}_2(D, \mathcal{M}_1(D))) = (a_1, a_2)\right]}{\Pr\left[(\mathcal{M}_1(D'), \mathcal{M}_2(D', \mathcal{M}_1(D'))) = (a_1, a_2)\right]} \tag{52}$$

$$= \frac{\Pr\left[\mathcal{M}_1(D) = a_1\right] \Pr\left[\mathcal{M}_2(D, \mathcal{M}_1(D)) = a_2 | \mathcal{M}_1(D) = a_1\right]}{\Pr\left[(\mathcal{M}_1(D') = a_1\right] \Pr\left[\mathcal{M}_2(D', \mathcal{M}_1(D'))) = a_2 | \mathcal{M}_1(D') = a_1\right]} \tag{53}$$

$$= \frac{\Pr\left[(\mathcal{M}_1(D) = a_1\right]}{\Pr\left[(\mathcal{M}_1(D') = a_1\right]} \frac{\Pr\left[\mathcal{M}_2(D, a_1) = a_2\right]}{\Pr\left[\mathcal{M}_2(D', a_1)) = a_2\right]} \tag{54}$$

$$\leq e^{\epsilon_1} e^{\epsilon_2} \tag{55}$$

□

More generally, for k mechanisms $\mathcal{M}_1, \ldots, \mathcal{M}_k$ where $\mathcal{M}_i$ is ϵ_i-DP, the sequence of all outputs is $\sum_i \epsilon_i$-DP. This worst-case bound follows from the multiplicative property of the ϵ factor. In particular, if k independent mechanisms each satisfy ϵ-DP, then k-fold composition satisfies $\epsilon_{\text{tot}} = k\epsilon$ (so ϵ scales linearly with k) [6].

4.2 Post-Processing Immunity

Differential privacy is immune to post-processing, meaning that no additional computations on the output can degrade the privacy guarantee.

Theorem 4 (Post-Processing [6]). *If $\mathcal{M} : D \to O$ is an ϵ-DP mechanism and $f : O \to O'$ is an arbitrary (possibly randomized) function, then the composed mechanism $f(\mathcal{M}(D))$ is also ϵ-DP.*

Proof. The intuition is that since $\mathcal{M}$'s output is already private, any further transformation f cannot introduce new dependence on any single individual's data. Formally, for neighboring D, D' and any subset of outputs $S \subseteq O'$, consider

$$\Pr[f(\mathcal{M}(D)) \in S] = \Pr[\mathcal{M}(D) \in f^{-1}(S)]\,.$$

Note that

$$f^{-1}(S) = \{o \in O : f(o) \in S\}. \tag{56}$$

Because $\mathcal{M}$ is ϵ-DP, this is at most

$$e^{\epsilon} \Pr[\mathcal{M}(D') \in f^{-1}(S)] = e^{\epsilon} \Pr[f(\mathcal{M}(D')) \in S]\,.$$

Thus $f(\mathcal{M}(D))$ satisfies the same DP inequality. In simple terms: whatever an adversary could infer from $f(\mathcal{M}(D))$, they could also infer (up to the DP bounds) from $\mathcal{M}(D)$ alone. □

Post-processing immunity implies that one can publish a DP-sanitized result and allow any downstream analysis on it without worrying about further privacy loss. This is critical: as long as the initial result is differentially private, even an adversary with arbitrary side computations or auxiliary data cannot worsen the privacy.

4.3 Group Privacy and Amplification by Subsampling

Differential privacy as defined protects a single individual's data. If we consider a group of individuals (of size t) all opting in or out of a dataset, the privacy guarantee scales roughly linearly with t.

Theorem 5 (Group Privacy [6])**.** *If $\mathcal{M}$ is ϵ-DP, then for any group of t individuals, $\mathcal{M}$ satisfies $t\epsilon$-DP with respect to changes in that entire group.*

This is conceptually straightforward: changing t people can be seen as t single-person changes in sequence, and by composition (or a direct coupling argument) the bound becomes $e^{t\epsilon}$ on likelihood ratio and $t\delta$ additive slack. Group privacy implies that if someone is concerned about a group of, say, 5 of their records (maybe the same person appears multiple times in the data), then the guarantee would degrade by that factor 5. Typically ϵ is kept small enough that even a moderate t yields an acceptable $t\epsilon$.

Privacy Amplification by Subsampling. An important positive property is that if a DP mechanism is applied to a random subset of the data (as opposed to the whole dataset), the effective privacy loss can be smaller. In many algorithms, we only use each individual's data with some probability (for example, in stochastic gradient descent, each mini-batch might include any given record with probability q). It turns out that this random selection provides additional privacy protection.

A simple form of this result: Suppose $\mathcal{M}$ is an ϵ-DP mechanism for a single individual's data. Now construct a new mechanism that first samples each individual with probability (rougly) q and applies $\mathcal{M}$ to the subsampled dataset (and outputs $\mathcal{M}$'s result). Then this overall procedure is ϵ'-DP for the full dataset, where $\epsilon' < \epsilon$. Specifically, one can show

$$\epsilon' = \ln(1 + q(e^{\epsilon} - 1)),$$

which for small q is roughly $q\epsilon$.

Here is a formal description. Suppose we have an algorithm $M : \mathcal{X}^m \to \mathcal{Y}$ which is (ϵ, δ)-DP. Consider the following algorithm $M' : \mathcal{X}^n \to \mathcal{Y}$, where

$n > m$. When run on an input $X \in \mathcal{X}^n$, it chooses a random subset of the input $X' \in \mathcal{X}^m$ of size m, and outputs $M(X')$. Let $q = m/n$, then M' is ϵ'-DP.

Proof. Let $X, X' \in \mathcal{X}^n$ be neighboring dataset. Without loss of generality, assume that $X = (x, x_2, x_3, ..., x_n)$ while $X' = (x', x_2, x_3, \dots, x_n)$ Let $I = (I_1, \dots, I_m)$ be a random vector of size m uniformly sampled (without replacement) from $\{1, \dots, n\}$. and sampling function $s(X; I) = (x_{I_1}, \dots, x_{I_m})$. Then,

$$\begin{aligned}\Pr[M'(X) \in T] =& \Pr[1 \in I) Pr(M(s(X;I)) \in T | 1 \in I] \\ &+ \Pr[1 \notin I) Pr(M(s(X;I)) \in T | 1 \notin I]\end{aligned}$$

Note that $\Pr[1 \in I] = m/n = q$. Due to ϵ-DP, we have

$$\begin{aligned}\Pr[M(s(X;I)) \in T | 1 \in I] \leq& e^\epsilon \Pr\left[M(s(X';I)) \in T | 1 \in I\right] \\ \Pr[M(s(X;I)) \in T | 1 \in I] \leq& e^\epsilon \Pr[M(s(X;I)) \in T | 1 \notin I] \\ \Pr[M(s(X;I)) \in T | 1 \notin I] =& \Pr[M(s(X';I)) \in T | 1 \notin I].\end{aligned}$$

Note that the first inequality and the last equality are trivial. For the second inequality, we can find a mapping between all subsets in $1 \in I$ to $1 \notin I$ so that the corresponding subsets are neighboring each other.

Exercise 4. *Note that the number of subsets in $1 \in I$ and the number of subsets in $1 \notin I$ are not the same. Prove the second inequality formally.*

Let $\alpha = q + (1-q)e^{-\epsilon} < 1$, then using the second properties of the above, we have

$$\begin{aligned}\Pr[M(s(X;I)) \in T | 1 \in I] \leq& \alpha e^\epsilon \Pr\left[M(s(X';I)) \in T | 1 \in I\right] \\ &+ (1-\alpha) e^\epsilon \Pr[M(s(X;I)) \in T | 1 \notin I]\end{aligned}$$

Finally,

$$\begin{aligned}\Pr[M'(X) \in T] =& \alpha q e^\epsilon \Pr\left[M(s(X';I)) \in T | 1 \in I\right] \\ &+ (1 - q + (1-\alpha)e^\epsilon) \Pr\left[M(s(X';I)) \in T | 1 \notin I\right] \\ =& (1 + q(e^\epsilon - 1)) \left\{ q \Pr\left[M(s(X';I)) \in T | 1 \in I\right] \right. \\ & \left. + (1-q) \Pr\left[M(s(X';I)) \in T | 1 \notin I\right] \right\} \\ =& (1 + q(e^\epsilon - 1)) \Pr\left[M'(X') \in T\right].\end{aligned}$$

□

This means that given a 1-DP algorithm, it can always be converted to a ϵ-DP algorithm by just expanding the dataset size by a factor of $O(1/\epsilon)$. In summary, using only a random sample of users for each computation can “amplify” privacy. This is highly useful in large-scale distributed or federated computations where each round only involves a subset of devices or data points.

4.4 Differentially Private k-means

k-Means clustering is a classical algorithm used to partition n data points into k clusters by minimizing within-cluster variance. The standard k-Means algorithm is as follows.

Algorithm 1 k-Means Clustering

```
Require: Dataset $X = \{x_1, x_2, \dots, x_n\} \subset \mathbb{R}^d$, number of clusters $k$
Ensure: Cluster assignments and centroids $\{\mu_1, \dots, \mu_k\}$
Initialize $k$ cluster centers $\mu_1, \dots, \mu_k$ randomly
for $t = 1$ to $T$ do
    for each cluster $j = 1$ to $k$ do
        Update the Cluster
            $$C_j \leftarrow \{x_i \in X : \|x_i - \mu_j\|^2 \leq \min_{m \in \{1,\dots,k\}} \|x_i - \mu_m\|^2\}$$
    end for
    for each cluster $j = 1$ to $k$ do
        Update the centroid:
            $$\mu_j \leftarrow \frac{1}{|C_j|} \sum_{x_i \in C_j} x_i$$
    end for
end for
```

The algorithm minimizes the total within-cluster variance:

$$\mathcal{L} = \sum_{j=1}^{k} \sum_{x_i \in C_j} \|x_i - \mu_j\|^2$$

We aim for ϵ-differential privacy: the algorithm’s output should not significantly differ when a single data point in X is changed.

We present a simplified algorithm based on the "Private k-Means via Output Perturbation" method described in [9, 7]. Let $\mathcal{U} = \{x \in \mathbb{R}^d : \|x\|_1 \leq 1\}$ and $\epsilon' = \epsilon/2T$.

Algorithm 2 k-Means Clustering

```
Require: Dataset $X = \{x_1, x_2, \ldots, x_n\} \subset \mathcal{U}^n$, number of clusters $k$
Ensure: Cluster assignments and centroids $\{\mu_1, \ldots, \mu_k\}$
Initialize $k$ cluster centers $\mu_1, \ldots, \mu_k$ randomly
for $t = 1$ to $T$ do
    for each cluster $j = 1$ to $k$ do
        Update the Cluster
            $$C_j \leftarrow \{x_i \in X : \|x_i - \mu_j\|^2 \leq \min_{m \in \{1,\ldots,k\}} \|x_i - \mu_m\|^2\}$$
    end for
    for each cluster $j = 1$ to $k$ do
        $n_j \leftarrow |C_j| + Y$ where $Y \sim \mathrm{Lap}(2/\epsilon')$
        $a_j \leftarrow \left(\sum_{x_i \in C_j} x_i\right) + Z^d$ where $Z^d \sim \mathrm{Lap}(2/\epsilon')$ i.i.d.
        Update the centroid:
            $$\mu_j \leftarrow \begin{cases} a_j/n_j & \text{if } n_j \geq 1 \\ \text{uniform in } \mathcal{U} & \text{if } n_j < 1 \end{cases}$$
    end for
end for
```

Theorem 6. *The above algorithm is ϵ-DP.*

Proof. In each iteration,

- Privatizing the sum of elements in each cluster:

$$\left(\sum_{x_i \in C_1} x_i, \ldots, \sum_{x_i \in C_k} x_i\right) \to (a_1, \ldots, a_k) \tag{57}$$

  has sensitivity 2 and is ϵ'-DP.

- Privatizing the number of elements in each cluster:

$$(|C_1|, \ldots, |C_k|) \to (n_1, \ldots, n_k) \tag{58}$$

  has sensitivity 2 and is ϵ'-DP.

Using a basic composition theorem, the overall algorithm is $T \times 2\epsilon' = \epsilon$-DP. □

Remark 6. - *Sensitivity estimation often uses assumptions about bounded data domains (e.g., all x_i in $[0,1]^d$).*

- *For approximate DP, we can use Gaussian mechanism.*

4.5 Additive Differential Privacy (Not Recommended)

So far, we discussed the differential privacy under multiplicative factor e^ϵ. You may wonder "can we define the similar notion of privacy under additive factor," i.e.,

$$\Pr[\mathcal{M}(D) \in O] \ \leq \ \Pr[\mathcal{M}(D') \in O] + \delta \,.$$

Indeed, the above definition satisfies some desirable properties, including composition, postprocessing, and group privacy. However, it is not so useful due to the following example. Consider a following mechanisim: For dataset $X = (X_1, \ldots, X_n)$, the mechanism outputs the whole set $Y = X$ with probability of δ, and release nothing with probability of $1 - \delta$.

Exercise 5. *Show that the above mechanism satisfies the δ-additive DP.*

This requires that the δ should be small. However, as we discussed in group privacy of ϵ-DP, the parameter δ should be larger than $1/n$ for any useful applications. Then, the question would be $\delta > 1/n$ (but still small) would be okay.

Consider the following mechanism for dataset $X = (X_1, \ldots, X_n)$. The mechanism outputs $Y = (Y_1, \ldots, Y_n)$ where

$$Y_i = \begin{cases} X_i & \text{w.p. } \delta \\ \perp & \text{w.p. } 1 - \delta \end{cases} \tag{59}$$

The dummy symbol $\perp$ represents "no information."

Exercise 6. *Show that the above mechanism satisfies the δ-additive DP.*

Note that the probability of leaking at least one private information is

$$1 - (1 - \delta)^n, \tag{60}$$

which is non-negligible in the case of $\delta > 1/n$.

4.6 Selection Problem: Exponential Mechanism

Not all queries are numeric; sometimes we wish to select an output from a discrete set in a privacy-preserving way. The Exponential Mechanism [6] is a general method to choose an output with probability biased towards high-utility results while preserving differential privacy (DP).

We have two motivaional examples.

Example 4 (Voting)**.** *Suppose there are d candidates, and n users. Each user can vote to multiple candidates, i.e., i-th user voted to $X_i \subset \{1, \dots, d\}$. The score of candidate j is*

$$q(j; D) = |\{i : j \in X_i\}|. \tag{61}$$

Without DP, we would pick $\arg\max_j q(j; D)$*, but we want to privatize this mechanism.*

Example 5 (Pricing)**.** *Suppose you want to decide the price p of the product. There are n users where i-th user is willing to buy a product if $p \leq X_i$. For price p, the revenue would be*

$$q(p; D) = p \times |\{i : p \leq X_i\}|. \tag{62}$$

Without DP, we would pick $\arg\max_j q(j; D)$*, but we want to privatize this mechanism. In this case, p is a real variable, and you may want to add a Laplacian noise. But consider the following case where $X_1 = X_2 = \cdots = X_n = 7$, where the optimal price is $p^\star = 7$. If the noisy price is larger than 7, then the revenue would be 0. In this case, we can discretize the mechanism and convert the problem to pick $\hat{p}$ from the price candidates $\{1, 2, ..., p_{\max}\}$. One can define the finer candidate set.*

In both cases, the problem is simply a "selection problem" where the usual Laplace mechanism may not applicable. In this selection problem,

Suppose we have an arbitrary set of possible outputs $\mathcal{Y}$. We define a utility (or score) function $q(y; D)$ that evaluates the quality of output y with respect to dataset D. We require that changing one individual's data changes the utility by at most Δ: This ensures that no single data point can significantly affect the output distribution.

The Exponential Mechanism $\mathcal{M}_{\mathrm{EM}}$ selects an output $y \in \mathcal{Y}$ with probability proportional to $\exp\Bigl(\frac{\epsilon\, q(D,o)}{2\,\Delta}\Bigr)$. Formally:

$$\Pr[\mathcal{M}_{\mathrm{EM}}(D) = y] \propto \exp\Bigl(\frac{\epsilon\, q(D, y)}{2\,\Delta}\Bigr).$$

This mechanism satisfies ϵ-DP.

Intuitively, outputs with higher utility on the true data are exponentially more likely, but the presence of the normalization by Δ and scaling by ϵ ensures that a single individual's effect on the utilities does not skew the probabilities too much.

Utility Guarantee. One of the most important features of the Exponential Mechanism is its utility guarantee: with high probability, the chosen output y has utility close to the maximum possible utility. Specifically, we have

$$\Pr\left[q(D,Y) < q_{\max} - \frac{2\Delta(\log d + t)}{\epsilon}\right] \leq \exp(-t).$$

Proof. Let the set of bad outputs:

$$B_t = \left\{q(D,y) < q_{\max} - \frac{2\Delta(\log d + t)}{\epsilon}\right\}$$

$$\begin{aligned}
\Pr\left[Y \in B_t\right] \leq & \frac{\Pr\left[Y \in B_t\right]}{\Pr\left[Y = y^\star\right]} && (63)\\
= & \frac{\sum_{y \in B_t} \Pr\left[Y = y\right]}{\Pr\left[Y = y^\star\right]} && (64)\\
= & \frac{\sum_{y \in B_t} \exp\left(\frac{\epsilon}{2\Delta} q(y; D)\right)}{\exp\left(\frac{\epsilon}{2\Delta} q_{\max}\right)} && (65)\\
= & \sum_{y \in B_t} \exp\left(\frac{\epsilon}{2\Delta}(q(y; D) - q_{\max})\right) && (66)\\
\leq & \sum_{y \in B_t} \exp\left(-(\log d + t)\right) && (67)\\
= & |B_t| \exp\left(-(\log d + t)\right) && (68)\\
\leq & \exp(-t). && (69)
\end{aligned}$$

where the last inequality is due to $|B_t| \leq d$. □

We can rewrite the above inequality by

$$\Pr\left[q_{\max} - q(D,Y) > \alpha\right] \leq \beta \qquad (70)$$

where $\alpha = (2\Delta/\epsilon)(\log d + t)$ is a threshold and $\beta = e^{-t}$ is a small probability. This shows that the utility loss grows logarithmically with the size of the output space $|\mathcal{Y}|$ and with $1/\beta$. Thus, the Exponential Mechanism is especially suited to large but finite domains where directly adding noise to the output (e.g., via Laplace or Gaussian mechanisms) is not feasible.

We have another utility guarantee in expectation.

$$\mathbb{E}\left[q_{\max} - q(D, Y)\right] \leq \frac{2\Delta}{\epsilon}(\log d + 1).$$

Proof. Let $Z = \frac{\epsilon}{2\Delta}(q_{\max} - q(D; Y))$, then with reparameterization of $z = \log d + t$, we have

$$\begin{aligned}
\mathbb{E}\left[Z\right] &= \int_0^\infty \Pr\left[Z > z\right] dz && (71)\\
&= \int_{-\log d}^\infty \Pr\left[Z > \log d + t\right] dt && (72)\\
&\leq \log d + \int_0^\infty \Pr\left[Z > \log d + t\right] dt && (73)\\
&\leq \log d + \int_0^\infty e^{-t}\, dt && (74)\\
&= \log d + 1. && (75)
\end{aligned}$$

□

Example: Selecting the Most Frequent Item Privately. For the voting example, the sensitivity is $\Delta = 1$. Suppose we have $d = 100$ with the target of $e^{-t} = 0.01$ and $\epsilon = 0.5$, then

$$\frac{2\Delta}{\epsilon}(\log d + t) \approx 36.8 \tag{76}$$

In other words,

$$\Pr\left[q_{\max} - q(D; Y) > 36.8\right] \leq 0.01. \tag{77}$$

Thus, if the winner and the runner-up has difference more than 36.8, the winner (with most votes) will be chosen with probability larger than 99%.

4.7 Selection Problem: Noisy Max

The *Noisy Max* mechanism (also known as *Report Noisy Max*) is a simple yet powerful tool for differentially private selection. It is designed to privately solve argmax-style queries: given multiple candidate outcomes (or queries) with real-valued scores, Noisy Max identifies the highest-scoring outcome under the protection of differential privacy.

Motivating Example (Private Voting). Imagine an election with several candidates. Each candidate y has $q(y; D)$ votes in the true dataset D. The Noisy Max mechanism adds Exponential noise to each vote count and outputs the candidate with the highest noisy count. This prevents small changes in individual votes from drastically affecting the result.

Definition 3 (Noisy Max Mechanism)**.** *Let $\mathcal{Y}$ be a finite set of candidate outputs and let $q : \mathcal{X}^n \times \mathcal{Y} \to \mathbb{R}$ be a utility function with sensitivity Δ. The Noisy Max mechanism $\mathcal{M}_{NM}$ operates as follows:*

1. *For each $y \in Y$, compute $s_y := q(y; D) + Z_y$, where $Z_y \sim \mathrm{Exp}(0, 2\Delta/\epsilon)$.*
2. *Output $y^* := \arg\max_{y \in Y} s_y$.*

Theorem 7 (Differential Privacy of Noisy Max)**.** *The Noisy Max mechanism $\mathcal{M}_{NM}$ is ϵ-differentially private.*

Proof. Let X and X' be neighboring datasets and $y \in \mathcal{Y}$ be any output. Then,

$$\Pr\left[\mathcal{M}_{NM}(X) = y\right] \tag{78}$$

$$= \Pr\left[q(y; X) + Z_y \geq \max_{j \neq y} q(j; X) + Z_j\right] \tag{79}$$

$$= \int_{z^{d \setminus j}} \Pr\left[Z_y \geq \max_{j \neq y} q(j; X) - q(y; X) + z_j | Z^{d \setminus j} = z^{d \setminus j}\right] f_{Z^{d \setminus j}}(z^{d \setminus j})\, dz^{d \setminus j} \tag{80}$$

$$\leq \int_{z^{d \setminus j}} \Pr\left[Z_y \geq \max_{j \neq y} q(j; X') - q(y; X') - 2\Delta + z_j | Z^{d \setminus j} = z^{d \setminus j}\right] f_{Z^{d \setminus j}}(z^{d \setminus j})\, dz^{d \setminus j} \tag{81}$$

$$\leq e^{\epsilon} \int_{z^{d \setminus j}} \Pr\left[Z_y \geq \max_{j \neq y} q(j; X') - q(y; X') + z_j | Z^{d \setminus j} = z^{d \setminus j}\right] f_{Z^{d \setminus j}}(z^{d \setminus j})\, dz^{d \setminus j} \tag{82}$$

$$= e^{\epsilon} \Pr\left[\mathcal{M}_{NM}(X') = y\right]. \tag{83}$$

where $Z^{d \setminus j}$ is a $d-1$ dimensional vector $(Z_1, \ldots, Z_d)$ execept Z_j.

The key step is

$$\Pr\left[Z_y \geq K - 2\Delta\right] \leq e^{\epsilon} \Pr\left[Z_y \geq K\right] \tag{84}$$

for a constant K when $Z_y \sim \mathrm{Exp}(2\Delta/\epsilon)$. □

Utility Guarantee. Let $y^\star = \arg\max_y q(y; X)$ and let y be the output of $\mathcal{M}_{NM}$. Then for $\alpha \geq 0$,

$$\Pr\left[q_{\max} - q(Y;X) > \frac{2\Delta(\log d + t)}{\epsilon}\right] \leq e^{-t}.$$

Proof. Since $q(Y;X) + Z_Y \geq \max_j q(j;X) + Z_j$, we have

$$\begin{aligned}
&\Pr\left[q_{\max} - q(Y;X) > \tfrac{2\Delta(\log d+t)}{\epsilon}\right] && (85)\\
&= \Pr\left[q_{\max} > (q(Y;X) + Z_Y) - Z_Y + \frac{2\Delta(\log d + t)}{\epsilon}\right] && (86)\\
&\leq \Pr\left[q_{\max} > \max q(j;X) + Z_j - Z_Y + \frac{2\Delta(\log d + t)}{\epsilon}\right] && (87)\\
&\leq \Pr\left[Z_Y - Z_j > \frac{2\Delta(\log d + t)}{\epsilon} \quad \text{for some } j\right] && (88)\\
&\leq d \times \exp(-(\log d + t)) && (89)\\
&= e^{-t}. && (90)
\end{aligned}$$

□

We have the exact same expected utility bound as before, since the probabilistic utility guarantee is the same.

5 Approximate Differential Privacy

Definition 4 ((ϵ,δ)-Differential Privacy [5])**.** *A randomized mechanism $\mathcal{M}$ (algorithm) operating on datasets is (ϵ,δ)-differentially private if for all pairs of neighboring datasets D and D', and for all measurable subsets of outputs $O \subseteq \mathit{Range}(\mathcal{M})$, we have:*

$$\Pr[\mathcal{M}(D) \in O] \;\leq\; e^{\epsilon}\,\Pr[\mathcal{M}(D') \in O] + \delta\,.$$

If $\delta = 0$, we say $\mathcal{M}$ satisfies ϵ-DP (often called pure DP*). If $\delta > 0$ (typically very small, e.g. 10^{-6} or 10^{-9}), it is called* approximate DP*.*

The parameter $\epsilon \geq 0$ quantifies the *privacy loss*: smaller ϵ means stronger privacy (since e^{ϵ} will be closer to 1). The parameter δ allows a small probability of failure of the pure DP guarantee. In particular, with probability at most δ, the inequality may not hold, but typically δ is chosen to be negligible in the dataset size or in practical terms. Differential privacy is a worst-case guarantee, quantifying risk over all possible outputs and all possible neighboring dataset pairs.

Interpretations: One way to interpret DP is that an adversary who sees the output of $\mathcal{M}$ cannot confidently tell whether any particular individual's data was used or not, up to an odds ratio of e^{ϵ} (and with probability δ of a larger deviation). Another interpretation in terms of hypothesis testing: for any individual i, consider the hypothesis H_0 that i's data was included in D versus H_1 that i's data was replaced (or removed) to obtain D'. Differential privacy guarantees that any test performed on the output of $\mathcal{M}$ cannot distinguish H_0 vs H_1 with advantage much more than δ, and even in the best case, the log-likelihood ratio is bounded by ϵ [10]. In simple terms, the presence or absence (or change) of a single individual's data has minimal effect on the distribution of outputs.

5.1 Privacy Loss Random Variable

Let the privacy loss by

$$I_{X,X'}(y) = \log \frac{\Pr\left[\mathcal{M}(X) = y\right]}{\Pr\left[\mathcal{M}(X) = y\right]}. \tag{91}$$

Then, one can define the privacy loss random variable by

$$I_{X,X'}(Y) \tag{92}$$

where $Y = \mathcal{M}(X)$.

The typical way of proving the mechanism $\mathcal{M}$ is (ϵ, δ)-DP is first divide $\mathcal{Y} = G \subset B$ where

$$\begin{align}
G &= \{y \in \mathcal{Y} I_{X,X'}(y) \leq \epsilon\} \tag{93}\\
B &= \{y \in \mathcal{Y} I_{X,X'}(y) > \epsilon\} \tag{94}\\
& \tag{95}
\end{align}$$

Then, for $S \subset \mathcal{Y}$, we have

$$\begin{align}
\Pr\left[\mathcal{M}(Y) \in S\right] =& \Pr\left[\mathcal{M}(X) \in S \cap G\right] + \Pr\left[\mathcal{M}(Y) \in S \cap B\right] \tag{96}\\
\leq& e^{\epsilon} \Pr\left[\mathcal{M}(X') \in S \cap G\right] + \Pr\left[\mathcal{M}(X) \in B\right] \tag{97}\\
\leq& e^{\epsilon} \Pr\left[\mathcal{M}(X') \in S\right] + \Pr\left[\mathcal{M}(X) \in B\right]. \tag{98}
\end{align}$$

Thus, it is enough to show that

$$\Pr\left[\mathcal{M}(X) \in B\right] = \Pr\left[I_{X,X'}(Y) > \epsilon\right] \leq \delta. \tag{99}$$

5.2 Truncated Laplacian Mechanism

Several standard mechanisms satisfy differential privacy, for example, the following truncated Laplacian mechanism. Let the truncated Laplacian random variable $Z \sim \mathrm{LapTr}(\lambda, \tau)$ where the pdf is

$$f_Z(z) = \begin{cases} \frac{1}{Z} e^{-|z|/\tau} & \text{if } |z| \leq \tau \\ 0 & \text{otherwise} \end{cases} \tag{100}$$

Then, the truncated Laplacian mechanism is simply adding truncated Laplacian, i.e.,

$$\mathcal{M}(X) = f(X) + (Z_1, \ldots, Z_d) \tag{101}$$

where Z_i is i.i.d. truncated laplacian with parameter $(\Delta/\epsilon, \tau)$ with appropriate τ. Note that the utility of the mechanism is more or less the same as standard Laplacian mechanism, we can guarantee the worst-case bound on the magnitude of the noise.

5.3 Gaussian Mechanism

When we allow a small $\delta > 0$, using Gaussian (normal) noise is another popular approach. The Gaussian mechanism adds Gaussian noise with variance proportional to Δ_2^2. Specifically, to achieve (ϵ, δ)-DP for a function f with ℓ_2-sensitivity Δ_2 (the maximum ℓ_2 change in f between neighbors), the mechanism outputs

$$\mathcal{M}(D) = f(D) + \mathcal{N}(0, \sigma^2 I)\,,$$

where $\mathcal{N}(0, \sigma^2 I)$ is a multivariate Gaussian with covariance $\sigma^2 I$ (identity) and σ is chosen as $\sigma = \frac{\Delta_2 \sqrt{2 \ln(1.25/\delta)}}{\epsilon}$. This choice of σ ensures (ϵ, δ)-DP [6]. The Gaussian mechanism is often used when the desired δ is not zero or when dealing with high-dimensional or real-valued outputs where Gaussian noise may be more natural. It requires slightly more noise than Laplace for a given ϵ if δ is extremely small, but for moderate δ it can be effective.

These classical mechanisms can be used as building blocks for more complex differentially private analyses. We emphasize that ϵ and δ are parameters that one generally wants to keep as small as possible to ensure strong privacy, but smaller values mean more noise or less accuracy. The art of designing DP algorithms lies in achieving useful accuracy (utility) for given privacy parameters.

Proof. Let first show that the 1-dimensional case. First, define $K = f(X) - f(X') \leq \Delta$, then

$$
\begin{aligned}
I_{X,X'}(y) =& \log \frac{\Pr\left[\mathcal{M}(X) = y\right]}{\Pr\left[\mathcal{M}(X') = y\right]} && (102)\\
=& \log \frac{\exp(-(y - f(X))^2/2\sigma^2)}{\exp(-(y - f(X'))^2/2\sigma^2)} && (103)\\
=& \frac{(y - f(X'))^2 - (y - f(X))^2}{2\sigma^2} && (104)\\
=& \frac{(f(X) - f(X'))(f(X) - f(X') + 2(y - f(X)))}{2\sigma^2} && (105)\\
=& \frac{K(K + 2(y - f(X)))}{2\sigma^2} && (106)
\end{aligned}
$$

From $\Pr\left[\mathcal{N}(0,1) \geq t\right] \leq e^{-\frac{1}{2}t^2}$, we have

$$
\begin{aligned}
\Pr\left[I_{X,X'}(Y) > \epsilon\right] \leq& \Pr\left[\tfrac{K(K+2(Y-f(X)))}{2\sigma^2} > \epsilon\right] && (107)\\
=& \Pr\left[\tfrac{K(Y-f(X))}{\sigma^2} > \epsilon - \tfrac{K^2}{2\sigma^2}\right] && (108)\\
=& \Pr\left[\tfrac{Y-f(X)}{\sigma} > \tfrac{\epsilon\sigma}{K} - \tfrac{K}{2\sigma}\right] && (109)\\
\leq& \exp\left(-\frac{1}{2}\left(\frac{\epsilon\sigma}{K} - \frac{K}{2\sigma}\right)^2\right) && (110)\\
\leq& \exp\left(-\frac{1}{2}\left(\frac{\epsilon\sigma}{\Delta} - \frac{\Delta}{2\sigma}\right)^2\right) && (111)\\
\leq& \exp\left(-\frac{1}{2}\frac{\epsilon^2\sigma^2}{\Delta^2}\right) && (112)\\
\leq& \exp\log(\delta/1.25) && (113)\\
\leq& \delta. && (114)
\end{aligned}
$$

□

For a multidimensional Gaussian, we need the following lemma

Lemma 2. *Let $Z = (Z_1, \ldots, Z_d)$ be multivariate Gaussian with zero mean and covariance $\sigma^2 I$, then, for any $v \in \mathbb{R}^d$, $v \cdot Z$ is a zero mean Gaussian with variance $\sigma^2\|v\|^2$.*

Proof. The proof of the multi-dimensional case is essentially the same. Let v = f(X) - f(X'), *then*

$$\begin{aligned}
I_{X,X'}(y) =& \log \frac{\Pr\left[\mathcal{M}(X) = y\right]}{\Pr\left[\mathcal{M}(X') = y\right]} && (115)\\
=& \log \frac{\exp(-\|y - f(X)\|^2/2\sigma^2)}{\exp(-\|y - f(X')\|^2/2\sigma^2)} && (116)\\
=& \frac{\|y - f(X')\|^2 - \|y - f(X)\|^2}{2\sigma^2} && (117)\\
=& \frac{(f(X) - f(X')) \cdot (f(X) - f(X') + 2(y - f(X)))}{2\sigma^2} && (118)\\
=& \frac{\|v\|^2}{2\sigma^2} + \frac{2v \cdot (y - f(X))}{2\sigma^2} && (119)
\end{aligned}$$

Using the fact that $v \cdot (y - f(X))$ is Gaussian with zero mean and variance $\|v\|^2\sigma^2$, the proof remains the same. □

Remark 7. *The ℓ_2 sensitivity is normally much smaller than the ℓ_1 sensitivity. For example, consider the case of binary queries $f(X) \in \{0, 1\}^d$ (answering d binary queries). The ℓ_1 sensitivity is d since a single alteration may effect the answer of all d queries. However, the ℓ_1 sensitivity is $\sqrt{d}$, which is smaller than d. Thus, the noise proportional to ℓ_2 in Gaussian mechanism is beneficial in most practical setup.*

6 Properties of Approximate DP

Differential privacy has several key properties that make it an appealing and robust privacy definition. We outline some fundamental properties:

6.1 Post-Processing Immunity

Differential privacy is immune to post-processing, meaning that no additional computations on the output can degrade the privacy guarantee.

Theorem 8 (Post-Processing [6])**.** *If $\mathcal{M} : D \to O$ is an (ϵ, δ)-DP mechanism and $f : O \to O'$ is an arbitrary (possibly randomized) function, then the composed mechanism $f(\mathcal{M}(D))$ is also (ϵ, δ)-DP.*

Proof. The intuition is that since $\mathcal{M}$'s output is already private, any further transformation f cannot introduce new dependence on any single individual's data. Formally, for neighboring D, D' and any subset of outputs

$O'_{sub} \subseteq O'$, consider

$$\Pr[f(\mathcal{M}(D)) \in O'_{sub}] = \Pr[\mathcal{M}(D) \in f^{-1}(O'_{sub})] \,.$$

Because $\mathcal{M}$ is (ϵ, δ)-DP, this is at most

$$e^{\epsilon} \Pr[\mathcal{M}(D') \in f^{-1}(O'_{sub})] + \delta = e^{\epsilon} \Pr[f(\mathcal{M}(D')) \in O'_{sub}] + \delta \,.$$

Thus $f(\mathcal{M}(D))$ satisfies the same DP inequality. In simple terms: whatever an adversary could infer from $f(\mathcal{M}(D))$, they could also infer (up to the DP bounds) from $\mathcal{M}(D)$ alone. □

Post-processing immunity implies that one can publish a DP-sanitized result and allow any downstream analysis on it without worrying about further privacy loss. This is critical: as long as the initial result is differentially private, even an adversary with arbitrary side computations or auxiliary data cannot worsen the privacy.

6.2 Composition Theorems

When we run multiple differentially private mechanisms on the same dataset, the privacy losses accumulate. DP provides theoretical guarantees for how privacy degrades under composition of multiple analyses. There are basic composition results as well as more advanced analyses that give tighter bounds.

Basic Composition. If we have two algorithms $\mathcal{M}_1$ and $\mathcal{M}_2$ that are (ϵ_1, δ_1)-DP and (ϵ_2, δ_2)-DP respectively, then releasing the outputs of both (either together or sequentially) is at most $(\epsilon_1 + \epsilon_2, \delta_1 + \delta_2)$-DP. More generally, for k mechanisms $\mathcal{M}_1, \ldots, \mathcal{M}_k$ where $\mathcal{M}_i$ is (ϵ_i, δ_i)-DP, the sequence of all outputs is $(\sum_i \epsilon_i,\ \sum_i \delta_i)$-DP. This worst-case bound follows from a union bound on the δ probabilities and the multiplicative property of the ϵ factor. In particular, if k independent mechanisms each satisfy ϵ-DP (pure), k-fold composition satisfies $\epsilon_{\text{tot}} = k\epsilon$ (so ϵ scales linearly with k) [6].

6.3 Group Privacy and Amplification by Subsampling

Differential privacy as defined protects a single individual's data. If we consider a group of individuals (of size t) all opting in or out of a dataset, the privacy guarantee scales roughly linearly with t.

Theorem 9 (Group Privacy [6])**.** *If $\mathcal{M}$ is (ϵ, δ)-DP, then for any group of t individuals, $\mathcal{M}$ satisfies $(t\epsilon,\ t\delta)$-DP with respect to changes in that entire group.*

This is conceptually straightforward: changing t people can be seen as t single-person changes in sequence, and by composition (or a direct coupling argument) the bound becomes $e^{t\epsilon}$ on likelihood ratio and $t\delta$ additive slack. Group privacy implies that if someone is concerned about a group of, say, 5 of their records (maybe the same person appears multiple times in the data), then the guarantee would degrade by that factor 5. Typically ϵ is kept small enough that even a moderate t yields an acceptable $t\epsilon$.

Privacy Amplification by Subsampling. An important positive property is that if a DP mechanism is applied to a random subset of the data (as opposed to the whole dataset), the effective privacy loss can be smaller. In many algorithms, we only use each individual's data with some probability (for example, in stochastic gradient descent, each mini-batch might include any given record with probability q). It turns out that this random selection provides additional privacy protection.

A simple form of this result: Suppose $\mathcal{M}$ is an ϵ-DP mechanism for a single individual's data. Now construct a new mechanism that first samples each individual independently with probability q and applies $\mathcal{M}$ to the subsampled dataset (and outputs $\mathcal{M}$'s result). Then this overall procedure is (ϵ', δ')-DP for the full dataset, where $\epsilon' < \epsilon$ and δ' are smaller than ϵ would be without sampling. Specifically, for pure DP, one can show

$$\epsilon' = \ln(1 + q(e^{\epsilon} - 1))\,,$$

which for small q is roughly $q\epsilon$. And δ' can be taken as 0 in this setting (if $\mathcal{M}$ was pure DP). For approximate DP, a similar scaling occurs: roughly speaking ϵ scales by q and δ scales by q as well (or $q\delta +$ an additional term). Precise theorems have been developed in, e.g., [1], which use a moments analysis to get tight bounds.

In summary, using only a random sample of users for each computation can "amplify" privacy. This is highly useful in large-scale distributed or federated computations where each round only involves a subset of devices or data points.

7 DP Empirical Risk Minimization

7.1 Empirical Risk Minimization (ERM)

Let $X = ((x_1, y_1), \ldots, (X_n, y_n))$ be a dataset. Suppose we want a model $f_\theta : \mathcal{X} \to \hat{\mathcal{Y}}$ that $f_\theta(x) \approx y$. It is common to train the model by minimizing the empirical loss

$$L(\theta, X) = \frac{1}{n} \sum_{i=1}^{n} \ell(f_\theta(x_i), y_i) \tag{120}$$

for some loss function $\ell : \hat{\mathcal{Y}} \times \mathcal{Y} \to [0, \infty)$. The empirical risk is defined by

$$R(\theta) = L(\theta, X) - \min_\theta L(\theta, X) \tag{121}$$

the cost of suboptimality. Note that the population loss is

$$L(\theta, P_X) = \mathbb{E}\left[\ell(f_\theta(X), Y)\right] \tag{122}$$

when the true distribution $(X, Y) \sim P_X$ is known. The corresponding population risk is

$$R_p(\theta) = L_p(\theta, P_X) - \min_\theta L_p(\theta, P_X). \tag{123}$$

In practice, we do not know the true distribution, and the goal is often to minimize the empirical risk.

To provide a theoretical guarantee, we have some additional assumptions.

- Constraint: θ is in a feasible set $\mathcal{C}$, i.e., $\theta \in \mathcal{C}$. For example, we can have bounded norm constraint, i.e., $\|\theta\|_2^2 \leq C$.
- Bounded loss: $\ell(f_\theta(x), y) \in [0, \Delta]$ for all θ, x, y.
- Lipschitzness: $\|\nabla_\theta \ell(f_\theta(x), y)\| \leq G$. This implies that the loss function is not varying too fast so that it is "predictable."

7.2 Exponential Mechanism

One way to solve ERM is via the exponential mechanism. The idea is simply sampling θ according to the probability distribution

$$\Pr\left[\Theta = \theta\right] \propto \exp\left(-\frac{\epsilon n}{2\Delta} L(\theta; X)\right) \tag{124}$$

Then, the following theorem provides a utility guarantee.

Theorem 10. *Suppose $\mathcal{C}$ is the ℓ_2-ball of radius R in $\mathbb{R}^d$ and ℓ is G-Lipschitz. Then, for every dataset X with $\Delta = GR$, exponential mechanism returns $\hat{\theta}$ such that*

$$\mathbb{E}\left[L(\hat{\theta}, X)\right] \le \min_{\theta \in \mathcal{C}} L(\theta; X) + O\left(\frac{dGR\log(\epsilon n/d)}{\epsilon n}\right) \tag{125}$$

Proof. Define the set A_1 (the good one), and the set A_2 (the bad one). Fix a small r, and

$$A_1 = \{\hat{\theta} \in \mathcal{C} : L(\theta) \le L(\theta^\star) + rG\} \tag{126}$$

$$A_2 = \{\hat{\theta} \in \mathcal{C} : L(\theta) > L(\theta^\star) + rG + t\} \tag{127}$$

Then,

$$\begin{aligned}
\frac{\Pr\left[\hat{\theta} \in A_2\right]}{\Pr\left[\hat{\theta} \in A_1\right]} &\le \frac{\mathrm{Vol}(A_2)\exp\left(-\frac{\epsilon n}{2RG}(L(\theta^\star) + rG + t)\right)}{\mathrm{Vol}(A_1)\exp\left(-\frac{\epsilon n}{2RG}(L(\theta^\star) + rG)\right)} && (128)\\
&= \frac{\mathrm{Vol}(A_2)}{\mathrm{Vol}(A_1)}\exp\left(-\frac{\epsilon n t}{2RG}\right) && (129)\\
&\le \left(\frac{R}{r/2}\right)^d \exp\left(-\frac{\epsilon n t}{2RG}\right) && (130)
\end{aligned}$$

where $r/2$ is from the case where $\theta^\star$ lies in the boundary of the set $\mathcal{C}$. Even in such case, the G should contain the ball of radius $r/2$.

By setting $t = \frac{2RG}{\epsilon n}(d\log\frac{2R}{r} + \log 1/\beta)$,

$$\begin{aligned}
\Pr\left[\hat{\theta} \in A_2\right] &\le \exp\left(d\log\frac{2R}{r} - \frac{\epsilon n t}{2RG}\right) && (131)\\
&= \beta. && (132)
\end{aligned}$$

Finally, by having $r = 2R\frac{d}{n\epsilon}$

$$\begin{aligned}
\mathbb{E}\left[L(\hat{\theta}) - L(\theta^\star)\right] =& \int_0^{rG} \Pr\left[L(\hat{\theta}) - L(\theta^\star) \ge rG + t\right] dt\\
&+ \int_{rG}^{\infty} \Pr\left[L(\hat{\theta}) - L(\theta^\star) \ge rG + t\right] dt && (133)\\
\le& rG + \int_{rG}^{\infty} \left(\frac{R}{r/2}\right)^d \exp\left(-\frac{\epsilon n t}{2RG}\right) dt && (134)\\
=& O\left(\frac{dGR\log(\epsilon n/d)}{\epsilon n}\right) && (135)
\end{aligned}$$

□

7.3 Advanced Composition Theorem

In regular composition theorem, the basic linear growth of ϵ with k can be pessimistic, especially when $\delta > 0$ is allowed. A more nuanced analysis shows that we can get sub-linear growth of privacy loss for ϵ if we tolerate a little increase in δ. One common formulation is:

Theorem 11 (Advanced Composition [6])**.** *If $\mathcal{M}_1, \ldots, \mathcal{M}_k$ are each (ϵ, δ)-DP, then for any $\delta' > 0$, the k-fold composition of these mechanisms is $(\epsilon', k\delta + \delta')$-DP, where*

$$\epsilon' = \sqrt{2k \ln(1/\delta')}\,\epsilon + k\epsilon \cdot \frac{e^\epsilon - 1}{e^\epsilon + 1}\,.$$

In particular, for small ϵ we have $\epsilon' \approx \sqrt{2k \ln(1/\delta')}\,\epsilon$ (the dominant term).

The advanced composition theorem (proved via the moments accounting method or by group privacy arguments) essentially says that ϵ grows like $O(\sqrt{k})$ rather than $O(k)$ if we allow a modest increase in δ. If ϵ is small (say $\epsilon < 0.1$), $e^\epsilon - 1 \approx \epsilon$, so the second term $k\epsilon(e^\epsilon - 1) \approx k\epsilon^2$, which for small ϵ is much smaller than the first term. Thus a simpler bound is $\epsilon' \approx \epsilon\sqrt{2k \ln(1/\delta')}$.

The intuition is the following. If we have mechanisms $A_1, \ldots, A_k$ where each $A_k : \mathcal{X} \times \mathcal{Y}_1 \times \cdots \times \mathcal{Y}_{k-1} \to \mathcal{Y}_k$ satisfies (ϵ_k, δ_k)-DP. Then, for $A = (A_1, \ldots, A_k)$, we have

$$\log \frac{\Pr\left[A(X) = (y_1, \ldots, y_k)\right]}{\Pr\left[A(X') = (y_1, \ldots, y_k)\right]} = \sum_{i=1}^{k} \frac{\Pr\left[A_i(X) = y_i | A^{i-1} = y^{i-1}\right]}{\Pr\left[A_i(X') = y_i | A^{i-1} = y^{i-1}\right]}.$$

The proof of basic composition theorem bounds each term with ϵ_i by treating each term independently. However, all probability terms shares the conditioning on the same $(y_1, \ldots, y_k)$, we can obtain a stricter bound with $\sqrt{k}\epsilon$ instead of $k\epsilon$.

This has practical importance: it means we can perform many DP operations on the same data and still maintain a reasonable overall privacy level, as long as each individual ϵ is small and we account for the composition properly.

7.4 Differentially Private Stochastic Gradient Descent (DP-SGD)

Stochastic Gradient Descent (SGD) is a core algorithm in machine learning for optimizing model parameters by iteratively updating on random subsets of data. DP-SGD [1] is a variant of SGD that incorporates noise to ensure differential privacy of the training process. It has become a standard method to train deep learning models with privacy guarantees.

Let $\Pi_{\mathcal{C}}$ is a projection method to a set $\mathcal{C}$.

Algorithm 3 Gradient Descent (GD)

```
Require: Dataset $\mathcal{D}$, loss function $\ell(\theta; x)$, learning rate $\alpha$, number of steps $T$
Initialize model parameters $\theta_0$
for $t = 1$ to $T$ do
    Compute gradient: $g_t = \frac{1}{n}\sum_{i=1}^{n} \nabla_\theta \ell(\theta_{t-1}; x_i)$
    Update parameters: $\theta_t = \theta_{t-1} - \alpha g_t$
    Update parameters: $\theta_t \leftarrow \Pi_{\mathcal{C}}(\theta_{t-1})$
end for
return Final parameters $\theta_T$
```

Remark 8. *With appropriate choice of learning rate,*

- *GD converges to a stationary point.*
- *If the loss function is convex, GD converges to a global optimal point.*

More precisely, the following theorem provides a convergence guarantee.

Theorem 12. *Consider a loss function $\mathcal{L} : \mathcal{C} \to \mathbb{R}$ where $\mathcal{L}$ is G-Lipschitz and convex. Suppose $\mathcal{C}$ is closed and convex set with diameter R. Then, for $\alpha = \frac{R}{G\sqrt{T}}$, the θ_T from gradient descent satisfies*

$$\mathcal{L}(\theta_T) \leq \mathcal{L}(\theta^\star) + \frac{RG}{\sqrt{T}}. \tag{136}$$

Remark 9.

- *The convexity of $\mathcal{L}$ means*

$$\mathcal{L}(\lambda\theta_1 + (1-\lambda)\theta_2) \leq \lambda\mathcal{L}(\theta_1) + (1-\lambda)\mathcal{L}(\theta_2) \tag{137}$$

for all $0 \leq \lambda \leq 1$ and $\theta_1, \theta_2 \in \mathcal{C}$.

- *The convexity of the set $\mathcal{C}$ means*

$$\lambda\theta_1 + (1-\lambda)\theta_2 \in \mathcal{C} \tag{138}$$

for all $0 \leq \lambda \leq 1$ and $\theta_1, \theta_2 \in \mathcal{C}$.

- *The diameter R means that*

$$\max_{\theta_1,\theta_2\in\mathcal{C}} \|\theta_1 - \theta_2\| \leq R. \tag{139}$$

However, computing gradients $\nabla\ell(\theta, x_i)$ for all x_i's are often impractical. One alternative is to use a sampling method.

Algorithm 4 Stochastic Gradient Descent (SGD)

```
Require: Dataset $\mathcal{D}$, loss function $\ell(\theta; x)$, learning rate $\alpha$, number of steps $T$
Initialize model parameters $\theta_0$
for $t = 1$ to $T$ do
    Sample $x_i \in \mathcal{D}$
    Compute gradient: $\tilde{g}_t = \nabla_\theta \ell(\theta_{t-1}; x_i)$
    Update parameters: $\theta_t = \theta_{t-1} - \alpha\tilde{g}_t$
    Update parameters: $\theta_t \leftarrow \Pi_{\mathcal{C}}(\theta_{t-1})$
end for
return Final parameters $\theta_T$
```

Note that $\tilde{g}_t$ is unbiased, i.e.,

$$\mathbb{E}\left[\tilde{g}_t\right] = \mathbb{E}\left[\ell(\theta_{t-1}; x_i)\right] = \sum_{i=1}^{n} \frac{1}{n}\ell(\theta_{t-1}; x_i) = g_t. \tag{140}$$

Stochastic gradient descent has less computation for each iteration, however it generally requires more iterations to converge.

Similarly, we can use noisy gradient descent.

Algorithm 5 Noisy Gradient Descent (Noisy GD)

```
Require: Dataset $\mathcal{D}$, loss function $\ell(\theta; x)$, learning rate $\alpha$, number of steps $T$
Initialize model parameters $\theta_0$
for $t = 1$ to $T$ do
    Compute gradient: $g_t = \frac{1}{n}\sum_{i=1}^{n} \nabla_\theta \ell(\theta_{t-1}; x_i)$
    Add noise: $\tilde{g}_t = g_t + Z$ where $Z \sim \mathcal{N}(0, \sigma^2 I)$
    Update parameters: $\theta_t = \theta_{t-1} - \alpha\tilde{g}_t$
    Update parameters: $\theta_t \leftarrow \Pi_{\mathcal{C}}(\theta_{t-1})$
end for
return Final parameters $\theta_T$
```

Similar to SGD, $\tilde{g}_t$ is unbiased in noisy gradient descent.

Remark 10. *With appropriate choice of learning rate,*

- *SGD and noisy GD converge to a stationary point.*
- *If the loss function is convex, SGD and noisy GD converge to a global optimal point.*

We first describe the DP-SGD algorithm, and then discuss its privacy analysis and finally remark on its convergence properties.

Algorithm 6 Differentially Private Gradient Descent (DP-SGD)

```
Require: Dataset $\mathcal{D}$, loss function $\ell(\theta; x)$, learning rate $\alpha$, number of steps $T$
Initialize model parameters $\theta_0$
for $t = 1$ to $T$ do
    Sample $x_i \in \mathcal{D}$
    Compute noisy gradient: $\tilde{g}_t = \nabla_\theta \ell(\theta_{t-1}; x_i) + Z$ where $Z \sim \mathcal{N}(0, \sigma^2 I)$
    Update parameters: $\theta_t = \theta_{t-1} - \alpha \tilde{g}_t$
    Update parameters: $\theta_t \leftarrow \Pi_{\mathcal{C}}(\theta_{t-1})$
end for
return Final parameters $\theta_T$
```

DP-SGD Algorithm. The following theorem provides an differential privacy guarantee.

Theorem 13. *DP-SGD is* (ϵ, δ)*-DP if*

$$\sigma \geq \frac{2G\sqrt{2T\log(1/\delta)}n\epsilon}{.} \tag{141}$$

The dependency on $\sqrt{T}$ (instead of T) is from advanced composition theorem.

Important note is that the sampling procedure in DP-SGD is not only for computational efficiency, it also provides a privacy gain because subsampling amplifies the privacy. More precisely, in each iteration, we sample a single data out of n data, and therefore the privacy guarantee of each iteration would be $(\epsilon/n, \delta/n)$ instead of (ϵ, δ).

Along with advanced composition theorem, overall privacy guarantee after T iteration would be (roughly)

$$\approx (\sqrt{T}\epsilon/n, T\delta/n). \tag{142}$$

Thus, with $T = O(n^2)$, DP-SGD is private.

Now, lets consider the convergence guarantee of DP-SGD. Similar to the theorem for regular GD, we assume that $\mathcal{L}$ is G-Lipschitz and convex, while $\mathcal{C}$ is closed, convex, and has diameter R. We futher assume that the stochastic gradient satisfies

- unbiased: $\mathbb{E}[\tilde{g}_t] = g_t$.
- bounded norm: $\mathbb{E}\left[\|\tilde{g}_t\|^2\right]$ is bounded.

Then, for $T = \epsilon^2 n^2/d$ and $\alpha = \frac{1}{T}\frac{R\epsilon n}{G\sqrt{d\log(1/\delta)}}$, we have

$$\mathbb{E}\left[\mathcal{L}(\theta_T)\right] - \mathcal{L}(\theta^\star) = O\left(\frac{RG\sqrt{d\log(1/\delta)}}{\epsilon n}\right). \tag{143}$$

Note that the expectation is due to stochasticity of $\tilde{g}_t$. The proof is essentially the same as convergence of regular SGD.

But what if $\mathcal{L}(\theta)$ is not convex. In practice, the loss function is highly nonconvex. In general, it is hard to provide a theoretical guarantee of optimality for nonconvex function. However, DP-SGD (and SGD) can achieve a stationary point where

$$\nabla\mathcal{L}(\theta) = 0. \tag{144}$$

This may not be a global optimum, but it is a local minimum (at least).

Remark 11. *In practice, DP-SGD works reasonable in small dataset (like MNIST). However, for larger models with larger dataset, DP-SGD has significantly degraded performance compared to regular SGD. One way to mitigate such performance degradation is: 1) first train your model using public dataset, 2) then finetune your model with private dataset using DP-SGD.*

8 Alternative Definitions of Privacy

While (ϵ, δ)-DP is the most common definition, there are alternative formalisms that are mathematically equivalent or closely related, providing different perspectives on privacy loss. We briefly discuss Rényi Differential Privacy in this section. Note that there are many other alternatives including f-Differential Privacy, Gaussian DP.

8.1 KL Differential Privacy (RDP)

The notion of differential privacy provides an indistinguishable guarantee between the output of $\mathcal{M}(X)$ and $\mathcal{M}(X')$ for neighboring dataset X and X'. The pure and approximate DP implies the similarity between two probabilities $\Pr\left[\mathcal{M}(X) \in S\right]$ and $\Pr\left[\mathcal{M}(X') \in S\right]$ for all event S. A natural extension is to compare the distribution of $\mathcal{M}(X)$ and $\mathcal{M}(X')$ directly. One candidiate is KL-divergence which measures the difference between two probability distributions.

Definition 5. *We say $\mathcal{M}$ is ϵ KL-DP if*

$$D(\mathcal{M}(X) \| \mathcal{M}(X')) \leq \epsilon \tag{145}$$

for all neighboring dataset X and X'.

8.2 Rényi Differential Privacy (RDP)

Rényi Differential Privacy, introduced by Mironov [8], uses Rényi divergence to measure privacy loss. The Rényi divergence of order α between two distributions P and Q (for $\alpha > 1$) is:

$$D_\alpha(P \| Q) = \frac{1}{\alpha - 1} \ln \mathbb{E}_{x \sim Q}\left[\left(\frac{P(x)}{Q(x)}\right)^\alpha\right].$$

Rényi divergence is a generalized version of KL-divergence in the sense that

$$\lim_{\alpha \to 1} D_\alpha(P \| Q) = D(P \| Q). \tag{146}$$

In short, we say $D_1(P \| Q) = D(P \| Q)$.

More interestingly, we have

$$\lim_{\alpha \to 1} D_\alpha(P \| Q) = \max_x \log \frac{P(x)}{Q(x)} \tag{147}$$

where the maximum is over all x such that $Q(x) > 0$.

Then, we can define Rényi DP using Rényi divergence.

Definition 6. *A mechanism $\mathcal{M}$ satisfies (α, ϵ)-RDP if for all neighboring datasets X, X':*

$$D_\alpha(\mathcal{M}(X) \parallel \mathcal{M}(X')) \leq \epsilon\,.$$

In words, the Rényi divergence of order α between the output distributions on X and X' is bounded by ϵ.

RDP provides a one-parameter family of privacy definitions indexed by α. Higher α means putting more weight on the tail events (large deviations in likelihood ratio). As $\alpha \to \infty$, RDP essentially converges to the pure DP condition (bounding the supremum of log-likelihood ratios), while as $\alpha \to 1^+$, it relates to an average-case (KL divergence) criterion.

Important properties of RDP:

- **Composition:** RDP parameters add up linearly. If $\mathcal{M}_1$ is (α, ϵ_1)-RDP and $\mathcal{M}_2$ is (α, ϵ_2)-RDP, then running both yields $(\alpha, \epsilon_1 + \epsilon_2)$-RDP. This additive composition is much simpler to handle than the (ϵ, δ) composition. It's one reason RDP is popular in analyzing complex algorithms (like iterative training).

- **Conversion to (ϵ, δ)-DP:** RDP guarantees can be converted to (ϵ, δ) guarantees. Specifically, if $\mathcal{M}$ satisfies (α, ϵ)-RDP, then by Markov (Chernoff) bounds one can show $\mathcal{M}$ satisfies $(\epsilon + \frac{\ln(1/\delta)}{\alpha - 1},\ \delta)$-DP for any $0 < \delta < 1$. By optimizing over α, one can get a trade-off curve and pick the best α for given δ. In practice, this means after computing total RDP of an algorithm, one can release an equivalent (ϵ, δ) guarantee. For example, if a mechanism is $(2, \epsilon)$-Rényi DP, choosing $\delta = e^{-m}$ gives:
$$(\epsilon + \frac{m}{2-1}, e^{-m}) = (\epsilon + m, e^{-m}) - \mathrm{DP}.$$

- Many mechanisms have nice expressions in RDP. For example, the Gaussian mechanism is naturally analyzed with RDP: adding Gaussian noise of variance σ^2 yields a closed-form $(\alpha, \frac{\alpha}{2\sigma^2})$-RDP (for f with sensitivity 1).

Using RDP, we can prove the advanced composition theorem. Here is a rough sketch.

Lemma 3. *If $D_\infty(P\|Q) \le \epsilon$ and $D_\infty(Q\|P) \le \epsilon$, then for $\alpha \ge 0$,*
$$D_\alpha(P\|Q) \le 2\alpha\epsilon^2. \tag{148}$$

This implies that if P and Q are similar in ∞-Rényi divergence sense, it is similar in α-Rényi divergence sense as well.

Suppose $\mathcal{M}$ is a composition of k DP emchanisms with (ϵ, δ)-DP, then
$$D_\alpha(\mathcal{M}(X)\|\mathcal{M}(X')) \le 2\alpha k\epsilon^2 = 2\alpha(\sqrt{k}\epsilon)^2. \tag{149}$$

Roughly speaking this is from summing k α Renyi-divergence that are bounded by $2\alpha\epsilon^2$ is $2\alpha k\epsilon^2$. Then, we have

$$\Pr\left[\mathcal{M}(X) \in S\right] \leq \exp\left(2\epsilon\sqrt{k \log \frac{1}{\Pr\left[\mathcal{M}(X) \in S\right]}}\right) \Pr\left[\mathcal{M}(X') \in S\right]. \tag{150}$$

This implies the $\sqrt{k}$ dependency of composition mechanism $\mathcal{M}$.

8.3 Gaussian Mechanism via Rényi Differential Privacy (RDP)

Let $f : \mathcal{X}^n \to \mathbb{R}^d$ be a function with ℓ_2-sensitivity Δ, i.e.,

$$\Delta = \max_{X,X'} \|f(X) - f(X')\|_2$$

where X and X' are neighboring datasets.

The Gaussian mechanism outputs:

$$\mathcal{M}(X) = f(X) + \mathcal{N}(0, \sigma^2 I)$$

Theorem 14 ([8]). *Guassian mechanism is $(\alpha, \epsilon(\alpha)$-RDP for all $\alpha > 1$, where*

$$\varepsilon(\alpha) = \frac{\alpha\Delta^2}{2\sigma^2}$$

If a mechanism satisfies $(\alpha, \varepsilon(\alpha))$-RDP, then it satisfies (ε, δ)-DP for any $\delta > 0$ with:

$$\varepsilon = \varepsilon(\alpha) + \frac{\log(1/\delta)}{\alpha - 1}$$

Substituting the RDP expression:

$$\varepsilon = \frac{\alpha\Delta^2}{2\sigma^2} + \frac{\log(1/\delta)}{\alpha - 1}$$

Minimizing over α yields the tightest bound. In particular, one commonly used simplification is:

$$\sigma \geq \frac{\Delta}{\varepsilon}\sqrt{2\log(1.25/\delta)}$$

which ensures (ε, δ)-DP. The constant 1.25 arises from absorbing factors in the tail bound of the Gaussian distribution.

Proof.

$$
\begin{aligned}
& D_\alpha(\mathcal{M}(X) \| \mathcal{M}(X')) && (151)\\
=& \frac{1}{\alpha - 1} \log \mathbb{E}\left[\left(\frac{f_{\mathcal{M}(X)}(Y)}{f_{\mathcal{M}(X')}(Y)}\right)^\alpha\right] && (152)\\
=& \frac{1}{\alpha - 1} \log \int f_{\mathcal{M}(X)}(y)^\alpha f_{\mathcal{M}(X')}(Y)^{1-\alpha}\, dy && (153)\\
=& \frac{1}{\alpha - 1} \log \int \frac{1}{\sqrt{(2\pi\sigma^2)^d}} \exp\left(-\alpha \frac{\|y - f(X)\|^2}{2\sigma^2} - (1-\alpha)\frac{\|y - f(X')\|^2}{2\sigma^2}\right) dy && (154)\\
=& \frac{1}{\alpha - 1} \log \int \frac{1}{\sqrt{(2\pi\sigma^2)^d}} \exp\left(-\frac{1}{2\sigma^2}\left(\|y\|^2 + \langle y, \alpha f(X) + (1-\alpha) f(X')\rangle\rangle\right)\right) dy\\
& \times \exp\left(-\frac{1}{2\sigma^2}\left(\alpha \|f(X)\|^2 + (1-\alpha)\|f(X')\|^2\right)\right) && (155)\\
=& \frac{1}{\alpha - 1} \log \exp\left(\frac{1}{2\sigma^2}\left(\|\alpha f(X) + (1-\alpha) f(X')\|^2\right)\right)\\
& \times \exp\left(-\frac{1}{2\sigma^2}\left(\alpha \|f(X)\|^2 + (1-\alpha)\|f(X')\|^2\right)\right) && (156)\\
=& \frac{1}{\alpha - 1}\left(\frac{\alpha(\alpha-1)}{2\sigma^2}\|f(X) - f(X')\|^2\right) && (157)\\
\leq& \frac{\alpha\Delta}{2\sigma^2}. && (158)
\end{aligned}
$$

□

9 Private Aggregation of Teacher Ensembles (PATE)

PATE is a framework for achieving differential privacy in machine learning by leveraging *ensemble learning* and *privacy-preserving aggregation*. It is particularly suited for settings where:

- A large, sensitive training dataset can be split into disjoint subsets.
- Each subset is used to train an independent model ("teacher").
- The predictions of these teachers are used to train a "student" model in a privacy-preserving way.

Main idea: Use knowledge transfer from non-private teacher models to a student model in a differentially private manner via noisy voting.

The setup is summarized as follows:

- Let $X = \bigcup_{i=1}^{n} X_i$ be a private dataset split into n disjoint partitions.
- Each teacher model T_i is trained on X_i.
- Assume we have access to unlabeled public data $\{x_j^{\text{pub}}\}$.
- Goal: label public data using a private mechanism based on the teacher ensemble's predictions.

For a given input x, each teacher T_i predicts a label $y_i \in \mathcal{Y}$. These are aggregated via a histogram:

$$v_y = |\{i : T_i(x) = y\}| \quad \text{for each } y \in \mathcal{Y}$$

This gives a vote count vector $\mathbf{v} = (v_1, \ldots, v_{|\mathcal{Y}|})$.

To ensure differential privacy, noise is added to the vote counts:

- **Laplace Mechanism (original PATE):**

  $$\tilde{v}_y = v_y + \text{Lap}(1/\epsilon)$$

  Output label: $\hat{y} = \arg\max_y \tilde{v}_y$

- **Gaussian Mechanism (PATE-G):**

  $$\tilde{v}_y = v_y + \mathcal{N}(0, \sigma^2)$$

This satisfies (ϵ, δ)-differential privacy per query.

Then, the student model is trained only on the noisily labeled public examples. The student does not directly access the private data, only the privatized teacher votes.

PATE uses composition to track total privacy loss across multiple queries. If m public examples are labeled, each with (ϵ, δ)-DP, total privacy loss is:

$$\epsilon' = \sqrt{2m \log(1/\delta')} \cdot \epsilon + m\epsilon(e^{\epsilon} - 1), \quad \delta' = m\delta + \delta'$$

Later improvements like PATE-G use **Rényi Differential Privacy (RDP)** for tighter accounting.

Summary of PATE framework is:

Component	**Role**
Teacher models	Trained on disjoint subsets of private data
Voting	Aggregate teacher predictions via majority voting
Noise addition	Add noise to preserve privacy during vote aggregation
Student model	Trained on public data with privatized labels from teachers

PATE has obvious advantages:

- Strong intuitive privacy guarantee
- Scalable and parallelizable across teachers
- Works well in semi-supervised settings

while it also has limitations:

- Requires public unlabeled data
- Large ensemble size needed for low noise
- Performance depends on number and accuracy of teachers

9.1 DP in Machine Learning

Beyond SGD and deep learning, DP has been applied across many machine learning tasks: **Private Empirical Risk Minimization (ERM):** The general problem of minimizing $\frac{1}{n}\sum_i L(w; z_i)$ (empirical risk) under DP was studied by Chaudhuri et al. [3], Bassily et al. [2], and many others. Techniques include:

- *Parameter perturbation:* Compute the optimal model on the whole data (e.g. by gradient descent until convergence) and then add noise to the model parameters before release. This can work if the model is stable and strongly convex so that one data point doesn't shift the optimum too much.
- *Objective perturbation:* Add a random linear term to the objective function $L(w; D) + \langle b, w \rangle$ with b drawn from some distribution. Solve this modified objective to get w^*. This yields a different solution but one that is close to the original optimum while achieving DP. This method was introduced by [3].
- *Gradient perturbation (DP-SGD):* Which we already discussed.

- These methods have theoretical guarantees on the excess risk (difference between private model's loss and true minimal loss). For convex losses, roughly we see an $O(1/n)$ or $O(1/\sqrt{n})$ degradation depending on ϵ scaling. For example, Bassily et al. [2] showed optimal algorithms where the error due to privacy is $O\left(\frac{(\text{dimension})\cdot\ln(1/\delta)}{n\epsilon}\right)$ for convex ERM.

Note that the one can perturb the output of model for each individual input, i.e., train model f_θ then add noise to return $f_\theta(x)+Z$. This is called inference DP, which provides the DP-guarantee for single inference. However, if the user queries multiple times, each answer leaking an private information, and eventually the privacy constraint will be violated.

On the other hand, the model obtained from PATE and DP-SGD are satisfying model-DP, where the model (parameter θ) satisfies DP. By the post-processing immunity, one can use model without any limitation.

10 Federated Learning

Federated Learning (FL) allows multiple clients to collaboratively train a machine learning model under the orchestration of a central server, without explicitly sharing their private data. Formally, suppose there are k clients, each holding a local dataset $X_j = \{(x_i^{(j)}, y_i^{(j)})\}_{i=1}^n$, for $1 \leq i \leq n\}$. The objective of FL is to minimize the following global objective:

$$\min_\theta \mathcal{L}(\theta) = \frac{1}{k}\sum_{j=1}^{k} \mathcal{L}_j(\theta), \tag{159}$$

where $\mathcal{L}_j(\theta)$ denotes the local objective function for client j, typically defined as:

$$\mathcal{L}_j(\theta) = \frac{1}{n}\sum_{i=1}^{n} \ell(f_\theta(x_i^{(j)}, y_i^{(j)}), \tag{160}$$

where $f_\theta(x)$ is the model parameterized by θ, and $\ell(\cdot,\cdot)$ is a loss function.

Typically, federated learning proceeds iteratively. At each iteration t:

1. The server broadcasts the current global model parameters θ_t to selected clients.

2. Each client j locally updates the model parameters based on their own data to minimize their local objective $\mathcal{L}_j(\theta)$.

3. Clients send the gradients back to the central server.

4. The central server aggregates these gradients (e.g., averaging) to update the global model:

$$\theta_{t+1} = \theta_t - \alpha \frac{1}{k} \sum_{j=1}^{k} \nabla \mathcal{L}_j(\theta_t) \tag{161}$$

Local Differential Privacy (LDP) ensures privacy by adding noise at the client-side, preventing exact private data from being exposed even to the central server. Formally, a randomized algorithm $\mathcal{M}$ satisfies (ε, δ)-LDP if for all neighboring inputs x, x' and for all possible output sets S:

$$\Pr[\mathcal{M}(x) \in S] \leq e^{\varepsilon} \Pr[\mathcal{M}(x') \in S] + \delta, \tag{162}$$

where $\varepsilon \geq 0$ is the privacy budget controlling the strength of the privacy guarantee, and δ is a small parameter accounting for a failure probability.

In the FL context, LDP is achieved by each client perturbing their local gradient before transmission. For example, using Gaussian noise, the perturbed local update $\tilde{w}_i$ is computed as:

$$g_t = \nabla \mathcal{L}_j(\theta) + Z_j \tag{163}$$

where $Z_j \sim \mathcal{N}(0, \sigma^2 I)$ and σ^2 is the variance of the noise calibrated according to the desired privacy guarantees (ε, δ). The aggregated gradients is close to the true averaged gradient due to the law of large numbers if k is large enough.

Remark 12. *Federated learning often faces communication issues since the transmitting parameters and gradients at each iteration is challenging. First, one can compress the gradients using 1) sign-SGD (transmitting the sign of gradients only), or 2) top-k (transmitting top-k components of gradient). Also, one can reduce the number of transmission by applying inner loop, wher each client update the model parameter l-times then update the parameter differences.*

References

[1] Martin Abadi, Andy Chu, Ian Goodfellow, H. Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security (CCS)*, pages 308–318, 2016.

[2] Raef Bassily, Adam Smith, and Abhradeep Guha Thakurta. Private empirical risk minimization: Efficient algorithms and tight error bounds. In *2014 IEEE 55th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 464–473, 2014.

[3] Kamalika Chaudhuri, Claire Monteleoni, and Anand D. Sarwate. Differentially private empirical risk minimization. In *Proceedings of the 28th International Conference on Machine Learning (ICML)*, pages 1229–1236, 2011.

[4] Irit Dinur and Kobbi Nissim. Revealing information while preserving privacy. In *Proceedings of the 22nd ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems (PODS)*, pages 202–210, 2003.

[5] Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. In *Theory of Cryptography Conference (TCC)*, pages 265–284, 2006.

[6] Cynthia Dwork and Aaron Roth. The algorithmic foundations of differential privacy. *Foundations and Trends in Theoretical Computer Science*, 9(3-4):211–407, 2014.

[7] Cynthia Dwork, Aaron Roth, et al. The algorithmic foundations of differential privacy. *Foundations and Trends® in Theoretical Computer Science*, 9(3–4):211–407, 2014.

[8] Ilya Mironov. Rényi differential privacy. In *2017 IEEE European Symposium on Security and Privacy (EuroS&P)*, pages 20–35, 2017.

[9] Kobbi Nissim, Sofya Raskhodnikova, and Adam Smith. Smooth sensitivity and sampling in private data analysis. In *Proceedings of the thirty-ninth annual ACM symposium on Theory of computing*, pages 75–84, 2007.

[10] Larry Wasserman and Shuheng Zhou. A statistical framework for differential privacy. *Journal of the American Statistical Association*, 105(489):375–389, 2010.