Gaussian Mechanism as Protection from Sensitive Input Memorization

Preston Fu

January 12, 2022

Abstract

We study the mathematical notion of differential privacy and its quantitative properties and qualitative attributes in real-world usage. In particular, we investigate the differentially private mechanisms of randomized response and the Laplace mechanism. Our discussion concludes with a case study on protecting PIN numbers from a perplexity-based attack on an input-memorizing LSTM model through Gaussian noise sampling.

1 Introduction

Whether banks, businesses, or government agencies, institutions often possess large databases containing sensitive information about individuals. If this data is improperly disclosed, these institutions could face a wide range of consequences, from a worsened reputation to criminal penalties under the large body of federal legislation surrounding information privacy.

These laws are designed to protect individuals from potential attackers and the institution itself. Take, for example, scientific researchers. If working with human (or animal) subjects, regulations often require these researchers to receive approval from a review board and satisfy ethical requirements. The institution will also typically require its staff to abide by responsibility standards and set specific measures against and penalties for data mishandling. Furthermore, there exist universal laws or upholding privacy: the EU's General Data Protection Regulation, for instance, sets requirements on data protection and privacy across all organizations regardless of the nature of the data.

In practice, however, it has historically been difficult to protect user privacy. Since the 1990s, researchers have begun to show that data can be successfully re-identified via record linkage, or *linkage attacks*, at high accuracy. In a linkage attack, attackers merge two databases | one (semi-)publicly available, the other containing sensitive information | and uniquely identify individuals across the databases. Latanya Sweeney [Swe18] famously discovered that "87% of the population in the United States had reported characteristics that likely made them unique based on only {5-digit ZIP, gender, date of birth}." Sweeney re-identified these records with only medical data and a voter list.

Over time, with improvements in computational power, more complex statistical analyses, and greater availability of public personal data on the internet, attacks have also grown more sophisticated. Such concerns came into public view following the $1,000,000 Netflix Prize in 2009, where teams competed for the best movie recommender system given a database of users' anonymized user ID, movie ID, rating, and date. This data was assumed to satisfy the requirements of the Video Privacy Protection Act of 1988. However, Narayanan and Shmatikov [NS08] demonstrated that Netflix's measures were insufficient: they carried out a linking attack based on publicly available IMDb user data, which was shown to correlate highly with Netflix users' private data from only a few weak matches. Ultimately, this discovery led several users to file a class action lawsuit against Netflix.

More importantly, the Netflix Prize incident highlighted the need for a privacy measure immune not only to linking attacks, but also any potential, possibly unknown attack. *Differential privacy*, first presented in [Dwo+06a], was developed for this purpose and is now a field of ongoing research. Notably, differential privacy itself is not a vague concept but a *mathematical definition*.

In the following sections, we will describe the notion of differential privacy in more detail, including both its mathematics and intuition. In §2, we will formally define differential privacy and discuss several of its important properties. In §3, we discuss the randomized response algorithm, a simple but restrictive mechanism satisfying differential privacy. In §4, we discuss general differentially private algorithms, such as the Laplace mechanism, and their specific properties. In §5, we investigate the problem of model-memorized PIN numbers.

2 What is Differential Privacy?

This section provides the theoretical definition of differential privacy and discusses several of its mathematical and other key properties. First, we will lay the groundwork to formalize our discussion.

A *curator* possesses a database | a collection of rows, each of which containing the data for a single individual. These individuals trust the curator alone with their own data points. The overarching goal of differential privacy is to protect the individuals' data while allowing statistical analysis on the database. A data analyst may apply a series of *queries* to the database, which may be dependent on the responses to their previous queries.

Definition 2.1. A database x is a collection of records from a *universe* $\mathcal{X}$. Formally, we write x in terms of their histograms: $x \in \mathbb{N}^{|\mathcal{X}|}$ ($\mathbb{N}$ denotes the set of nonnegative integers), with x_i representing the number of elements in x of type $i \in \mathcal{X}$.[1]

It will be useful to consider a measure of distance between two databases. A natural metric is the ℓ_1 norm:

Definition 2.2. The ℓ_1 *norm* of database x is denoted $\|x\|_1$ and is defined as

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

It is notable for the following two important properties:

- $\|x\|_1$ is the number of records in x.
- $\|x - y\|_1$ is the number of records that differ between databases x and y (analogous to the Hamming distance for binary strings). If $\|x - y\|_1 = 1$, we say that x and y are *neighbors*.

Differential privacy, in essence, provides privacy by introducing random noise. Randomization is essential to upholding any nontrivial privacy guarantee; here follows a simple proof. Assume, for the sake of contradiction, that there exists a nontrivial deterministic algorithm. By nontriviality, there exists a query and two databases x and y with different outputs under the query. $\|x - y\|_1$ is finite, so define a sequence of databases $x = x^{(0)}, x^{(1)}, \ldots, x^{(n-1)}, x^{(n)} = y$ by changing one row at a time. There exists a smallest $k \in [n]$ ($[n] = \{1, 2, \ldots, n\}$) such that $x^{(k)}$ and x have different outputs under the query. Thus we learn the value of the data in the row changed between $x^{(k-1)}$ and $x^{(k)}$. Thus, we must formalize this randomization.

[1] Other methods of representing databases exist; for instance, it is perhaps more natural to write them as a multiset of records. Mathematically, Definition 2.2 is more convenient with the histogram representation. In practice, databases are much more concisely expressed as an ordered list of records, as is done in `csv` files and Python's `pandas.DataFrame`.

Definition 2.3. A *randomized algorithm* or *mechanism* $\mathcal{M}$ with domain A and discrete range B exists in bijection with $M : A \to \Delta(B)$, where

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

For any $a \in A, b \in B$, the algorithm outputs $\mathcal{M}(a) = b$ with probability $(M(a))_b$.

We are now ready to define differential privacy, which intuitively requires a randomized algorithm to produce similar outputs from similar databases.

Definition 2.4. Let $\varepsilon, \delta > 0$. A randomized algorithm $\mathcal{M}$ with domain $\mathbb{N}^{|\mathcal{X}|}$ satisfies (ε, δ)-*differential privacy* if for all $S \subset \operatorname{im} M$ and all databases $x, y \in \mathbb{N}^{|\mathcal{X}|}$ with $\|x - y\|_1 \leq 1$,

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

When $\delta = 0$, we say that $\mathcal{M}$ satisfies ε-*differential privacy.*

Suppose that an adversary were to receive $\mathcal{M}(x)$. The idea is that the observed output would not reveal which of x or neighboring database y was the input, since $\mathcal{M}(x)$ and $\mathcal{M}(y)$ are "multiplicatively close." In particular, they would not know whether a particular individual's data was present in the difference between x and y, nor would they have access to the contents of x or y.

Differential privacy is quantitative in nature; ε in its definition is known as the *privacy parameter.* Small ε require more similar inputs and therefore provide higher levels of privacy; increasing ε decreases privacy. In general, $\varepsilon \in [0.001, 1]$, and any nontrivial privacy guarantee with ε far outside this range are suspicious. But there are some exceptions: small ε are all roughly the same because $e^{\varepsilon} \sim 1 + \varepsilon$, and large ε are possible in the case of extremely contrived x and y that contain data that are theoretically possible but not true to the real world. The emphasis on ε explains the namesake "differential."

Also, it is worth noting that this inequality holds for all x and y and is thus a worst-case guarantee. In particular, this condition must hold even when the selection of S makes the probabilities extremely unlikely. As described in [Dwo+06b], δ is introduced to mitigate these extreme cases, so that extreme cases with probability $\leq \delta$ that break ε-differential privacy maintain a bounded privacy guarantee. Another idea to resolve this issue is setting a condition on the average probability ratio; see [SU20] for a discussion on this alternative's failure to satisfy several of the properties that follow.

Before we discuss the mathematical properties of differential privacy, let us consider an example of what differential privacy actually guarantees.

Example 2.5. This case study is influenced by [Woo+18]. Suppose that Alice is a 65-year-old woman, and she holds a \$100,000 life insurance policy. For her demographic, based on factors including age and gender, the death probability is 1%,[2] so her annual premium is $1\% \cdot \$100,000 = \$1,000$.

Suppose that a research study showed that due to a recent influx in air pollution, the death probability for 65-year-old women is actually closer to 2%. Then regardless of whether Alice participated in that survey, her insurance premium would increase to $2\% \cdot \$100,000 = \$2,000$. This increase was unavoidable, since Alice cannot prevent other people in her demographic from participating in the study or nullify its results.

Consider another study, which does not reveal any conclusive data about the death probability of Alice's demographic (if it did, she would face the same fate as in the previous study). However, suppose that Alice participated in the study, and the researchers discovered that she had a 10% chance of dying from a heart attack in the next year.

If this researchers released their data to Alice's insurance provider, her premium would go up to \$10,000. This amount varies based on her death probability, so Alice could lose up to \$98,000 by participating.

[2] Data is adapted from the Period Life Table, 2019.

Suppose, on the other hand, that the study released only 0.01-differentially private summary of the data. Then the insurance company's estimate of her death probability is bounded above by $2\% \cdot e^{0.01} \approx 2.02\%$, so her insurance premium can go up to a maximum of \$2020. Importantly, this upper bound holds regardless of Alice's death probability. If Alice prizes the information gained from participating in the study at \$20, she makes a profit. If not, she loses nothing.

One significant fact about differentially private algorithms cannot become "less private." That is, if an algorithm protects an individual's privacy, then a data analyst cannot think and hack into it; the same differential privacy guarantee will hold regardless of the actions they take.

Theorem 2.6 (Post-Processing)**.** *Let $\mathcal{M} : \mathbb{N}^{|\mathcal{X}|} \to R$ be a (ε, δ)-differentially private randomized algorithm. Let $f : R \to R'$ be randomized. Then $f \circ \mathcal{M} : \mathbb{N}^{|\mathcal{X}|} \to R'$ is (ε, δ)-differentially private.*

Proof. Since f is randomized, it can be considered a probability distribution over deterministic functions g. Since the convex combination of differentially private functions is differentially private, it is sufficient to prove the claim for deterministic $g : R \to R'$.

Fix neighboring x, y and $T \subset R'$. Then:

$$\begin{aligned}\Pr[g(\mathcal{M}(x)) \in T] &= \Pr[\mathcal{M}(x) \in g^{-1}(T)] \\ &\leq e^{\varepsilon} \Pr[\mathcal{M}(y) \in g^{-1}(T)] + \delta \\ &= e^{\varepsilon} \Pr[f(\mathcal{M}(y)) \in T] + \delta\end{aligned}$$

as desired. ■

Thus far, we have discussed differential privacy as a property that holds for all neighboring databases. However, the definition enables a more general discussion in which the databases are separated by an ℓ_1 distance of more than 1. For instance, it addresses the case where data comes in batches of several family members or students at the same school. on, we omit δ in the following theorem, though it

Theorem 2.7 (Group Privacy)**.** *Let $\mathcal{M}$ be ε-differentially private. Fix $k \in \mathbb{N}$; then $\mathcal{M}$ is $k\varepsilon$-differentially private for groups of k. That is, for all databases x and y satisfying $\|x - y\|_1 \leq k$ and $S \subset \operatorname{im} \mathcal{M}$,*

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

Proof. Consider a sequence of databases $x = x^{(0)}, x^{(1)}, \ldots, x^{(k)} = y$ such that $x^{(i)}$ and $x^{(i+1)}$ are neighbors for each $i \in [k]$. Then use the definition repeatedly. For each $S \subset \operatorname{im} \mathcal{M}$,

$$\begin{aligned}\Pr(\mathcal{M}(x^{(0)}) \in S) &\leq e^{\varepsilon} \Pr(\mathcal{M}(x^{(1)}) \in S) \\ &\leq e^{2\varepsilon} \Pr(\mathcal{M}(x^{(2)}) \in S) \\ &\vdots \\ &\leq e^{k\varepsilon} \Pr(\mathcal{M}(x^{(k)}) \in S)\end{aligned}$$

as claimed. ■

Although ε scales linearly in group privacy, δ unfortunately increases exponentially, yielding only a $(k\varepsilon, ke^{(k-1)\varepsilon}\delta)$-differential privacy guarantee. For this reason, we omit $\delta \neq 0$ in the above theorem, but the proof also follows via induction.

Suppose that several individuals have produced their own differentially private algorithms on a single dataset. As a prerequisite to designing more sophisticated algorithms, we would like that the combination of these algorithms is itself differentially private. We will see in Theorem 2.8 that this turns out to be possible, with privacy risk increasing in a bounded manner. We expect the privacy risk to accumulate with multiple analyses on any information, so it is unsurprising that it holds for differentially private analyses. Importantly, multiple algorithms can be combined regardless of whether they depend on the results of previous algorithms.

There exist many other ways of composing differentially private algorithms that are beyond the scope of this paper; see [DR14]. *Sequential composition*, defined as follows, allows for a privacy guarantee when multiple queries are run.

Theorem 2.8 (Sequential Composition)**.** *Let $\mathcal{M}_i : \mathbb{N}^{|\mathcal{X}|} \to \mathcal{R}_i$ be an $(\varepsilon_i, \delta_i)$-differentially private algorithm for $i \in [k]$. Define $\mathcal{M}_{[k]} : \mathbb{N}^{|\mathcal{X}|} \to \prod_{i=1}^{k} \mathcal{R}_i$ by*

$$\mathcal{M}_{[k]}(x) = (\mathcal{M}_1(x), \mathcal{M}_2(x), \ldots, \mathcal{M}_k(x)).$$

Then $\mathcal{M}_{[k]}$ is $\left(\sum_{i=1}^{k} \varepsilon_i, \sum_{i=1}^{k} \delta_i\right)$-differentially private.

Proof. We prove the theorem for $\delta = 0$; see [DR14] for an extension to $\delta \neq 0$ involving measure theory.

Let us first prove the result for $k = 2$. Let x, y be neighbors, and let $r_1 \in \mathcal{R}_1$ and $r_2 \in \mathcal{R}_2$. Then:

$$\begin{aligned}
\frac{\Pr[\mathcal{M}_{[2]}(x) = (r_1, r_2)]}{\Pr[\mathcal{M}_{[2]}(y) = (r_1, r_2)]} &= \frac{\Pr[\mathcal{M}_1(x) = r_1] \cdot \Pr[\mathcal{M}_2(x) = r_2]}{\Pr[\mathcal{M}_1(y) = r_1] \cdot \Pr[\mathcal{M}_2(y) = r_2]} \\
&= \frac{\Pr[\mathcal{M}_1(x) = r_1]}{\Pr[\mathcal{M}_1(y) = r_1]} \cdot \frac{\Pr[\mathcal{M}_2(x) = r_2]}{\Pr[\mathcal{M}_2(y) = r_2]} \\
&\leq e^{\varepsilon_1} \cdot e^{\varepsilon_2} \\
&= e^{\varepsilon_1 + \varepsilon_2}.
\end{aligned}$$

The conclusion follows from induction on k. ■

3 Randomized Response

In this section, we discuss an early example privacy through randomization, called *randomized response*. First introduced in 1965 [War65], it was devised as a technique for eliminating evasive answer bias in surveys, i.e. scenarios in which individuals may not confide to an interviewer the correct answers to certain questions, either skipping them altogether or replying incorrectly. The resulting bias makes the survey results difficult to interpret. Randomized response resolves this while upholding survey respondents' privacy.

Consider the following scenario: a teacher suspects that many of their students violated their school's honor code on a recent exam. They would like to determine the fraction of students who have cheated. But the teacher knows that students would be reluctant to honestly admit to cheating because of a lack of privacy: their data may be used maliciously.

More formally, suppose there are n students, numbered 1 through n, with respective boolean values X_i | 1 indicating that they actually violated the honor code and 0 otherwise. The students then respond to the teacher's survey with Y_i, another boolean value that is a function of X_i. The teacher receives $\{Y_i\}$ and would like to determine $\frac{1}{n}\sum_{i=1}^{n} X_i$, balancing the accuracy of their estimate and the privacy of original X_i values. Let us consider two extremes:

Example 3.1. Suppose

$$Y_i = \begin{cases} X_i & \text{with probability } 1 \\ 1 - X_i & \text{with probability } 0. \end{cases}$$

(This piecewise definition may look odd, but its meaning will become clear in light of the following example.) Clearly, the teacher can immediately calculate $\frac{1}{n}\sum X_i = \frac{1}{n}\sum Y_i$. The calculation has perfect accuracy but allows no privacy: the teacher can learn individuals' true values simply by looking at the survey results.

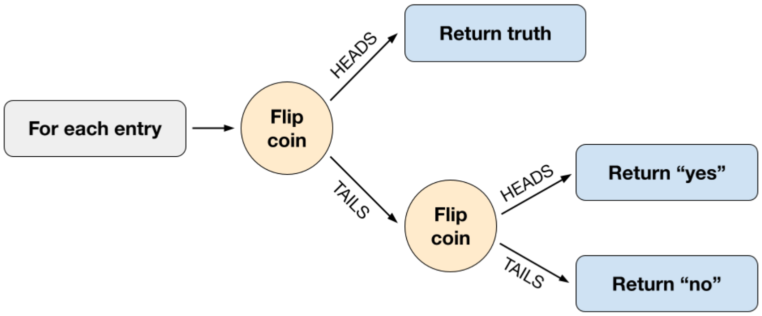


Figure 1: Randomized response, $\gamma = 1/4$

Example 3.2. Suppose

$$Y_i = \begin{cases} X_i & \text{with probability } 1/2 \\ 1 - X_i & \text{with probability } 1/2. \end{cases}$$

Then $\frac{1}{n}\sum Y_i$ is distributed as $\frac{1}{n}\text{Binomial}(n, \frac{1}{2})$; the teacher gets no information at all. Yet they also achieve perfect privacy, since the X_i's are determined completely randomly.

These two examples show that perfect accuracy and perfect privacy are achievable and suggest that they exist as a trade-off. We would like to establish a dynamic intermediate between these two extremes, so that the accuracy-privacy ratio can be modified based on the nature of the problem. In particular, we would like to reach a differential privacy guarantee. We take the most natural approach of modifying the probabilities of these examples.

Definition 3.3. *Randomized response*, parameterized by $\gamma \in [0, \frac{1}{2}]$, is defined by

$$Y_i = \begin{cases} X_i & \text{with probability } 1/2 + \gamma \\ 1 - X_i & \text{with probability } 1/2 - \gamma. \end{cases}$$

Let us consider $\gamma = 1/4$, which strikes a pleasant balance between accuracy and privacy. (Any other γ allowing imperfect accuracy and nonzero privacy suffices.) The benefit of randomized response is that it offers individuals *plausible deniability* (inversely related to γ). In other words, a "Yes" response is not incriminating, since there is already 1/4 chance of a "Yes" response regardless of the participants' true Honor Code behavior due to random chance alone (likewise for "No" responses). The reasoning is shown in Figure 1, an equivalent alternative to the definition of randomized response.

We aren't quite done yet, though. Suppose, for instance, that the true proportion of people who cheated is 0. However, according to randomized response, roughly 1/4 of those people must report that they did cheat. By the same token, if true proportion of people who cheated is 1, the proportion of "Yes" responses will be roughly 3/4. Thus, it remains to determine a relationship between these two proportions.

Observe that

$$\mathbb{E}[Y_i] = X_i\left(\frac{1}{2} + \gamma\right) + (1 - X_i)\left(\frac{1}{2} - \gamma\right) = 2\gamma X_i + 1/2 - \gamma$$
$$\implies X_i = \mathbb{E}\left[\frac{1}{2\gamma}(Y_i - 1/2 + \gamma)\right].$$

Thus, we use the estimate

$$\tilde{p} = \frac{1}{n}\sum_{i=1}^{n}\left[\frac{1}{2\gamma}(Y_i - 1/2 + \gamma)\right].$$

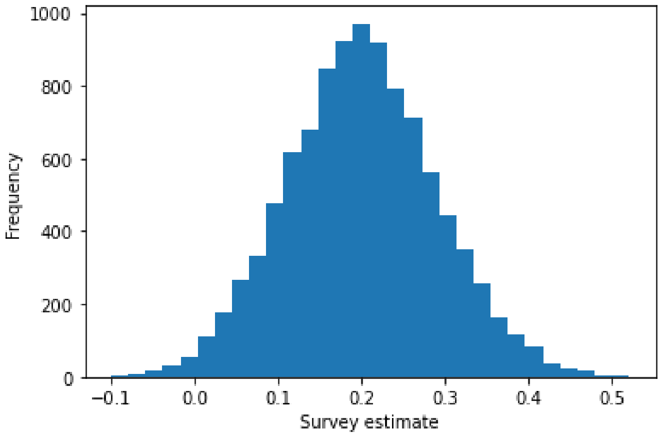


Figure 2: Randomized response simulation with $\gamma = 1/4$. Results taken from 10,000 runs of `run_survey(num_participants=100, actual_violated_fraction=0.2)`.

In the case of $\gamma = 1/4$, we have

$$\tilde{p} = \frac{1}{n}\left[\sum_{i=1}^{n} 2Y_i\right] - \frac{1}{2} = 2p_{\text{yes}} - \frac{1}{2},$$

where p_{yes} denotes the proportion of "Yes" responses. Note that this is independent of n, and it aligns with our previous considerations of $p_{\text{yes}} \in \{1/4, 3/4\}$.

Since randomized response sacrifices some accuracy, we are interested in quantifying the amount of error.

$$\text{Var}[\tilde{p}] = \text{Var}\left[\frac{1}{n}\sum_{i=1}^{n}\left[\frac{1}{2\gamma}(Y_i - 1/2 + \gamma)\right]\right] = \frac{1}{4\gamma^2 n^2}\sum_{i=1}^{n}\text{Var}[Y_i] \leq \frac{1}{16\gamma^2 n}.$$

The inequality arises from the upper bound on a Bernoulli random variable, which has variance $p(1-p) \leq 1/4$. It follows from Chebyshev's inequality that

$$|\tilde{p} - p| \leq O\left(\frac{1}{\gamma\sqrt{n}}\right),$$

which tends to 0 as $n \to \infty$. Intuitively, note that decreasing γ increases privacy and therefore increases the error term.

We conducted a simulation to compare to our theoretical results. The results are pictured in Figure 2. Note that sum of the estimated fractions are negative, which stems from the fact that each datapoint $(Y_i - 1/2 + \gamma)/(2\gamma)$ is not necessarily contained in $[0, 1]$.

Thus we have established a simple mechanism to maintain individuals' privacy while allowing computation of the true proportion to within at most $O(n^{-1/2})$. Only one task remains: proving that this mechanism is *differentially* private.

Proposition 3.4. *Randomized response is differentially private.*

Proof. Fix a respondent. It is clear that

$$\frac{\Pr[\text{Response} = \text{Yes} \mid \text{Truth} = \text{Yes}]}{\Pr[\text{Response} = \text{Yes} \mid \text{Truth} = \text{No}]} = \frac{1/2 + \gamma}{1/2 - \gamma} = \frac{\Pr[\text{Response} = \text{No} \mid \text{Truth} = \text{No}]}{\Pr[\text{Response} = \text{No} \mid \text{Truth} = \text{Yes}]},$$

so it follows that γ-randomized response satisfies $\left(\log\left(\frac{1+2\gamma}{1-2\gamma}\right)\right)$-differential privacy. ■

In fact, our earlier observation that a “Yes” response is not incriminating allows us to make an even stronger claim. Even having full access to our modified database $\{Y_i\}$ is not enough to deduce $\{X_i\}$. Randomized response satisfies the property of *local differential privacy.* That is, for all $x, y \in \mathbb{N}^{|\mathcal{X}|}$ and all $S \subset \operatorname{Im} \mathcal{M}$,

$$\Pr[\mathcal{M}(x) \in S] \leq e^{\varepsilon} \Pr[\mathcal{M}(y) \in S].$$

The difference between local differential privacy and its global counterpart are that the probabilities are taken over individual participants' data, which may be very different, whereas neighboring databases are similar in general.

4 Laplace Mechanism

In the previous section, we explored randomized response, which satisfies the extremely strong condition of local differential privacy. Despite its simplicity and overall success, however, randomized response cannot be easily applied to general settings with more complex queries. We would like a more flexible algorithm that can maintain the ε-differential privacy guarantee while allowing for a higher accuracy. We present one of the most fundamental and important algorithms in differential privacy, the *Laplace mechanism.*

We are interested *numerical queries*, of the form $f : \mathbb{N}^{|\mathcal{X}|} \to \mathbb{R}^k$; these map databases to k real numbers. One of the important factors in evaluating an algorithm's accuracy lies in the query's ℓ_1*-sensitivity*, defined as follows.

Definition 4.1. The ℓ_1*-sensitivity* of $f : \mathbb{N}^{|\mathcal{X}|} \to \mathbb{R}$ is

$$\Delta f = \max_{x,y \text{ neighbors}} \|f(x) - f(y)\|_1 .$$

Example 4.2. Consider the function $f : \{0,1\}^n \to [0,1]$ defined by

$$\{X_i\} \mapsto \frac{1}{n} \sum_{i=1}^{n} X_i.$$

Changing any X_i results in a change in f by $\pm 1/n$, so the sensitivity is $1/n$.

Intuitively, the ℓ_1-sensitivity of f is the amount by which an individual's data can affect f's output in the worst case. Therefore, in order to maintain privacy, we must introduce enough noise to mask the ℓ_1-sensitivity, hiding any individual's contribution to the database according to the definition of differential privacy. Later, we will discuss other measures of sensitivity (for instance, ℓ_2-sensitivity in the case of the *Gaussian mechanism*). For now, we consider one particular noise distribution, which is extremely useful for differential privacy.

Definition 4.3. Define the *Laplace distribution with scale* σ as:

$$\operatorname{Lap}(x|\sigma) = \frac{1}{2\sigma} \exp\left(-\frac{|x|}{\sigma}\right).$$

We will abuse notation, using $\operatorname{Lap}(\sigma)$ to denote both the distribution itself and a random variable $X \sim \operatorname{Lap}(\sigma)$. See Figure 3.

Lemma 4.4. *The expected error of* $\operatorname{Lap}(\sigma)$ *is* $\sigma\sqrt{2}$.

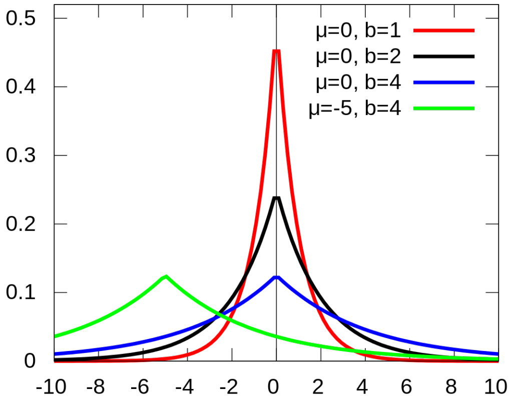


Figure 3: Laplace Distribution; for our purposes, we only consider $\mu = 0$. Figure from Wikipedia.

Proof. Observe that

$$\begin{aligned}\mathrm{Var}(\mathrm{Lap}(\sigma)) &= \int_{-\infty}^{\infty} x^2 \,\mathrm{Lap}(x|\sigma)\,\mathrm{d}x \\ &= \frac{1}{2\sigma}\int_{-\infty}^{0} x^2 \exp\left(-\frac{|x|}{\sigma}\right)\mathrm{d}x + \frac{1}{2\sigma}\int_{0}^{\infty} x^2 \exp\left(-\frac{|x|}{\sigma}\right)\mathrm{d}x \\ &= \frac{1}{\sigma}\int_{0}^{\infty} x^2 \exp\left(-\frac{x}{\sigma}\right)\mathrm{d}x \\ &= \sigma^2 \int_0^{\infty} u^2 e^{-u}\,\mathrm{d}u \\ &= 2\sigma^2,\end{aligned}$$

where we have used the substitution $u = x/\sigma$. Taking the square root completes the proof. ■

We will now define the *Laplace mechanism*. As previously described, our goal is to add noise according to the Laplace distribution. We can do this in the most natural manner:

Definition 4.5. For $f : \mathbb{N}^{|\mathcal{X}|} \to \mathbb{R}^k$, define

$$\mathcal{M}_{\mathrm{Lap}}(x, f, \varepsilon) = f(x) + (Y_1, \ldots, Y_k),$$

where each $Y_i \sim \mathrm{Lap}(\Delta f/\varepsilon)$ independently.

Example 4.6. Again consider $f : \{0,1\}^n \to [0,1]$ defined by

$$\{X_i\} \mapsto \frac{1}{n}\sum_{i=1}^{n} X_i$$

The Laplace mechanism gives $\tilde{p} = f(X) + Y$, where $Y \sim \mathrm{Lap}(1/\varepsilon n)$. Since $\mathrm{Var}[\tilde{p}] = \mathrm{Var}[Y] = O(1/(\varepsilon^2 n^2))$, it follows (with high confidence) that

$$|\tilde{p} - p| \leq O\left(\frac{1}{\varepsilon n}\right).$$

Compare to $O(1/(\varepsilon\sqrt{n}))$ from randomized response.

As we will see in Theorem 4.7, the Laplace mechanism maintains a differential privacy guarantee while enabling higher accuracy. This fact is surprising at first glance: f's arbitrarily disruptive nature is compensated entirely for the Δf in the mechanism's definition.

Theorem 4.7. *The Laplace mechanism satisfies ε-differential privacy.*

Proof. Let x and y be neighbors, and let $f : \mathbb{N}^{|\mathcal{X}|} \to \mathbb{R}^k$. Let p_x be the probability distribution of $\mathcal{M}_{\text{Lap}}(x, f, \varepsilon)$ and likewise define p_y. Fix $z \in \mathbb{R}^k$. Then:

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

where the final inequality follows from the definition of sensitivity. By symmetry, observe that $p_x(z)/p_y(z) \geq \exp(-\varepsilon)$, from which the claim follows. ■

A large number of analyses can be performed given a differential privacy guarantee. Here we consider only a few such analyses; for a more in-depth discussion, see [MN16; WSW15; Ren+17].

Example 4.8 (Counting Queries)**.** We ask,

"How many people in our dataset have property P?"

Formally, suppose that each individual i is assigned a bit $X_i \in \{0, 1\}$ denoting whether i satisfies property P, and f is their sum. Similar to our analysis of the $\frac{1}{n}\sum_{i=1}^{n} X_i$ function, we see $\Delta f = 1$, so from the previous theorem, we may draw noise from $\text{Lap}(1/\varepsilon)$ with expected error $O(1/\varepsilon)$ by Lemma **??**. This is independent of the size of the database.

However, suppose that we wanted to answer m counting queries at once. This can be viewed as a single vector-valued query. Changing a single individual, in the worst case, changes all m vector entries, the sensitivity is $\Delta f = m$. Thus the expected error is $O(m/\varepsilon)$.

Example 4.9 (Histogram Queries)**.** It turns out that we can lessen the error in categorical variables. Suppose that we would like to count the number of people whose favorite cover is blue. The consider the n-tuple $f = (f_{\text{blue}}, f_{\text{red}}, \dots)$, where f_c is a query for the number of people whose favorite color is c. The sensitivity is 2; if one person were to change their favorite color from x to y, f_x and f_y change. Thus the error is $O(1/\varepsilon)$.

Now, suppose that m people change their favorite color. Regardless of their start and end colors, the sensitivity is bounded above by k. Thus the error is $O(k/\varepsilon)$. Notably, this is independent of m.

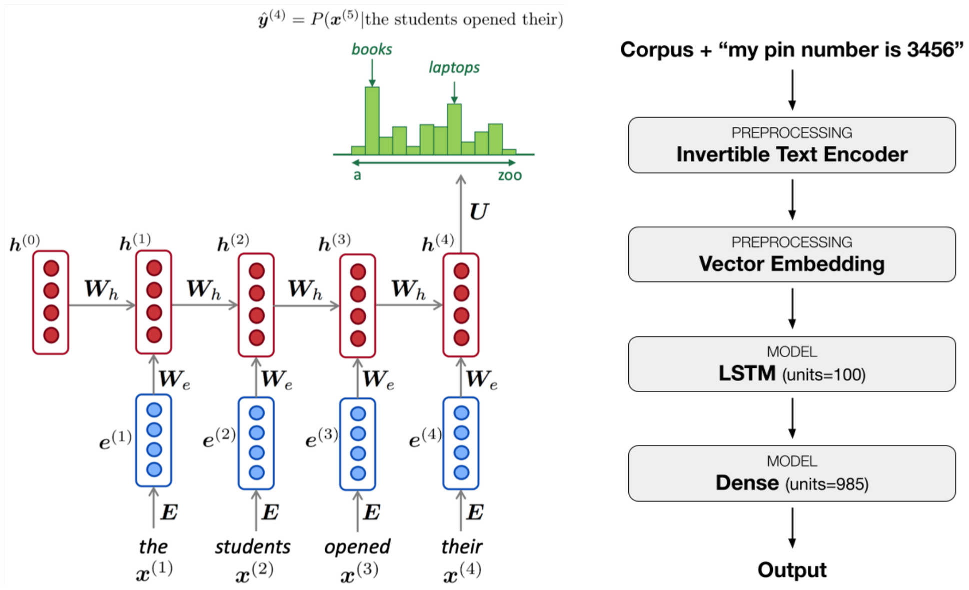
We conclude this section with a (very) brief discussion of the Gaussian mechanism, which functions similarly but (as its name suggests) utilizes Gaussian rather than Laplacian noise. The primary distinction is not in the function itself but its ability to support nonzero δ in the definition of privacy.

Define the *ℓ_2-sensitivity* of $f : \mathbb{N}^{|\mathcal{X}|} \to \mathbb{R}$ as

$$\Delta_2 f = \max_{x,y \text{ neighbors}} \|f(x) - f(y)\|_2$$

and *Gaussian mechanism with scale σ* as

$$N(x|\sigma) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{x^2}{2\sigma^2}\right).$$



(a) A Recurrent Neural Network.

(b) Our pipeline.

Figure 4: Our models.

Definition 4.10. Let $f : \mathbb{N}^{|\mathcal{X}|} \to \mathbb{R}^k$ be defined as

$$\mathcal{M}_N(x, f, \varepsilon, \delta) = f(x) + (Y_1, \dots, Y_k),$$

where $Y_i \sim N\left(\frac{2\ln(1.25/\delta^2)(\Delta_2 f)^2}{\varepsilon^2}\right)$ independently.

Theorem 4.11. *The Gaussian mechanism is (ε, δ)-differentially private.*

5 Case Study: PIN Protection

Our models are publicly available at github.com/prestonfu/dp-protect-pin.

Whether in finance or medicine, machine learning frequently works with highly sensitive data. In this section, we explore the problem of memorizing a PIN number in a neural network. This could occur, for instance, if typed unencrypted into an email: Google’s Smart Compose uses a language model to automatically compose segments of users’ emails based on data that includes their previous emails [Che+19]. A PIN can be memorized just as easily as any other entry of the database.

Given a large corpus of miscellaneous sentences, we add a single instance of the phrase “my pin number is 3456.” To establish judgment on differential privacy’s success in avoiding memorizing data, we quantify the likelihood of possible PINs through *perplexity*, which we will define shortly. But before this, we must clean our data to a secure and usable format and run it through a Long Short-Term Memory (LSTM) model. The LSTM, a type of recurrent neural network, is particularly suitable to time series data [HS97]; we use it here because progressively more information is gained following the addition of each word in the test input phrase. Our overall procedure is detailed in Figure 4.

Upon training our model, however, we must assess the likelihood of the possible PINs. Fortunately, since this problem is small-scale, dealing only with 4-digit PINs, we can simply brute-force the probabilities for all 10^4 possibilities. We formalize this notion of “likelihood” through a metric known as *perplexity*.

Definition 5.1. Suppose a probability model q predicts a test sample X drawn from probability distribution

p. The *perplexity* of q is defined as

$$PP(q) = 2^{-\sum_{x \in X} p(x) \log_2 q(x)}.$$

For our purposes, there are $n = 985$ words in the encoding vocabulary, and they are roughly equally distributed among our dataset. Then

$$PP(q) = 2^{-\sum_{i=1}^{n} \frac{1}{n} \log_2 q(x_i)} = \prod_{i=1}^{n} q(x_i)^{-1/n} = \sqrt[n]{\frac{1}{q(x_1) \cdots q(x_n)}}.$$

Put into words, the perplexity is the inverse probability of the test set $\{x_1, \ldots, x_n\}$, normalized by the number of words in the test set. Thus, high probabilities correspond to low perplexities, and vice-versa. TensorFlow has sparse cross entropy built in; this is the exponent in the definition of perplexity. Our implementation follows.

```python
def perplexity(labels, logits):

  all_losses = tf.nn.sparse_softmax_cross_entropy_with_logits(labels=labels, logits=logits)
  per_example_losses = tf.reduce_mean(all_losses, axis=-1)
  per_example_perplexities = tf.math.exp(per_example_losses)

  return tf.metrics.mean(per_example_perplexities, name='perplexity')
```

After training our model, we may begin attacking it. For each 4-digit pin n, we feed the encoding of "my pin number is n" into our LSTM. Based on our model's predictions, we can evaluate the perplexity corresponding to n. Their ranking in increasing perplexity order is shown in Table 1.

Rank	PIN	Perplexity
1	**3456**	**2.397483**
2	3356	2.702384
3	3453	2.706784
4	3436	2.730098
5	3466	2.741481
6	3450	2.762213
7	3452	2.781174
8	3451	2.786884
9	3656	2.792649
10	3454	2.796821

Table 1: PINs by increasing perplexity, pre-differential privacy.

3456, our PIN, has the lowest perplexity by a significant margin. To make matters worse, our model memorized the data to such an extreme that the rest of the first 10 PINs are all extremely similar | even if a mistake was made, simply testing the next few options would almost certainly reveal the PIN. Although our data was encoded, we were unable to provide sufficient privacy.

We would like to provide our algorithm with differential privacy so that our database will yield a similar output with or without the PIN datapoint. We will use a `DPAdamGaussianOptimizer`, a replacement for the traditional Adam optimizer for differentially private stochastic gradient descent using the Gaussian mechanism defined above [SCS13]. Instantiating the optimizer requires several differential privacy–related arguments along with the standard arguments for the Adam optimizer. Our hyperparameters are detailed below.

```python
def custom_optimizer():
  ledger = privacy_ledger.PrivacyLedger(
    population_size=len(data),
```

Rank	PIN	Perplexity
1	1919	96.488
2	2001	98.116
3	1200	99.450
4	2002	99.733
5	3200	99.814
6	2003	99.871
7	6200	100.101
8	1819	100.832
⋮	⋮	⋮
3050	**3456**	**137.996**

Table 2: PINs by increasing perplexity, post-differential privacy.

```
    selection_probability=100/len(data)
  )

  optimizer = dp_optimizer.DPAdamGaussianOptimizer(
    l2_norm_clip=1.0,
    noise_multiplier=0.3,
    num_microbatches=10,
    ledger=ledger,
    learning_rate=0.001,
    unroll_microbatches=True
  )

  return optimizer
```

We develop and train our new differentially private model and carry out the same perplexity ranking detailed above. Our results are shown in Table 2. We make a few key observations:

- Most importantly, our PIN (rank 3050) is not among the lowest in perplexity. Thus, an attacker would not be able to use a perplexity-based attack to deduce our PIN, since our model is not memorizing its inputs.
- The PINs ranked in the top 10 indicate high variation. In contrast to our pre-differential privacy results, in which the first and third digits can be deduced with nearly complete certainty, there are multiple possible options for each of the four digits. These numbers are also dissimilar from the true PIN, making trial-and-error an infeasible option.
- The perplexity values themselves are significantly higher than those in Table 1. This is due to increased variation: the algorithmic confidence in each PIN decreases considerably due to random noise.
- As a disclaimer, our comparison of privacy does not enable us to compare the accuracies of our two models. This may be a source of future work on this case study.

6 Conclusion

In this paper, we formally defined differential privacy, discussed its mathematical and intuitive properties, and investigated its usage in real-world scenarios. Here we highlight its key properties:

- Differential privacy is quantitative in nature, with ε as a parameter of both the privacy risk and the degree of accuracy.

- Differential privacy protects against arbitrary risks, beyond just linking attacks, with the presence of past, present, or future auxiliary information. In particular, it is immune to post-processing.
- Multiple differentially private mechanisms can be combined through *composition* with bounded privacy loss.
- Differential privacy also applies to group privacy.

As we saw in the example of Alice, while differential privacy provides an extremely powerful security guarantee, it does not protect individuals from all harm. Instead, it provides that one's participation in a survey can never be disclosed, nor will participation reveal any specific data from the individual's contribution to the survey.

For our future work, we intend to investigate *advanced* composition [DRV10]. It improves on sequential composition in two primary ways. First, it allows for the repeated use of differentially private algorithms on different databases that contain information about the same individual. Second, it enables lower privacy loss due to an improved bound on δ. This process is much more involved.

References

[War65] Stanley L. Warner. "Randomized Response: A Survey Technique for Eliminating Evasive Answer Bias". In: *Journal of the American Statistical Association* 60.309 (1965), pp. 63–69. ISSN: 01621459. URL: http://www.jstor.org/stable/2283137.

[HS97] Sepp Hochreiter and Jürgen Schmidhuber. "Long Short-term Memory". In: *Neural computation* 9 (Dec. 1997), pp. 1735–80. DOI: 10.1162/neco.1997.9.8.1735.

[Dwo+06a] Cynthia Dwork et al. "Calibrating Noise to Sensitivity in Private Data Analysis". In: *Theory of Cryptography*. Ed. by Shai Halevi and Tal Rabin. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, pp. 265–284. ISBN: 978-3-540-32732-5.

[Dwo+06b] Cynthia Dwork et al. "Our Data, Ourselves: Privacy Via Distributed Noise Generation". In: *Advances in Cryptology (EUROCRYPT 2006)*. Vol. 4004. Lecture Notes in Computer Science. Springer Verlag, May 2006, pp. 486–503. URL: https://www.microsoft.com/en-us/research/publication/our-data-ourselves-privacy-via-distributed-noise-generation/.

[NS08] Arvind Narayanan and Vitaly Shmatikov. "Robust De-anonymization of Large Sparse Datasets". In: *2008 IEEE Symposium on Security and Privacy (sp 2008)*. 2008, pp. 111–125. DOI: 10.1109/SP.2008.33.

[DRV10] Cynthia Dwork, Guy N. Rothblum, and Salil Vadhan. "Boosting and Differential Privacy". In: *2010 IEEE 51st Annual Symposium on Foundations of Computer Science*. 2010, pp. 51–60. DOI: 10.1109/FOCS.2010.12.

[SCS13] Shuang Song, Kamalika Chaudhuri, and Anand D. Sarwate. "Stochastic gradient descent with differentially private updates". In: *2013 IEEE Global Conference on Signal and Information Processing*. 2013, pp. 245–248. DOI: 10.1109/GlobalSIP.2013.6736861.

[DR14] Cynthia Dwork and Aaron Roth. *The Algorithmic Foundations of Differential Privacy*. Foundations and trends in theoretical computer science. Now, 2014. URL: https://books.google.com/books?id=Z3p8swEACAAJ.

[WSW15] Yue Wang, Cheng Si, and Xintao Wu. "Regression Model Fitting under Differential Privacy and Model Inversion Attack". In: *Proceedings of the 24th International Conference on Artificial Intelligence*. IJCAI'15. Buenos Aires, Argentina: AAAI Press, 2015, pp. 1003–1009. ISBN: 9781577357384.

[MN16] Daniel Muise and Kobbi Nissim. *Differential Privacy in CDFs*. 2016. URL: https://privacytools.seas.harvard.edu/files/dpcdf_user_manual_aug_2016.pdf.

[Ren+17] Jun Ren et al. "DPLK-Means: A Novel Differential Privacy K-Means Mechanism". In: *2017 IEEE Second International Conference on Data Science in Cyberspace (DSC)*. 2017, pp. 133–139. DOI: 10.1109/DSC.2017.64.

[Swe18] Latanya Sweeney. *Simple Demographics Often Identify People Uniquely*. June 2018. DOI: 10.1184/R1/6625769.v1. URL: https://kilthub.cmu.edu/articles/journal_contribution/Simple_Demographics_Often_Identify_People_Uniquely/6625769/1.

[Woo+18] Alexandra Wood et al. "Differential privacy: A primer for a non-technical audience". In: *Vanderbilt Journal of Entertainment & Technology Law* 21.1 (2018), pp. 209–275. URL: http://www.jetlaw.org/journal-archives/volume-21/volume-21-issue-1/differential-privacy-a-primer-for-a-non-technical-audience/.

[Che+19] Mia Xu Chen et al. "Gmail Smart Compose: Real-Time Assisted Writing". In: *CoRR* abs/1906.00080 (2019). arXiv: 1906.00080. URL: http://arxiv.org/abs/1906.00080.

[SU20] Thomas Steinke and Jonathan Ullman. *The pitfalls of average-case differential privacy*. 2020. URL: https://differentialprivacy.org/average-case-dp/.