Differential Privacy: How to not say too much, but just enough

Maruth Goyal

December 2020

1 Introduction

Gossiping: we all pretend to hate it, and tell ourselves we do not partake in it, but human beings almost inherently have a weirdly strong desire to learn information about each other. In the process you may not just collect information, but also distribute it in the hopes of gaining more through collective addition of knowledge. However, you must be careful. Suppose Alice told you in confidence she's a big fan of the Star Wars prequel series. Now George asks you if you know anyone at UT who likes The Revenge of the Sith. Alice told you her secret in confidence, so you decide to say "Oh, I know a math major who's a big fan". Great, Alice's secret is safe. The next day Jack might ask you if you know anyone taking M367K, and since this wasn't said in confidence you might say "Oh yeah, Alice is taking that class!". Aha! Do you see the problem? Jack and George can now potentially talk, and put 2 and 2 together, and uncover that there is a math major at UT named Alice who is friends with you, who likes the Revenge of the Sith! This is despite you never revealing any private information about Alice in particular! Your friendship with Alice is now forever ruined.

"Jeez, fine, I'll be more careful when I talk with people. We done?", you say. Well, while you might not have been able to tell, I was simply using this story to secretly motivate a more general phenomenon (surprise!). If you take a step back, fundamentally the following is occuring: (1) a party collects public, and private information about someone, (2) the party releases a (supposedly) sufficiently obfuscated version of the private information, (3) the party releases the public information, (4) adversaries are able to combine the public and obfuscated private information to uncover the private information. We knowingly, and unkowingly participate in this sequence on an almost daily basis. Companies collect everything from our location data, to our search history, movie preferences, health data, and everything in between on a daily basis. They promise you your data is secure, and only sell anonymized versions of your data to their clients. However, it has been shown on multiple occasions that just like yourself, companies aren't so good at anonymizing private data.

In 2006, Netflix announced a competition where participants would be given access to anonymized Netflix user data (over $500,000$ records) with the goal of improving the performance of Netflix's recommendation system by at least 10% over multiple iterations of the competition. However, Arvind Naryanan, a PhD student at UT Austin at the time (Hook 'em!), and his advisor Vitaly Shmatikov demonstrated how to identify individual users in the anonymized Netflix data by linking reviews with reviews on IMDB [NS06]. This led to Netflix shutting down the competition. The US Census Beaureau also conducted an internal attack where just from the publicly published summary data they were able to exactly reconstruct microdata for 46% of the participants in the 2010 census, and 71% with small error. More importantly, however, they were able to identify over **50 million** participants **by name** using this attack combined with commercial databases [Abo19].

Importantly, it turns out that these are not just one-off instances where the attackers got lucky. In 2003, Irit Dinur and Kobbi Nissim published a paper [DN03] which introduced the "Fundamental Law of Information Recovery", which provided a mathematical formalism the phenomenon we talked about above. i.e., roughly that overly accurate answers to too many questions destroys privacy. At this point you might be having a crisis, wondering if anything in this world is private anymore. Thankfully, a lot of very smart people have been thinking about this question, and we've made a lot of progress on it. In 2006 Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith introduced the notion of *Differential Privacy* [DMNS06], a new way to formally reason about the privacy guarantees of algorithms. Their work has led to a vast array of work in creating algorithms which are provably private.

As much as I would like to say that you can now safely sleep at night knowing your data is being operated on exclusively by private algorithms, that is unfortunately not the case. Private algorithms have some almost inherent performance limitations, which we will talk about later in this work [PCS$^+$20, HSLA20]. Moreover, they also present some interesting engineering challenges [GL20] (yes, filthy engineering challenges are far below your honor, oh supreme pure mathematician, we know. Go play with your cohomologies on $5-$manifolds or whatever.). It might, however, bring you some peace in knowing that the US 2020 Census will be utilizing differentially private algorithms for their statistics [Bur20].

In the rest of this work, we will follow the following structure: In Section 2 we'll briefly talk about reconstruction attacks more concretely, focusing on some key parts from Dinur and Nissim's work, then in Section 3 we motivate a formal definition of privacy, and define differential privacy, and in Section 4 try get a flavor for some basic differentially private algorithms. From there we take a step back and gander at the limitations of current definitions, and some attempts at bypassing them. We will also briefly explore some out-there work by peeking into differential privacy in the world of Quantum Computers [AR19]!

2 Reconstruction Attacks

2.1 Just Encrypt It?

Before we talk about reconstruction attacks, I want to address the encryption-, I mean elephant, in the room. Why don't we just encrypt our data? That guarantees privacy right? Well, there are 2 problems with that:

- Suppose we only gave companies access to encrypted data. Well, they wouldn't be able to do anything with it unless they are able to decrypt it! Alright, well technically that's a lie since there are schemes for Fully Homomorphic Encryption (FHE) [Gen09] which allow arbitrary computation over encrypted data without decrypting it (these schemes, however, are currently not practical). But even assuming FHE, we have the following issue:

- We are looking for a different guarantee of privacy. With encryption, it is guaranteed that a (perhaps computationally constrained) adversary cannot learn anything from your encrypted message that it didn't already know. However, in the notion of privacy we'll be considering in this work we want to be able to learn information and statistics from the data, but do so in a manner that doesn't reveal private information about the data.

2.2 Back to attacks

In the introduction we discussed a perhaps somewhat juvenile potential "real" instance of a reconstruction attack, in addition to ones performed on real-world data such as the Netflix Prize data set, and the US 2010 Census. In this section we'll take a slightly closer look at some underlying formalisms introduced by Dinur and Nissim for this class of attacks, and even briefly discuss how these attacks are conducted.

In our setup we will have 3 players: (1) you, the nosy black-hoodie wearing adversary, (2) the curator who answers questions, and (3) the analyst who asks the question. The curator has access to a database with private information, and wants to answer questions in an at least somewhat private manner. Your goal is to get the analyst to ask certain questions, and from the answers reconstruct as much of the private data as accurately as you can. Further suppose you have access to UT's supercomputers, so you have just about arbitrarily powerful computation power.

Now, what remains is to model a database, and a query. For simplicity, we will assume the following model:

- A database is a (potentially ordered) collection $\langle c_1, \dots, c_m \rangle$ of vectors in $\{0,1\}^{\beta_i \cdot n}$, where $c \in \mathbb{Z}^+$, and n is the number of rows and m is the number of columns. You may visualize consecutive blocks of β_i bits in a column as being the entries.
- There is a column c_i which contains private information. For simplicity, we will assume $\beta_i = 1$. (for instance, this can encode whether someone has a disease).
- Queries (1) specify conditions to specify which rows to pick, and (2) what data to extract from these rows. We will wlog assume the adversary always asks to extract the private information. Thus, we may denote the row selection criteria as a vector $S \in \{0,1\}^n$, where the i^{th} row is selected iff the i^{th} bit is 1.

We are now ready to state and prove the following theorem:

Theorem 1. *If you can ask 2^n queries, and the curator responds with answers within some error bound E, then you can reconstruct the database in all but $4E$ positions.*

Proof. We would first like to show that the adversary will be able to produce some database, and then bound the error on it. Suppose that the real secret data column is $d \in \{0,1\}^n$. Then, for any candidate database $c \in \{0,1\}^n$, we rule it out iff there's some query S such that $\sum_{i \in S} |c_i - r(S)| > E$, where $r(S)$ is the noisy output from the curator. Observe this is sound since we know the curator will add at most E noise. Output any candidate that is not ruled out (Note at least one database is not ruled out, in particular the real one).

Now consider the query $I_0 \subset \{0,1\}^n$ defined such that $I_0 = \{i \mid d_i = 0\}$. Then, again by assumption on the curator and our candidate database, we have that $\sum_{i \in I_0} |c_i - r(I_0)| \leq E$. Moreover, once again by assumption on the curator we have the same relation with the real database $\sum_{i \in I_0} |d_i - r(I_0)| \leq E$. Intuitively, in the worst case each of them differ from $r(I_0)$ in exactly E positions, but moreover these E positions are disjoint for the candidate database c, and d. Thus, in the worst case they differ in at most $2E$ entries (this may also be seen by triangle inequality). Now you can similarly define $I_1 = \{i \mid d_i = 1\}$, and get that c and d differ in at most $2E$ entries which are supposed to be 1, for a grand total upper bound of at most $4E$ entries where you are incorrect. □

Recall that here n is the number of rows in the database. Observe that if E is sufficiently small in n, you entirely lose privacy! In particular, even if E is linear in n, for instance $E = \frac{n}{500}$, then you are not able to reconstruct only $\approx 1\%$ of the database! i.e., 99% of the database can be recovered even with an error that grows linearly in the number of entries! Thus for error that is asymptotically dominated by n (i.e., significantly smaller), there is basically no privacy. (Note also if you are feeling particularly enthusiastic that with $E = n$ you cannot recover 25% of the database, but then the noisy data is not very useful anyway)

You might not be very convinced since 2^n queries is a lot of queries. Rest assured, there is a weaker but similar results where when you are allowed $O(n)$ queries, with say $\alpha\sqrt{n}$ error, you can recover the database except at most $O(\alpha^2)$ entries. Note this is remarkable since α^2 is independent of n. It has also been shown that under this setting if a constant fraction of responses have *arbitrary* error, you can still succeed.

3 Formal notion of Privacy

3.1 Arriving at a definition

Alright, wonderful. So even if we add some noise to our output (akin to changing a few details in the gossip analogy), the adversary can still recover most of the private data (most of the "tea" in the analogy). Is there any hope? Well, in order for us to determine if there's hope, we need to first define precisely what it is we are hoping for. Informally, as we said in section 2.1, we want to be able to compute and release statistics about our data, *but* also do so in a way that does not reveal private information. Let's try and formalize this.

To get some intuition, let's say the adversary sends us subsets (containing at least 3 students, say) of students as queries, and we report the mean of their test scores. Suppose for simplicity that Alice scored 100%, and John scored 5%, and everyone else is within [45%, 55%]. Then any subset containing Alice will have a mean very close to 100, and any set containing John will have a mean very close to 5. i.e., the presence of Alice, or John in our data set may make outcomes in a certain range significantly more likely than if they weren't (in which case most answers would be somewhere around, say, 50). This can let an adversary learn information about Alice or John's scores! Intuitively a good starting point then seems like defining our statistic as private if certain outcomes do not become much more likely because of a particular entry in the database (say, Alice). This is exactly what *Differential Privacy* tries to capture!

Ok, based on our starting point let's try out the following definition:

Definition 1. *An algorithm $\mathcal{M} : \mathcal{X} \to \mathcal{Y}$ where $\mathcal{X}$ is the space of databases, and $\mathcal{Y}$ is some output space, is said to be ε-"private" iff for $\varepsilon \geq 0$ and any subset $S \subset \mathcal{Y}$, and "similar" databases $X, X' \in \mathcal{X}$ we have that*

$$\Pr[\mathcal{M}(X) \in S] \leq \Pr[\mathcal{M}(X') \in S] + \varepsilon$$

where randomness is over any coin flips made by $\mathcal{M}$

Let's see if this is a good definition. Well, we wanted our definition to capture the idea that changing the database a bit shouldn't really change the likelihood of an outcome by much. Our definition tries to do that by bounding the difference by some additive error. This seems potentially problematic, because by our definition an algorithm $\mathcal{M}$ would be private even if $\Pr[\mathcal{M}(X) \in S] = 10^{-6}$, while $\Pr[\mathcal{M}(X) \in S] = 0.99$ even for $\varepsilon = 10^{-5}$! I don't know about you but that does not look private to me. Since additive error doesn't seem to work, let's try something else. How about multiplicative error?

Definition 2. *An algorithm $\mathcal{M} : \mathcal{X} \to \mathcal{Y}$ where $\mathcal{X}$ is the space of databases, and $\mathcal{Y}$ is some output space, is said to be ε-"private" iff for $\varepsilon \geq 0$ and any subset $S \subset \mathcal{Y}$, and "similar" databases $X, X' \in \mathcal{X}$ we have that*

$$\Pr[\mathcal{M}(X) \in S] \leq \varepsilon \Pr[\mathcal{M}(X') \in S]$$

where randomness is over any coin flips made by $\mathcal{M}$

This seems to look a bit better since now the bound scales according to the probabilities. Let's say we pick $\varepsilon = 1.1$. Our definition then essentially says the algorithm is at most 10% more likely to output any given outcome on a similar database. If we pick $\varepsilon = 1$, then any outcome occurs with at most equal probability. If we're a bit more careful about our analysis we can see this $\varepsilon = 1$ case essentially corresponds to an algorithm that doesn't particularly care about the database at all. Given this, $\varepsilon < 1$ doesn't really make much sense to think about. So let's just shift things a bit:

Definition 3. *An algorithm $\mathcal{M} : \mathcal{X} \to \mathcal{Y}$ where $\mathcal{X}$ is the space of databases, and $\mathcal{Y}$ is some output space, is said to be ε-"private" iff for $\varepsilon \geq 0$ and any subset $S \subset \mathcal{Y}$, and "similar" databases $X, X' \in \mathcal{X}$ we have that*

$$\Pr[\mathcal{M}(X) \in S] \leq (1 + \varepsilon) \Pr[\mathcal{M}(X') \in S]$$

where randomness is over any coin flips made by $\mathcal{M}$

Now $\varepsilon = 0$ corresponds to the case where the algorithm doesn't particularly care about the database. Notationally that seems to make a lot more sense. Since this definition seems pretty good, let's just clean it up by making the notion of "similarity" more concrete. Suppose we define it as follows:

Definition 4. *Two databases X, X' are similar (denoted $X \sim X'$) iff they differ in at most one row.*

Et voila! This is (pretty much) the definition of Differential Privacy introduced in the seminal 2006 paper by Dwork-McSherry-Nissim-Smith [DMNS06]! The only difference is instead of defining the error as $(1 + \varepsilon)$, they picked it to be e^{ε}. This is not much different, since for small values of ε, by a simple Taylor series approximation $e^{\varepsilon} \approx 1 + \varepsilon$. It is, however, much more mathematically convenient to be able to work with just exponents. So here's the final thing:

Definition 5. (ε-Differential Privacy [DMNS06]) *An algorithm $\mathcal{M} : \mathcal{X} \to \mathcal{Y}$ where $\mathcal{X}$ is the space of databases, and $\mathcal{Y}$ is some output space, is said to be ε-differentially private iff for any subset $S \subset \mathcal{Y}$, and databases $X, X' \in \mathcal{X}$ differing in at most one row, we have that*

$$\Pr[\mathcal{M}(X) \in S] \leq e^{\varepsilon} \Pr[\mathcal{M}(X') \in S]$$

where randomness is over any coin flips made by $\mathcal{M}$

We will note, however, that additive error is not entirely useless. In fact, it is very common to consider (ε, δ)-differential privacy where there's an extremely small additive error δ as well as the multiplicative error. While it is out of scope for this work, we refer the reader to [DKM+06, Kam20] for more on approximate differential privacy.

3.2 What's in a definition?

Exactly how useful is this definition though? Would privacy by any other definition smell as sweet? Well, a good way to do this might be to look at what other properties our definition might provide about algorithms which satisfy it. For instance, perhaps the first scenario that pops into your head is "Wait, what if I take the output of this ε-whatever private thingamabob and then apply some other algorithm to it? Can I lose privacy?". That's a great, and important question to ask. It wouldn't be particularly nice if something like adding 1 to our output would somehow destroy any semblance of privacy. The wonderful thing is differential privacy, as it turns out, is immune to post-processing! If you take the output of a ε-differentially private algorithm and do something to it, it will still remain ε-differentially private. Let's try and prove this.

Theorem 2. *Suppose $\mathcal{M} : \mathcal{X} \to \mathcal{Y}$ is a ε-differentially private algorithm. Then for any algorithm $\mathcal{A} : \mathcal{Y} \to \mathcal{Z}$ we have that $\mathcal{A} \circ \mathcal{M} : \mathcal{X} \to \mathcal{Z}$ is also ε-differentially private.*

Proof. Since $\mathcal{M}$ is ε-DP, For any databases $X, X' \in \mathcal{X}$ such that $X \sim X'$, and subset $S \subset \mathcal{Y}$ we know the following inequality is satisfied:

$$\Pr[\mathcal{M}(X) \in S] \leq e^{\varepsilon} \Pr[\mathcal{M}(X') \in S]$$

We would like to show the following holds for any subset $T \subset \mathcal{Z}$:

$$\Pr[\mathcal{A} \circ \mathcal{M}(X) \in T] \leq e^{\varepsilon} \Pr[\mathcal{A} \circ \mathcal{M}(X') \in T]$$

Define $A^{-1}(T)$ as the set such that for $x \in A^{-1}(T)$ there is some choice of coin flips so that $A(x) \in T$. Then we may observe

$$\begin{aligned} \Pr[\mathcal{A} \circ \mathcal{M}(X) \in T] &= \Pr[\mathcal{M}(X) \in A^{-1}(T)] \\ \Pr[\mathcal{A} \circ \mathcal{M}(X') \in T] &= \Pr[\mathcal{M}(X') \in A^{-1}(T)] \end{aligned}$$

But since $\mathcal{M}$ is ε-DP, we have

$$\Pr[\mathcal{A} \circ \mathcal{M}(X) \in T] = \Pr[\mathcal{M}(X) \in A^{-1}(T)] \leq e^{\varepsilon} \Pr[\mathcal{M}(X') \in A^{-1}(T)] = e^{\varepsilon} \Pr[\mathcal{A} \circ \mathcal{M}(X') \in T]$$

Thus, $\mathcal{A} \circ \mathcal{M}$ is also ε-DP! □

Whew! That's pretty good, right? We don't have to worry about what an adversary might do with our results as such since we're still guaranteed privacy that's at least as strong. Ok great, so our definition is safe with respect to post-processing. What else? Let's think about what other scenarios we would like to have some guarantees about. Well, odds are we do not just want to run just one algorithm, or just run an algorithm once on our database. For instance if our database represents a histogram, we might want to know the mean, but also the entries in a given bucket. This naturally leads to the question of what, if any, privacy guarantees we have if we run a ε_1-DP and ε_2-DP algorithm, and return both of their outputs?

To get some intuition for what should happen, let's go back to our gossip analogy. If someone tells you something (potentially with omissions or changes to preserve privacy), and then tells you another thing then at the end of the day you have learned more information about the private data than you did from just the one thing. Thus, it would make sense that returning the output of 2 differentially private algorithms would result in a potential loss (well, decrease) in privacy.

Theorem 3. *Given $\mathcal{M}_1 : \mathcal{X} \to \mathcal{Y}_1$, $\mathcal{M}_2 : X \to \mathcal{Y}_2$ where $\mathcal{M}_1$ is ε_1-DP, and $\mathcal{M}_2$ is ε_2-DP, the algorithm $\mathcal{M} : \mathcal{X} \to \mathcal{Y}_1 \times \mathcal{Y}_2$ defined as $\mathcal{M}(X) = (\mathcal{M}_1(X), \mathcal{M}_2(X))$ is $\varepsilon_1 + \varepsilon_2$-DP.*

Proof.

$$\begin{aligned}\Pr[\mathcal{M}(X) = (r_1, r_2)] &= \Pr[\mathcal{M}_1(X) = r_1 \wedge \mathcal{M}_2(X) = r_2] = \Pr[\mathcal{M}_1(X) = r_1] \Pr[\mathcal{M}_2(X) = r_2)] \\ &\leq e^{\varepsilon_1} \Pr[\mathcal{M}_1(X') = r_1] e^{\varepsilon_2} \Pr[\mathcal{M}_2(X') = r_2] = e^{\varepsilon_1 + \varepsilon_2} \Pr[\mathcal{M}(X') = (r_1, r_2)]\end{aligned}$$

□

Observe now by a standard induction argument this theorem can be generalized as follows:

Theorem 4. *Given $\mathcal{M}_i : \mathcal{X} \to \mathcal{Y}_i$ with $i \in \{1, \ldots, k\}$, where$\mathcal{M}_i$ is ε_i-DP, the algorithm $\mathcal{M} : \mathcal{X} \to \prod_{i=1}^{k} \mathcal{Y}_i$ defined as $\mathcal{M}(X) = \prod \mathcal{M}_i(X)$ is $\sum_{i=1}^{k} \varepsilon_i$-DP.*

Essentially our loss of privacy is additive. While in some sense it's not "amazing", it's still pretty good! It could have been multiplicative (the horror!). At this point, our definition seems pretty solid since (1) it satisfies all the criteria we'd asked for, (2) it is immune to post-processing, (3) it composes almost naturally. The choice of e^{ε} over $(1 + \varepsilon)$ also becomes more apparent now given how clean all the calculation has been (imagine having to multiply multivariate polynomials!).

4 "Just add noise" Privacy

"Jeez! I get it, things are not private, and your fancy definition is the cure to cancer. What do you do with it though?". Yes, yes, you're right. We have spent a bunch of time motivating, and setting up things but it's finally time to try and actually take a stab at making things private. We'll take a quick look at essentially the first general mechanism to generate differentially private statistics introduced in [DMNS06].

This algorithm is popularly known as the "Laplace Mechanism", the reason for which shall be apparent soon. Going back to the idea of making the probability of certain outcomes similar with respect to similar databases, the idea with the Laplace mechanism was to add an element of randomness. In particular, suppose you have some algorithm $f : \mathcal{X} \to \mathbb{R}^n$. Then, the privatized version $f^* : \mathcal{X} \to \mathbb{R}^n$ is defined as

$$f^*(X) = f(X) + L(\rho)$$

Where ρ is a parameter we'll elaborate on later, and $L(\rho)$ represents a sample of n i.i.d samples from a particular random distribution. The idea is that we can tailor this distribution to some property of the function we are computing, such that the probability of a given outcome is bounded for neighboring databases. In particular, we will be sampling from the Laplacian distribution which as the following density:

$$\frac{1}{2b} \exp(\frac{-x}{b})$$

Observe this is essentially the symmetric version of the exponential distribution. The parameter ρ from earlier will correspond to b in this density. Intuitively, if we want to have a distribution that essentially masks the effect of changing at most one entry in the underlying database a good quantity to focus on might be how much our function changes when we perturb the input. i.e., if you change the input slightly, how much does the output change? Thus we define the following quantity:

Definition 6. *The ℓ_1-sensitivity of a function $f : \mathcal{X} \to \mathbb{R}^n$, $\Delta(f)$ is defined as*

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

Thus, it is the maximum change in f over all neighboring databases. We then define our parameter ρ as $\rho = \frac{\Delta(f)}{\varepsilon}$. Thus, in total we have the following:

Definition 7. (Laplace Mechanism) *Given a function $f : \mathcal{X} \to \mathbb{R}^n$, and privacy parameter ε we define the privatized version $f^* : \mathcal{X} \to \mathbb{R}^n$ as*

$$f^*(X) = f(X) + L(\frac{\Delta(f)}{\varepsilon}$$

where L represents n i.i.d samples from the Laplacian distriubtion with the respective parameter.

Naturally, as you would expect we have the following theorem about this procedure:

Theorem 5. *The Laplace mechanism is ε-differentially private.*

Proof. This fact is seen from the following calcualtion:

$$\begin{aligned} \frac{\Pr[f^*(X) = z]}{\Pr[f^*(X') = z]} &= \frac{\prod_{i=1}^n \exp\left(-\frac{\varepsilon|f(X)_i - z_i|}{\Delta(f)}\right)}{\prod_{i=1}^n \exp\left(-\frac{\varepsilon|f(X')_i - z_i|}{\Delta(f)}\right)} \leq \prod_{i=1}^n \exp\left(-\varepsilon|f(X)_i - f(X')_i|\right) \\ &= \exp(\frac{-\varepsilon \sum_{i=1}^n f(X)_i - f(X')_i}{\Delta(f)} \leq \exp(-\varepsilon) \end{aligned}$$

The first inequality follows from the reverse triangle inequality, and the latter from the definition of ℓ_1 sensitivity. □

This is actually a pretty impressive result. We are now able to take any function for which we can compute the sensitivity, and almost automatically produce a differentially private version! Rather remarkable if you ask me. Note it also does not do this in a trivial manner since the accuracy loss is bounded by $O(1/\varepsilon n)$, though we do not prove this here. This, however, is not perfect. You might think, for instance, what if the noisy answer just absolutely throws off this other high sensitivity function I'm computing? A good example of this might be auction prices. Even a slight fluctuation could be the difference between a sale and a bust. Fear not dear auctioneer, for the "Exponential Mechanism" [MT07] is here to save you (we omit discussion, but refer the reader to the cited material).

You might also be scratching your head as to why the Laplacian was chosen as opposed to a more normal distribution (pun absolutely intended) like the Gaussian. The short answer is, the Laplacian just naturally operates very well with our definitions. However, the Gaussian can most certainly be used, and is indeed used in the context of approximate differential privacy, where the guarantee need only hold with some probability $1 - \delta$ for choice of parameter δ. We refer the interested reader to [Kam20, DKM+06].

5 Benefits and Limitations

Differential Privacy is awesome, I can now privately compute anything in the world, ... right? Well, not really. While as we have seen above the definition certainly works quite well and has a lot of desirable properties in addition to having very intuitive realizations, it has some (almost inherent in some sense) limitations. A rather clear one is loss of accuracy. By restricting the distribution of outcomes, we are inherently introducing some error in the output which restricts our ability to learn arbitrary statistics to some degree. In fact, for some common machine learning tasks such as image classification the best differentially private implementations have accuracies of about 75% [PCS+20], which is strikingly less than the about 99% accuracy achieved by state-of-the-art ML models! However, we say that some degree of such loss in performance is almost seemingly inherent as attempts to sidestep the strong guarantees of DP have mostly failed. A recent, and rather famous and controversial instance is InstaHide [HSLA20]. This paper claimed to introduce a technique which allowed private learning without losing accuracy, or causing any slow down unlike DP using some fancy encoding. However, among the multitude of issues this paper had [Car20], perhaps the most glaring was that it wasn't really private! Researchers managed to recover the original images from the output of InstaHide [CDG$^+$20], violating any sense of privacy.

However, accuracy is not the only issue with DP. One thing we took for granted above, for instance, while talking about the Laplace mechanism was being able to sample from some random distribution. Random sampling is a very well-explored and nonetheless highly non-trivial computational task. For large enough tasks, a truly differentially private implementation would require copious amounts of high quality random data, as was discovered by people working on the 2020 US Census [GL20]. They had to generate over 90 Terrabytes of high quality random bits! To give the reader a sense of how hard this problem is, companies like Cloudflare have resorted to live videos of a wall of lava lamps to reliably generate random bits as opposed to just principled psuedo-random-generators (PRNGs).

At the end of the day, however, this field is barely over a decade old. While there will probably always be an accuracy-privacy tradeoff, what remains is the philosophical argument of how much is your privacy worth? Would you accept worse search results if at least Google guaranteed your data was used in a privacy-preserving manner (you bet your bottom dollar I would!).

6 Future Directions

The story of differential privacy, interestingly enough does not stop within the paradigm we have been discussing. Recently, UT's own Scott Aaronson published a paper discussing differential privacy in the setting of Quantum computing! Interestingly, the paper was born of coincidence. Dr. Aaronson was giving a talk about "gentle measurements", wherein you attempt to measure a quantum state in a manner that doesn't totally destroy the superposition (for instance, measuring in a basis which contains the state itself, and states orthogonal to it would work). Guy Rothblum, a foremost expert on Differential Privacy was in the audience and observed the actual technique was similar in nature to the Laplace mechanism, and thus was born quantum differential privacy [AR19]. There is now increasing interest in this sub-genre of DP too [BO20].

Besides other paradigms, DP has also managed to pique the interests of law schools around the country [CD].

7 Acknowledgements

I am really grateful to my DRP mentor Hunter Vallejos. Getting to talk with him about differential privacy and math in general over the past semester was super fun, and enriching. I'd also really like to thank Gautam Kammath for making his excellent lecture notes on the subject public. I also owe gratitude to Cynthia Dwork, whom I had the great honor of getting to listen to deliver a presentation of differential privacy at FCRC 2019 in Phoenix!

References

[Abo19] John Abowd. Tweetorial: Reconstruction-abetted re-identification attacks and other traditional vulnerabilities. https://twitter.com/john$_{a}bowd/status/$1114942180278272000, 2019.

[AR19] Scott Aaronson and Guy N Rothblum. Gentle measurement of quantum states and differential privacy. In *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing*, pages 322–333, 2019.

[BO20] Costin Bădescu and Ryan O'Donnell. Improved quantum data analysis. *arXiv preprint arXiv:2011.10908*, 2020.

[Bur20] US Census Bureau. Disclosure avoidance and the 2020 census, 2020.

[Car20] Nicholas Carlini. Instahide disappointingly wins bell labs prize, 2nd place. https://nicholas.carlini.com/writing/2020/instahide$_disappointingly_wins_bell_labs_prize.html$, 2020.

[CD] Rachel Cummings and Deven Desai. The role of differential privacy in gdpr compliance.

[CDG$^+$20] Nicholas Carlini, Samuel Deng, Sanjam Garg, Somesh Jha, Saeed Mahloujifar, Mohammad Mahmoody, Shuang Song, Abhradeep Thakurta, and Florian Tramer. An attack on instahide: Is private learning possible with instance encoding? *arXiv preprint arXiv:2011.05315*, 2020.

[DKM$^+$06] Cynthia Dwork, Krishnaram Kenthapadi, Frank McSherry, Ilya Mironov, and Moni Naor. Our data, ourselves: Privacy via distributed noise generation. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 486–503. Springer, 2006.

[DMNS06] Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. In *Theory of cryptography conference*, pages 265–284. Springer, 2006.

[DN03] Irit Dinur and Kobbi Nissim. Revealing information while preserving privacy. In *Proceedings of the twenty-second ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, pages 202–210, 2003.

[Gen09] Craig Gentry. Fully homomorphic encryption using ideal lattices. In *Proceedings of the forty-first annual ACM symposium on Theory of computing*, pages 169–178, 2009.

[GL20] Simson L. Garfinkel and Philip Leclerc. *Randomness Concerns When Deploying Differential Privacy*, page 73–86. Association for Computing Machinery, New York, NY, USA, 2020.

[HSLA20] Yangsibo Huang, Zhao Song, Kai Li, and Sanjeev Arora. InstaHide: Instance-hiding schemes for private distributed learning. In Hal Daumé III and Aarti Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 4507–4518, Virtual, 13–18 Jul 2020. PMLR.

[Kam20] Gautum Kamath. Lecture 5 – approximate differential privacy. http://www.gautamkamath.com/CS860notes/lec5.pdf, 2020.

[MT07] Frank McSherry and Kunal Talwar. Mechanism design via differential privacy. In *48th Annual IEEE Symposium on Foundations of Computer Science (FOCS'07)*, pages 94–103. IEEE, 2007.

[NS06] Arvind Narayanan and Vitaly Shmatikov. How to break anonymity of the netflix prize dataset. *arXiv preprint cs/0610105*, 2006.

[PCS$^+$20] Nicolas Papernot, Steve Chien, Shuang Song, Abhradeep Thakurta, and Ulfar Erlingsson. Making the shoe fit: Architectures, initializations, and tuning for learning with privacy, 2020.