Introduction to Differential Privacy
Lesson 1 : The notion of Differential Privacy

Clément Lalanne

December 3, 2020

Abstract

In this lesson, we investigate the information leaked by the results of an algorithm. We use a basic poll example to introduce the need for information control allowed by Differential Privacy. We then introduce and study its most common variants, Probabilistic Differential Privacy and Approximate Differential Privacy.

1 Introduction

Let us introduce this topic by taking a concrete example. Let us say you have acces to the results of the polls of your target and you want to gather information about this target. This poll is binary with a Yes or No answer. You gain access to the answer of your target and now what ? Well if the answer is not encrypted and is fully deterministic, you have the full knowledge, GG, you won. Now, let us say that the process that gives the answer is somewhat randomized. You cannot be certain of your target's answer. But you can still gain information from this answer.

Let us formalize this a bit. The true result can be either D_1: YES, or D_2: NO. And you have access to $O = A(D)$ which is the output of the true answer D by the random answer mechanism A.

The knowledge that you're interested in is $\mathbb{P}(D = D_1|A(D) = O)$ or $\mathbb{P}(D = D_2|A(D) = O) = 1 - \mathbb{P}(D = D_1|A(D) = O)$. Indeed, it means "what's the probability of the true answer being D_1 or D_2 knowing the output that we observed is O". In order the have a symetrical and synthetic interpretation, we are going to look at the odd ratio or betting ratio.

$$\frac{\mathbb{P}(D = D_1|A(D) = O)}{\mathbb{P}(D = D_2|A(D) = O)}$$

If this ratio is small, you know that $(D = D_2)$ is very likely. Conversely, if it is high, $(D = D_1)$ is very likely. So what's a good ratio ? There is no absolute answer to that question except a ratio that doesn't leak too much information compared to what you already knew. If you had prior information on $\mathbb{P}(D = D_1)$ and $\mathbb{P}(D = D_2)$, a good algorithm is an algorithm that bounds the information gained by what you knew,

$$a\frac{\mathbb{P}(D = D_1)}{\mathbb{P}(D = D_2)} \leq \frac{\mathbb{P}(D = D_1|A(D) = O)}{\mathbb{P}(D = D_2|A(D) = O)} \leq b\frac{\mathbb{P}(D = D_1)}{\mathbb{P}(D = D_2)} \tag{1}$$

for $0 \leq a \leq b$. This means that your updated guess on the true answer cannot be too far from your initial one. Note that if you do not have any initial guess, you can always take an uninformative prior $\mathbb{P}(D = D_1) = \mathbb{P}(D = D_2) = 1/2$.

2 ϵ-Differential Privacy (ϵ-DP)

2.1 Definition

More generally, in a setup where D is not necessarilly a binary input but a database. we are interested in a definition of privacy that limits the information leaked on an individual level with the same type of guarantees as presented in the last example: Not allowing any substantial gain of information. A good definition for that is the notion of Differential Privacy that was introduce by [DMNS06, DKM$^+$06].

Definition 1 (Differential Privacy). *A randomized function $\mathcal{K}$ gives ϵ-differential privacy if for all data sets D_1 and D_2 differing on at most one element, and all $S \subseteq Range(\mathcal{K})$*

$$\mathbb{P}(\mathcal{K}(D_1) \in S) \leq e^{\epsilon} \times \mathbb{P}(\mathcal{K}(D_2) \in S) \tag{2}$$

The probability is taken over the coin tosses of $\mathcal{K}$.

Let us clarify a bit the meaning of this definition. By database, we mean the input data of the algorithm. It can be a simple scalar input like in the example from the introduction. But it can be more complicated, for instance with a set of values from which we would like to perform some sort of statistics of even the training set of a machine learning algorithm. By differing on one element, we mean a value that changes or that is removed or added from or to the database. What defines a value is what we consider as the individual level that we want to protect.

Remark 1. *The definition is symetrical in D_1 and D_2 and hence yields to the following inequality:*

$$e^{-\epsilon} \times \mathbb{P}(\mathcal{K}(D_2) \in S) \leq \mathbb{P}(\mathcal{K}(D_1) \in S) \leq e^{\epsilon} \times \mathbb{P}(\mathcal{K}(D_2) \in S) \tag{3}$$

Let us get back to the introduction example. We can check that, if the algorithm A is ϵ-differentially private for some ϵ, we have the expected properties. By Bayes rule, we have:

$$\frac{\mathbb{P}(D = D_1 | A(D) = O)}{\mathbb{P}(D = D_2 | A(D) = O)} = \frac{\mathbb{P}(D = D_1)}{\mathbb{P}(D = D_2)} \frac{\mathbb{P}(A(D_1) = O)}{\mathbb{P}(A(D_2) = O)}$$

Furthermore, if A is ϵ-DP, the ratio $\frac{\mathbb{P}(A(D_1)=O)}{\mathbb{P}(A(D_2)=O)}$ is bounded by $e^{-\epsilon}$ and e^{ϵ}. So, we have:

$$e^{-\epsilon} \frac{\mathbb{P}(D = D_1)}{\mathbb{P}(D = D_2)} \leq \frac{\mathbb{P}(D = D_1 | A(D) = O)}{\mathbb{P}(D = D_2 | A(D) = O)} \leq e^{\epsilon} \frac{\mathbb{P}(D = D_1)}{\mathbb{P}(D = D_2)} \tag{4}$$

We get the expected form. And if we want to have a better representation of this gained knowledge, we can get rid of $\mathbb{P}(D = D_2)$ and $\mathbb{P}(D = D_2 | A(D) = O)$ which gives the following:

$$\frac{\mathbb{P}(D = D_1)}{e^{\epsilon} + (1 - e^{\epsilon})\mathbb{P}(D = D_1)} \leq \mathbb{P}(D = D_1 | A(D) = O) \leq \frac{e^{\epsilon}\mathbb{P}(D = D_1)}{1 + (e^{\epsilon} - 1)\mathbb{P}(D = D_1)} \tag{5}$$

1 gives a comprehensive understanding of the information leaked depending on multiple values of ϵ. The case

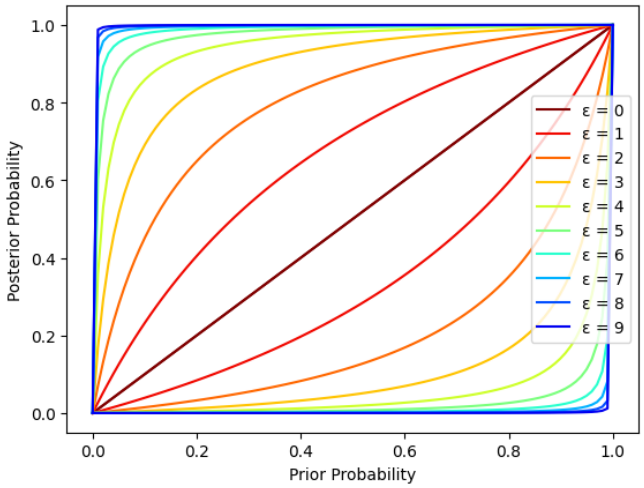


Figure 1: Lower and Uppber bounds of Prior/Posterior updates for different values of ϵ

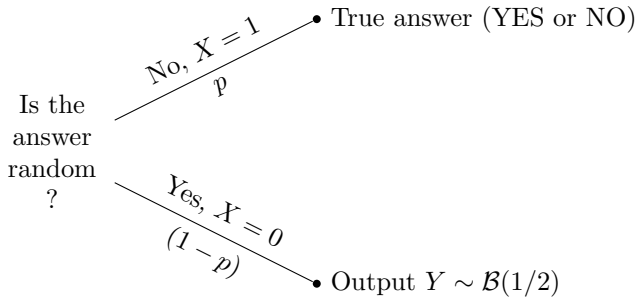
ϵ=0 corresponds to no information leaked at all. In this case, the prior and posterior distributions are the same. Realistically, only small values of ϵ can be considered in order to protect privacy. For values of $\epsilon \sim 10$, we can see that the algorithm can potentially leak almost certainty in only one run.

2.2 Discussing the definition

Let us investigate the definition a bit more. We saw that the database can be a single user input. In this setup, the notion of privacy seems natural. Indeed this is the scenario where you do not trust the information-gathering agent. It can leak information on you and so on. But this database can be more complex and in this case the need of privacy is not so obvious. For instance, if the database is the salaries of the employees of a company and the algorithm returns their mean, how can one get back to an individual salary ? First there are the outliers. If a salary is really different compared to the others, it can have a huge impact on the mean. And then there are the more hacky attacks where an attacker has a lot of prior knowledge on the database. The more extreme example of this is if the entire database except the target leagues to communicate on their salaries and hence can deduce the salary of the target. In a less trivial setup, [BDK07] shows that it is possible to reidentify with high probability any community of size n in a social graph from a coalition of users of size $\log(n)$. Long story short, the interesting part of differential privacy is that it makes no assumption on the database or on the data distribution. The individual is protected no matter the prior knowledge on the data base which results in a high protection to these types of attacks.

2.3 Building a Differentially Private poll algorithm

Now that we saw why it is important to have privacy preserving algorithms, we can wonder how can we build one ? A method known and often used to make polls is to aks a real answer with probability p and to answer at random with probability $(1-p)$.



If $D = 1$ corresponds to YES and $D = 0$ corresponds to NO, we can write:

$$A(D) = XD + (1 - X)Y \tag{6}$$

where $X \sim \mathcal{B}(p)$ and $Y \sim \mathcal{B}(1/2)$ are independent and drawn independently among all the runs of A. Intuitivelly, it is easy to see why this method is more private than directly asking the answer. Indeed, even by knowing the output, it is impossible to know for sure if the answer has been given because it is the truth or because it has been drawn randomly. But is this approach differentially private ?

2.3.1 What about Differential Privacy ?

Let us consider the two following ratios:

$$\frac{\mathbb{P}(A(D) = 1|D = 1)}{\mathbb{P}(A(D) = 1|D = 0)} = \frac{p + (1-p)/2}{(1-p)/2} = 1 + 2\frac{p}{1-p} \tag{7}$$

$$\frac{\mathbb{P}(A(D) = 0|D = 0)}{\mathbb{P}(A(D) = 0|D = 1)} = \frac{p + (1-p)/2}{(1-p)/2} = 1 + 2\frac{p}{1-p} \tag{8}$$

Since there are only two possible outputs and inputs, we covered all the cases. It showns that A is $\log\left(1 + 2\frac{p}{1-p}\right)$-DP. Quite intuitivelly, this also confirms that the smaller p is, the more private this survey is. Furthermore, for small values of p, A is almost $2p$-DP.

2.3.2 The accuracy of the data

By doing such polls, we ensure that the data collection is differentially private, but for what ? Can this data be exploited ? At an individual level and with only one run of the algorithm, the answer is "not really". However, one might want to gather statistical information about a population from these results. Can it be done and how much do we lose by protecting privacy ? A statistician typically models D by a random variable that follows a Bernoulli law of unknown parameter μ that has to be estimated from the real data. If we take the expectancy of $A(D)$ with respect to the joint distribution on A and D, we get:

$$\begin{aligned}\mathbb{E}\left(A(D)\right) &= \mathbb{E}\left(XD + (1-X)Y\right) \\ &= \mathbb{E}\left(X\right)\mathbb{E}\left(D\right) + \mathbb{E}\left((1-X)\right)\mathbb{E}\left(Y\right) \\ &= p\mu + (1-p)/2\end{aligned} \tag{9}$$

So, if we have access to independent samples $O_1, ..., O_n$ of $A(D)$, we have, thank to the strong law of large numbers, a strongly consistent estimator of μ by taking:

$$\hat{\mu}_n = \frac{1}{p}\left(\frac{1}{n}\sum_{i=1}^{n} O_i - \frac{1-p}{2}\right) \tag{10}$$

This means that $\mathbb{E}[\hat{\mu}_n] = \mu$ and that $\hat{\mu}_n$ converges almost surely to μ. In order to get information on the convergence, let us compute the variance of O.

$$\begin{aligned}V(O) &= V(A(D)) \\ &= V(XD + (1-X)Y) \\ &= V(XD) + V((1-X)Y) + 2Cov(XD, (1-X)Y) \\ &= \mathbb{E}[X^2]\mathbb{E}[D^2] - \mathbb{E}[X]^2\mathbb{E}[D]^2 \\ &+ \mathbb{E}[(1-X)^2]\mathbb{E}[Y^2] - \mathbb{E}[(1-X)]^2\mathbb{E}[Y]^2 \\ &+ 2\left(\mathbb{E}\left(XD(1-X)Y\right) - \mathbb{E}[XD]\mathbb{E}[(1-X)Y]\right) \\ &= p\mu - p^2\mu^2 + (1-p)/2 - (1-p)^2/4 - 2p\mu(1-p)/2\end{aligned} \tag{11}$$

This variance makes sense since in the case $p = 0$, we have $V(O) = 1/4$ which is the variance on a Bernoulli of parameter $1/2$ (fully randomized answer) and the case $p = 1$ (totally accurate answer) gives $V(O) = \mu(1-\mu)$ which is the variance of a Bernoulli of parameter μ.

The Central limit theorem allows us to write that

$$\sqrt{n}(\hat{\mu} - \mu) \rightsquigarrow \mathcal{N}(0, v(p, \mu)) \tag{12}$$

$$\sqrt{n}\frac{\hat{\mu} - \mu}{\sqrt{v(p, \mu)}} \rightsquigarrow \mathcal{N}(0, 1) \tag{13}$$

where

$$v(p, \mu) = \mu/p - \mu^2 + (1-p)/2p^2 - (1-p)^2/4p^2 - \mu(1-p)/p. \tag{14}$$

Let $\alpha \in (0, 1)$ and Φ the cummulative distribution function of $\mathcal{N}(0, 1)$.

$$\mathbb{P}\left(\sqrt{n}\frac{\hat{\mu} - \mu}{\sqrt{v(p, \mu)}} \in (-\Phi(1-\alpha/2), \Phi(1-\alpha/2))\right) \rightarrow_{n\to\infty} 1 - \alpha \tag{15}$$

This formula gives us asympotic confidence bounds for μ and can be used to have an idea on the feasability of our poll. We can compare it to the natural setup without privacy in which we simply ask the truth (case $p = 1$).

$$\mathbb{P}\left(\sqrt{n}\frac{\hat{\mu}_{p=1} - \mu}{\sqrt{\mu(1-\mu)}} \in (-\Phi(1-\alpha/2), \Phi(1-\alpha/2))\right) \rightarrow_{n\to\infty} 1 - \alpha \tag{16}$$

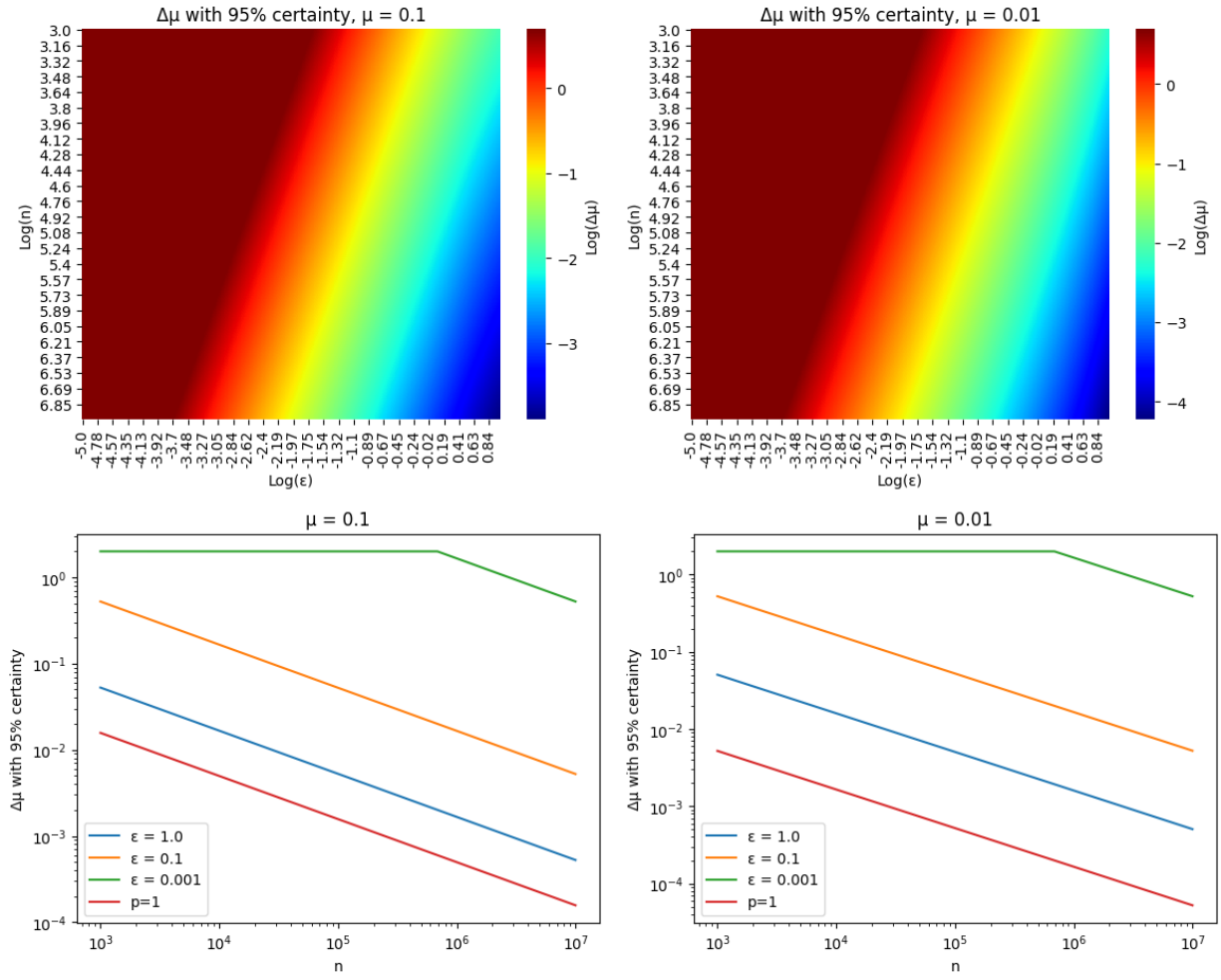


Figure 2: Asymptotic incertitudes on μ for multiple sample sizes and values of ϵ

2.3.3 Tradeoffs between Privacy and Accuracy

Thank to 16, we can start to quantify the question that we had in mind since the begining: What do we lose in order to protect privacy ? The figure 2 confirms our intuition. The more privacy we want (smaller values of epsilon), the higher the size of the sample should be in order to get the same incertainty. This idea will araise many times when talking about privacy protection. The size of the data, the level of privacy and the quality of the results are the three core parameters to keep in mind and many tradeoffs involving those three variables will have to be considered.

2.4 Properties

Now that we defined Differential Privacy and studied it on an example, let us look at its general properties.

2.4.1 The vulterability of a group

We said that differential privacy is good if you want to protect privacy at an individual level. But what happens if you want to protect groups of people ? Do you lose all the nice guarantees you had ? Luckilly, no, you only lose privacy proportinally to the size of the group. This is logical since the data you're trying to protect has significantly more importance relatively to the whole dataset and hence influences the output more.

Proposition 1 (Group Scaling [DR$^+$14]). *If $\mathcal{K}$ is ϵ-DP and D_1 and D_2 differ on k inputs, then*

$$\mathbb{P}(\mathcal{K}(D_1) \in S) \leq e^{k\epsilon} \times \mathbb{P}(\mathcal{K}(D_2) \in S) \tag{17}$$

For all $S \subset Range(\mathcal{K})$.

Proof. We decompose the path between D_1 and D_2 in k segments that only differ on one input and apply the ϵ-DP property. □

All in all, this means that for groups of size k, you get $k\epsilon$-differential privacy. This property is really useful since it allows to know what degree of privacy leaks in the worst case for groups of inputs, communities that would try to stay hidden... This way, privacy at an individual level provides privacy at a group level.

2.4.2 It is impossible to hack the results

Now, a really important question of security that you have to ask yourself is: "Can your results be hacked somehow ?". Even though my output is ϵ-DP, can someone modify it and make it less private ? A very strong property of differential privacy is that the answer is no. At least on a mathematical point of view.

Proposition 2 (Post Processing [DR+14])**.** *If $\mathcal{K}$ is ϵ-DP and f is a random function independent of $\mathcal{K}$, then $f(\mathcal{K})$ is also ϵ-DP.*

Proof. First, supose that f is deterministic. Let D_1 and D_2 be two possible inputs and $S \subset \text{Range}(\mathcal{K})$. We note $T = f^{-1}(S)$.

$$\begin{aligned} \mathbb{P}(f(\mathcal{K}(D_1)) \in S) &= \mathbb{P}(\mathcal{K}(D_1) \in T) \\ &\leq e^{\epsilon}\mathbb{P}(\mathcal{K}(D_2) \in T) \\ &= e^{\epsilon}\mathbb{P}(f(\mathcal{K}(D_2)) \in S) \end{aligned} \tag{18}$$

Now, f is a random variable (built uppon Ω). We can write,

$$\begin{aligned} \mathbb{P}(f(\mathcal{K}(D_1)) \in S) &= \int_{\Omega} \mathbb{P}(f_\omega(\mathcal{K}(D_1)) \in S)\mathbb{P}(d\omega) \\ &\leq \int_{\Omega} e^{\epsilon}\mathbb{P}(f_\omega(\mathcal{K}(D_2)) \in S)\mathbb{P}(d\omega) \\ &= e^{\epsilon}\mathbb{P}(f(\mathcal{K}(D_2)) \in S) \end{aligned} \tag{19}$$

□

This makes the task of releasing data really easy since people do not have to figure out how their algorithm can be hacked. None can process the information to gain more knowledge.

2.4.3 One can control the information leaked over time

If we consider the poll example again. Imagine that we run the algorithm multiple times on the same person. What happens to privacy? Well for the first iterations, nothing too scary will happen. However, after a few runs, the true answer will start to become dominant. It is possible to quantify this statistically but this is also pretty intuitive. This shows that privacy is deteriored by running the algorithm multipe times. Luckilly, it is possible to quantify the privacy loss.

Proposition 3 (Composition Theorem [DR+14])**.** *If $\mathcal{K}_1, ..., \mathcal{K}_k$ are respectivelly $\epsilon_1, ..., \epsilon_k$-DP and are independent, then $(\mathcal{K}_1, ..., \mathcal{K}_k)$ is $\epsilon_1 + ... + \epsilon_k$-DP.*

Proof. Let D_1 and D_2 be two databases that only differ on one input. Let $S_1 \subset \text{Range}(\mathcal{K}_1), ..., S_k \subset \text{Range}(\mathcal{K}_k)$.

$$\begin{aligned} &\mathbb{P}\left((\mathcal{K}_1, ..., \mathcal{K}_k)(D_1) \in (S_1 \times ... \times S_k)\right) \\ &= \mathbb{P}(\mathcal{K}_1(D_1) \in S_1)...\mathbb{P}(\mathcal{K}_k(D_1) \in S_k) \\ &\leq e^{\epsilon_1}\mathbb{P}(\mathcal{K}_1(D_2) \in S_1)...e^{\epsilon_k}\mathbb{P}(\mathcal{K}_k(D_2) \in S_k) \\ &= e^{\epsilon_1+...+\epsilon_k}\mathbb{P}\left((\mathcal{K}_1, ..., \mathcal{K}_k)(D_2) \in (S_1 \times ... \times S_k)\right) \end{aligned} \tag{20}$$

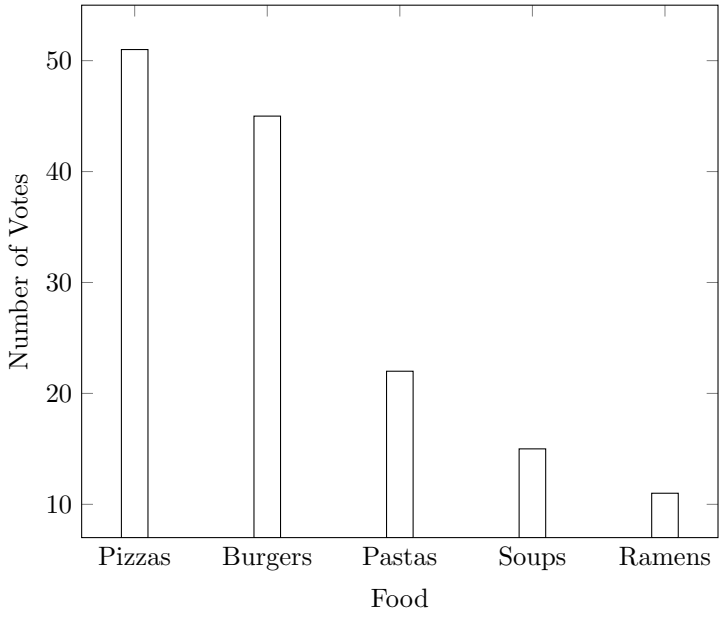
□

Remark 2. *This property is often called Composition Theorem since in practice the result of a $\mathcal{K}_i$ depends on the results of the previous ones but these results should not affect the differentially private property with respect to the database variable.*

This property is really usefull in practice since it allows to combine DP algorithms and have end to end guarantees. For instance, a campany would like to release multiple informations on a database. Each piece of information is released differential privately but can an attacker get more informations by joining the multiple outputs ? Well yes but it is possible to have a control on this information and even better, this is still differentially private !

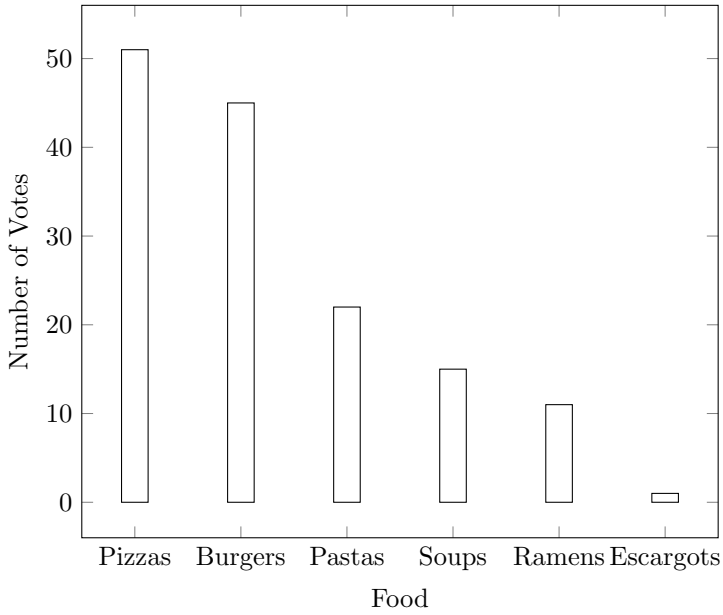
3 Probabilistic Differential Privacy (PDP)

Although we saw the power of Differential Privacy, it can sometimes be hard to achieve in practice. This is the reason why araised the notion of probabilistic differential privacy in which we allow (with small probability) the possibility that everything goes wrong. But first, let us consider a small example. Let us say you want to share an histogram that represents for instance people's favorite food.



In order to respect the privacy of the voters, you cannot release the histogram like that. It is not as clear as for the poll collection but an attacker can have a really strong knowledge of the database that would lead to your answer. For instance if there is a coalition of the entire database against you. A commonly used method to achieve differential privacy on such queries is to add noise to each column and return the result. It might not be clear right now that it achieves differential privacy but it will be covered in the next lesson. For now, let us stick to the idea that blurring the result adds privacy.

However, in the case where the categories are not fixed and the user can input whatever they want, this doesn't work. Indeed, let us consider the following histogram.



It has been made from the same database as the first one with one differing input ("Escargots"). However, applying noise to the columns will still reveal that new category. Hence, it leaks that "Escargots" is the last input in the database and the differential privacy property has no chance to be satisfied. Furthermore, it seems hard to achieve differential privacy without considering all the different foods, misspellings and possible trolls in the survey. Indeed, if the input is left free to the user, the aim of the survey is to be adaptative. So the output has to be adaptative and one cannot decide to hide some outputs all the time.

A solution is to threshold: After applying the noise to the columns, you only release the ones that are above a certain number. The idea is that, this way, only the categories that have many votes will be released resulting in an almost invariant support for databases that only differ one one input and hence acting almost like a differentially private algorithm. What does almost meant ? It means that this process has a chance to fail with a (hopefully small) probability. What is that probability ? In the previous example, it is the probability that 1 (the count on "Escargots") plus the noise exceeds the threshold. This led to the definition of Probabilistic Differential Privacy.

Definition 2 (Probabilistic Differential Privacy [Mei18])**.** *A randomized function $\mathcal{K}$ gives (ϵ, δ)-probabilistic differential privacy if for all data sets D_1 and D_2 differing on at most one element, there exists a set $S^\delta \subset Range(\mathcal{K})$ such that $\mathbb{P}(\mathcal{K}(D_1) \in S^\delta) \leq \delta$ and for all $S \subset Range(\mathcal{K})$*

$$\mathbb{P}(\mathcal{K}(D_1) \in S \setminus S^\delta) \leq e^\epsilon \times \mathbb{P}(\mathcal{K}(D_2) \in S \setminus S^\delta) \tag{21}$$

The probability is taken over the coin tosses of $\mathcal{K}$.

This definition has the advantage of being intuitive: In some cases we have differential privacy and in others, that are bounded in probability, things can go wrong. However, it is way harder to minipulate than the notion of differential privacy. For instance, this property is not preserved under post processing.

Proposition 4 (Post Processing Hack [Mei18])**.** *Probabilistic Differential Privacy is not preserved under post processing.*

Proof. Let us consider the following randomized function.

$$\mathcal{K}(0) = \begin{cases} 0 \text{ with probability } \delta \\ 1 \text{ with probability } \frac{e^\epsilon}{1+e^\epsilon}(1-\delta) \\ 2 \text{ with probability } \frac{1}{1+e^\epsilon}(1-\delta) \\ 3 \text{ with probability } 0 \end{cases} \tag{22}$$

$$\mathcal{K}(1) = \begin{cases} 0 \text{ with probability } 0 \\ 1 \text{ with probability } \frac{1}{1+e^\epsilon}(1-\delta) \\ 2 \text{ with probability } \frac{e^\epsilon}{1+e^\epsilon}(1-\delta) \\ 3 \text{ with probability } \delta \end{cases} \tag{23}$$

It satisfies (ϵ, δ)-PDP. We define the function T as:

$$T(x) = \begin{cases} 4 \text{ if } x = 0 \\ 4 \text{ if } x = 1 \\ 2 \text{ if } x = 2 \\ 3 \text{ if } x = 3 \end{cases} \tag{24}$$

Then, $\mathbb{P}\left(T(\mathcal{K}(0)) = 4\right) > e^{\epsilon}\mathbb{P}\left(T(\mathcal{K}(1)) = 4\right)$ and $\mathbb{P}\left(T(\mathcal{K}(0)) = 4\right) > \delta$, which shows that $T(\mathcal{K})$ cannot be (ϵ, δ)-PDP. □

However, the following property gives a much cleaner relation between probabilities.

Proposition 5 (Approximate Differential Privacy Relaxation)**.** *Let $\mathcal{K}$ be a (ϵ, δ)-probabilistic differentially private algorithm. For all D_1 and D_2 differing on only one input we have*

$$\mathbb{P}(\mathcal{K}(D_1) \in S) \leq e^{\epsilon} \times \mathbb{P}(\mathcal{K}(D_2) \in S) + \delta \tag{25}$$

Proof. Let D_1 and D_2 differ on only one input. Let $S \subset \text{Range}(\mathcal{K})$.

$$\begin{aligned} \mathbb{P}(\mathcal{K}(D_1) \in S) &= \mathbb{P}(\mathcal{K}(D_1) \in S \setminus S^{\delta}) + \mathbb{P}(\mathcal{K}(D_1) \in S^{\delta}) \\ &\leq e^{\epsilon}\mathbb{P}(\mathcal{K}(D_2) \in S \setminus S^{\delta}) + \delta \\ &\leq e^{\epsilon}\mathbb{P}(\mathcal{K}(D_2) \in S) + \delta \end{aligned} \tag{26}$$

where S^{δ} is the set given by the definition. □

It is not equivalent to PDP but it is much more handy. The study of this inequality is the objective of the next section.

4 Approximate Differential Privacy (ADP)

We saw that in some setups, pure differential privacy is hard to get and hence the concept of probabilistic differential privacy araised. This concept is not as handy to manipulate as pure differential privacy is, but satisfies the same pobability distortion ratio with an extra additive term. This inequality is used to define Approximate Differential Privacy that can be seen as a relaxation of PDP.

Definition 3 (Approximate Differential Privacy [DR+14])**.** *A randomized function $\mathcal{K}$ gives (ϵ, δ)-approximate differential privacy if for all data sets D_1 and D_2 differing on at most one element, and all $S \subseteq$ Range$(\mathcal{K})$*

$$\mathbb{P}(\mathcal{K}(D_1) \in S) \leq e^{\epsilon} \times \mathbb{P}(\mathcal{K}(D_2) \in S) + \delta \tag{27}$$

The probability is taken over the coin tosses of $\mathcal{K}$.

Remark 3. *We shown in last section that every (ϵ, δ)-PDP mechanism is also (ϵ, δ)-ADP.*

This notion is often vulgarized the same way as PDP is. Namelly, that everything is fine with probability $1 - \delta$ but can be problematic with probability δ. This can be convenient for the reasoning but it is not generally true. Be careful when you use such an argument.

So why do we use this notion instead of PDP ? It is because it is easy to use. Indeed, most of the properties satisfied by pure DP are also satisfied by ADP.

Proposition 6 (Group Scaling [DR+14])**.** *If $\mathcal{K}$ is ϵ-DP and D_1 and D_2 differ on k inputs, then*

$$\mathbb{P}(\mathcal{K}(D_1) \in S) \leq e^{k\epsilon} \times \mathbb{P}(\mathcal{K}(D_2) \in S) + \delta\left(\frac{e^{k\epsilon} - 1}{e^{\epsilon} - 1}\right) \tag{28}$$

For all $S \subset$ Range$(\mathcal{K})$.

Proof. Let us decompose the path between D_1 and D_2 in databases that differ only on one input pairwise $D^{(0)} = D_1, ..., D^{(k)} = D_2$.

$$\begin{aligned}
\mathbb{P}(\mathcal{K}(D_1) \in S) &= \mathbb{P}(\mathcal{K}(D^{(0)}) \in S) \\
&\leq e^{\epsilon}\mathbb{P}(\mathcal{K}(D^{(1)}) \in S) + \delta \\
&\leq e^{\epsilon}\left(e^{\epsilon}\mathbb{P}(\mathcal{K}(D^{(2)}) \in S) + \delta\right) + \delta \\
&\leq e^{\epsilon}\left(e^{\epsilon}...\left(e^{\epsilon}\mathbb{P}(\mathcal{K}(D^{(k)}) \in S) + \delta\right)...+\delta\right) + \delta \\
&= e^{k\epsilon}\mathbb{P}(\mathcal{K}(D^{(k)}) \in S) + \delta + e^{\epsilon} + ... + e^{(k-1)\epsilon} \\
&\leq e^{k\epsilon} \times \mathbb{P}(\mathcal{K}(D_2) \in S) + \delta\left(\frac{e^{k\epsilon}-1}{e^{\epsilon}-1}\right)
\end{aligned} \tag{29}$$

□

Proposition 7 (Post Processing [DR$^+$14])**.** *If $\mathcal{K}$ is (ϵ, δ)-DP and f is a random function independent of $\mathcal{K}$, then $f(\mathcal{K})$ is also (ϵ, δ)-DP.*

Proof. Same proof as for pure DP. □

Proposition 8 (Composition Theorem [DR$^+$14])**.** *If $\mathcal{K}_1, ..., \mathcal{K}_k$ are respectivelly $(\epsilon_1, \delta_1), ..., (\epsilon_k, \delta_k)$-DP and are independent, then $(\mathcal{K}_1, ..., \mathcal{K}_k)$ is $(\epsilon_1 + ... + \epsilon_k, \delta_1 + ... + \delta_k)$-DP.*

Proof. Let us consider the case with only $\mathcal{K}_1$ and $\mathcal{K}_2$. The rest will follow by induction. Let D_1 and D_2 be two databases that differ on only one input. For any $C_1 \subseteq \text{Range}(\mathcal{K}_1)$, we define,

$$\mu(C_1) = \mathbb{P}(\mathcal{K}_1(D_1) \in C_1) - e^{\epsilon_1}\mathbb{P}(\mathcal{K}_1(D_2) \in C_1) \tag{30}$$

Then μ is a measure on $\text{Range}(\mathcal{K}_1)$ that satisfies $\mu(\text{Range}(\mathcal{K}_1)) \leq \delta_1$. So,

$$\mathbb{P}(\mathcal{K}_1(D_1) \in ds_1) \leq e^{\epsilon_1}\mathbb{P}(\mathcal{K}_1(D_2) \in ds_1) + \mu(ds_1) \tag{31}$$

Furthermore,

$$\begin{aligned}
\mathbb{P}((s_1, \mathcal{K}_2(D_1)) \in S) &\leq (e^{\epsilon_2}\mathbb{P}((s_1, \mathcal{K}_2(D_2)) \in S) + \delta_2) \wedge 1 \\
&\leq (e^{\epsilon_2}\mathbb{P}((s_1, \mathcal{K}_2(D_2)) \in S) \wedge 1) + \delta_2
\end{aligned} \tag{32}$$

As a consequence,

$$\begin{aligned}
\mathbb{P}((\mathcal{K}_1, \mathcal{K}_2)(D_1) \in S) &\leq \int_{S_1} \mathbb{P}((s_1, \mathcal{K}_2(D_1)) \in S)\mathbb{P}(\mathcal{K}_1(D_1) \in ds_1) \\
&\leq \int_{S_1} ((e^{\epsilon_2}\mathbb{P}((s_1, \mathcal{K}_2(D_2)) \in S) \wedge 1) + \delta_2)\mathbb{P}(\mathcal{K}_1(D_1) \in ds_1) \\
&\leq \int_{S_1} ((e^{\epsilon_2}\mathbb{P}((s_1, \mathcal{K}_2(D_2)) \in S) \wedge 1))\mathbb{P}(\mathcal{K}_1(D_1) \in ds_1) + \delta_2 \\
&\leq \int_{S_1} ((e^{\epsilon_2}\mathbb{P}((s_1, \mathcal{K}_2(D_2)) \in S) \wedge 1))(e^{\epsilon_1}\mathbb{P}(\mathcal{K}_1(D_2) \in ds_1) + \mu(ds_1)) + \delta_2 \\
&\leq e^{\epsilon_1+\epsilon_2}\int_{S_1} \mathbb{P}((s_1, \mathcal{K}_2(D_2)) \in S)\mathbb{P}(\mathcal{K}_1(D_2) \in ds_1) + \mu(S_1) + \delta_2 \\
&= e^{\epsilon_1+\epsilon_2}\mathbb{P}((\mathcal{K}_1, \mathcal{K}_2)(D_2) \in S) + \delta_1 + \delta_2
\end{aligned} \tag{33}$$

□

Remark 4. *As in the pure DP case, this theorem is called composition theorem since the result of an algorithm can depend on the results of the previous ones but it cannot affect the differential privacy property with respect to the database.*

It is possible to obtain tighter bounds as shown in [KOV15]. However, this one is still really handy and shows that the joint information of multiple outputs is still ruled by differential privacy.

These three properties were the core of differential privacy and the fact that they still hold for approximate differential privacy is amazing. Indeed it means that this definition allows to have quantitative control over the information leaked by the algorithm when an attacker is able to process the output, join it with other sources made from the same database and even try to attack a group. This is why ADP is used so much in practice.

The following proposition (even if it comes from a brutal majoration) gives a good understanding of the relation between ϵ and δ: The smaller ϵ is, the bigger δ has to be for the same process. It acts like a dump parameter for the exessive quarantees on the probability measure.

Proposition 9 ([Mei18]). *If $\mathcal{K}$ is (ϵ, δ)-DP, then for any $0 \leq \epsilon' \leq \epsilon$, $\mathcal{K}$ is $(\epsilon', \delta + e^{\epsilon} - e^{\epsilon -})$-DP.*

Proof.

$$\begin{aligned} \mathbb{P}(\mathcal{K}(D_1) \in S) &\leq e^{\epsilon}\mathbb{P}(\mathcal{K}(D_2) \in S) + \delta \\ &= e^{\epsilon'}\mathbb{P}(\mathcal{K}(D_2) \in S) + (e^{\epsilon} - e^{\epsilon'})\mathbb{P}(\mathcal{K}(D_2) \in S) + \delta \\ &\leq e^{\epsilon'}\mathbb{P}(\mathcal{K}(D_2) \in S) + e^{\epsilon} - e^{\epsilon'} + \delta \end{aligned} \tag{34}$$

□

A δ is considered good if it scales in a way that is dominated by any inverse of a polynomial function in the size of the database [DR+14]. It might appear unclear at this stage why these tradeoffs have some importance right now but it will be clear in the next lessons.

5 A quick look at Encryption

To end this first lesson that aimed at presenting the basics of differential privacy, we can look at encryption. The point of encryption is to create a function that is inrevertible in reasonable (e.g. polynomial) time without a key. Usually, none considers encryption as a leak of privacy. However, according to the definition of differential privacy, this process has no chance to be private at all. Indeed, encryption is or is almost injective. In order to take this case into account, it is possible to redefine differential privacy. Instead of taking strong guarantees on the probability distribution, we impose that this distribution is indistiguishable from a differentially private one by any adversary A in a given calss.

Definition 4 (Computational Differential Privacy [Mei18]). *A randomized function $\mathcal{K}$ gives (ϵ, δ)-computational differential privacy if for all data sets D_1 and D_2 differing on at most one element, and all polynomial time probabilistic Turing machine A,*

$$\mathbb{P}(A(\mathcal{K}(D_1)) = 0) \leq e^{\epsilon} \times \mathbb{P}(A(\mathcal{K}(D_2)) = 0) + \delta \tag{35}$$

The probability is taken over the coin tosses of $\mathcal{K}$.

Remark 5. *It is possible to obtain all the properties of ADP for CDP.*

6 Conclusion

We saw that differential privacy and its variants, that allow some flexibility, give strong privacy guarantees at an individual level. This means that even with a strong prior knowledge on the database, no attacker can learn too much from the algorithm. Furthermore, we saw that DP and ADP have some handy group preserving, hack free and information tractability over joint information properties making them really usable and tractable in practice. In the next lesson, we are goint to see some generic methods in order to achieve differential privacy on a wide range of problems. Finally, if you want some more information on differential privacy, you can look at the really good blog [Des] by Damien Desfontaines which was also really useful to write this lesson.

References

[BDK07] Lars Backstrom, Cynthia Dwork, and Jon Kleinberg. Wherefore art thou r3579x? anonymized social networks, hidden patterns, and structural steganography. In *Proceedings of the 16th international conference on World Wide Web*, pages 181–190, 2007.

[Des] Damien Desfontaines. Ted is writing things. `https://desfontain.es/privacy/index.html`.

[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.

[DR+14] 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.

[KOV15] Peter Kairouz, Sewoong Oh, and Pramod Viswanath. The composition theorem for differential privacy. In *International conference on machine learning*, pages 1376–1385. PMLR, 2015.

[Mei18] Sebastian Meiser. Approximate and probabilistic differential privacy definitions. *IACR Cryptol. ePrint Arch.*, 2018:277, 2018.