An Nguyen
Jul 1, 2019 · 15 min read ★ · Listen

Understanding Differential Privacy

From Intuitions behind a Theory to a Private AI Application.

Privacy by Nick Youngson CC BY-SA 3.0 Alpha Stock Images

This article is a concise introduction to Differential Privacy. By reading, you will go from intuitions and reasons behind important concepts of Differential Privacy, such as privacy loss, the relationship between privacy loss and accuracy (of differential-privacy outputs). These intuitions, then, will be explained by concrete illustrative stories and reinforced by qualitative and quantitative analyses via programming. What's expected at the end of the article is an application of Private AI with *Private Aggregation of Teacher Ensembles* algorithm (PATE) on MNIST handwritten digits dataset.

Note: The application will require the reader must be familiar with Pytorch.

> "Differential privacy makes it possible for tech companies to collect and share aggregate information about user habits, while maintaining the privacy of individual users."
>
> The Conversation

What are the practical applications of Differential Privacy?

In the 21st century, we were confronting many big data breaches that necessitate governments, organizations, and companies to give a reconsideration of privacy. In contrast to that, almost breakthroughs in Machine Learning come from learning techniques that require a large amount of training data. Besides, research institutions often use and share data containing sensitive or confidential information about individuals. Improper disclosure of such data can have adverse consequences for a data subject's private information, or even lead to civil liability or bodily harm.

The development of formal privacy models likes Differential Privacy was helping in solving the problem. Thus, there is an increasing number of organizations and companies were applying Differential Privacy to protect sensitive information, such as personal information, user's events, individual real-time location, as mentioned in this post: A High-level Introduction to Differential Privacy. There is even an open-source differential privacy project for executing differential privacy queries on any standard SQL

In short, Differential Privacy permits:
— Companies access a large number of sensitive data for researching and business without privacy breaches.
— Research institutions can develop differential privacy technology to automate privacy processes within cloud-sharing communities across countries. Thus, they could protect the privacy of users and resolve the data-sharing problem.

Introduction to Differential Privacy

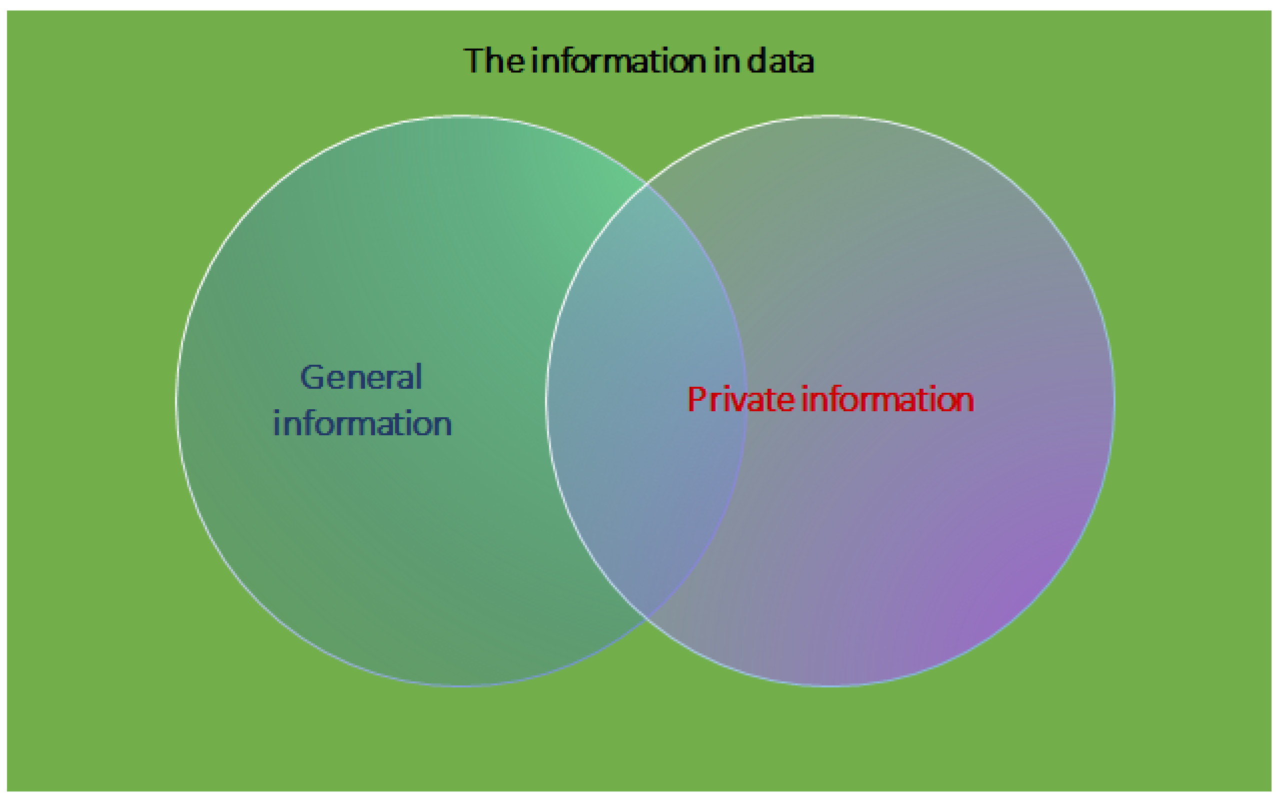


Figure 1: Information in data under Differential Privacy view.

What is Differential Privacy?
— Differential privacy (DP) is a strong, mathematical definition of privacy in the context of statistical and machine learning analysis. According to this mathematical definition, DP is a criterion of privacy protection, which many tools for analyzing
sensitive personal information have been devised to satisfy. [1].

The above diagram represents the information contained in data under the view of DP. Thus, general information is any information, that doesn't specific to any individual data subject. General information can be understood as the information of the entire population in the data (not only an individual or a group of data subjects). The contrast of general information is private information, what specifics to any individual data subject.

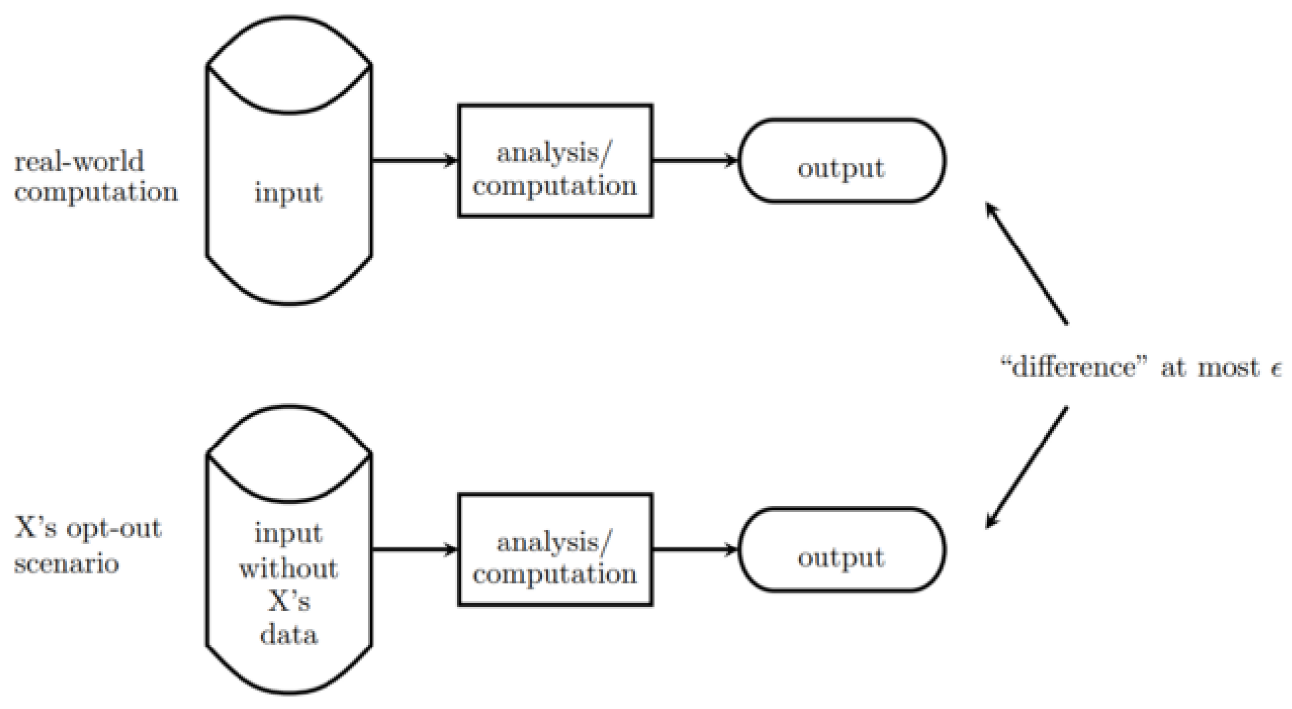


Figure 2 [1]: Differential privacy.

How can we distinguish between private information and general information?
— In the view of DP, private information is the change of information in data when before and after opting-out of an individual data subject (illustrated in Figure 2). This also explains the "differential" word in the name.

- **What does it guarantee?**
  — Differential privacy mathematically guarantees that anyone seeing the result of a differentially private analysis will essentially make the same inference about any individual's private information, whether or not that individual's private information is included in the input to the analysis. [1]
  — DP provides a mathematically provable guarantee of privacy protection against a wide range of *privacy attacks (*include *differencing attack*, *linkage attacks,* and *reconstruction attacks)* [2].
- **What does it not guarantee?**
  DP does not guarantee that one believes to be one's secrets will remain secret.

It's important to identify which is general information and which is private information to get benefits from DP umbrella and reduce harm. DP guarantees to protect only private information (mentioned above). So, if one's secret is general information, it will be not protected!

To understand this, let's consider a scenario when you, a smoker, decided to be included in a survey. Then, analysis on the survey data reveals that smoking causes cancer. Will you, as a smoker, be harmed by the analysis? Perhaps — Based on the fact that you're a smoker, one may guess at your health status. It is certainly the case that he knows more about you after the study than was known before (this is also the reason behind saying it is "general information", not "public information"), but was your information leaked? Differential privacy will take the view that it was not, with the rationale that the impact on the smoker is the same independent of whether or not he was in the study. It is the conclusions reached in the study that affect the smoker, not his presence or absence in the data set. [2]

How does it work?

> *Let consider a canonical example to see how a DP algorithm, what satisfies DP criterion, works: Image that you are a social data scientist, who wants to perform an analysis on a survey data about a very taboo behavior. Each entry in the data is an answer (the truth) of individuals, "yes" or "no", in the surveyed population. Because of a privacy policy, the data holder, or curator, never permits you for direct access to the data.*

You, a DP expert, suggested to the curator a DP algorithm for removing private information in the data, whereby you can perform the analysis on the data. Thus, for each entry, the curator will apply this algorithm:

- Flip a coin (the coin's bias is the probability that its outcome is head and it will be denoted as *p_head*.).
- If heads, return the answer in the entry.
- If tails, then flip a second coin and return "yes" if heads and "no" if tails.

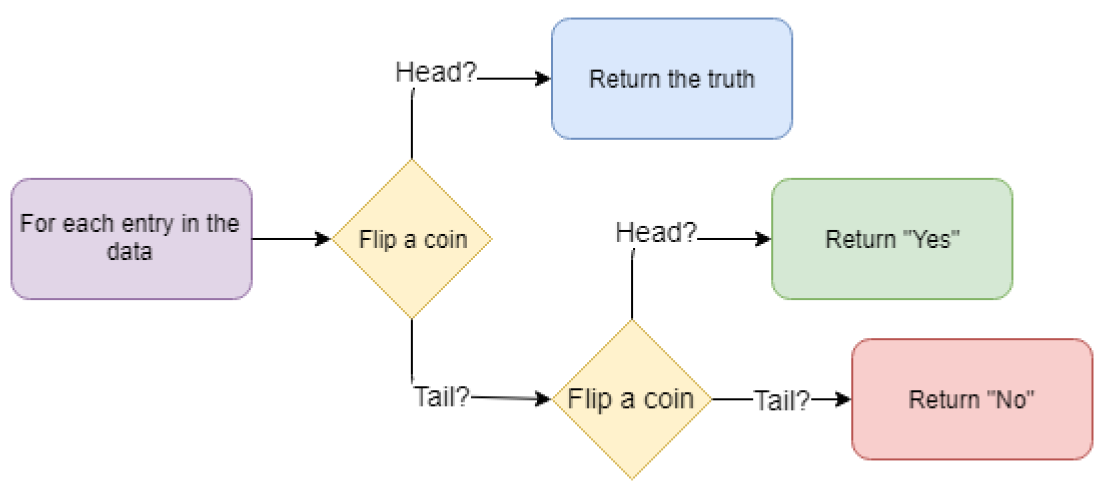
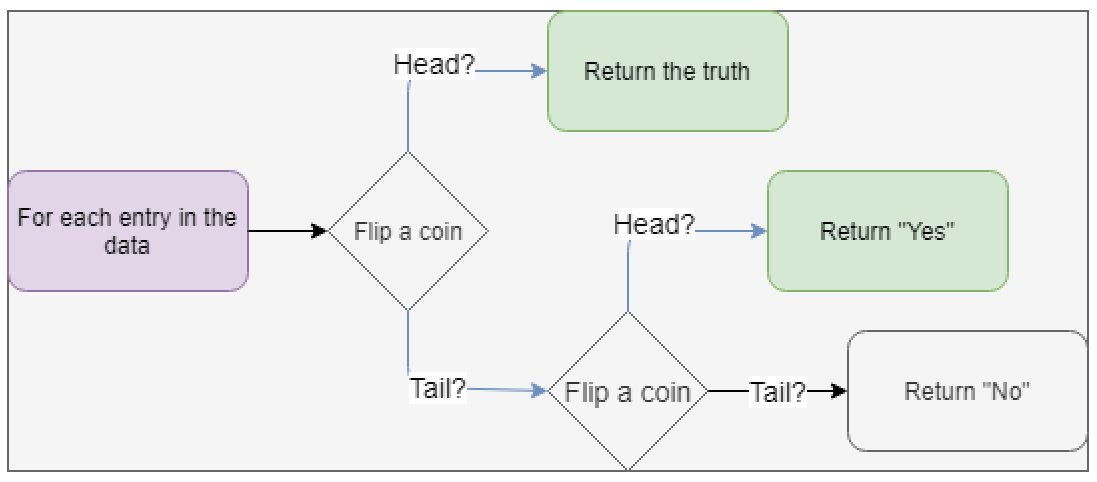


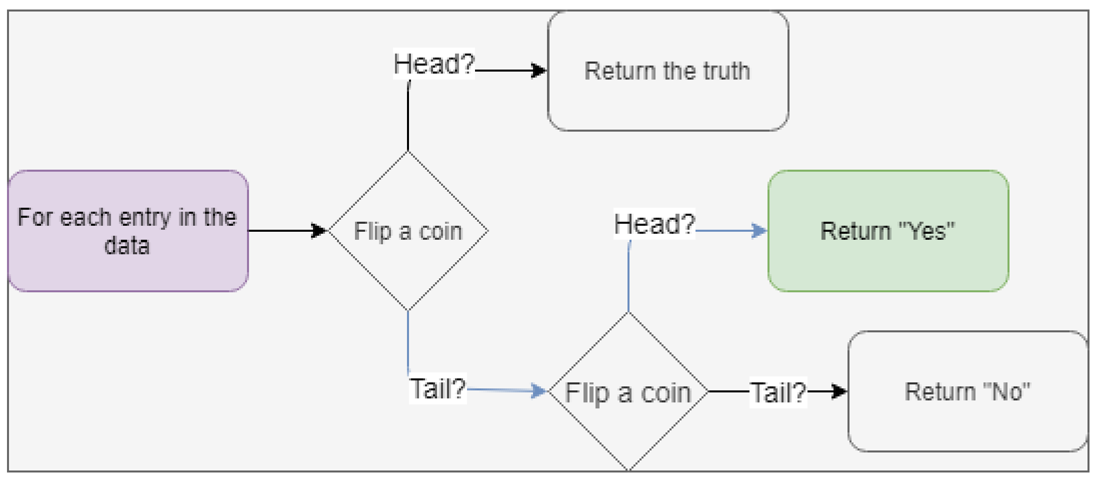
Figure 3. Flow diagram of the Differential privacy algorithm.

Now, each person is protected with "plausible deniability", because a person is plausible to deny the answer by the randomness of flipping a coin. Assume you want to infer the percentage of innocents in the population (*p_innocent*) from that noisy data. It can be done by these steps:

- Compute probability of returning "yes" given that individual isn't an innocent: P("yes"|not innocent) = *p_head+(1-p_head)*p_head.*



- Compute probability of returning "yes" given that individual is an innocent: P("yes"|innocent) = (1-*p_head*)**p_head*.



- Compute *p_innocent*:
p_innocent = (P("yes")-P("yes"|not innocent))/(P("yes"|innocent)-P("yes"|not innocent)) = 1-(P("yes")-(1-*p_head*)**p_head*)/*p_head*. (proof by deduction)

Note: Above result is an asymptotic result guaranteed by the Law of Large Numbers.

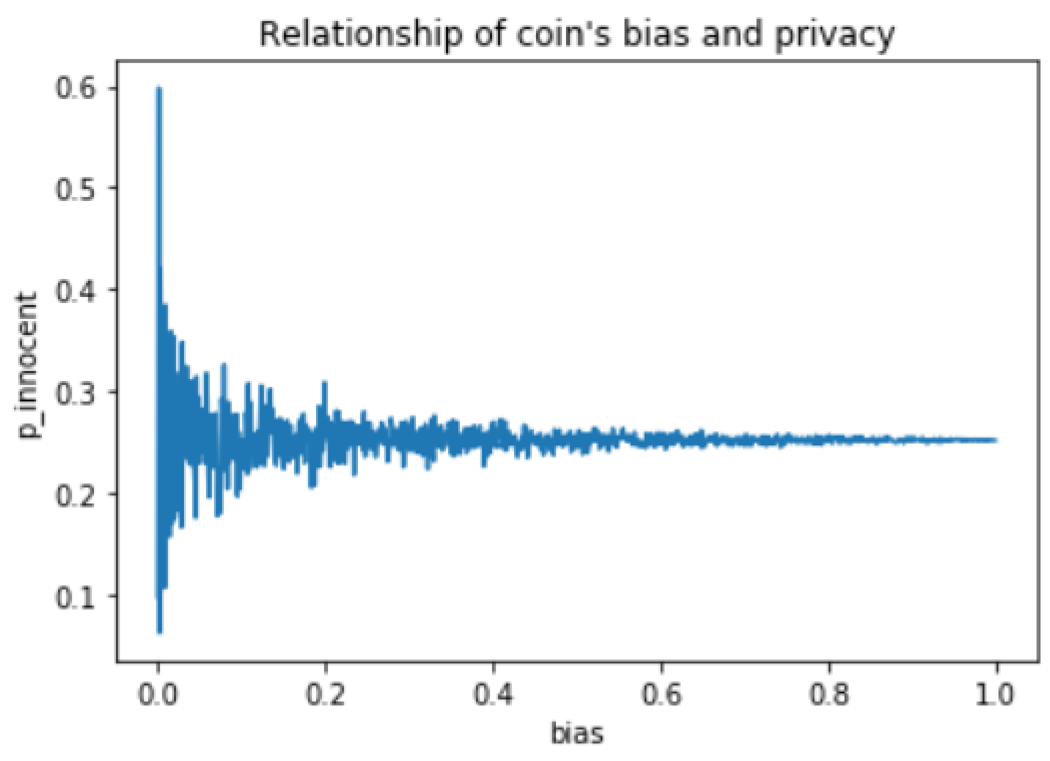


Figure 4: Compute on a simulated survey data of 20,000 individuals. Low bias coins add more noise to the data. A consequence of adding noise is the decrease in privacy loss.

- What does DP tell us?
As you can see in Figure 4, the variance of *p_innocent* increases dramatically and approaches infinity when *p_head* approaches 0, this leads to a rapid decrease in privacy loss. DP also gives us the same conclusion. Thus, when *p_head* is 0, the distribution of returned result is identical, no matter an individual is innocent or not (the distance of 2 distributions is P("yes"|not innocent)-P("yes"|innocent)=*p_head*, the bias). If the number of innocents participating in the data changed, it does not lead to any changes in information in the noisy returned data. It means that there is no private information in the noisy returned data.

Reasons for Differential Privacy

DP has valuable properties [2] makes it becomes a rich framework for analyzing sensitive personal information and privacy protection:

1. ***Quantification of privacy loss.***
Privacy loss is a measure in any DP mechanisms and algorithms. It permits comparisons among different techniques. Privacy loss is controllable that ensures a trade-off between it and the accuracy of general information.
2. ***Composition.***
The quantification of loss permits the analysis and control of cumulative privacy loss over multiple computations. Understanding the behavior of differentially private mechanisms under composition enables the design and analysis of complex differentially private algorithms from simpler differentially private building blocks.
3. ***Group Privacy.***
DP permits the analysis and control of privacy loss incurred by groups, such

as families.

4. ***Closure Under Post-Processing.***
DP is immune to post-processing: A data analyst, without additional knowledge about the private database, cannot compute a function of the output of a differentially private algorithm and make it less differentially private.

Measure Differential Privacy and Report Noisy Max

Now, we are ready for a mathematical definition of Differential Privacy. Let's take a look at it:

> A randomized algorithm K gives *ε-differential privacy* if for all data sets D and D′ differing on at most one row, and any S ⊆ Range(K),

$$Pr[K(D) \in S] \leq exp(\varepsilon) \times Pr[K(D') \in S]$$

These are 2 quantities that must be considered in DP algorithms are:

- Epsilon (ε): A metric of privacy loss at a differentially change in data (adding, removing 1 entry). The smaller the value is, the better privacy protection.
- Accuracy: The closeness of the output of DP algorithms to the pure output. In the *Private Machine Learning with PATE* part, I will use the classification accuracy on the test set as a statistic for evaluating accuracy.

Note: There is another quantity, for mechanisms other than the Laplace mechanism, isn't mentioned is the delta (δ) [2].

There is something that you need to catch up on:

1. Decreasing in ε leads to a decrease in the accuracy.
If your algorithm is 0-differential privacy, which protects privacy well, then it has very low accuracy, it would be useless. Because you will get nothing other than the noise. This is also shown in Figure 4. You can verify it here: https://georgianpartners.shinyapps.io/interactive_counting/.
2. ε=0 (and δ=0 in the general case) is equivalent to absolute privacy.
It can be derived directly from the definition of Differential Privacy. *Briefly, ε=0 is equivalent to Pr[K(D)] = Pr[K(D')], this leads to the algorithm K being independent of the data and thus protects privacy, perfectly.*

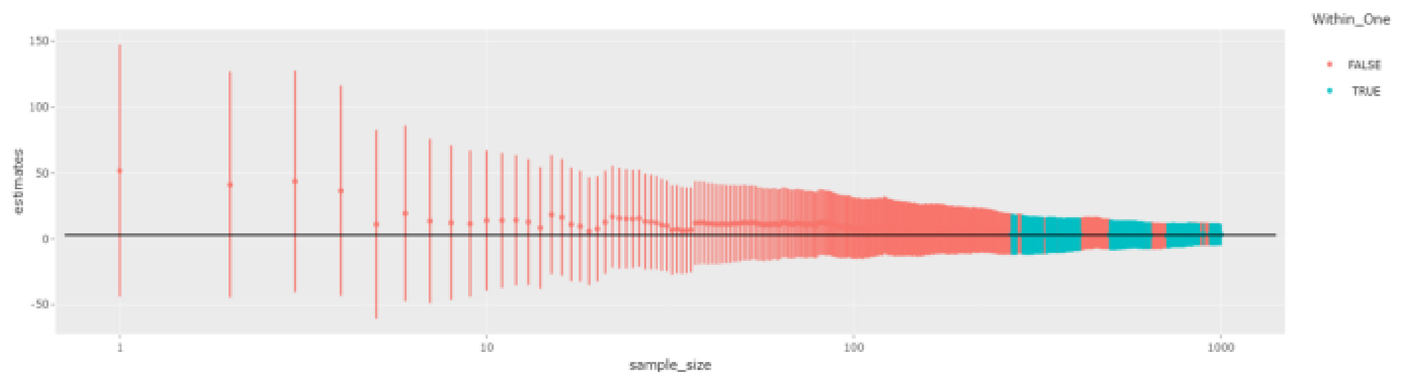


Figure 5: Confidence intervals of the average of results of an ε=0.03-differential privacy mechanism over 1000 queries. Note the confidence level is 95%, and that "Within_one" means the estimate is in the interval [2.5, 3.5]. Take from https://georgianpartners.shinyapps.io/interactive_counting.

Caution!

> *The more times applying the algorithm on the data, the more privacy loss. It's similar to playing "I Spy." . Your mission is guessing the object the Spy saw. With each answer from the Spy, you will get more information about the object. So, after a certain number of queries, you will be able to know what thing is it. The* **DP's Composition property** *will be useful for analyzing sensitive personal information in this scenario. The readers who are curious about this fact can read up on "privacy budget" concept [1] in the reference and this story.*

Report Noisy Max algorithm

For the explanation of the PATE algorithm, I will tell about Report Noisy Max (RNM) algorithm.

Why RNM?
RNM algorithm (introduced in The Algorithmic Foundations of Differential Privacy of Cynthia Dwork, the inventor of *Differential privacy)* applied when:

> *… "we wish to know which condition is (approximately) the most common in the medical histories of a set of respondents, so the set of questions is, for each condition under consideration, whether the individual has ever received a diagnosis of this condition. Since individuals can experience many conditions, the sensitivity of this set of questions can be high. Nonetheless, as we next describe, this task can be addressed using addition of Lap(1/ε) noise to each of the counts (note the small scale of the noise, which is independent of the total number of conditions). Crucially, the m noisy counts themselves will not be released (although the "winning" count can be released at no extra privacy cost)."*
>
> ***The Algorithmic Foundations of Differential Privacy, page 35.***

Naive pseudo implementation of RNM algorithm:

Input: A vector of labels: X, Privacy loss level: ε, number of labels: n
Output: A Noisy Max Answer
Compute label counts from X, assign the result to Y.
Draw a random n-dimension vector L from a zero median Laplace distribution has variance is 2/(ε*ε) (elements in L are mutually independent).

Return arg max(Y + L)

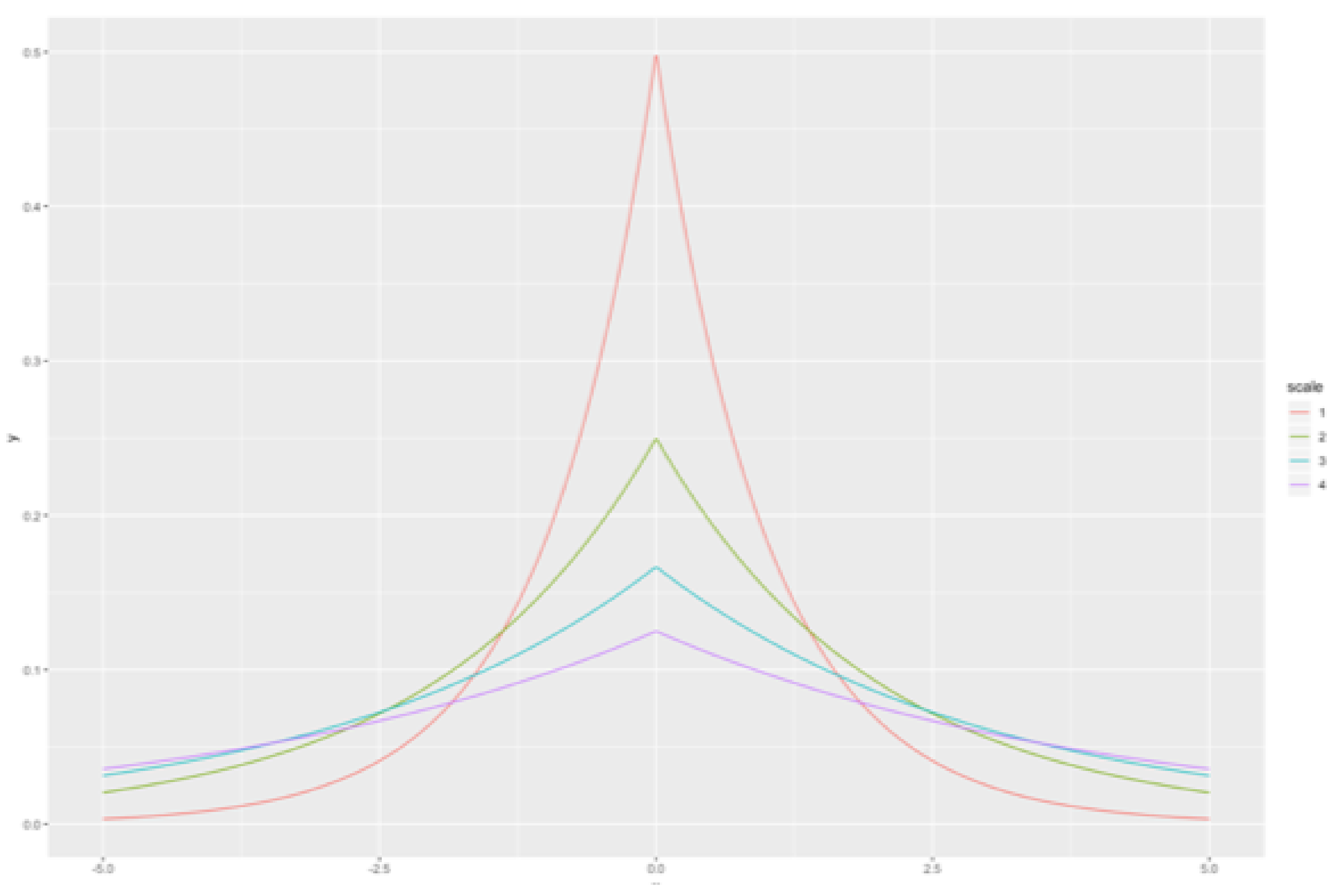


Figure 6. PDF of zero median Laplace distributions.

The RNM algorithm is proved that is ε-differential privacy, with ε is the privacy loss level (Source: The Algorithmic Foundations of Differential Privacy, page 35).

Practice Differential Privacy with PATE

Private Aggregation of Teacher Ensembles algorithm

In this part, I will explain the *Private Aggregation of Teacher Ensembles* algorithm (PATE) [3] and use it to reinforce the understanding of DP.

- **Why PATE?**
  The **Model Inversion Attacks that Exploit Confidence Information and Basic Countermeasures** [4] paper showed that it's possible for exploiting private information from the output of Machine learning algorithms. PATE was invented to solve the problem, which is a generally applicable approach to providing strong privacy guarantees for training data.
- **How does it work?**
  The idea behind PATE is applying a DP aggregation (RNM algorithm) on outputs of sensitive models, called "teachers" (sensitive models are trained directly on labeled sensitive data). The aggregated result will be used for training another public model on unlabeled public data. By that, applying DP on teachers' responses can be viewed as a proxy for preserving the privacy of the sensitive data and **teachers must be trained with disjoint subsets of the data**.

Why must they be disjoint?
Image that if each teacher is trained on the whole data then removing one teacher doesn't have any effects on participating of any individual data subject in the aggregated result, as that individual data subject still participates in training other teachers. This will make applying Differential privacy on teachers' responses is no longer a valid proxy for preserving privacy of individual data subjects in the sensitive data. Thus, training data sets must be disjoint.

Install PySyft

Analyzing *Differential privacy* of PATE, or performing PATE analysis, can be done with Pysyft. You can follow these steps to install Pysyft and related libraries.

- The easiest way to install the required libraries is with Conda. Create a new environment, then install the dependencies in that environment. In your terminal:

- If you have any errors relating to zstd — run the following (if everything above installed fine then skip this step):

- Retry installing syft (pip install syft).

APIs of PATE Differential Privacy Analysis

We will use Numpy and *perform_analysis* function of the "pate" package in Pysyft:

Description of the *perform_analysis* function:
Input (parameters are simplified for the purpose of the article):

— A 2-dimensional array of (m, n) teacher models' outputs.
m: number of teachers or number of outputs.
n: number of times querying.
— A vector of most voted labels.
— ε of RNM algorithm.

Output:

— Data-independent epsilon [3]: privacy loss in the worst case.
— Data-dependent epsilon [3]: a tight bound of privacy loss based on real values of teacher models' outputs.
Compute data-dependent epsilon:
For each query, calculate a tight bound privacy loss of the aggregated data, which is generated by applying the RNM algorithm to teacher models' outputs.
Sum tight bound privacy losses up, then assign the result to the data-dependent epsilon.

Return: The data-independent epsilon and the data-dependent epsilon

Description of *cal_max* and *noisy_max* functions:

The *cal_max* has the function of finding the most voted labels for each query.
And, *noisy_max* is the RNM algorithm.

PATE Analysis

I will use the word "data" for teacher models' outputs.

Let's consider these scenarios and compare our expectations with analysis results:

In the first three scenarios, I will use an RNM algorithm with a large privacy loss level (ε=5), to reduce its effects on the data. Thus, permit us to be able to "observe" intrinsic data's private information.

- **First, there is no information in the data:**

Create the data:

```
array([[0., 0., 0., ..., 0., 0., 0.],
       [0., 0., 0., ..., 0., 0., 0.],
       [0., 0., 0., ..., 0., 0., 0.],
       ...,
       [0., 0., 0., ..., 0., 0., 0.],
       [0., 0., 0., ..., 0., 0., 0.],
       [0., 0., 0., ..., 0., 0., 0.]])
```

Expectation:
There would be no general information and no private information in the aggregated results. Thus, the data-dependent epsilon should be low.

Perform PATE analysis:

```
Warning: May not have used enough values of l. Increase 'moments'
variable and run again.
Data Independent Epsilon: 10000.164470363787
Data Dependent Epsilon: 0.16447036378528898
```

The data-dependent epsilon is pretty low that reflected the expectation.

- **Second, there is rich information in the data and a strong agreement among the teachers:**

Create the data:

```
array([[  0,   1,   2, ..., 997, 998, 999],
       [  0,   1,   2, ..., 997, 998, 999],
       [  0,   1,   2, ..., 997, 998, 999],
       ...,
       [  0,   1,   2, ..., 997, 998, 999],
       [  0,   1,   2, ..., 997, 998, 999],
       [  0,   1,   2, ..., 997, 998, 999]])
```

Expectation:
Information in the data is rich may lead to the richness of both general information and private information. But, in this case, there is a strong

agreement among the teachers. Thus, opting-out of a teacher doesn't affect the last result. It means that there is no private information. By that, the data-dependent epsilon should be low, even with a large privacy loss level (ε=5) of the RNM algorithm.

PATE analysis (the same codes):

```
Warning: May not have used enough values of l. Increase 'moments'
variable and run again.
Data Independent Epsilon: 10000.57564627325
Data Dependent Epsilon: 0.5756462732485115
```

A nice catch!

- **Third, random data:**

Create the data:

```
array([[74, 60, 81, ..., 44, 95, 90],
       [77, 50, 72, ..., 40, 14, 49],
       [69, 60, 54, ..., 57, 60,  9],
       ...,
       [57, 47, 67, ..., 40, 59, 55],
       [26, 74, 21, ..., 27, 88, 57],
       [85, 39, 39, ...,  7, 84, 30]])
```

Expectation:
This case contrasts with the previous that there is no longer a strong agreement among the teachers. Thus, the richness of information is shared with both general information and private information. Thereby, the data-dependent epsilon should be high.

PATE analysis (the same codes):

```
Data Independent Epsilon: 10000.57564627325
Data Dependent Epsilon: 8135.9030202753
```

The data-dependent epsilon is high as expected.

- **Fourth, random data and small privacy loss level (ε=0.001) RNM**

Create the data (the same as the third scenario):

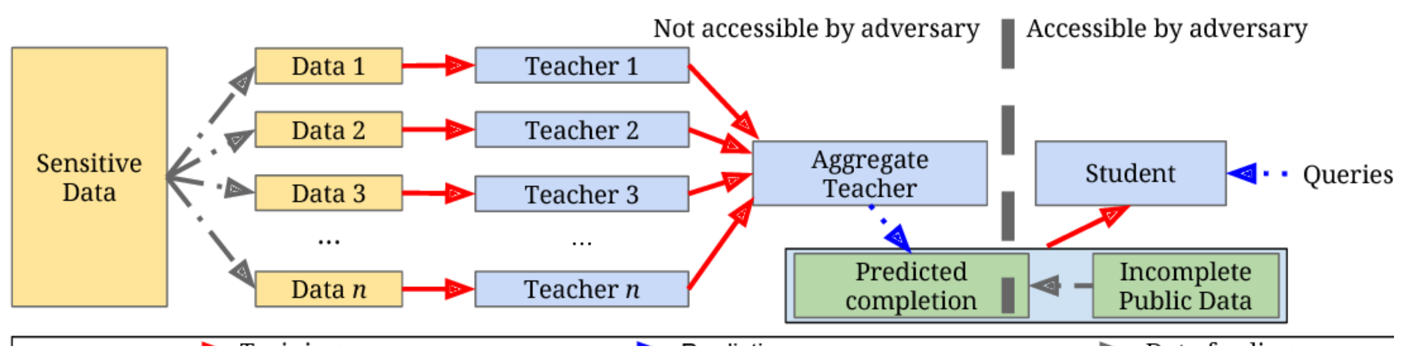
Expectation:
The decrease in the privacy loss level of RNM will reduce privacy loss, in all cases. Thus, the data-independent epsilon and the data-dependent epsilon should be low.

PATE analysis (the same codes):

```
Data Independent Epsilon: 0.3054858613811872
Data Dependent Epsilon: 0.3054858613811865
```

Again, we were right. The data-independent epsilon and the data-dependent epsilon are low.

Private Machine Learning with PATE (in Pytorch)



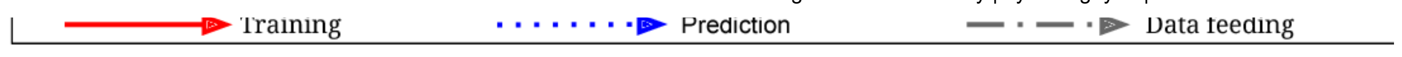


Figure 7 [3]: Private Machine Learning with PATE.

How does it work?
With PATE, we will use aggregated results of teacher models for training another model, a "student", on unlabeled (incomplete) public data. This helps us to get benefits from sensitive data sources.

Problem Description

- Data:

We'll use the MNIST dataset, which consists of greyscale handwritten digits, in the "torchvision" package. Each image is 28x28 pixels of a digit image, as you can see a sample below:

Thus, the training dataset represents the labeled sensitive dataset, while the test dataset represents the unlabeled public data.

- Goal:
  — Providing strong privacy guarantees for the labeled sensitive dataset.
  — Training a model for classifying an input image (28x28 pixels) of a handwritten single digit (0–9) from the unlabeled public data.

Implementation

Import libraries

Download *aggregation.py* and import it and related libraries.

Load MNIST data

Train student model

I will assume that you've had predictions of 250 fine-tuned teacher models-2048 queries for 2048 input samples from the unlabeled public data and it's stored in a 2-dimensional array as mentioned in the *PATE Analysis* part:

```
array([[7, 2, 1, ..., 3, 4, 6],
       [7, 2, 1, ..., 3, 9, 5],
       [7, 2, 1, ..., 3, 4, 6],
       ...,
       [7, 9, 7, ..., 9, 9, 8],
       [2, 2, 2, ..., 7, 1, 7],
       [6, 6, 6, ..., 6, 6, 6]])
```

PATE analysis:

```
Data Independent Epsilon: 751.0955105579642
Data Dependent Epsilon: 0.7707510458439465
```

Use inputs and RNM outputs (privacy loss level ε=0.3) of 2048 queries for training this student model:

Training codes:

Result:
The model has **82.5%** accuracy and less than **0.771** total privacy loss.

You can find the source code in this *GitHub project*.

Final Thoughts

Differential privacy is a powerful tool for quantifying and solving practical problems related to privacy. Its flexible definition gives it the potential to be applied in a wide range of applications, including Machine Learning applications. All is just a starting point, but I hope it's also a sign that it's possible for getting all benefits from big data techniques without any compromise on privacy.

References

[1] Kobbi Nissim, et al. Differential Privacy: A Primer for a Non-technical Audience. February 14, 2018.

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

[3] Nicolas Papernot, et al. Semi-supervised Knowledge Transfer for Deep Learning from Private Training Data. 2017.

[4] Fredrikson, Matt & Jha, Somesh & Ristenpart, Thomas. (2015). Model Inversion Attacks that Exploit Confidence Information and Basic Countermeasures. 1322–1333. 10.1145/2810103.2813677.