Differential Privacy

6.1600 Course Staff

Fall 2023

Taking advantage of our growing capability to collect all kinds of data, computation allows us to learn from this data and use it to make predictions. For example, machine learning algorithms are trained on huge datasets and then used to predict features of new input data.

In order for these predictions to be meaningfully useful, the data they base their predictions on must be based in reality. If researchers are designing a system to predict whether a patient has a certain disease, they need real patient data to base their model on. However, this data is very sensitive: the US has all kinds of laws protecting medical data with strict requirements. Ideally, we would like to use the data itself without the ability to tell who the data corresponds to. This way, we can alleviate privacy concerns.

However, this problem of using data while protecting the privacy of the individuals that the data comes from proves to be very hard.

Disclaimer: This set of notes is a work in progress. It may have errors and be missing citations. It is certainly incomplete. Please let the staff know of any errors that you find.

1 *Approach 1: Anonymize the Data*

An obvious solution may seem to be to remove explicit identifiers like name, address, and phone number. This way, it won't be possible to just glance at the data and see who a record corresponds to. However, this approach does not work since these datasets do not exist in isolation.

Example: Re-identification by Linking (Sweeney 1997) An example of this has to do with health data. In Massachusetts, the Group Insurance Commission released a dataset that they believed to be anonymous. This dataset consists of health data including patient ethnicity, ZIP code, birth date, sex, date of visit, diagnosis, procedure, and medication.

In 1997, Sweeney purchased (for $20!) another public dataset from Cambridge, MA: the voter registration list. This dataset included voter name, address, ZIP code, birth date, and gender.

Importantly, this other database included each voter's zip code, birth date, and gender! From linking the data in these two datasets, Sweeney was able to completely de-anonymize the GIC dataset and reveal private medical information for everyone up to the governor of Massachusetts.

Netflix Competition In 2006, Netflix aimed to improve their movie recommendation system. To do this, they aimed to have researchers do the work for them: they published an "anonymized" dataset that included a randomized user id, movie id, rating, and date. They released a portion of the dataset, and the group that could produce a model that best predicted ratings for the unreleased set of movies would win a million dollars.

One group of researchers was able to link individual records in the Netflix database to records from the public IMDb database. This allowed them to de-anonymize the data from Netflix even though the database contained no identifying information whatsoever!

Attempts to publish only part of a dataset to make the entries anonymous simply does not work due to the availability of other data that may be correlated.

2 *Approach 2: Publish Only Statistics*

Another approach we might consider is to publish only summary statistics with the aim that these summary statistics compress the data so much that it is impossible to learn anything meaningful about an individual from them. However, we have to be careful: even a few statistics can reveal sensitive information.

For example, consider a company that released the average salary of its employees regularly. If the company releases this average before and after the resignation of one individual, anyone who knows that that individual resigned can learn his salary.

So, releasing only statistics does not cleanly protect privacy either.

3 *Differential Privacy*

In order to achieve some measure of privacy, we need to define what privacy means: a natural definition may be something like "an algorithm is private if before and after it runs, no one learns anything about a given individual in the dataset". However, a definition like this removes all utility from the algorithm: it does not meaningfully compromise the privacy of an individual to reveal that smoking cigarettes causes cancer for them, since it does for every human. However, this first definition attempt would prevent this.

Since we are interested in protecting the privacy of *individuals* while learning about the whole group, it should be satisfactory if it is impossible to tell whether a given individual was included in the dataset or not. To formalize this, we will require that a pair of datasets that differ in only a single row—that is, one contains a row and the other does not—are *close*. We can define this as follows:

Definition 3.1 (ϵ-differential privacy)**.** An algorithm A is ϵ-differentially private if, for all neighboring datasets x and x' that differ in only one row, for all subsets S of outputs of A:

$$\Pr[A(x) \in S] - \Pr[A(x') \in S] \leq \epsilon$$

or, equivalently,

$$\Pr[A(x) \in S] \leq e^{\epsilon} \Pr[A(x') \in S]$$

Note that $e^{\epsilon} \approx 1 + \epsilon$. This is approximately saying that whether or not a given row is included in the dataset, the probability of the output being detectably different is less than ϵ.

In order to achieve this, the algorithm must be randomized. Take our salary example from before: if we want to reveal an average salary of all employees, achieving differential privacy requires adding *noise* in order to bring the probability of detection down to ϵ.

3.1 Adding Noise

A common approach to achieving DP is exactly this: adding random noise to the data in order to add uncertainty about the real data. The first instance of this was by Warner in 1965, who proposed using random noise to allow individuals to be truthful.

For example, consider a professor that wants to learn the fraction of students who cheated on a test. Obviously asking students to directly answer whether they cheated or not would result in students lying and saying that they did not cheat. Instead, using Warner's approach, the professor could ask students to answer honestly with probably $2/3$, but to lie with probability $1/3$. This allows a student who did cheat to claim that they were lying, as per the directions. However, with many students, the noise should average out, and the professor can still learn approximately how many students cheated on the test.

We can apply this same approach if we want to make a certain algorithm differentially private. Instead of publishing $A(x)$ directly, we can publish $A(x) + \text{noise}$.

Choosing Noise Level. In order to determine how much noise needs to be added to the function to achieve a given ϵ, we need to consider the function we are computing. For example, if our function returns a constant no matter what the data is, no noise is necessary. However, if the function returns one individual's record exactly, the record needs to be completely randomized so a large amount of noise is necessary. For an aggregate statistic like an average, something in the middle is required.

To formalize this, we will use a notion called *global sensitivity* of the function. This aims to capture the maximal change in f that results from changing one of the function's n inputs.

Definition 3.2 (Global Sensitivity)**.** The global sensitivity of a funcion f is given by:

$$GS_f = \max_{\text{neighbors}x,x'} \|f(x) - f(x')\|_1$$

Where "neighbors" are sets of inputs that differ in only a single value.

This gives us a sense of how much noise is necessary: if the function changes by a large amount when a single input changes, a large amount of noise is necessary, but if it changes only slightly then less noise is required.

In order to specify exactly how much noise to add, the Laplace distribution is used. The laplace distribution is:

$$h(y) = \frac{1}{2b} e^{-\frac{|y|}{b}}$$

For a function f, if A adds noise to f from the Laplace distribution with $b \geq \frac{GS_f}{\epsilon}$, then A is ϵ-DP.