Last time we showed that given a database x with sufficiently many rows, and a large collection Q of counting queries, there exists a distribution Y over synthetic databases such that on the one hand, a database y sampled from Y is differentially private; and on the other hand, all the counting queries Q can be accurately approximated if y was used as a reference database instead of x.

While we proved the *existence* of the distribution Y, we did not say how to go about sampling a synthetic database from Y efficiently. It turns out that in general, this task is computationally infeasible. (We may come back to this point in the future.) For this reason, the mechanism of Blum, Ligett, and Roth may be of limited practical significance.

In this lecture we will describe a different mechanism that provides similar guarantees. Unlike other mechanisms we have seen so far, this one is *interactive*: The state of the mechanism, and the answer to a query, will in general depend on the previous queries.

An *interactive data release protocol* is an interactive protocol that involves two parties: A *mechanism* and an *inquirer*. We model both of these parties as randomized algorithms.[1] The mechanism takes as private input a database $x \in D^n$. In each round i of interaction, the inquirer generates and sends a query q_i and the mechanism outputs an answer a_i, possibly followed by the special symbol `halt` that determines the end of the interaction. In this model both the inquirer and the mechanism may be *adaptive*; that is, the i-th query q_i may depend on the answers to the previous queries $a_1, \ldots, a_{i-1}$, and the i-th answer a_i may depend on $q_1, \ldots, q_{i-1}$.

The *view* of the inquirer in a given interaction consists of its private randomness and the sequence of answers that it receives. We say a mechanism M is *ε-differentially private* if for every inquirer I, every pair of adjacent databases $x, x' \in D^n$, and every possible view v, the probability that the view of I in its interaction with $M(x)$ equals v is at most e^ε times the probability that the view of I in its interaction with $M(x')$ equals v.

1 Threshold queries

One practical mechanism that mitigates certain privacy violations is to refuse providing answers to counting queries if the answer is too small. To see why this might be desirable, suppose you want to make a case that Andrej is an unfair teacher, so you query the CSE administrator how many of the (say) 30 students in CSCI 5520 failed the class. Should she give you this information?

If the number of students that failed CSCI 5520 is indeed large, then there is no real shame in having failed as the fault lies in the teacher. However, the situation is very different if only one or two students failed. The mechanism in question would reveal the correct answer to the query as long as the answer lies above a certain publicly known threshold t, and output the special symbol $\perp$ otherwise. This mechanism is not randomized, so it is not differentially private.

[1] If we do not concern ourselves with the inquirer's efficiency — as we won't in this lecture — we can think, without loss of generality, of the inquirer as deterministic.

We now describe an interactive mechanism that is a parametrized by a threshold t and a quota k. The mechanism takes a sequence of counting queries, approximately answers those queries whose answers are approximately above the threshold, and halts after providing k numerical answers. The salient feature of this mechanism is that its privacy deteriorates with the number of above-threshold answers k, not the total number of queries.

Let us first consider the case $k = 1$. The mechanism Threshold_t first generates a private threshold T by adding noise to the public threshold t. Upon receiving a query q, it answers this query independently by the Laplace mechanism as long as the answer is at least as large as T. Here is a detailed description:

Mechanism $Thr_t(x)$:
  Let $T = t + N$, where N is a $\text{Lap}(1/\varepsilon)$ random variable.
  Upon receiving the i-th counting query q_i:
    Sample an independent $\text{Lap}(1/\varepsilon)$ random variable N_i.
    If $q_i(x) + N_i < T$, output $\perp$.
    Otherwise output $a_i = q_i(x) + N_i'$, where N_i' is a $\text{Lap}(1/\varepsilon)$ random variable, and `halt`.

Lemma 1. *Mechanism Thr_t is 4ε-differentially private.*

Proof. Let x and x' be two adjacent databases. Without loss of generality, we will assume that the sequence of queries issued by the inquirer is infinite. To show differential privacy, we need to argue that on inputs x and x', the mechanism halts at the same step and produces the same answer with similar probability, or more precisely

$$\Pr[Thr_{t,1}(x) \text{ halts at time } h \text{ with answer } a_h] \le e^{\varepsilon} \Pr[Thr_{t,1}(x') \text{ halts at time } h \text{ with answer } a_h].$$

for every possible value of i and a_i. The event on the left happens if (1) $T > \max_{i<h}\{q_i(x) + N_i\}$; (2) $q_h(x) + N_h \ge T$ and (3) $a_h = q_h(x) + N_h'$. Then

$$\begin{aligned}
&\Pr[Thr_{t,1}(x) \text{ halts at time } h \text{ with answer } a_h] \\
&\quad = \Pr[T > \max_{i<h}\{q_i(x) + N_i\} \text{ and } q_h(x) + N_h \ge T \text{ and } a_h = q_h(x) + N_h'] \\
&\quad = \Pr[T > \max_{i<h}\{q_i(x) + N_i\} \text{ and } N_h \ge T - q_h(x) \text{ and } N_h' = a_h - q_h(x)] \\
&\quad = \mathrm{E}_{N_{-h}}\left[\Pr[T > \max_{i<h}\{q_i(x) + N_i\} \text{ and } N_h \ge T - q_h(x) \text{ and } N_h' = a_h - q_h(x)]\right] \\
&\quad = \mathrm{E}_{N_{-h}}\left[\Pr[T > \max_{i<h}\{q_i(x) + N_i\} \text{ and } N_h \ge T - q_h(x)] \cdot \Pr[N_h' = a_h - q_h(x)]\right]
\end{aligned}$$

where the last line follows from the fact that for every fixing of N_{-h}, N_h' is independent of N_h and T. We now fix N_{-h} and analyze the change in the two probabilities when x is replaced by x'. We will show that they change by at most a factor of $e^{3\varepsilon}$ and e^{ε}, respectively, so the total change in probability is bounded by a factor of $e^{4\varepsilon}$ as desired.

Since q_h is a counting query, and therefore 1-Lipschitz, we can say

$$\Pr[N_h' = a_h - q_h(x)] \le e^{\varepsilon} \Pr[N_h' = a_h - q_h(x')].$$

The value $m(x) = \max_{i<h}\{q_i(x) + N_i\}$ is also 1-Lipschitz and so

$$\begin{aligned}
\Pr[T > m(x) \text{ and } N_h \geq T - q_h(x)] &= \sum_{t\colon t>m(x)} \Pr[N_h \geq t - q_h(x)] \Pr[T = t] \\
&\leq e^{\varepsilon} \sum_{t\colon t>m(x)} \Pr[N_h \geq t - q_h(x')] \Pr[T = t] \\
&\leq e^{2\varepsilon} \sum_{t\colon t>m(x)} \Pr[N_h \geq t - q_h(x')] \Pr[T = t+1] \\
&\leq e^{3\varepsilon} \sum_{t\colon t>m(x)} \Pr[N_h \geq (t+1) - q_h(x')] \Pr[T = t+1] \\
&= e^{3\varepsilon} \sum_{t'\colon t'>m(x)+1} \Pr[N_h \geq t' - q_h(x')] \Pr[T = t'] \\
&\leq e^{3\varepsilon} \sum_{t'\colon t'>m(x')} \Pr[N_h \geq t' - q_h(x')] \Pr[T = t'] \\
&= e^{3\varepsilon} \Pr[T > m(x') \text{ and } N_h \geq T - q_h(x')].
\end{aligned}$$

Here, the first and fourth inequality use the fact that q_h and m are 1-Lipschitz, respectively, and the third inequality follows from Lemma 4 from Lecture 2. □

For larger values of k, we apply the following variant of the product construction for online mechanism M:

Mechanism $M^k(x)$:
    While $k > 0$,
        Emulate a new independent copy of $M(x)$ until it halts.
        Decrease k by 1.
    `halt`.

You will prove the following theorem in the homework.

Theorem 2. *If M is ε-differentially private, then M^k is $k\varepsilon$-differentially private.*

From Lemma 1 and Theorem 2 it follows that the mechanism Thr_t^k is $4k\varepsilon$-differentially private.

2 Interactive control of privacy loss

We will now begin our description of the interactive mechanism of Hardt and Rothblum for a sequence of counting queries. For the purposes of describing and analyzing this mechanism, it will be easier to work in a slightly different formal setting. To each database $x \in D^n$ we can associate a probability distribution over D obtained by sampling every row of x with probability $1/n$, which (abusing notation) we also denote by x. For example the database

name	**favorite fruit**
Alice	orange
Bob	banana
Alice	orange
Charlie	banana
Erica	apple

yields the probability distribution that assigns the entry (Alice, orange) probability 2/5, and the entries (Bob, banana), (Charlie, banana), and (Erica, apple) probability 1/5 each. The probability distribution x contains enough information to answer all averaging queries The answer to an averaging query $\overline{q}_P$ corresponding to predicate P (e.g. "favorite fruit is a banana") equals exactly

$$\overline{q}_P(x) = \Pr_{r\sim x}[P(r)] = \sum_{r:\ P(r)=1} x(r)$$

where we write $x(r)$ for the probability of outcome r in distribution x.

The Hardt-Rothblum mechanism answers queries using a proxy public distribution y in lieu of the true distribution x. The distribution y will change over the course of the interaction. To understand the privacy of this mechanism, it will be useful to think of the distribution y as being jointly maintained by the mechanism and the inquirer. In an actual implementation, the work of maintaining y can be fully emulated by the mechanism.

Initially, y is set to the uniform distribution over the whole domain D. Upon receiving a counting query $\overline{q}$, the mechanism does roughly the following:

1. If the values $\overline{q}(x)$ and $\overline{q}(y)$ are "close",[2] the inquirer is told to calculate $\overline{q}(y)$ by himself as an approximation of $\overline{q}(x)$. The protocol does this formally by sending the message $\perp$.
2. If the values $\overline{q}(x)$ and $\overline{q}(y)$ are "far", the mechanism outputs the answer $\overline{q}(x)$ plus some noise and tells the inquirer to perform a public, joint *update* to the distribution y.

The view of the inquirer consists of a sequence of symbols: The symbol $\perp$ in case 1 and a numerical approximation to its query in case 2, when an update is also performed. In analogy with the threshold mechanism we may expect that its privacy deterioration is governed not by the total number of queries, but by the number of updates. Thus the update is not relevant for the current query $\overline{q}$ evaluated on the current distribution Y, but the accuracy of typical *future* queries and future distributions.

This sounds incredulous: The mechanism is not clairvoyant, so how can it optimize for future queries? There is a general technique for this exact purpose called the method of multiplicative weights. Let us first explain how this technique works in a simpler model that does not incorporate privacy.

[2] I put close in quotes because in order to preserve privacy the actual test for closeness is randomized, so it may sometimes output the wrong answer.

3 Multiplicative weights update

Suppose you have a predictor that receives a sequence of averaging queries (possibly adaptively chosen) and wants to approximate their values on a secret probability distribution x over D. Upon receiving query $\overline{q}_P$, the predictor produces a guess a of the value $\overline{q}_P(x)$ and obtains one of the following three answers from an estimate checker $E(x)$ (with private access to x) which given an averaging query $\overline{q}$ and an estimate a, outputs

$$E(x) \text{ on input } (\overline{q}, a) = \begin{cases} \texttt{correct}, & \text{if } |\overline{q}(x) - a| < 2\alpha, \\ \texttt{too low}, & \text{if } a \leq \overline{q}(x) - 2\alpha, \\ \texttt{too high}, & \text{if } a \geq \overline{q}(x) + 2\alpha. \end{cases}$$

The objective of the predictor is to maximize the number of correct guesses.

In the multiplicative weights algorithm, the predictor maintains an empirical distribution y that approximates x in a certain sense. Initially, he sets y to the uniform distribution over D. His guess to a given query $\overline{q}_P$ is the value $\overline{q}_P(y)$. If his guess is correct, he does not change y. If his guess is too low, he reasons that his distribution assigns too little probability to the elements $r \in D$ that satisfy P, so he increases all these probabilities by a factor proportional e^α. If his guess is too low, he applies an analogous update to those r that do not satisfy P.

Algorithm MW^C, where C is an estimate checker:
- Set y to be the uniform distribution over D.
- Upon receiving counting query $\overline{q}_P$,
  - Set $a = \overline{q}_P(y)$.
  - Query C on input $(\overline{q}_P, a)$.
    - If `too low`, mutliply $y(r)$ by e^α for every r that satisfies P.
    - If `too high`, mutliply $y(r)$ by e^α for every r that does not satisfy P.
    - Normalize y so that $\sum_{r \in D} y(r) = 1$.

Theorem 3. *Assume $\alpha \leq 1.79$. For every distribution x, $MW^{E(x)}$ fails to answer* `correct` *at most $\ln|D|/\alpha^2$ times.*

To prove this theorem we make use of a notion from information theory. Given two distributions x and y over a finite domain D, the *information divergence* $\mathrm{Div}(x\|y)$ is the quantity

$$\mathrm{Div}(x\|y) = \mathrm{E}_{r \sim x}[\ln(x(r)/y(r))].$$

(If $y(r) = 0$ for some r in the support of x, then this quantity is undefined; this won't happen in our application.) We need the following two facts about information divergence:

Fact 1. *For every pair of distributions x and y, $\mathrm{Div}(x, y) \geq 0$.*

Fact 2. *If y is the uniform distribution, then for every x, $\mathrm{Div}(x, y) \leq \ln|D|$.*

Proof of Theorem 3. We will show that every time y is updated, $\mathrm{Div}(x, y)$ decreases by at least α^2. Since it starts at $\ln|D|$ or below and it cannot dip below zero, there can be at most $\ln|D|/\alpha^2$ updates.

An update to the distribution y happens whenever $|\overline{q}_P(x) - \overline{q}_P(y)| \geq 2\alpha$. Let y' denote the distribution after the update. We will show that $\mathrm{Div}(x\|y') \leq \mathrm{Div}(x\|y) - \alpha^2$. We only work out the case when E_x outputs `too low`, i.e. $\overline{q}_P(x) - \overline{q}_P(y) \geq 2\alpha$. The other case is completely analogous. All expectations are taken with respect to r sampled from x:

$$\begin{aligned}\mathrm{Div}(x\|y) - \mathrm{Div}(x\|y') &= \mathrm{E}\left[\ln \frac{x(r)}{y(r)}\right] - \mathrm{E}\left[\ln \frac{x(r)}{y'(r)}\right] = \mathrm{E}\left[\ln \frac{y'(r)}{y(r)}\right] \\ &= \mathrm{E}\left[\ln \frac{y(r)e^{\alpha P(r)}/Y'}{y(r)}\right] = \mathrm{E}\left[\ln \frac{e^{\alpha P(r)}}{Y'}\right] = \alpha\,\mathrm{E}_{r\sim x}[P(r)] - \ln Y'\end{aligned}$$

where $P(r)$ takes value 1 if r satisfies P and 0 if not, and $Y' = \sum_{r\in D} y(r)e^{\alpha P(r)}$ is the normalization factor for the distribution y'. We can rewrite the term $\ln Y'$ as

$$\ln Y' = \ln \sum_{r\in D} y(r)e^{\alpha P(r)} = \ln \sum_{r\in D} y(r)\big(1 + (e^\alpha - 1)P(r)\big) = \ln\big(1 + (e^\alpha - 1)\,\mathrm{E}_{r\sim y}[P(r)]\big).$$

By the lemma below, the last expression is at most $\alpha\,\mathrm{E}_{r\sim y}[P(r)] - \alpha^2$. Therefore

$$\mathrm{Div}(x\|y) - \mathrm{Div}(x\|y') \geq \alpha(\mathrm{E}_{r\sim x}[P(r)] - \mathrm{E}_{r\sim y}[P(r)]) - \alpha^2 \geq \alpha \cdot 2\alpha - \alpha^2 = \alpha^2. \qquad \square$$

Lemma 4. *For every* $0 \leq \alpha \leq 1.79$ *and every* $0 \leq t \leq 1$, $\ln(1 + (e^\alpha - 1)t) \leq \alpha t + \alpha^2$.

Proof Sketch. We can write

$$\ln(1 + (e^\alpha - 1)t) \leq (e^\alpha - 1)t \leq (\alpha + \alpha^2)t \leq \alpha t + \alpha^2.$$

the first inequality holds because $\ln(1 + x) \leq x$ for every $x \geq 0$, and the second one is valid for every α that falls between the two roots of the equation $e^x = 1 + x + x^2$, which are $x = 0$ and $x = 1.79328\ldots$ Both of these facts can be proven using basic calculus. $\square$

4 The interactive mechanism for counting queries

We now have all the elements to describe the mechanism of Hardt and Rothblum. The private input to the mechanism is a database x with n rows, which we also view as a probability distribution over D.

The first component is a *noisy estimate checker* $N(x)$ with private access to the database x. This estimator takes as input an averaging query $\overline{q}$ and an estimate a, samples a random variable N from the distribution $\mathrm{Lap}(\varepsilon n)$ and outputs the value

$$\begin{cases} \texttt{correct}, & \text{if } |\overline{q}(x) + N/n - a| < 3\alpha, \\ \texttt{too low}, & \text{if } a \leq \overline{q}(x) + N/n - 3\alpha, \\ \texttt{too high}, & \text{if } a \geq \overline{q}(x) + N/n + 3\alpha. \end{cases}$$

The second component is an *approximation mechanism* for averaging queries. Given an averaging query $\overline{q}$ and an estimate a for $\overline{q}(x)$, this mechanism outputs $\perp$ if the estimate is accurate, and an approximation of the estimate otherwise. It is very similar to the threshold mechanism.

Mechanism $Apx^C(x)$ where C is an estimate checker:
Let $T = t + N$, where N is a $\text{Lap}(\varepsilon n)$ random variable.
Upon receiving the i-th query $(\overline{q}_i, a_i)$:
If $C(\overline{q}_i, a_i)$ outputs `correct`, output $\perp$.
Otherwise, output $\overline{q}_i(x) + N_i'/n$, where N_i' is a $\text{Lap}(\varepsilon n)$ random variable and `halt`.

In your homework you will prove that the Apx is $O(1/(\varepsilon n))$ differentially private. Using Theorem 2, we have the following privacy guarantee for the product mechanism Apx^k.

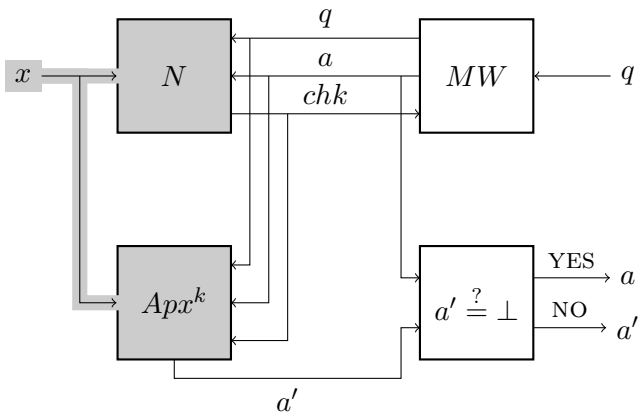
Theorem 5. *Mechanism $Apx^k(x)$ with estimate checker $N(x)$ is $O(k/(\varepsilon n))$-differentially private.*

The final component is the multiplicative update algorithm MW, but with access to the noisy estimate checker $N(x)$ instead of $E(x)$.

Here is how these components are interconnected. We set the parameter k to $\ln|D|/\varepsilon^2$. The estimate checker $N(x)$ and the approximation mechanism $Apx^k(x)$ are private to the mechanism. The algorithm MW is shared publicly between the mechanism and the inquirer.

When the inquirer issues an averaging query $\overline{q}$, this query is first forwarded to MW, which produces a guess a for the value $\overline{q}(x)$. MW then queries the noisy checker $N(x)$ if the guess is correct. The query, answer and the output of the noisy checker are also forwarded to $Apx^k(x)$. If this algorithm outputs $\perp$, then the answer to the query $\overline{q}(x)$ is a. Otherwise, the output of $Apx^k(x)$ is taken as the answer.

Here is a somewhat complicated diagram. The shaded boxes and pathways are the private components of the mechanism. All other items are public.



In the next lecture we will analyze the privacy and utility of this mechanism.

References

These notes are based on Chapters 3 and 4 of the survey *The Algorithmic Foundations of Differential Privacy* by Cynthia Dwork and Aaron Roth. My presentation of the interactive mechanism for counting queries also borrows from these lecture notes of Jonathan Ullman.

For formal definitions of interactive computation, see for example the book *Computational Complexity* by Oded Goldreich. For more on information divergence and proofs of its basic properties, see the book *Information Theory* by Cover and Thomas.