The interactive online mechanism that we discussed in the last two lectures allows efficient approximation of answers to arbitrary averaging queries. We showed this mechanism is differentially private as long as the database has sufficiently many rows.

If we disregard efficiency considerations, this mechanism can be generalized to handle not only counting queries, but arbitrary queries that take values in $\mathbb{R}$ or $\mathbb{R}^d$ (e.g. histograms; see Chapter 5 of the survey). However, the privacy loss grows with the sensitivity of the queries.

In general, the mechanisms that we discussed so far do not give any privacy guarantee for queries of large or unbounded sensitivity. One example of such a query is the median. In the worst case, this number can change dramatically if we change one row of our data: If 51 of our database participants make zero HKD and 50 make 1 million HKD then changing the salary of one person can cause the value of the median to go up by 1 million.

What should a data release mechanism do in such a case? One option is to refuse providing an answer when the answer reveals too much private information. One might hope that such cases happen rarely and do not affect the typical utility of the mechanism. In this setting, the role of the mechanism is to determine whether providing an answer to a given query might harm the privacy of the participants in the database and to provide an answer if no harm is caused.

The design of these mechanisms from first principles is not particularly difficult, although I am not sure how realistic the typicality assumption they make about their data is.

1 Median queries

As an instructive example let's talk about the median query. Recall that the median of a sequence of numbers is the middle number in the sequence after the sequence is ordered. (We'll assume that the sequence has an odd number of entries.) Intuitively, if we have a sequence of numbers like

$$1, 1, 2, 3, 3, 3, 3, 4, 5, 5, 5$$

then releasing the median should not be particularly harmful to privacy because changing a single entry in this sequence does not change the value of the median.

This observation suggests the following mechanism for median queries given a database x consisting of a sequence of numbers: If for all rows of the database, changing the value of this row does not change the median then output median(x), otherwise output $\perp$ (i.e. refuse to answer). This mechanism is deterministic, so it is not differentially private. Where does the privacy loss occur?

The reason this mechanism is not private is that making the decision whether to release or refuse public entails a loss of privacy. One solution is to randomize this decision. The mechanism below incorporates a tradeoff between privacy and stability, i.e. the minimum number of rows that need to be replaced in order to change the value of the median.

Stable median mechanism $SMed$**.** Given a database x of numbers:
Calculate the minimum number $\Delta(x)$ of rows of x that need to be replaced in order to change median(x).
Sample a Lap($1/\varepsilon$) random variable N.
If $\Delta(x) + N > t/\varepsilon$, output median($x$).
Otherwise, output $\perp$.

We show that this mechanism is almost always differentially private:

Theorem 1. *For $\varepsilon \leq 1$, the stable median mechanism is $(\varepsilon, O(e^{-t}))$-differentially private.*

Proof. If x and x' are adjacent databases. Then median(x) can be changed by first replacing one row of x to get x', then replacing $\Delta(x')$ rows of x', so $\Delta(x) \leq \Delta(x') + 1$. By symmetry we obtain $|\Delta(x) - \Delta(x')| \leq 1$, so Δ is 1-Lipschitz.

We now consider two cases. If $\Delta(x) > 1$, then x and x' have the same median m so the possible answers of the mechanism on databases x and x' are m and $\perp$. In particular

$$\Pr[SMed(x) = m] = \Pr[\Delta(x) + N > t/\varepsilon] \leq \Pr[(\Delta(x') + 1) + N > t/\varepsilon] \leq e^{\varepsilon} \Pr[SMed(x') = m]$$

and by similar reasoning $\Pr[SMed(x) = \perp] \leq e^{\varepsilon} \Pr[SMed(x') = \perp]$.

If $\Delta(x) = 1$, then

$$\Pr[SMed(x) \neq \perp] = \Pr[1 + N > t/\varepsilon] = \Pr[N > t/\varepsilon - 1] = O(e^{-t})$$

and

$$\Pr[SMed(x') \neq \perp] \leq \Pr[2 + N > t/\varepsilon] = O(e^{-t})$$

by the tail bound on the Laplace distribution. By going over all possible values over the set S $(\varnothing, \{\text{median}(x)\}, \{\perp\}, \{\text{median}(x), \perp\})$ we conclude that for all S,

$$\Pr[SMed(x) \in S] \leq \Pr[SMed(x') \in S] + O(e^{-t}).$$

We conclude that in either case, the mechanism is $(\varepsilon, O(e^{-t}))$-differentially private. □

The utility of this mechanism on a particular database x is determined by the parameter $\Delta(x)$. The probability that the mechanism outputs the median equals the probability that a $Lap(1/\varepsilon)$ random variable takes value greater than $t/\varepsilon - \Delta(x)$. For example, if $\Delta(x) \geq 2t/\varepsilon$, then by the large deviation bound from Homework 1 the median is output with probability at least $1 - O(e^{-t})$.

More generally, this type of mechanism can be used on any type of query for which the value $\Delta(x)$ tends to be reasonably large. One computational issue that can come up in the general setting is that $\Delta(x)$ may be difficult to calculate (or approximate to within sufficient accuracy). For median queries, this calculation can be done in time roughly linear in the size of x.

2 The subsample and aggregate mechanism

The propose-test-release mechanism is a clever conceptual solution to the problem of statistical query release that bypasses the sensitivity issue altogether. So far we have viewed all queries to the data as legitimate, and the goal of the mechanism was to answer every query as accurately as possible without compromising data privacy. Let us go one step further and ask what is the point of querying a data set? Often the reason we query data is because we need to draw some statistical conclusion (e.g. New Territories districts tend to vote for the DAB; cancer is more likely among smokers).

It is often assumed that the more data we have, the more accurate our conclusions should be. A consequence of this belief is that there comes a point at which having additional data does not improve the accuracy of the results; that is, results of a similar quality can be obtained even if some of the data was discarded. Thus, a query whose answer is affected substantially by subsampling the data may not be particularly useful in the first place, and so there shouldn't be much harm in refusing to answer such a query. The subsample and aggregate mechanism answers all other queries – that is, those that are robust to subsampling – in a differentially private way.

We begin with a definition that captures this notion of "robustness". Our defintion requires that the answer stays the same with high probability when the data is subsampled. A more realistic notion might allow for a small change in the answer. One should be able to prove similar results in that case, so for simplicity I'll stick to the more basic definition.

Definition 2. Query q is *m-stable* for database $x \in D^n$ if

$$\Pr_y[q(x) = q(y)] \geq 3/4$$

where y is obtained by sampling m independent uniformly distributed rows of x.

The constant $3/4$ is somewhat arbitrary; the discrepancy between this constant and $1/2$ only affects various constants in the description of the mechanism.

The subsample and aggregate mechanism takes several independent subsamples of the data, queries the data on each subsample, and releases the answer to the query if there tends to be agreement among the subsamples.

The *mode* of a sequence of numbers is the most frequent value occurring among them (breaking ties arbitrarily). The *frequency* of the mode is the number of times it occurs. For instance the mode of $(1, 2, 4, 3, 4, 4, 2)$ is 4 and its frequency is 3.

Subsample and aggregate mechanism SA**.** Given a database $x \in D^n$,
Create $k = \varepsilon(n/m)^3$ i.i.d. databases $x_1, \ldots, x_k \in D^m$,
where x_i is a uniform sample of m rows of x with repetition.
Let f be frequency of the mode of $(q(x_1), \ldots, q(x_k))$.
Sample a $\mathrm{Lap}(2km/\varepsilon n)$ random variable N.
If $f + N > 5k/8$, output the mode of $(q(x_1), \ldots, q(x_k))$.
Otherwise output $\perp$.

In the analysis, we will assume that εn is larger than Km for a sufficiently large constant K.

The utility of this mechanism will follow from the stability of q: Typically about 3/4 of the answers $q(x_i)$ will be the same as $q(x)$, and in such a case the stable mode mechanism is likely to output the correct answer.

Theorem 3. *If q is m-stable for x then $SA(x)$ outputs $q(x)$ with probability at least $1 - e^{-\Omega(\varepsilon n/m)}$.*

Proof. Let X_i be an indicator random variable for the event $q(x_i) = q(x)$. These are i.i.d. random variables with mean at least 3/4. By the Chernoff bound, the probability that fewer than $5k/8$ of them are equal to $q(x)$ is at most $2^{-\Omega(k)} = 2^{-\Omega(\varepsilon n/m)}$.

Assuming this is the case, the probability that $f + N \leq 5k/8$ is at most the probability that $N \leq k/8$, which is at most $e^{-\Omega(\varepsilon n/m)}$. The theorem follows by taking a union bound. □

For privacy, we will argue that any specific row of x is unlikely to be represented in too many of the samples $x_1, \ldots, x_k$, so it is unlikely too many values in the sequence $(q(x_1), \ldots, q(x_k))$ are affected by the change of a single row of x. We then apply a similar analysis to the one for the stable median mechanism.

Theorem 4. *Assume $m \leq n/64$. Then SA is $(\varepsilon, e^{-\Omega(\varepsilon n/m)})$-differentially private.*

Proof. Let $x, x' \in D^n$ be adjacent databases. Let X_i be an indicator random variable for the event that the row r on which x and x' differ is present in the database x_i. These are independent random variables with mean less than m/n, so by the Chernoff bound the probability that their sum exceeds $(2m/n)k$ is at most $e^{-2km^2/n^2} = e^{-\Omega(\varepsilon n/m)}$ by our choice of k.

For the rest of the proof we will assume that r is present in at most $(2m/n)k$ of the databases x_i. Under this assumption the sequences $s = (q(x_1), \ldots, q(x_k))$ and $s' = (q(x'_1), \ldots, q(x'_k))$ can differ in at most $(2m/n)k$ entries. Let f and f' denote the frequency of the mode of s and s', respectively. We now consider two cases.

If $f > 9k/16$ then the mode of s and s' must be the same as the two sequences differ in at most $(2m/n)k \leq k/16$ of their entries. In this case, the only possible output of the mechanism on x and x' is either the mode m of both sequences or $\perp$. Then

$$\begin{aligned}\Pr[SM(x) = m] = \Pr[f + N > 5k/8] &\leq \Pr[f' - (2m/n)k + N > 5k/8] \\ &\leq e^{\varepsilon} \Pr[f' + N > 5k/8] = e^{\varepsilon} \Pr[SM(x') = m]\end{aligned}$$

because N is a Laplace random variable with parameter $2km/\varepsilon n$. Using a similar calculation we conclude that $\Pr[SM(x) = \perp] \leq e^{\varepsilon} \Pr[SM(x) = \perp]$.

If $f \leq 9k/16$, then $SA(x)$ outputs $\perp$ whenever $N > -k/16$, which happens with probability at least $1 - 2^{-\Omega(\varepsilon n/m)}$. Also, $f' \leq f + (2m/n)k \leq 31k/32$, so $SM(x')$ outputs $\perp$ whenever $N > -k/32$, which also happens with probability at least $1 - e^{-\Omega(\varepsilon m/n)}$. As in the proof of Theorem 1, we can conclude that for every event S,

$$\Pr[SA(x) \in S] \leq \Pr[SA(x') \in S] + e^{-\Omega(\varepsilon n/m)}$$

and so SA is $(\varepsilon, e^{-\Omega(\varepsilon n/m)})$-differentially private. □

References

These notes are based on Chapter 7 of the survey *The Algorithmic Foundations of Differential Privacy* by Cynthia Dwork and Aaron Roth.