6.889 New Developments in Cryptography May 6, 2011

Answering Many Queries with Differential Privacy

Instructors: Shafi Goldwasser, Yael Kalai, Leo Reyzin, Boaz Barak, and Salil Vadhan

Lecturer: Jonathan Ullman *Scribe: Jonathan Ullman*

Last week we introduced the notion of *differential privacy* and discussed both the definitional aspects and basic techniques for privately querying a database. In this lecture we'll study more sophisticated mechanisms that allow us to ask exponentially more queries while preserving the accuracy of the answers and the privacy of the database.

1 Recap

First we'll review some basic definitions. We model the database as a collection of entries ("rows") from n individuals, who each specify a their type from an arbitrary *data universe*. Specifically, we use $\mathcal{X}$ to denote the universe of types and denote an *n-row database* $D = (x_1, x_2, \ldots, x_n) \in \mathcal{X}^n$. We say two databases D, D' are *neighbors* (denoted $D \sim D'$) if they differ on only one row. That is, $D \sim D'$ if $D = (x_1, \ldots, x_i, \ldots, x_n)$ and $D' = (x_1, \ldots, x_i', \ldots, x_n)$ for some $i \in [n]$. Let $\mathcal{M} : \mathcal{X}^* \to \mathcal{R}$ be a function from databases to any output range, often referred to as a *mechanism.*

Definition 1 ((ϵ, δ)-differential privacy) *A mechanism $\mathcal{M} : \mathcal{X}^* \to \mathcal{R}$ is* (ϵ, δ)-differentially private *if for every pair of neighboring databases $D \sim D'$ and every set of outputs $T \subseteq \mathcal{R}$,*

$$\Pr[\mathcal{M}(D) \in T] \leq e^{\epsilon} \cdot \Pr[\mathcal{M}(D') \in T] + \delta.$$

where the probability is taken only over the coins of $\mathcal{M}$.

In this lecture we're interested in sanitizers that evaluate *counting queries* on the database. Intuitively, a counting query asks "What fraction of the database satisfies a property q?". Formally, a counting query is specified by a boolean predicate $q : \mathcal{X} \to \{0, 1\}$. Abusing notation, we use

$$q(D) = \frac{1}{n} \sum_{i=1}^{n} q(x_i)$$

to denote the evaluation of a counting query on a database. We will use $Q = \{q_1, q_2, \ldots, q_k\}$ to denote a set of queries and frequently write $k = |Q|$.

Previous we saw the simplest way of answering counting queries with differential privacy, the *Laplace mechanism.* The mechanism $\mathcal{M}_{Q,\mathrm{Lap}} : \mathcal{X}^* \to \mathbb{R}^k$ is defined as

$$\mathcal{M}_{Q,\mathrm{Lap}}(D) = (q_1(D) + \mathrm{Lap}(k/n\epsilon), \ldots, q_k(D) + \mathrm{Lap}(k/n\epsilon))$$

where each $\mathrm{Lap}(k/n\epsilon)$ denotes an independent sample from the Laplace distribution with standard deviation $O(k/n\epsilon)$. See the previous lecture for the exact specification of the Laplace distribution and analysis. This mechanism turns out to give accurate answers for relatively small sets of queries.

Theorem 2 *The mechanism $\mathcal{M}_{Q,\mathrm{Lap}}(D)$ is $(\epsilon, 0)$-differentially private. Moreover, let $\mathcal{M}_{Q,\mathrm{Lap}}(D) = (a_1, \ldots, a_k)$. Then with probability $1 - \beta$ over the coins of $\mathcal{M}_{Q,\mathrm{Lap}}$, we have*

$$\max_{q_i \in Q} |q_i(D) - a_i| \leq \frac{k \log(k/\beta)}{n\epsilon}.$$

Recall that the way we defined a counting query, the true answer lies in $[0,1]$. So the bound on the error is only non-trivial if $k \ll n$. In the next section we will see how to accurately answer counting queries when $k \gg n$, while still satisfying differential privacy. In what follows, we will refer to the quantity

$$\max_{q_i \in Q} |q_i(D) - a_i| \leq \frac{k \log(k/\beta)}{n\epsilon}$$

as the *error* and when speaking informally or qualitatively we will often refer to mechanisms with non-trivial error as *accurate* (or as being more or less accurate than another mechanism).

2 The BLR Mechanism

In this section we will present a very elegant mechanism for answering counting queries with differential privacy due to Blum, Ligett, and Roth [**?**] that will be accurate even when $k \gg n$. The idea is to carefully correlate the noise we use in order to introduce less error, while still achieving privacy. Notice that in the Laplace mechanism, we use a different component of the output vector to answer each query, and add independent noise to each dimension of the output vector. Thus our error degrades linearly in the number of queries, since we are requiring that each query be accurate and private independently. One way to correlate the noise is to embed the answers to the queries in a space of lower dimension. BLR showed how to achieve this using a *synthetic database*, that is a new database that approximately preserves the answers to all the queries.

We start with the observation that for every D and Q, there exists a synthetic database with a small number of rows that preserves the answers to every query in Q.

Lemma 3 *For every $D \in \mathcal{X}^n$ and every set of counting queries Q, there exists a* synthetic database *$\widehat{D} \in \mathcal{X}^m$, for $m = \frac{8 \log k}{\alpha^2}$, such that*

$$\max_{i \in [k]} \left| q_i(D) - q_i(\widehat{D}) \right| \leq \alpha.$$

Proof: Consider a database $\widehat{D} \in \mathcal{X}^m$ formed by taking a random sample of m rows from D, chosen with replacement. Then by a union bound and a Chernoff bound

$$\begin{aligned}\Pr\left[\max_{q_i \in Q} \left| q_i(D) - q_i(\widehat{D}) \right| > \alpha\right] \leq k \cdot \Pr\left[\left| q_1(D) - q_1(\widehat{D}) \right| > \alpha\right] &\leq k \cdot \exp\left(\frac{-\alpha^2 m}{4}\right) \\ &= k \cdot \exp(-2 \log k) \\ &< 1\end{aligned}$$

There there exists some $\widehat{D} \in \mathcal{X}^m$ that preserves the answer to every query $q \in Q$ up to an additive error term of α. □

Notice that the dimension of $\widehat{D}$ in the previous lemma is roughly $\log |\mathcal{X}^m| = \frac{8 \cdot \log k \cdot \log |\mathcal{X}|}{\alpha^2}$, which is much smaller than k when the number of queries is large (and the other parameters $|\mathcal{X}|$ and $1/\alpha$ are not too large). However, in order to use this fact to our advantage, we need to know how to output an accurate synthetic database while satisfying differential privacy.

2.1 The Exponential Mechanism

McSherry and Talwar gave an elegant mechanism for producing useful, differentially private output from an arbitrary range [**?**]. More precisely, they give a generic specification of a distribution over any support that is simultaneously private and concentrated around elements of the range that maximize an arbitrary "score function" that depends on the input database.

Specifically:

- Let $\mathcal{R}$ be an arbitrary range of outputs.
- Let $s : \mathcal{X}^* \times \mathcal{R} \to \mathbb{R}$ be an arbitrary *score function*.
- Let
$$\mathrm{GS}_s = \max_{D \sim D', r \in \mathcal{R}} |s(D,r) - s(D',r)|$$
be the *global sensitivity of* s.

Then we define the *exponential mechanism* $\mathcal{M}_s(D) : \mathcal{X}^* \to \mathcal{R}$ so that

$$\Pr[\mathcal{M}_s(D) = r] \propto \exp\left(\frac{\epsilon \cdot s(D,r)}{2 \cdot \mathrm{GS}_s}\right).$$

Before we see how the BLR Mechanism makes use of the exponential mechanism, we will verify that the exponential mechanism is *always* $(\epsilon, 0)$-differentially private, regardless of the choice of range and score function.

Theorem 4 *([**?**]) The mechanism $\mathcal{M}_s$ satisfies $(\epsilon, 0)$-differential privacy.*

Proof: Consider any pair of neighboring databases $D \sim D'$ and any output $r \in \mathcal{R}$. Let

$$N_D = \sum_{r \in \mathcal{R}} \exp\left(\frac{\epsilon \cdot s(D,r)}{2 \cdot \mathrm{GS}_s}\right)$$

be the normalizing constant for the database D. Then we have

$$\begin{aligned}
\frac{\Pr[\mathcal{M}_s(D) = r]}{\Pr[\mathcal{M}_s(D') = r]} &= \frac{N_{D'}}{N_D} \cdot \exp\left(\frac{\epsilon \cdot s(D,r)}{2 \cdot \mathrm{GS}_s}\right) \cdot \exp\left(\frac{-\epsilon \cdot s(D,r)}{2 \cdot \mathrm{GS}_s}\right) \\
&= \frac{N_{D'}}{N_D} \cdot \exp\left(\frac{\epsilon \cdot (s(D,r) - s(D',r))}{2 \cdot \mathrm{GS}_s}\right) \\
&\leq \frac{N_{D'}}{N_D} \cdot \exp(\epsilon/2)
\end{aligned}$$

where the last inequality follows from the definition of global sensitivity. Now we consider the ratio of the normalizing constants.

$$\frac{N_{D'}}{N_D} = \frac{\sum_{r \in \mathcal{R}} \exp\left(\frac{\epsilon \cdot s(D',r)}{2 \cdot \mathrm{GS}_s}\right)}{\sum_{r \in \mathcal{R}} \exp\left(\frac{\epsilon \cdot s(D,r)}{2 \cdot \mathrm{GS}_s}\right)} \leq \frac{\sum_{r \in \mathcal{R}} \exp(\epsilon/2) \cdot \exp\left(\frac{\epsilon \cdot s(D,r)}{2 \cdot \mathrm{GS}_s}\right)}{\sum_{r \in \mathcal{R}} \exp\left(\frac{\epsilon \cdot s(D,r)}{2 \cdot \mathrm{GS}_s}\right)} = \exp(\epsilon/2)$$

So we conclude that

$$\frac{\Pr[\mathcal{M}_s(D) = r]}{\Pr[\mathcal{M}_s(D') = r]} \leq \exp(\epsilon)$$

where the inequality follows from the definition of global sensitivity. A similar argument would show that this ratio is also at least $\exp(-\epsilon)$. So the exponential mechanism is indeed $(\epsilon, 0)$-differentially private. □

Notice that the exponential mechanism is actually a generalization of the Laplace mechanism. If we set $\mathcal{R} = \mathbb{R}^k$ and $s(D, r) = \sum_{q_i \in Q} |q_i(D) - r_i|$ then we'd recover the Laplace mechanism exactly. However, we will now see how BLR instantiate the exponential mechanism to output a private and accurate synthetic database.

2.1.1 BLR Mechanism and its Analysis

The BLR mechanism uses the exponential mechanism to sample a synthetic database privately. We have already seen that the set of m-row databases contains at least one "good' synthetic database that will preserve the answers to all of the queries in Q. The analysis of the BLR mechanism will show that the output of the exponential mechanism is strongly concentrated around the set of "good" synthetic databases.

The BLR Mechanism, $\mathcal{M}_{Q,\mathrm{BLR}}$, instantiates the exponential mechanism as follows:

- Let $\mathcal{R} = \mathcal{X}^m$ for $m = \frac{32 \log k}{\alpha^2}$.
- Let $s(D, \widehat{D}) = -\max_{q \in Q} \left| q(D) - q(\widehat{D}) \right|$ for every $D \in \mathcal{X}^n, \widehat{D} \in \mathcal{X}^m$. Note that we make the score function inversely related to the error so that *better accuracy* implies *higher score.*
- Note that $\mathrm{GS}_s \leq 1/n$, since the score function is simply the maximum error over a set of counting queries, and a counting query has global sensitivity $1/n$.

Now we show that the BLR mechanism is also accurate with high probability

Theorem 5 ([?]) *Let* $\widehat{D} = \mathcal{M}_{Q,\mathrm{BLR}}(D)$. *With probability* $1 - \beta$

$$\max_{q \in Q} \left| q(D) - q(\widehat{D}) \right| \leq \alpha = \left(\frac{256 \cdot \log k \cdot \log |\mathcal{X}|}{\epsilon n} \right)^{1/3} + \left(\frac{8 \log(1/\beta)}{\epsilon n} \right)$$

Proof: We want to analyze the ratio

$$\frac{\Pr\left[s(D, \widehat{D}) \geq -\alpha/2 \right]}{\Pr\left[s(D, \widehat{D}) < -\alpha \right]}$$

Note that $\widehat{D}$ is accurate if the event in the numerator occurs, and fails to be accurate only when the event in the denominator occurs. So we can establish the theorem by showing this ratio is lower-bounded by $1/\beta$. Lemma 3, and our choice of m, ensures that there is at least one $\widehat{D} \in \mathcal{X}^m$ with $s(D, \widehat{D}) \geq -\alpha/2$. To bound the denominator we will simply use a union bound over all $\widehat{D} \in \mathcal{X}^m$. Thus we have

$$\frac{\Pr\left[s(D, \widehat{D}) \geq -\alpha/2 \right]}{\Pr\left[s(D, \widehat{D}) < -\alpha \right]} \geq \frac{\exp\left(-\epsilon n \alpha/4\right)}{|\mathcal{X}|^m \exp\left(-\epsilon n \alpha/2\right)} = \frac{\exp(\epsilon n \alpha/4)}{|\mathcal{X}|^m}$$

and we wish to choose α such that

$$\frac{\exp(\epsilon n\alpha/4)}{|\mathcal{X}|^m} \geq \frac{1}{\beta}.$$

This condition is satisfied when

$$n \geq \frac{128 \log k \cdot \log|\mathcal{X}|}{\epsilon\alpha^3} + \frac{4\log(1/\beta)}{\epsilon\alpha}$$

Thus we can choose

$$\alpha = \left(\frac{256 \cdot \log k \cdot \log|\mathcal{X}|}{\epsilon n}\right)^{1/3} + \left(\frac{8\log(1/\beta)}{\epsilon n}\right),$$

as desired[1]. □

2.2 Remarks about the BLR Mechanism

The first thing to note about the BLR Mechanism is that it is considerably more accurate when $k \gg n$. For a concrete example, we will consider the case of *monotone conjunction queries*. Let $\mathcal{X} = \{0,1\}^d$. For every $S \subseteq [d]$ we define $q_S(x) = \bigwedge_{j\in S} x_j$. Think of each individual as giving answers to a set of d yes/no questions and the queries will ask what fraction of individuals answers yes to some subset of the questions.

For this set of queries, the Laplace mechanism would have error

$$O\left(\frac{2^d \cdot d}{\epsilon n}\right),$$

which is only non trivial if $n \gg 2^d$. However, the BLR mechanism would have error

$$O\left(\left(\frac{d^2}{\epsilon n}\right)^{1/3}\right),$$

which is non-trivial if $n \gg d^2$! This is an extremely remarkable fact—we can give accurate answers to a very large and rich set of counting queries while satisfying a very strong notion of individual privacy, even with a database of size d^2!

However, the BLR mechanism is highly inefficient. A naive implementation of the mechanism would require enumerating all datasets in $(\{0,1\}^d)^m$, of which there are at least $2^{d\cdot\log k} = 2^{d^2}$. In general, the running time will be at least polynomial in $|\mathcal{X}|$, and, as this example shows, it is natural to think of $\mathcal{X}$ as being exponentially large. It is a significant open question to understand whether or not we can achieve accuracy and differential privacy in time polylog $|\mathcal{X}|$ for various classes of queries (cf. [**?**, **?**, **?**]).

Finally, a remark about the proof of Theorem 5. Notice that the proof basically only requires the following facts about BLR's instantiation of the exponential mechanism:

1. For our choice of parameters, there always exists at least one "good" output $r \in \mathcal{R}$ for the appropriate definition of "good."
2. The size of $\mathcal{R}$ is relatively small, so that we can take a union bound of all the "bad" outputs $r \in \mathcal{R}$.

[1] We've made no attempt to optimize constants. See [**?**] for a slightly more careful analysis.

In general this is a powerful paradigm for achieving differential privacy, at least showing that differential privacy and accuracy are consistent for a particular task: Find a small set of outputs that will always contain at least one “good” output and use the exponential mechanism to select an element from that set.

3 The Private Multiplicative Weights (PMW) Mechanism

In this section we will see a different mechanism for answering many counting queries with differential privacy, due to Hardt and Rothblum [**?**]. Their mechanism is based on the multiplicative weights algorithm [**?**, **?**] for online learning, and has some advantages over the BLR Mechanism:

- The PMW Mechanism is *interactive*. While the BLR Mechanism needs to know all the queries up front, the PMW Mechanism can take the queries one at a time, even if they are adaptively and adversarially chosen.
- The PMW Mechanism achieves better quantitative parameters in some settings. In particular the PMW Mechanism has $O(1/\sqrt{n})$ error (holding all other parameters fixed), which is optimal for answering $\omega(n)$ queries [**?**].
- The PMW Mechanism runs in *nearly-linear* time as a function of $|\mathcal{X}|$, as opposed the BLR Mechanism, which may run in quasi-polynomial time as a function of $|\mathcal{X}|$.
- Less formally, the PMW Mechanism feels “more direct,” in that it specifies a concrete procedure to perform when answering each query, compared to the BLR Mechanism which specifies a distribution to sample from.

The exact specification and analysis of the PMW Mechanism is a bit delicate, so we refer the reader to [**?**] for a complete presentation. In these notes we will attempt to communicate the intuition for the privacy and accuracy guarantees of the mechanism. First we will consider a non-private version of the algorithm to show how the multiplicative weights algorithm helps us in this setting.

The main idea is that we want to use the answers to previous queries in order to “guess” the answers to new queries. While this may seem difficult, there are clearly some cases where we expect to do well. For instance, if we see the same query multiple times we can give the same answers. For a slightly harder instance, suppose we already answered the queries that asked “What fraction of the database likes musicals?” and “What fraction of the database likes tragedies?” Then we are well-prepared to answer the query “What fraction of the database likes musicals *or* tragedies?” Although these are simplistic examples, the multiplicative weights algorithm shows how we can do a good job of guessing the answers to most queries, for any sequence of queries.

Before we specify the non-private algorithm, we will make a small notation shift. Instead of thinking of the database as a collection of n rows from $\mathcal{X}$, we will think of the database as a continuous distribution/function $D : \mathcal{X} \to [0,1]$ (normalized so that $\sum_{i\in\mathcal{X}} D(i) = 1$). The notion of a counting query $q : \mathcal{X} \to \{0,1\}$ can be adapted to this type of database as $q(D) = \sum_{i\in\mathcal{X}:q(i)=1} D(i)$.

3.1 Non-Private Multiplicative Weights

The idea of the multiplicative weights update step is to maintain a sequence of databases $D^{(0)}, D^{(1)}, \ldots, D^{(k-1)}$ that we use to answer queries $q^{(1)}, q^{(2)}, \ldots, q^{(k)}$, respectively. If $D^{(t-1)}$ fails to give a good answer to $q^{(t)}$ (which we determine by checking the real answer $q^{(t)}(D)$), then we will modify $D^{(t)}$ in a way that brings it closer to the true database D. A potential argument will show that we cannot do too many modifications. We are now ready to state the non-private algorithm.

Algorithm 1 Non-private multiplicative weights

```
**Input:** A database $D$
**Parameters:** An update parameter $\eta$ and a threshold $\tau := 2\eta$.
Let $D^{(0)}(i) = 1/|\mathcal{X}|$ for every $i \in \mathcal{X}$.
**For** $t = 1, 2, \ldots, k$**:**
    1. Receive query $q^{(t)}$ and compute $q^{(t)}(D^{(t-1)})$.
    2. Compute $q^{(t)}(D)$.
    3. If $|q^{(t)}(D^{(t-1)}) - q^{(t)}(D)| \leq \tau$ then set $D^{(t)}(i) = D^{(t-1)}(i)$ for every $i \in \mathcal{X}$ and continue.
       Otherwise do an update.
    4. **Update:** if $q^{(t)}(D^{(t-1)}) - q^{(t)}(D) < \tau$ set
           $D^{(t)}(i) \propto \exp\left(\eta \cdot q^{(t)}(i)\right) \cdot D^{(t-1)}(i)$
       and finally if $q^{(t)}(D^{(t-1)}) - q^{(t)}(D) > \tau$ set
           $D^{(t)}(i) \propto \exp\left(-\eta \cdot q^{(t)}(i)\right) \cdot D^{(t-1)}(i).$
```

In this algorithm, if we get a query $q^{(t)}$ and discover that the "guess", $q^{(t)}(D^{(t-1)})$, is much smaller than the "true answer", $q^{(t)}(D)$ then we increase the probability mass of elements $i \in \mathcal{X}$ such that $q^{(t)}(i) = 1$. Similarly if the guess is much smaller than the true answer we decrease the probability mass of such rows.

The goal is to show that we cannot do too many updates, and this is formalized in the following way. Consider the potential function

$$\Phi^{(t)} = \mathrm{RE}(D \| D^{(t)}) = \sum_{i \in \mathcal{X}} D(i) \log\left(\frac{D(i)}{D^{(t)}(i)}\right)$$

given by the *relative entropy* between the true database and the database $D^{(t)}$. Observe that $\Phi^{(0)} \leq \log|\mathcal{X}|$ and $\Phi^{(t)} \geq 0$ for every $t = 1, 2, \ldots, k$. In addition, every update step must significantly *decrease* the potential.

Lemma 6 ([?]) *In every update round*

$$\Phi^{(t-1)} - \Phi^{(t)} \geq \eta \left| q^{(t)}(D^{(t-1)}) - q^{(t)}(D) \right| - \eta^2.$$

The proof of this lemma follows by a direct calculation of the potential. See [?] for details. By our choice of $\tau = 2\eta$, we have

$$\Phi^{(t-1)} - \Phi^{(t)} \geq \eta^2.$$

Since the potential is never larger than $\log|\mathcal{X}|$ and is always positive, we have the following corollary:

Corollary 7 *Algorithm 1 performs at most* $\frac{\log|\mathcal{X}|}{\eta^2}$ *updates.*

3.2 Private Multiplicative Weights

Now we will describe the main ideas required to make Algorithm 1 satisfy differential privacy. A natural first idea would be to leave the algorithm unchanged in steps 1-4 and add a step 5 where we either output $q^{(t)}(D^{(t-1)})$ if we don't update, and output $q^{(i)}(D) + \mathrm{Lap}(\sigma)$, for an appropriate standard deviation σ if we do update. Since we only do $\frac{\log|\mathcal{X}|}{\eta^2}$ updates, we should be able to choose $\sigma = \frac{\log|\mathcal{X}|}{\epsilon n \eta^2}$ to get privacy (see Theorem 2).

The problem with this approach is that it reveals whether or not we do an update, which itself may not be private! For instance, consider the case where $q^{(1)}(D^{(0)}) = 0$ and $q^{(1)}(D) = \tau$. Then we would not do an update and would output 0. However for a neighboring database D' we may have $q^{(1)}(D') = \tau + 1/n$ and would output $\tau + 1/n + \mathrm{Lap}(\sigma)$, which will reveal that the input was D' and not D.

The solution to this problem is to add noise to $q^{(t)}(D)$ in step 2 of the algorithm. However, in light of the previous discussion, we might be worried that at every step just revealing whether or not we do an update discloses as much as revealing $q^{(t)}(D) + \mathrm{Lap}(\sigma)$. In which case we would have to choose $\sigma = \frac{k}{\epsilon n}$ (as if we were answering all k queries) and would get no improvement over the Laplace mechanism. However, what Hardt and Rothblum show (roughly) is that very little information about the database can leak just from the decision whether or not to do an update. See their paper [?] for a full analysis. In the remainder of these notes we will give a specification of the algorithm with slightly relaxed parameters and the intuition behind their analysis.

We will now state the main results about this algorithm

Theorem 8 ([?], modified parameters) *Algorithm 2 satisfies (ϵ, δ)-differential privacy. Moreover, with probability at least $1 - \beta$, the outputs $a^{(1)}, a^{(2)}, \ldots, a^{(k)}$ satisfy*

$$\max_{t \in [k]} |a^{(t)} - q^{(t)}(D)| \leq \alpha = O\left(\frac{(\log|\mathcal{X}|)^{1/3} \cdot \log(1/\delta) \cdot \log(k/\beta)}{\epsilon n^{1/3}}\right).$$

Theorem 9 ([?], actual parameters) *Algorithm 2 (with properly adjusted parameters) satisfies (ϵ, δ)-differential privacy. Moreover, with probability at least $1 - \beta$, the outputs $a^{(1)}, a^{(2)}, \ldots, a^{(k)}$ satisfy*

$$\max_{t \in [k]} |a^{(t)} - q^{(t)}(D)| \leq \alpha = O\left(\frac{(\log|\mathcal{X}|)^{1/4} \cdot \log(1/\delta) \cdot \log(k/\beta)}{\epsilon n^{1/2}}\right).$$

Finally, we will sketch what needs to be shown to prove Theorem 8:

- First, we need to show that every time we perform an update, the potential does actually decrease. Notice that if $z^{(t)}$ is very far from $q^{(t)}(D)$ then we may perform an update when our guess was actually correct. Worse, we may even perform an update in the wrong direction. However, we can show that our choice of τ, σ, η ensures that the noise we add in each iteration of step 2 is never more than $\tau/2$ (except with probability β). Thus whenever we do an update we have

  $$|q^{(t)}(D^{(t-1)}) - q^{(t)}(D)| > \tau/2$$

  and since $\tau \geq 4\eta$, we do get a potential decrease of η^2 by Lemma 6.

- Second, we want to argue that we can ensure the privacy of all the rounds in which we output $z^{(t)}$. However, the number of updates we do is at most $\frac{\log|\mathcal{X}|}{\eta^2} = n^{2/3} \cdot (\log|\mathcal{X}|)^{1/3}$. Thus by

Algorithm 2 Private multiplicative weights([**?**], modified parameters)

```
Input: A database $D$
Parameters: An update parameter $\eta := \left(\frac{\log|\mathcal{X}|}{n}\right)^{1/3}$, a standard deviation
    $$\sigma := O\left(\frac{(\log|\mathcal{X}|)^{1/3}\log(1/\delta)}{\epsilon n^{1/3}}\right)$$
, and a threshold
    $$\tau := O\left(\frac{(\log|\mathcal{X}|)^{1/3}\cdot\log(1/\delta)\cdot\log(k/\beta)}{\epsilon n^{1/3}}\right)$$
.
Let $D^{(0)}(i) = 1/|\mathcal{X}|$ for every $i \in \mathcal{X}$.
For $t = 1, 2, \ldots, k$:
    1. Receive query $q^{(t)}$ and compute $q^{(t)}(D^{(t-1)})$.
    2. Compute $z^{(t)} = q^{(t)}(D) + \mathrm{Lap}(\sigma)$.
    3. If $|q^{(t)}(D^{(t-1)}) - z^{(t)}| \leq \tau$ then set $D^{(t)}(i) = D^{(t-1)}(i)$ for every $i \in \mathcal{X}$ and output $a^{(t)} = q^{(t)}(D^{(t-1)})$. Otherwise output $a^{(t)} = z^{(t)}$ and do an update.
    4. Update: if $q^{(t)}(D^{(t-1)}) - z^{(t)} < \tau$ set
           $$D^{(t)}(i) \propto \exp\left(\eta \cdot q^{(t)}(i)\right) \cdot D^{(t-1)}(i)$$
       and finally if $q^{(t)}(D^{(t-1)}) - z^{(t)} > \tau$ set
           $$D^{(t)}(i) \propto \exp\left(-\eta \cdot q^{(t)}(i)\right) \cdot D^{(t-1)}(i).$$
```

Theorem 2, it is sufficient to choose $\sigma \geq \frac{(\log|\mathcal{X}|)^{1/3}}{\epsilon n^{1/3}}$ to ensure the privacy of these values, which is what we have done.

- Finally, we need to argue that, in most rounds, revealing whether or not we have done an update[2] does not depend on whether we used a database D or a neighboring database D'. Actually we will show that with high probability over the coins of the mechanism, such a statement is true. The idea is to show that if we are sufficiently close to performing an update, then we have a constant probability of actually doing an update. Thus there cannot be too many more "almost updates" than actual updates.

  In slightly more detail, we consider three possible cases for each round:

  - **Case 1:** In this case we are so far from performing an update that we don't reveal any information. Specifically, suppose that $|q^{(t)}(D^{(t-1)}) - z^{(t)}| < \tau - 2/n$. In this case, we will would not perform an update even if we had switched to a neighboring database D', since this could only affect the value $z^{(t)}$ by $1/n$. Thus we don't have to pay any privacy cost in these rounds.
  - **Case 2:** In this case we are always performing an update. Here we do pay a privacy cost, but it's the privacy cost we already accounted for when we bounded the number of update rounds and chose σ appropriately. Specifically, suppose that $|q^{(t)}(D^{(t-1)}) - z^{(t)}| > \tau + 2/n$. Here we would perform an update even if we had switched to a neighboring database D' that changes the value $z^{(t)}$ by $1/n$.
  - **Case 3:** These are the rounds where we are "borderline". Specifically, $|q^{(t)}(D^{(t-1)} - z^{(t)}| = \tau \pm 2/n$. In these rounds we may find that we would do an update if the database were D but not if the database were D', and thus we reveal information. In principle every round where we do not perform and update could be such a borderline round, and thus we may pay a privacy cost in every step. However, we can show that by our choice of σ, conditioning on the event that we are in case 2 or case 3, we actually have a constant probability of being in case 3. Thus, with probability $1 - \delta$, we are in case 2 or case 3 at most $O\left(\frac{\log|\mathcal{X}|\cdot\log(1/\delta)}{\eta^2}\right)$ times[3]. Thus, by chosing σ to be larger than we need to protect the update rounds by a factor of $\log(1/\delta)$, we still ensure differential privacy with probability $1 - \delta$, which is equivalent to (ϵ, δ)-differential privacy.

References

[AHK] S. Arora, E. Hazan, S. Kale. *The multiplicative weights update method: a meta algorithm and applications.* Technical report, Princeton University, 2005.

[BLR08] A. Blum, K. Ligett, A. Roth. *A Learning Theory Approach to Non-Interactive Data Privacy.* In STOC2008.

[DN03] I. Dinur, K. Nissim. *Revealing Information While Preserving Privacy.* In PODS2003.

[DNRRV09] C. Dwork, M. Naor, O. Reingold, G. Rothblum, S. Vadhan. *On the Complexity of Differentially-Private Data Release: Efficient Algorithms and Hardness Results.* In STOC2009.

[2] Note that the algorithm does not actually output whether or not it decided to update. However, the analysis of the mechanism shows that it would still be private even if we were to output this information.

[3] Notice that $q^{(t)}(D^{(t-1)})$ is fixed when conditioning on the previous rounds of the mechanism. Thus which case we fall into in round t is determined *only by the value of the random variable* Lap(σ) *in round* t, which is independent of the value of this random variable in all other rounds.

[GHRU11] A. Gupta, M. Hardt, A. Roth, J. Ullman. *Privately Releasing Conjunctions and the Statistical Queries Barrier.* In STOC2011

[HR10] M. Hardt, G. Rothblum. *A Multiplicative Weights Update Mechanism for Privacy-Preserving Data Analysis.* In FOCS2010.

[LW94] N. Littlestone, M.K. Warmuth. *The Weighted Majority Algorithm.* Information and Computation 1994.

[UV11] J. Ullman, S. Vadhan. *PCPs and the Hardness of Generating Private Synthetic Data.* In TCC2011