A Teaser for Differential Privacy

Mark Bun

November 11, 2019

Contents

Differential privacy is a formal mathematical definition of privacy in the context of privacy-preserving data analysis. It enables a trusted data curator to release global statistical information about a sensitive dataset, while provably protecting individual-level information.

The focus of this tutorial will be on privately answering "counting queries," which are some of the most basic statistics one might ask about a sensitive dataset. A counting query simply asks, "what fraction of the dataset satisfies a given predicate q." By now, there are a host of simple techniques, sophisticated algorithms, and tight lower bounds known for the query release problem. We'll only scratch the surface here, but it will be enough to illustrate many of the core principles of differentially private data analysis.

Most of the material here is covered in much greater depth and detail in the monograph of Cynthia Dwork and Aaron Roth [**?**]. Another excellent resource is Salil Vadhan's survey [**?**].

1 Datasets and counting queries

We model a sensitive dataset D as a tuple of n elements $(x_1, \ldots, x_n)$, where each $x_i \in X$ for some "data universe" X. It will be without loss of generality to think of $X = \{0,1\}^d$, where d should be thought of as the dimensionality of the data.

Definition 1. A *counting query* is a predicate $q : X \to \{0,1\}$. We abuse notation by extending a counting query to an entire dataset by defining

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

That is, the value of a counting query on a dataset D is the fraction of the rows of D that satisfy the predicate q.

We will of course not only be interested in answering a single counting query on a dataset D, but will seek to answer many different queries simultaneously. We use the notation $\mathcal{Q}$ to denote such a family of counting queries.

Example 2 (Histograms)**.** For every $y \in X$, define a predicate q_y by $q_y(x) = 1$ if $x = y$ and $q_y(x) = 0$ otherwise. That is, q_y is the indicator (or point function) for the universe element y. The vector $(q(D))_{q \in \mathcal{Q}}$ represents the histogram of D.

Example 3 (k-way Conjunctions)**.** Given a subset $S \subseteq [d]$ with $|S| \leq k$, define $q_S(x) = \wedge_{j \in S} x^j$. The family $\mathcal{Q}$ of all such queries denotes the k-way conjunctions (marginals).

While extremely simple, counting queries capture a broad spectrum of statistical tasks. A slight generalization lets one perform any analysis that is captured by the statistical query model.

Counting queries are also a natural object of study for privacy-preserving data analysis. After all, a counting query captures a global statistical property, for which the impact of any given individual is small (only $1/n$). Based on this intuition, one might ask: What is wrong with simply answering counting queries in the clear?

One reason is that, while the answer to a single counting query may not be disclosive, the answers to many counting queries can interact in privacy compromising ways. For example, consider the following pair of counting queries: 1) What fraction of the dataset are Justin Bieber fans?[1] and 2) What fraction of the dataset are Justin Bieber fans and are not named "Matt Weinberg"? While each query individually seems innocuous, answering both exactly on the same dataset and taking the difference allows one to conclude whether Matt is a Belieber. While artificial, one must take care with definitions to prevent this kind of "differencing attack." A more general and far-reaching attack comes from the work of Dinur and Nissim [**?**], which showed that given (approximate) answers to an almost linear collection of counting queries involving public attributes and a single sensitive attribute, one is able to very accurately reconstruct the vector of sensitive attributes.

In light of these attacks, we appeal to ideas that are now very familiar in the design and analysis of algorithms: We introduce randomization and approximation.

Definition 4. A randomized algorithm $\mathcal{M} : X^n \to [0,1]^{|\mathcal{Q}|}$ is α-accurate if for every dataset D, with high probability $\mathcal{M}$ produces an answer vector $(a_q)_{q \in \mathcal{Q}}$ such that $|a_q - q(D)| \leq \alpha$ for every $q \in \mathcal{Q}$.

2 Definition and interpretation of differential privacy [?, §2]

The definition of differential privacy captures the following: A *randomized algorithm* protects individual-level privacy if no individual has "too much" of an effect on the distribution of the output of the algorithm. This idea is formalized by introducing the concept of "neighboring" or "adjacent" datasets. We say that two datasets D, D' are neighboring, written $D \sim D'$, if they differ in at most one row, corresponding to one individual's data.

Definition 5. A randomized algorithm $\mathcal{M} : X^n \to \mathcal{R}$ provides ε-differential privacy if, for all pairs of neighboring datasets $D \sim D'$ and all (measurable) sets of outcomes $S \subseteq \mathcal{R}$, we have

$$\Pr[\mathcal{M}(D) \in S] \leq e^{\varepsilon} \Pr[\mathcal{M}(D') \in S].$$

A few remarks are in order about the definition:

1. A moment's reflection shows that the definition of ε-differential privacy amounts to requiring a notion of similarity between the distributions $\mathcal{M}(D)$ and $\mathcal{M}(D')$. By swapping the roles of D and D' in the definition, it requires that the probability of landing in any set of outcomes S increases by at most e^{ε}, or decreases by at most $e^{-\varepsilon}$, when the input dataset is changed from D to D'.

[1] Thanks to Kobbi Nissim for this compelling example.

2. An equivalent way of formulating the definition is to require that for every pair of neighboring datasets $D \sim D'$, the associated *privacy loss function* $L_{D,D'}(r)$ defined by

$$L_{D,D'}(r) = \ln \frac{\Pr[\mathcal{M}(D) = r]}{\Pr[\mathcal{M}(D') = r]}$$

satisfies $|L_{D,D'}(r)| \leq \varepsilon$ for all $r \in \mathcal{R}$.

3. Establishing an acceptable choice for the parameter ε is a delicate and context-dependent question. But it is reasonable to think of it as being a small constant (independent of the universe X or the size of the dataset n), e.g. $\varepsilon = 0.1$, justifying the approximation $e^{\varepsilon} \approx (1+\varepsilon)$.

The literature on differential privacy makes a number of interpretive claims about what the definition means. The great thing about having a formal mathematical definition is that one can actually prove theorems to back these claims up. We explore just a few of these claims here.

A Bayesian interpretation To quote Dwork and McSherry (2006), differential privacy offers the following promise to a person whose data is included in a differentially private data release: "Regardless of external knowledge, an adversary with access to the sanitized database draws the same conclusions whether or not my data was included." We can justify such a statement by considering how a Bayesian adversary's inference is affected by an individual's participation, or non-participation, in a differentially private data release. For a much more in-depth discussion of this approach, see [**?**].

For an index $i \in [n]$ and a dataset $D \in X^n$, let D_{-i} denote the dataset with individual i's information removed (or replaced by a junk symbol $\perp$ to stay compatible with the definition). Let $\mathcal{P}$ denote an adversary's prior beliefs, and for concreteness, let's say this is a distribution over the space of datasets X^n. After seeing the outcome r of a differentially private computation, she updates her prior to a posterior distribution $\bar{\mathcal{P}}[\cdot \mid r]$. Let $\bar{\mathcal{P}}_{-i}[\cdot \mid r]$ denote her posterior distribution if, instead of observing $r \leftarrow \mathcal{M}(D)$ for some dataset D, she observes $r \leftarrow \mathcal{M}(D_{-i})$.

Proposition 6. *Let $\mathcal{M} : X^n \to \mathcal{R}$ be ε-differentially private. Then for any prior distribution $\mathcal{P}$, any transcript r, and any dataset D,*

$$\bar{\mathcal{P}}[D \mid r] \in e^{\pm 2\varepsilon} \cdot \bar{\mathcal{P}}_{-i}[D \mid r].$$

In other words, regardless of an adversary's prior beliefs (modeling arbitrary side information about the dataset D), her posterior distribution does not change too much depending on the inclusion or non-inclusion of any individual's data.

Proof. By Bayes' Theorem,

$$\begin{aligned}
\bar{\mathcal{P}}[D \mid r] &= \frac{\Pr[\mathcal{M}(D) = r]}{\sum_{\hat{D} \in X^n} \Pr[\mathcal{M}(\hat{D}) = r] \Pr[\mathcal{P} = \hat{D}]} \cdot \Pr[\mathcal{P} = D] \\
&\in e^{\pm 2\varepsilon} \cdot \frac{\Pr[\mathcal{M}(D_{-i}) = r]}{\sum_{\hat{D} \in X^n} \Pr[\mathcal{M}(\hat{D}_{-i}) = r] \Pr[\mathcal{P} = \hat{D}]} \cdot \Pr[\mathcal{P} = D] \\
&\in e^{\pm 2\varepsilon} \cdot \bar{\mathcal{P}}_{-i}[D \mid r].
\end{aligned}$$

□

An economic interpretation Differential privacy captures the idea that participating in a differentially private data release does not result in too much additional harm to any individual. This can be formalized in a utility-theoretic framework. Fix an individual $i \in [n]$, and suppose this individual has a utility function $u : \mathcal{R} \to \mathbb{R}_{\geq 0}$ defined over outcomes of a differentially private analysis. Then

$$\begin{aligned}\mathop{\mathbb{E}}_{r \leftarrow \mathcal{M}(D)}[u(r)] &= \sum_{r \in \mathcal{R}} u(r) \cdot \Pr[\mathcal{M}(D) = r] \\ &\geq \sum_{r \in \mathcal{R}} u(r) \cdot e^{-\varepsilon} \Pr[\mathcal{M}(D_{-i}) = r] \\ &= e^{-\varepsilon} \mathop{\mathbb{E}}_{r \leftarrow \mathcal{M}(D_{-i})}[u(r)] .\end{aligned}$$

That is, the expected utility enjoyed by user i decreases by a factor of at most $e^{-\varepsilon}$ by participating in a private data release.

3 Basic techniques: Randomized response and noise addition [?, §3.2, 3.3]

So how can we achieve the definition of differential privacy? Let's focus right now on the task of answering a single counting query q.

3.1 Randomized response

The technique of randomized response was introduced by Warner in 1965 (predating the definition of differential privacy by over 40 years) in the context of conducting surveys where the results should be kept hidden even from the surveyor. Let $p > 1/2$ be a parameter. Each individual $i \in [n]$ is surveyed independently as follows, producing a report r_i. With probability p, she reports the true value $q(x_i)$ on her data. With the remaining probability $1 - p$, she flips her answer, reporting $1 - q(x_i)$. To estimate the average value $q(D)$, the surveyor computes

$$a_q = \frac{1}{2p-1}\left(\left(\frac{1}{n}\sum_{i=1}^{n} r_i\right) - (1-p)\right).$$

It is a simple calculation to show that this estimator is unbiased (i.e. $\mathbb{E}[a_q] = q(D)$) and has standard deviation $\sqrt{p(1-p)}/(2p-1)\sqrt{n}$. Note that for constant p, this is comparable to the sampling error when D is thought of as independent random samples from a larger population.

Proposition 7. *Randomized response provides* $\ln(p/(1-p))$*-differential privacy.*

Proof. Let $D \sim D'$ be neighboring datasets, differing in the data of individual i. We wish to bound the privacy loss function $L_{D,D'}(r)$. Since each individual acts independently, the privacy loss decomposes as a product where all terms except those corresponding to individual i are 1. By symmetry, there is only one interesting case for the term corresponding to i, which is

$$\ln \frac{\Pr[r_i = 1 | q(x_i) = 1]}{\Pr[r_i = 1 | q(x_i) = 0]} = \ln\left(\frac{p}{1-p}\right).$$

□

Thus, in order to guarantee ε-differential privacy, one should set $p = e^\varepsilon/(1+e^\varepsilon) \approx (1+\varepsilon)/2$.

3.2 The "Laplace mechanism"

We next describe what is perhaps a simpler algorithm that works very well in the case of answering a single counting query. The strategy now will be to answer a query $q(D)$ by simply adding random noise to the true answer. The tradeoff is that this mechanism crucially uses the existence of a trusted central curator.

Define the Laplace distribution via the density function $\mathrm{Lap}(\lambda) \sim \frac{1}{2\lambda}\exp(-|y|/\lambda)$. The Laplace mechanism answers a query q by reporting $a_q = q(D) + \mathrm{Lap}(1/\varepsilon n)$. Again, this is an unbiased estimator, but now the Laplace distribution has standard deviation $\sqrt{2}\lambda = \sqrt{2}/n\varepsilon$ – quadratically better than the sampling error.

Proposition 8. *The Laplace mechanism provides ε-differential privacy.*

Proof. Let $D \sim D'$, and note that for any counting query q, we have $|q(D) - q(D')| \le 1/n$. Then for any $r \in \mathbb{R}$, we have

$$\frac{\Pr[\mathcal{M}(D) = r]}{\Pr[\mathcal{M}(D') = r]} = \frac{\exp(-\varepsilon n|r - q(D)|)}{\exp(-\varepsilon n|r - q(D')|)} \le \exp(\varepsilon n|q(D) - q(D')|) \le e^\varepsilon.$$

□

4 Properties: Postprocessing, group privacy, composition [?, §3.5]

Differential privacy enjoys a number of properties that make it amenable to algorithm design and analysis.

4.1 Postprocessing

The definition of differential privacy is robust to postprocessing. This means that performing additional computations on the outcome of a differentially private algorithm does not weaken its privacy guarantees. While a simple idea, this property is surprisingly useful for interpreting and extracting useful statistical information, especially when coupled with the fact that one often explicitly knows the noise distribution introduced by a differentially private algorithm.

Proposition 9. *Let $\mathcal{M} : X^n \to \mathcal{R}$ be ε-differentially private, and let $\mathcal{A} : \mathcal{R} \to \mathcal{S}$ be a randomized algorithm. Then $\mathcal{A} \circ \mathcal{M}$ defined by $(\mathcal{A} \circ \mathcal{M})(D) = \mathcal{A}(\mathcal{M}(D))$ is also ε-differentially private.*

Proof. By convexity, it is enough to show this for deterministic algorithms $\mathcal{A}$. In this case, we have for any $D \sim D'$ and any $S \subseteq \mathcal{S}$ that

$$\Pr[(\mathcal{A} \circ \mathcal{M})(D) \in S] = \Pr[\mathcal{M}(D) \in \mathcal{A}^{-1}(S)] \le e^\varepsilon \Pr[\mathcal{M}(D') \in \mathcal{A}^{-1}(S)] = e^\varepsilon \Pr[(\mathcal{A} \circ \mathcal{M})(D) \in S].$$

□

4.2 Group privacy

Differential privacy automatically extends to guarantee privacy for small groups of individuals.

Proposition 10. *Let D and D' be datasets which differ in at most k entries. Then if $\mathcal{M} : X^n \to \mathcal{R}$ is ε-differentially private, we have for every $S \subseteq \mathcal{R}$*

$$\Pr[\mathcal{M}(D) \in S] \leq e^{k\varepsilon} \Pr[\mathcal{M}(D') \in S].$$

To prove this, just iterate the definition of differential privacy k times.

4.3 Composition

Perhaps the hallmark property of differential privacy is that it degrades gracefully and predictably under composition, i.e. when multiple differentially private algorithms are performed on the same (or overlapping) datasets. It is an important property for several reasons. The first is that privacy-preserving analyses do not occur in isolation; one does not simply answer a single counting query on a dataset and then throw it away. Instead, many different analyses are performed on each dataset, and moreover, an individual may choose to participate in many different sensitive datasets. Composition is key to thwarting differencing attacks when many such analyses are conducted independently.

The second reason is that composition enables differentially private programming. Algorithms are almost always designed and analyzed in a modular way, by putting together smaller building blocks and reasoning about the computational resources (e.g. time, space, randomness) required by the algorithm as whole in terms of those required by the individual building blocks. Composition theorems allow one to reason about privacy as yet another computational resource in exactly the same way.

The "basic" composition theorem states that running k differentially private algorithms on the same datasets guarantees $(k\varepsilon)$-differential privacy.

Proposition 11. *Let $\mathcal{M}_1, \ldots, \mathcal{M}_k$ be ε-differentially private. Then $\mathcal{M}(D) = (\mathcal{M}_1(D), \ldots, \mathcal{M}_k(D))$ (where each mechanism uses independent coin tosses) is $(k\varepsilon)$-differentially private.*

Proof. Let $D \sim D'$ be neighboring datasets. Fix a transcript $\vec{r} = (r_1, \ldots, r_k)$. Then we can write the privacy loss of the combined mechanism $\mathcal{M}$ as

$$\begin{aligned} L_{D,D'}(\vec{r}) &= \ln\left(\frac{\Pr[\mathcal{M}(D) = \vec{r}]}{\Pr[\mathcal{M}(D') = \vec{r}]}\right) \\ &= \sum_{i=1}^{k} \ln\left(\frac{\Pr[\mathcal{M}_i(D) = \vec{r}]}{\Pr[\mathcal{M}_i(D') = \vec{r}]}\right) \\ &\leq \sum_{i=1}^{k} \varepsilon = k\varepsilon. \end{aligned}$$

□

It turns out that the basic composition theorem is too pessimistic in its prediction about cumulative privacy loss. Although each individual mechanism might incur privacy loss ε with reasonable

probability, it turns out to be exceedingly unlikely that all of the mechanisms $\mathcal{M}_1, \ldots, \mathcal{M}_k$ will conspire against you to attain the worst-case cumulative bound of $k\varepsilon$. Most of the time, the cumulative bound looks more like $\sqrt{k}\varepsilon$. To formalize an improvement, we need to introduce a slight relaxation of the definition of differential privacy.

Definition 12. A randomized algorithm $\mathcal{M} : X^n \to \mathcal{R}$ provides (ε, δ)-differential privacy if, for all pairs of neighboring datasets $D \sim D'$ and all (measurable) sets of outcomes $S \subseteq \mathcal{R}$, we have

$$\Pr[\mathcal{M}(D) \in S] \leq e^{\varepsilon} \Pr[\mathcal{M}(D') \in S] + \delta.$$

A more-or-less equivalent way of thinking about (ε, δ)-differential privacy is that it guarantees that with probability all but δ (over the coin tosses of the mechanism), the privacy loss is bounded by ε. While we usually think of ε as a small constant, δ should be significantly smaller (e.g. $n^{-\omega(1)}$ in theory and 2^{-30} in practice).

Proposition 13. *Let $\mathcal{M}_1, \ldots, \mathcal{M}_k$ be ε-differentially private. Then for every $\delta > 0$, the composition $\mathcal{M}(D) = (\mathcal{M}_1(D), \ldots, \mathcal{M}_k(D))$ is $(k\varepsilon^2/2 + \sqrt{2k \ln(1/\delta)} \cdot \varepsilon, \delta)$-differentially private.*

The main idea behind the proof is to reason about the privacy loss *random variable* $Y_i = L^i_{D,D'}(\mathcal{M}_i(D))$ associated to each mechanism $\mathcal{M}_i$ and the datasets D, D'. While ε-differential privacy guarantees that each Y_i is bounded in $[-\varepsilon, \varepsilon]$, a (miraculous) calculation shows that the *expected* privacy loss $\mathbb{E}[Y_i]$ is much smaller, only at most $\frac{1}{2}\varepsilon^2$. Applying, say, Hoeffding's inequality shows that the total privacy loss $Y = \sum_{i=1}^k Y_i$ concentrates around its expectation $\frac{1}{2}k\varepsilon^2$, exceeding this expectation by more than $O(\sqrt{\ln(1/\delta)})$ standard deviations with probability less than δ.

The composition theorems allow us to extend the strategies from Section **??** to handle multiple queries. For instance, by adding Laplace noise at the scale of roughly $\sqrt{k}/\varepsilon n$, we can answer k arbitrary counting queries subject to differential privacy.

Proposition 14. *Answering each of k counting queries with independent noise Laplace noise of scale $O(\sqrt{k \log(1/\delta)}/\varepsilon n)$ yields (ε, δ)-differential privacy and α-accuracy for*

$$\alpha \gtrsim \frac{\sqrt{k \log(1/\delta)} \log k}{\varepsilon n}.$$

Proof. Privacy follows from the advanced composition theorem taking $\varepsilon_0 \approx \varepsilon/\sqrt{k \log(1/\delta)}$. The Laplace distribution has exponentially decaying tails, i.e. the probability that $|\operatorname{Lap}(\lambda)|$ exceeds $t\lambda$ falls as $\exp(-t)$. Therefore, by a union bound, w.h.p. all k noise variables are bounded by $O(\lambda \log k)$. □

5 Private multiplicative weights [?, §4.2]

I mentioned an amazing result of Hardt and Rothblum [**?**] for answering many counting queries with differential privacy. This algorithm gives a nice taste of how one actually does more complex algorithm design using the composition theorem and basic differentially private primitives. (Plus, it's gives another application of the versatile multiplicative weights framework.) It is also the state-of-the-art algorithm for releasing answers to an arbitrary family of counting queries $\mathcal{Q}$, and is now known to be matched by lower bounds [**?**]. (However, for some specific families of queries $\mathcal{Q}$, better algorithms are known.)

The private multiplicative weights algorithm requires a fairly dramatic way of rethinking how one answers queries. We'll cast the query release problem as an *online learning* problem as follows. Given a stream of queries $q_1, \ldots, q_k$ and a (hidden) sensitive dataset D, we will try to learn a predictor $\hat{D}$ for the values $q_1(D), \ldots, q_k(D)$ using only differentially private access to the real dataset D. That is, we will try to learn $\hat{D}$ such that $q_t(\hat{D}) \approx q_t(D)$ for every $t = 1, \ldots, k$.

It will be useful to take a different view of a dataset D as a histogram over the data universe X. With this view, we write $D[x]$ to denote the probability mass that D assigns to universe element x. The value of a counting query can now be written as $q(D) = \mathbb{E}_{x \leftarrow D}[q(x)]$.

We begin by presenting a *non*-differentially private algorithm that uses the multiplicative weights update rule to learn a good predictor.

Algorithm 1 A non-private MW query release algorithm

- Initialize $\hat{D}_0 = \text{Unif}(X)$
- For $t = 1, 2, \ldots, k$:
  - If $|q_t(\hat{D}_{t-1}) - q_t(D)| \leq \alpha$: Output $a_t = q_t(\hat{D}_{t-1})$, update $D_t = D_{t-1}$
  - Else:
    - Output $a_t = q_t(D)$
    - For each $x \in X$, let

$$m_t(x) = \begin{cases} q_t(x) & \text{if } q_t(\hat{D}_{t-1}) > q_t(D) + \alpha, \\ 1 - q_t(x) & \text{if } q_t(\hat{D}_{t-1}) < q_t(D) - \alpha \end{cases}$$

    - Update each weight $\hat{D}_t[x] = \hat{D}_{t-1}[x] \cdot (1 - \alpha m_t(x))$
    - Renormalize $\hat{D}_t$

The key idea for turning this into a good differentially private algorithm is that the number of "update" rounds is small. The point will be that only the update rounds require a costly access to the sensitive dataset, so the cumulative privacy loss incurred by running the algorithm looks more like what is incurred by answering queries in only the update rounds, rather than what would be incurred by answering each query independently.

Claim 15. *For any dataset D and any sequence of queries $q_1, \ldots, q_k$, the number of update rounds in the MW query release algorithm is at most $O(\log |X|/\alpha^2) = O(d/\alpha^2)$.*

Proof Sketch. Consider the potential function $\Psi_t = \text{KL}(D \| D_t)$. Then $\Psi_0 \leq \log |X|$ and $\Psi_t \geq 0$ for every t. If t is an update round, then a calculation shows that

$$\Psi_{t-1} - \Psi_t \geq \frac{\alpha}{2}|q_t(D_{t-1}) - q_t(D)| - \frac{\alpha^2}{4} \geq \frac{\alpha^2}{4}.$$

□

With this in mind, we can state a differentially private version of the MW query release algorithm, which ensures that its accesses to the original dataset D are privacy-preserving.

Algorithm 2 Private MW query release algorithm

- Initialize $\hat{D}_0 = \text{Unif}(X)$
- Initialize noisy threshold $T = \alpha + \text{Lap}(\lambda)$
- For $t = 1, 2, \ldots, k$:
  - If $q_t(\hat{D}_{t-1}) - q_t(D) \leq T + \text{Lap}(\lambda)$ and $q_t(D) - q_t(\hat{D}_{t-1}) \leq T + \text{Lap}(\lambda)$: Output $a_t = q_t(\hat{D}_{t-1})$, update $D_t = D_{t-1}$
  - Else:
    - Output $a_t = q_t(D) + \text{Lap}(\lambda)$
    - For each $x \in X$, let

$$m_t(x) = \begin{cases} q_t(x) & \text{if } q_t(\hat{D}_{t-1}) \gg q_t(D), \\ 1 - q_t(x) & \text{if } q_t(\hat{D}_{t-1}) \ll q_t(D) \end{cases}$$

    - Update each weight $\hat{D}_t[x] = \hat{D}_{t-1}[x] \cdot (1 - \alpha m_t(x))$
    - Renormalize $\hat{D}_t$

Proposition 16. *The private multiplicative weights algorithm is (ε, δ)-differentially private, and provides $O(\alpha)$-accurate answers for*

$$\alpha = \frac{\sqrt{\log k}(d \log(1/\delta))^{1/4}}{\sqrt{\varepsilon n}}$$

Proof Sketch. Set $\lambda = O(\sqrt{d \log(1/\delta)}/\alpha\varepsilon n)$. Intuitively, one only needs to "pay" for privacy in each of the update rounds, of which there are at most $O(d/\alpha^2)$. So by advanced composition, this setting of the noise parameter yields (ε, δ)-differential privacy. (The formal analysis is more subtle, since it must deal with the noisy comparisons made in non-update rounds. The general paradigm for doing this is referred to as the "sparse vector" technique in [**?**].)

By a union bound over the k rounds of the algorithm, all of the Laplace noise variables have magnitude at most α as long as $\alpha \gtrsim \lambda \log k = O(\log k \sqrt{d \log(1/\delta)}/\alpha\varepsilon n)$. Rearranging gives the desired bound on α. □

6 Differential Privacy and False Discovery

An important interpretation of the definition of differential privacy is that it is a strong *stability guarantee*. Namely, differentially private algorithms have outputs which are "stable" with respect to changes in their inputs that are localized to a few rows. This makes differential privacy quite a useful concept in other areas of algorithm design, even when privacy is not a concern itself.

The first applications of differential privacy outside the realm of privacy were in economic mechanism design. Since then, a few more connections have been discovered in learning theory and cryptography. Today, we will focus on one miraculous application of differential privacy to the problem of *false discovery control* in interactive data analysis [**?**, **?**].

Let $\mathcal{P}$ be a distribution over a data domain X. The definition of a counting query naturally extends to such a distribution by taking

$$q(\mathcal{P}) = \mathop{\mathbb{E}}_{x \sim \mathcal{P}} [q(x)] .$$

A basic statistical task is to estimate the values of a sequence of counting queries $q_1, \ldots, q_k$ on a distribution $\mathcal{P}$ given access to a finite sample $D = (x_1, \ldots, x_n)$ of i.i.d. draws from $\mathcal{P}$. The natural way to estimate each value $q_i(\mathcal{P})$ is via the empirical fractional count $q_i(D)$. A *generalization* argument shows that as long as the sample size n is large enough, then these estimates will be accurate. Specifically, a Chernoff bound plus a union bound can be used to show that for every *fixed* collection $q_1, \ldots, q_k$ of counting queries, this strategy guarantees that w.h.p. every $q_i(\mathcal{P})$ can be estimated to within $\pm\alpha$ as long as the number of samples $n \geq O(\log k/\alpha^2)$.

For this analysis to go through, it is important to make the assumption that the queries $q_1, \ldots, q_k$ are fixed in advance, i.e., independently of the choice of the sample D. But it real life, statistical analyses are often conducted adaptively, with each query q_{i+1} chosen after using D to estimate $q_i(\mathcal{P})$. Not only does adaptivity cause the Chernoff bound + union bound analysis to break down, but it can be shown that using the above strategy to answer some sequences of adaptively chosen counting queries requires $n \geq \Omega(k/\alpha^2)$—an exponential gap! In other words, an adaptive analyst can find a query that fails to generalize to the underlying population after asking about $O(n)$ queries.

This problem can be addressed by estimating each $q_i(\mathcal{P})$ not necessarily with $q_i(D)$ itself, but with some perturbed estimate coming from a randomized algorithm $\mathcal{M}(D)$. It turns out that if $\mathcal{M}$ is differentially private, then it automatically protects the adaptive data analyst from making false discoveries.

Theorem 17 (Informal, [**?**])**.** *Let $\mathcal{Q}$ be a collection of counting queries, and let $\mathcal{M}$ be an (ε, δ)-differentially private and α-accurate algorithm capable of answering k adaptively chosen queries from $\mathcal{Q}$. Then the answers $\mathcal{M}$ provides are $(\alpha+\varepsilon)$-accurate with respect to an underlying population.*

Combining the above theorem with the guarantees of Private Multiplicative Weights gives the following result.

Theorem 18. *Let $\mathcal{Q}$ be a collection of counting queries over data universe X. Then there is an algorithm $\mathcal{M}$ which answers k adaptively chosen with answers that are α-accurate with respect to the population as long as*

$$n \gtrsim \frac{\log k \sqrt{\log |X|}}{\alpha^3}.$$

7 Negative Results via Cryptography

Sophisticated approaches for answering counting queries, including private multiplicative weights, show that in principle we can extract a lot of useful statistical information about sensitive data. However, algorithms for answering exponentially many queries are quite slow, running in time $\text{poly}(n, |\mathcal{Q}|, |X|)$, which is in particular exponential in the dimensionality $d = \log |X|$ of the data.

Ideally, given a large but concisely described family $\mathcal{Q}$ of queries, we would like to design an algorithm $\mathcal{M}$ that runs in time $\text{poly}(n, \log |X|)$ and is capable of outputting a small data structure that can be used to accurately answer all of the queries in $\mathcal{Q}$. The most intuitively appealing such data structure is a "synthetic dataset", which is a small collection $\hat{D}$ of "fake" records from the data universe X with the property that $q(\hat{D}) \approx q(D)$ for every $q \in \mathcal{Q}$.

7.1 Hardness of Generating Synthetic Data

Unfortunately, under standard cryptographic assumptions, it turns out to be computationally infeasible to privately generate useful synthetic data. This is true even for very simple classes $\mathcal{Q}$, such as the class of two-way conjunctions:

Theorem 19 (Informal, [**?**])**.** *Assuming the existence of a secure* digital signature scheme*, for every $n = \text{poly}(d)$ there is no algorithm running in time $\text{poly}(d)$ producing useful synthetic data for 2-way conjunctions with differential privacy.*

7.2 Hardness of Sanitization from Traitor-Tracing

While appealing, synthetic data is not the only type of data structure one might imagine using to answer counting queries. Nevertheless, a connection between differential privacy and cryptographic *traitor-tracing schemes* shows that it is infeasible to answer some collections of queries using any data structure whatsoever.

Theorem 20 (Informal, [**?**])**.** *Assuming the existence of a secure* traitor-tracing scheme *with "ciphertext length" c and "key length" d, then there is a collection $\mathcal{Q}$ of 2^c counting queries over a domain of size 2^d that cannot be answered efficiently with differential privacy.*

The above theorem can be instantiated using various traitor-tracing schemes which in turn can be built from various cryptographic assumptions. While the resulting infeasibility results hold for arbitrary query-answering data structures, the tradeoff is that the collections of queries they apply to are no longer "natural." (Roughly speaking, there is a counting query for every possible ciphertext, which when evaluated on a secret key, asks whether that key would decrypt the ciphertext to a 1.)

References

[BNSSSU16] Raef Bassily, Kobbi Nissim, Adam Smith, Thomas Steinke, Uri Stemmer, and Jonathan Ullman. Algorithmic stability for adaptive data analysis. In *STOC* 2016.

[BUV14] Mark Bun, Jonathan Ullman, and Salil Vadhan. Fingerprinting codes and the price of approximate differential privacy. In *STOC* 2014.

[DFHPRR15] Cynthia Dwork, Vitaly Feldman, Moritz Hardt, Toniann Pitassi, Omer Reingold, and Aaron Roth. Preserving statistical validity in adaptive data analysis. In *STOC* 2015.

[DN03] Irit Dinur and Kobbi Nissim. Revealing information while preserving privacy. In *PODS* 2003.

[DNRRV09] Cynthia Dwork, Moni Naor, Omer Reingold, Guy Rothblum, and Salil Vadhan. On the complexity of differentially private data release: Efficient algorithms and hardness results. In *STOC* 2009.

[DR14] Cynthia Dwork and Aaron Roth. The algorithmic foundations of differential privacy. Foundations and Trends in Theoretical Computer Science. `https://www.cis.upenn.edu/~aaroth/Papers/privacybook.pdf`

[HR10] Moritz Hardt and Guy Rothblum. A multiplicative weights mechanism for privacy-preserving data analysis. In *FOCS* 2010.

[HU14] Moritz Hardt and Jonathan Ullman. Preventing false discovery in interactive data analysis is hard. In *FOCS* 2014.

[KS14] Shiva Kasiviswanathan and Adam Smith. On the 'semantics' of differential privacy: A Bayesian formulation. Journal of Privacy and Confidentiality, 2014.

[UV11] Jonathan Ullman and Salil Vadhan. PCPs and the hardness of generating private synthetic data. In *TCC* 2011.

[Vad16] Salil Vadhan. On the complexity of differential privacy. `https://privacytools.seas.harvard.edu/publications/complexity-differential-privacy`