Chapter 7
The Complexity of Differential Privacy

Salil Vadhan

Abstract Differential privacy is a theoretical framework for ensuring the privacy of individual-level data when performing statistical analysis of privacy-sensitive datasets. This tutorial provides an introduction to and overview of differential privacy, with the goal of conveying its deep connections to a variety of other topics in computational complexity, cryptography, and theoretical computer science at large. This tutorial is written in celebration of Oded Goldreich's 60th birthday, starting from notes taken during a minicourse given by the author and Kunal Talwar at the 26th McGill Invitational Workshop on Computational Complexity [1].

To Oded, my mentor, role model, collaborator, and friend.
Your example gives me a sense of purpose as a researcher.

Salil Vadhan
Center for Research on Computation & Society, School of Engineering & Applied Sciences, Harvard University, Cambridge, Massachusetts, USA, e-mail: salil@seas.harvard.edu. webpage: http://seas.harvard.edu/~salil. Written in part while visiting the Shing-Tung Yau Center and the Department of Applied Mathematics at National Chiao-Tung University in Hsinchu, Taiwan. Supported by NSF grant CNS-1237235, a grant from the Sloan Foundation, and a Simons Investigator Award.

Y. Lindell (ed.), *Tutorials on the Foundations of Cryptography*,
Information Security and Cryptography, DOI 10.1007/978-3-319-57048-8_7

7.1 Introduction and Definition

7.1.1 Motivation

Suppose you are a researcher in the health or social sciences who has collected a rich dataset on the subjects you have studied, and want to make the data available to others to analyze as well. However, the dataset has sensitive information about your subjects (such as disease diagnoses, financial information, or political affiliations), and you have an obligation to protect their privacy. What can you do?

The traditional approach to such privacy problems is to try to "anonymize" the dataset by removing obvious identifiers, such as name, address, and date of birth, and then share the anonymized dataset. However, it is now well understood that this approach is ineffective, because the data that remains is often still sufficient to determine who is who in the dataset, given appropriate auxiliary information. This threat is not hypothetical; there have now been many high-visibility demonstrations that such "re-identification" attacks are often quite easy to carry out in practice, using publicly available datasets as sources of auxiliary information [84].

A more promising approach is to mediate access to the data through a trusted interface, which will only answer queries posed by data analysts. However, ensuring that such a system protects privacy is nontrivial. Which queries should be permitted? Clearly, we do not want to allow queries that target a particular individual (such as "Does Sonny Rollins have sensitive trait X?"), even if they are couched as aggregate queries (e.g., "How many people in the dataset are 84-year-old jazz saxophonists with trait X?"). Even if a single query does not seem to target an individual, a combination of results from multiple queries can do so (e.g., "How many people in the dataset have trait X?" and "How many people in the dataset have trait X and are *not* 84-year-old jazz saxophonists?"). These attacks can sometimes be foiled by only releasing *approximate* statistics, but Dinur and Nissim [31] exhibited powerful "reconstruction attacks" which showed that, given sufficiently many approximate statistics, one can reconstruct almost the entire dataset. Thus, there are fundamental limits to what can be achieved in terms of privacy protection while providing useful statistical information, and we need a theory that can assure us that a given release of statistical information is safe.

Cryptographic tools such as secure function evaluation and functional encryption do not address these issues. The kind of security guarantee such tools provide is that nothing is leaked *other than the outputs of the functions being computed*. Here we are concerned about the possibility that the outputs of the functions (i.e., queries) already leak too much information. Indeed, addressing these privacy issues is already nontrivial in a setting with a trusted data curator, whereas the presence of a trusted third party trivializes most of cryptography.

Differential privacy is a robust definition of privacy protection for data-analysis interfaces that:

- ensures meaningful protection against adversaries with arbitrary auxiliary information (including ones that are intimately familiar with the individuals they are targeting),

- does not restrict the computational strategy used by the adversary (in the spirit of modern cryptography), and
- provides a quantitative theory that allows us to reason about how much statistical information is safe to release and with what accuracy.

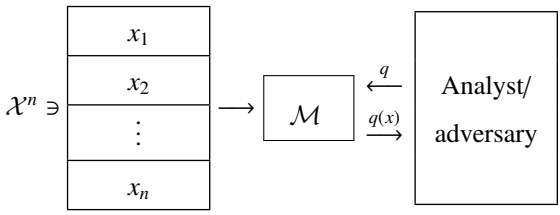
Following the aforementioned reconstruction attacks of Dinur and Nissim [31], the concept of differential privacy emerged through a series of papers by Dwork and Nissim [35], Blum, Dwork, McSherry, and Nissim [13], and Dwork, McSherry, Nissim, and Smith [48], with the latter providing the elegant indistinguishability-based definition that we will see in the next section.

In the decade since differential privacy was introduced, a large algorithmic literature has developed showing that differential privacy is compatible with a wide variety of data-analysis tasks. It also has attracted significant attention from researchers and practitioners outside theoretical computer science, many of whom are interested in bringing differential privacy to bear on real-life data-sharing problems. At the same time, it has turned out to be extremely rich from a theoretical perspective, with deep connections to many other topics in theoretical computer science and mathematics. The latter connections are the focus of this tutorial, with an emphasis on connections to topics in computational complexity and cryptography. For a more in-depth treatment of the algorithmic aspects of differential privacy, we recommend the monograph of Dwork and Roth [36].

7.1.2 The Setting

The basic setting we consider is where a trusted curator holds a dataset x about n individuals, which we model as a tuple $x \in \mathcal{X}^n$, for a *data universe* $\mathcal{X}$. The interface to the data is given by a (randomized) *mechanism* $\mathcal{M} : \mathcal{X}^n \times \mathcal{Q} \to \mathcal{Y}$, where $\mathcal{Q}$ is the *query space* and $\mathcal{Y}$ is the *output space* of $\mathcal{M}$. To avoid introducing continuous probability formalism (and to be able to discuss algorithmic issues), we will assume that $\mathcal{X}$, $\mathcal{Q}$, and $\mathcal{Y}$ are discrete.

The picture we have in mind is as follows:



for a dataset $x = (x_1, \ldots, x_n)$.

7.1.3 Counting Queries

A basic type of query that we will examine extensively is a *counting query*, which is specified by a predicate on rows $q \colon \mathcal{X} \to \{0, 1\}$, and is extended to datasets $x \in \mathcal{X}^n$ by counting the fraction of people in the dataset satisfying the predicate:

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

(Note that we abuse notation and use q for both the predicate on rows and the function that averages q over a dataset.) The examples mentioned above in Section 7.1.1 demonstrate that it is nontrivial to ensure privacy even when answering counting queries, because answers to several counting queries can be combined to reveal information about individual rows.

There are several specific families of counting queries that are important for statistical analysis and will come up many times in this tutorial:

Point Functions (Histograms): Here $\mathcal{X}$ is an arbitrary set and for each $y \in \mathcal{X}$ we consider the predicate $q_y : \mathcal{X} \to \{0,1\}$ that evaluates to 1 only on input y. The family $\mathcal{Q}^{\text{pt}} = \mathcal{Q}^{\text{pt}}(\mathcal{X})$ consists of the counting queries corresponding to all point functions on data universe $\mathcal{X}$. (Approximately) answering all of the counting queries in $\mathcal{Q}^{\text{pt}}$ amounts to (approximately) computing the *histogram* of the dataset.

Threshold Functions (CDFs): Here $\mathcal{X}$ is a totally ordered set, and we consider the set $\mathcal{Q}^{\text{thr}} = \mathcal{Q}^{\text{thr}}(\mathcal{X})$ of threshold functions. That is, for each $y \in \mathcal{X}$, $\mathcal{Q}^{\text{thr}}$ contains counting query corresponding to the function $q_y(z)$ that outputs 1 iff $z \le y$. (Approximately) answering all of the counting queries in $\mathcal{Q}^{\text{thr}}$ is tantamount to (approximating) the *cumulative distribution function* of the dataset.

Attribute Means (1-way Marginals): Here $\mathcal{X} = \{0,1\}^d$, so each individual has d Boolean attributes, and $\mathcal{Q}^{\text{means}} = \mathcal{Q}^{\text{means}}(d)$ contains the counting queries corresponding to the d coordinate functions $q_j : \{0,1\}^d \to \{0,1\}$ defined by $q_j(w) = w_j$ for $j = 1, \ldots, d$. Thus, (approximately) answering all of the queries in $\mathcal{Q}^{\text{means}} = \mathcal{Q}^{\text{means}}(d)$ amounts to (approximately) computing the fraction of the dataset possessing each of the d attributes. These are also referred to as the *(1-way) marginal statistics* of the dataset.

Conjunctions (Contingency Tables): Here again $\mathcal{X} = \{0,1\}^d$, and for an integer $t \in \{0,1,2,\ldots,d\}$, we consider the family $\mathcal{Q}^{\text{conj}}_t = \mathcal{Q}^{\text{conj}}_t(d)$ of counting queries corresponding to conjunctions of t literals. For example, $\mathcal{Q}^{\text{conj}}_2(5)$ contains the function $q(w) = w_2 \wedge \neg w_4$, which could represent a query like "what fraction of individuals in the dataset have lung cancer and are non-smokers?". Notice that $\mathcal{Q}^{\text{conj}}_1(d)$ consists of the queries in $\mathcal{Q}^{\text{means}}(d)$ and their negations, and $\mathcal{Q}^{\text{conj}}_d(d)$ contains the same queries as $\mathcal{Q}^{\text{pt}}(\{0,1\}^d)$. We have $|\mathcal{Q}^{\text{conj}}_t(d)| = \binom{d}{t} \cdot 2^t = d^{\Theta(t)}$ when $t \le d^{1-\Omega(1)}$. We also consider the family $\mathcal{Q}^{\text{conj}} = \mathcal{Q}^{\text{conj}}(d) = \cup_{t=0}^{d} \mathcal{Q}^{\text{conj}}_t(d)$, which is of size 3^d. The counting queries in $\mathcal{Q}^{\text{conj}}_t$ are also called *t-way marginals* and answering all of them amounts to computing the *t-way contingency table* of the dataset. These are important queries for statistical analysis, and indeed the answers to all queries in $\mathcal{Q}^{\text{conj}}$ is known to be a "sufficient statistic" for "logit models."

Arbitrary Queries: Sometimes we will not impose any structure on the data universe $\mathcal{X}$ or query family $\mathcal{Q}$ except possibly to restrict attention to families of

efficiently computable queries. For the latter, we encode elements of both $\mathcal{X}$ and $\mathcal{Q}$ as strings, so $\mathcal{X} = \{0,1\}^d$, $\mathcal{Q} = \{q_y : \mathcal{X} \to \{0,1\}\}_{y \in \{0,1\}^s}$ for some $s, d \in \mathbb{N}$, where $q_y(w) = \mathsf{Eval}(y, w)$ for some polynomial-time evaluation function $\mathsf{Eval} : \{0,1\}^s \times \{0,1\}^d \to \{0,1\}$.

7.1.4 Differential Privacy

The definition of differential privacy requires that no individual's data has much effect on what an adversary sees. That is, if we consider any two datasets x and x' that differ on one row (which we will denote $x \sim x'$), the output distribution of $\mathcal{M}$ on x should be "similar" to that of $\mathcal{M}$ on x'. Specifically, we require that

$$\forall T \subseteq \mathcal{Y},\ \Pr[\mathcal{M}(x,q) \in T] \leq (1+\varepsilon) \cdot \Pr[\mathcal{M}(x,q) \in T]\,.$$

The reverse relationship ($\Pr[\mathcal{M}(x',q) \in T] \leq (1+\varepsilon) \cdot \Pr[\mathcal{M}(x,q) \in T]$) follows by symmetry, swapping x and x'. The choice of a multiplicative measure of closeness between distributions is important, and we will discuss the reasons for it later. It is technically more convenient to use e^{ε} instead of $(1+\varepsilon)$, because the former behaves more nicely under multiplication ($e^{\varepsilon_1} \cdot e^{\varepsilon_2} = e^{\varepsilon_1+\varepsilon_2}$). This gives the following formal definition:

Definition 7.1.1 ((Pure) differential privacy [48]). *For $\varepsilon \geq 0$, we say that a randomized mechanism $\mathcal{M} \colon \mathcal{X}^n \times \mathcal{Q} \to \mathcal{Y}$ is* ε-differentially private *if, for every pair of neighboring datasets $x \sim x' \in \mathcal{X}^n$ (i.e., x and x' differ in one row) and every query $q \in \mathcal{Q}$, we have*

$$\forall T \subseteq \mathcal{Y},\ \Pr[\mathcal{M}(x,q) \in T] \leq e^{\varepsilon} \cdot \Pr[\mathcal{M}(x',q) \in T]\,.$$

Equivalently,

$$\forall y \in \mathcal{Y},\ \Pr[\mathcal{M}(x,q) = y] \leq e^{\varepsilon} \cdot \Pr[\mathcal{M}(x',q) = y]\,.$$

Here we typically take ε as small, but nonnegligible (not cryptographically small), for example, a small constant, such as $\varepsilon = 0.1$. Smaller ε provides better privacy, but as we will see, the definition is no longer useful when $\varepsilon < 1/n$. We will also think of n as known and public information, and we will study asymptotic behavior as $n \to \infty$.

We will often think of the query as fixed, and remove q from notation. In this section, we consider answering only one query; a major focus of subsequent sections will be the problem of answering many queries.

7.1.5 Basic Mechanisms

Before discussing the definition further, let us see some basic constructions of differentially private mechanisms.

Randomized response. Let $q : \mathcal{X} \to \{0,1\}$ be a counting query, and $x \in \mathcal{X}^n$ be a dataset. For each row x_i, let

$$y_i = \begin{cases} q(x_i) & \text{with prob. } (1+\varepsilon)/2\,, \\ \neg q(x_i) & \text{with prob. } (1-\varepsilon)/2 \end{cases}$$

and

$$\mathcal{M}(x_1, \ldots, x_n) = (y_1, \ldots, y_n)\,.$$

If $x \sim x'$ are datasets that differ on the i-th row, their output distributions differ only if $q(x_i) \neq q(x'_i)$, in which case the outputs differ only in the i-th components, denoted y_i and y'_i, respectively. We have

$$\frac{\Pr[y_i = q(x_i)]}{\Pr[y'_i = q(x_i)]} = \frac{(1+\varepsilon)/2}{(1-\varepsilon)/2} = \mathrm{e}^{O(\varepsilon)}\,.$$

And $\Pr[y_i = q(x'_i)] \leq \Pr[y'_i = q(x'_i)]$. Thus, randomized response is $O(\varepsilon)$-differentially private.

We can use the result of randomized response to estimate the value of the counting query $q(x)$ as follows. Note that $\mathrm{E}[y_i] = \varepsilon \cdot q(x_i) + (1-\varepsilon)/2$. Thus, by the Chernoff bound, with high probability we have

$$\left| \frac{1}{n} \sum_i \frac{1}{\varepsilon} \cdot \left(y_i - \frac{(1-\varepsilon)}{2} \right) - q(x) \right| \leq O\left(\frac{1}{\sqrt{n} \cdot \varepsilon} \right).$$

As $n \to \infty$, we get an increasingly accurate estimate of the average.

An advantage of randomized response is that it does not require a trusted, centralized data curator; each subject can carry out the randomization on her own and publicly announce her noisy bit y_i. Indeed, this method was introduced in the 1960s by Warner [108] for carrying out sensitive surveys in the social sciences, where participants may not feel comfortable revealing information to the surveyor. In Section 7.9, we will discuss the "local model" for differential privacy, which encompasses general mechanisms and interactive protocols where subjects ensure their own privacy and need not trust anyone else.

The Laplace mechanism [48]. Let q be a counting query; it is natural to try to protect privacy by simply adding noise. That is, $\mathcal{M}(x) = q(x) + \text{noise}$. But how much noise do we need to add, and according to what distribution?

Note that, if $x \sim x'$, we have $|q(x) - q(x')| \leq 1/n$. This suggests "noise" of magnitude $1/(\varepsilon n)$ should be enough to make $\mathcal{M}(x)$ and $\mathcal{M}(x')$ "ε-indistinguishable" in the sense required by differential privacy.

Which distribution will satisfy the multiplicative definition of differential privacy? Recall that, at every output y, the density of the output distribution should be the same under x and x' up to a factor of e^{ε}. The density of $\mathcal{M}(x)$ at y is the density of the noise distribution at $z = y - q(x)$, and the density of $\mathcal{M}(x')$ at y is the density of the noise distribution at $z' = y - q(x')$; again $|z - z'| \leq 1/n$. So we see that it

suffices for the density of the noise distribution to change by a factor of at most e^{ε} over intervals of length $1/n$.

This leads us to the *Laplace distribution* $\mathrm{Lap}(\sigma)$:

$$\text{the density of } \mathrm{Lap}(\sigma) \text{ at } z \propto e^{-|z|/\sigma}.$$

If we set $\sigma = 1/\varepsilon n$, then we see that the ratio of densities is as we want: for $z \geq 0$, we have

$$\frac{\text{density of } \mathrm{Lap}(1/\varepsilon n) \text{ at } z + 1/n}{\text{density of } \mathrm{Lap}(1/\varepsilon n) \text{ at } z} = e^{1/(n\sigma)} = e^{-\varepsilon}.$$

(For $z \leq -1/n$, the ratio of densities is e^{ε}, and for $z \in (-1/n, 0)$, it is between $e^{-\varepsilon}$ and e^{ε}.)

It may seem more natural to use Gaussian noise, but it does not quite achieve the definition of differential privacy that we have given: in the tail of a Gaussian, the density changes by an unbounded multiplicative factor over intervals of fixed width. Later, we will see a relaxation of differential privacy (called (ε, δ)-differential privacy) that is achieved by adding Gaussian noise of appropriate variance.

$\mathrm{Lap}(\sigma)$ has mean 0 and standard deviation $\sqrt{2} \cdot \sigma$, and has exponentially vanishing tails:

$$\Pr[|\mathrm{Lap}(\sigma)| > \sigma t] \leq e^{-t}.$$

The Laplace mechanism is not specific to counting queries; all we used was that $|q(x) - q(x')| \leq 1/n$ for $x \sim x'$. For an arbitrary query $q : \mathcal{X}^n \to \mathbb{R}$, we need to scale the noise to its *global sensitivity*:

$$\mathrm{GS}_q = \max_{x \sim x'} |q(x) - q(x')|.$$

Then we have:

Definition 7.1.2 (The Laplace mechanism). *For a query $q : \mathcal{X}^n \to \mathbb{R}$, a bound B, and $\varepsilon > 0$, the* Laplace mechanism $\mathcal{M}_{q,B}$ *over data universe $\mathcal{X}$ takes a dataset $x \in \mathcal{X}^n$ and outputs*

$$\mathcal{M}_{q,B}(x) = q(x) + \mathrm{Lap}(B/\varepsilon).$$

From the discussion above, we have:

Theorem 7.1.3 (Properties of the Laplace mechanism).

1. *If $B \geq \mathrm{GS}_q$, the Laplace mechanism $\mathcal{M}_{q,B}$ is ε-differentially private.*
2. *For every $x \in \mathcal{X}^n$ and $\beta > 0$,*

$$\Pr[|\mathcal{M}_{q,B}(x) - q(x)| > (B/\varepsilon) \cdot \ln(1/\beta)] \leq \beta.$$

As noted above, for a counting query q, we can take $B = 1/n$, and thus with high probability we get error $O(1/(\varepsilon n))$, which is significantly better than the bound of $O(1/\varepsilon\sqrt{n})$ given by randomized response.

Global sensitivity is also small for a variety of other queries of interest:

1. For $q(x) = \max\{q_1(x), q_2(x), \ldots, q_t(x)\}$, we have $\mathrm{GS}_q \leq \max_i\{\mathrm{GS}_{q_i}\}$.
2. For $q(x) = d(x, H)$ where $H \subseteq \mathcal{X}^n$ and d is Hamming distance,[1] we have $\mathrm{GS}_q \leq 1$. ("Is my data set close to one that satisfies my hypothesis H?").
3. A *statistical query* (sometimes called a *linear query* in the differential privacy literature) is a generalization of a counting query to averaging a real-valued function on the dataset. That is, we are given a bounded function $q : \mathcal{X} \to [0, 1]$, and are interested in the query:

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

Then $\mathrm{GS}_q \leq 1/n$.

We promised that we would only work with discrete probability, but the Laplace distribution is continuous. However, one can discretize both the query values $q(x)$ and the Laplace distribution to integer multiples of B (yielding a scaled version of a geometric distribution) and Theorem 7.1.3 will still hold. We ignore this issue in the rest of the tutorial for the sake of simplicity (and consistency with the literature, which typically refers to the continuous Laplace distribution).

7.1.6 Discussion of the Definition

We now discuss why differential privacy utilizes a multiplicative measure of similarity between the probability distributions $\mathcal{M}(x)$ and $\mathcal{M}(x')$.

Why not statistical distance? The first choice that one might try is to use statistical difference (total variation distance). That is, we require that, for every $x \sim x'$, we have

$$\mathrm{SD}(\mathcal{M}(x), \mathcal{M}(x')) \stackrel{\text{def}}{=} \max_{T \subseteq \mathcal{Y}} \left|\Pr[\mathcal{M}(x) \in T] - \Pr[\mathcal{M}(x') \in T]\right| \leq \delta.$$

ε-Differential privacy implies the above definition with $\delta = 1 - \mathrm{e}^{-\varepsilon} \leq \varepsilon$, but not conversely.

We claim that, depending on the setting of δ, such a definition either does not allow for any useful computations or does not provide sufficient privacy protection.

$\delta \leq 1/2n$: Then by a hybrid argument, for all pairs of datasets $x, x' \in \mathcal{X}^n$ (even nonneighbors), we have $\mathrm{SD}(\mathcal{M}(x), \mathcal{M}(x')) \leq n\delta \leq 1/2$. Taking x' to be a fixed (e.g., all-zeroes) dataset, this means that, with probability $1/2$ on $\mathcal{M}(x)$, we get an answer independent of the dataset x and the mechanism is useless.

$\delta \geq 1/2n$: In this case, the mechanism "with probability $1/2$, output a random row of the dataset" satisfies the definition. We do not consider a mechanism that outputs an individual's data in the clear to be protecting privacy.

[1] The *Hamming distance* $d(x, x')$ between two datasets $x, x' \in \mathcal{X}^n$ is the number of rows on which x and x' differ.

However, it turns out to be quite useful to consider the following relaxation of differential privacy, which incorporates a negligible statistical distance term δ in addition to the multiplicative ε:

Definition 7.1.4 ((Approximate) differential privacy). *For $\varepsilon \geq 0, \delta \in [0,1]$, we say that a randomized mechanism $\mathcal{M} \colon \mathcal{X}^n \times \mathcal{Q} \to \mathcal{Y}$ is* (ε, δ)-differentially private *if, for every two neighboring datasets $x \sim x' \in \mathcal{X}^n$ (x and x' differ in one row) and every query $q \in \mathcal{Q}$, we have*

$$\forall T \subseteq \mathcal{Y},\ \Pr[\mathcal{M}(x,q) \in T] \leq \mathrm{e}^{\varepsilon} \cdot \Pr[\mathcal{M}(x',q) \in T] + \delta\,. \tag{7.1}$$

Here, we will insist that δ is cryptographically negligible (in particular, $\delta \leq n^{-\omega(1)}$); it can be interpreted as an upper bound on the probability of catastrophic failure (e.g., the entire dataset being published in the clear). This notion is often called *approximate differential privacy*, in contrast with *pure differential privacy* as given by Definition 7.1.1. Note that, unlike pure differential privacy, with approximate differential privacy it is *not* sufficient to verify Inequality (7.1) for sets T of size 1. (Consider a mechanism that outputs the entire dataset along with a random number from $\{1, \ldots, \lceil 1/\delta \rceil\}$; then $\Pr[\mathcal{M}(x,q) = y] \leq \delta \leq \mathrm{e}^{\varepsilon} \cdot \Pr[\mathcal{M}(x',q) = y] + \delta$ for all y, but clearly does not provide any kind of privacy or satisfy Definition 7.1.4.)

More generally, we will call two random variables Y and Y' taking values in $\mathcal{Y}$ (ε, δ)-*indistinguishable* if:

$$\forall T \subseteq \mathcal{Y},\ \Pr[Y \in T] \leq \mathrm{e}^{\varepsilon} \cdot \Pr[Y' \in T] + \delta, \text{ and}$$
$$\Pr[Y' \in T] \leq \mathrm{e}^{\varepsilon} \cdot \Pr[Y \in T] + \delta$$

Setting $\varepsilon = 0$ is equivalent to requiring that $\mathrm{SD}(Y, Y') \leq \delta$. (ε, δ)-Indistinguishability has the following nice characterization, which allows us to interpret (ε, δ)-differential privacy as "ε-differential privacy with probability at least $1 - \delta$":

Lemma 7.1.5 (Approximate DP as smoothed[2] DP [19]). *Two random variables Y and Y' are (ε, δ)-indistinguishable if and only if there are events $E = E(Y)$ and $E' = E'(Y')$ such that:*

1. *$\Pr[E], \Pr[E'] \geq 1 - \delta$, and*
2. *$Y|_E$ and $Y'|_{E'}$ are $(\varepsilon, 0)$-indistinguishable.*

Proof: We prove the "if" direction, and omit the converse (which is rather technical). For every set T, we have

$$\begin{aligned}
\Pr[Y \in T] &\leq \Pr[Y \in T|E] \cdot \Pr[E] + \Pr[\overline{E}] \\
&\leq \Pr[Y \in T|E] \cdot (1 - \delta) + \delta \\
&\leq \mathrm{e}^{\varepsilon} \cdot \Pr[Y' \in T|E'] \cdot (1 - \delta) + \delta \\
&\leq \mathrm{e}^{\varepsilon} \cdot \Pr[Y' \in T|E'] \cdot \Pr[E'] + \delta \\
&\leq \mathrm{e}^{\varepsilon} \cdot \Pr[Y' \in T] + \delta
\end{aligned}$$

■

[2] The terminology "smoothed" was coined by [91] for similar variants of entropy measures.

A Bayesian interpretation. Although statistical distance is not a good choice (on its own), there are many other choices of distance measures, and we still have not justified why a multiplicative measure is a particularly good choice. One justification comes from a Bayesian interpretation of the definition of differential privacy [48, 33, 65]. Consider a prior distribution (X, X') on neighboring datasets, modeling an adversary's prior on a real dataset X and a dataset X' that would have been obtained if a particular individual had not participated. Given an output $y \leftarrow \mathcal{M}(X)$, the adversary will have a posterior belief on the dataset, given by the conditional distribution $X|_{\mathcal{M}(X)=y}$. We will argue that differential privacy implies that this posterior is close to the posterior that would have been obtained if the mechanism had been run on X' instead, which we think of as capturing "ideal" privacy for the individual.

Proposition 7.1.6 (DP implies Bayesian privacy). *Let $\mathcal{M} : \mathcal{X}^n \to \mathcal{Y}$ be any ε-differentially private mechanism and let (X, X') be any joint distribution on $\mathcal{X}^n \times \mathcal{X}^n$ such that $\Pr[X \sim X'] = 1$. Then for every dataset $x \in \mathcal{X}^n$ and output $y \in \mathrm{Supp}(\mathcal{M}(X)) = \mathrm{Supp}(\mathcal{M}(X'))$,*[3]

$$\mathrm{SD}(X|_{\mathcal{M}(X)=y}, X|_{\mathcal{M}(X')=y}) \leq 2\varepsilon.$$

A special case of the proposition is when we fix $X' = x'$ to be constant (so that there is nothing to learn from X') and $X = (X_i, x'_{-i})$ is varying only in the data of one individual. Then the proposition says that in such a case (where the adversary knows all but the i-th row of the dataset), the adversary's posterior on X_i is close to its prior. Indeed,

$$\mathrm{SD}(X_i|_{\mathcal{M}(X)=y}, X_i) = \mathrm{SD}(X_i|_{\mathcal{M}(X)=y}, X_i|_{\mathcal{M}(X')=y'}) = \mathrm{SD}(X|_{\mathcal{M}(X)=y}, X|_{\mathcal{M}(X')=y'}) \leq 2\varepsilon.$$

That is, whatever an adversary could have learned about an individual, it could have learned from the rest of the dataset.

Proof: By Bayes' rule,

$$\begin{aligned}
\Pr[X = x|\mathcal{M}(X) = y] &= \frac{\Pr[\mathcal{M}(X) = y|X = x] \cdot \Pr[X = x]}{\Pr[\mathcal{M}(X) = y]} \\
&\leq \frac{e^{\varepsilon} \cdot \Pr[\mathcal{M}(X') = y|X = x] \cdot \Pr[X = x]}{e^{-\varepsilon} \cdot \Pr[\mathcal{M}(X') = y]} \\
&= e^{2\varepsilon} \cdot \Pr[X = x|\mathcal{M}(X') = y].
\end{aligned}$$

By symmetry (swapping X and X'), we also have $\Pr[X = x|\mathcal{M}(X') = y] \leq e^{2\varepsilon} \cdot \Pr[X = x|\mathcal{M}(X) = y]$. Having all probability masses equal up to a multiplicative factor of $e^{2\varepsilon}$ implies that the statistical distance is at most $1 - e^{-2\varepsilon} \leq 2\varepsilon$. ■

There is also a converse to the proposition: if $\mathcal{M}$ guarantees that the two posterior distributions are close to each other (even in statistical difference), then $\mathcal{M}$ must be differentially private. In fact, this will hold even for the special case mentioned above where X' is constant.

Proposition 7.1.7 (Bayesian privacy implies DP). *Let $\mathcal{M} : \mathcal{X}^n \to \mathcal{Y}$ be any randomized mechanism, and let $x_0 \sim x_1 \in \mathcal{X}^n$ be two neighboring datasets. Define the*

[3] $\mathrm{Supp}(Z)$ is defined to be the *support* of random variable Z, i.e., $\{z : \Pr[Z = z] > 0\}$.

joint distribution (X, X') *to equal* (x_0, x_0) *with probability 1/2 and to equal* (x_1, x_0) *with probability 1/2. Suppose that, for some* $y \in \mathrm{Supp}(\mathcal{M}(x_0) \cap \mathrm{Supp}(\mathcal{M}(x_1))$*,*

$$\mathrm{SD}(X|_{\mathcal{M}(X)=y}, X|_{\mathcal{M}(X')=y}) \leq \varepsilon \leq 1/4. \tag{7.2}$$

Then

$$\mathrm{e}^{-O(\varepsilon)} \cdot \Pr[\mathcal{M}(x_1) = y] \leq \Pr[\mathcal{M}(x_0) = y] \leq \mathrm{e}^{O(\varepsilon)} \cdot \Pr[\mathcal{M}(x_1) = y].$$

In particular, if for all pairs $x_0 \sim x_1$ *of neighboring datasets, we have that* $\mathrm{Supp}(\mathcal{M}(x_0)) = \mathrm{Supp}(\mathcal{M}(x_1))$ *and (7.2) holds for all outputs* $y \in \mathrm{Supp}(\mathcal{M}(x_0))$*, then* $\mathcal{M}$ *is* $O(\varepsilon)$*-differentially private.*

Note that, for the joint distributions (X, X') in Proposition 7.1.7, we have $\Pr[X \sim X'] = 1$, so this is indeed a converse to Proposition 7.1.7.

Proof: Since X' is constant, $X|_{\mathcal{M}(X')=y}$ is the same as the prior X (namely, uniformly random from $\{x_0, x_1\}$). Thus, by hypothesis, for $b = 0, 1$, we have

$$\frac{1}{2} - \varepsilon \leq \Pr[X = x_b | \mathcal{M}(X) = y] \leq \frac{1}{2} + \varepsilon.$$

On the other hand, by Bayes' rule,

$$\begin{aligned}
\Pr[\mathcal{M}(x_b) = y] &= \Pr[\mathcal{M}(X) = y | X = x_b] \\
&= \frac{\Pr[X = x_b | \mathcal{M}(X) = y] \cdot \Pr[\mathcal{M}(X) = y]}{\Pr[X = x_b]} \\
&\in \left[\frac{(1/2) - \varepsilon}{1/2} \cdot \Pr[\mathcal{M}(X) = y], \frac{(1/2) + \varepsilon}{1/2} \cdot \Pr[\mathcal{M}(X) = y] \right].
\end{aligned}$$

Thus, $\Pr[\mathcal{M}(x_0) = y] / \Pr[\mathcal{M}(x_1) = y]$ is between $(1/2 - \varepsilon)/(1/2 + \varepsilon) = \mathrm{e}^{-O(\varepsilon)}$ and $(1/2 + \varepsilon)/(1/2 - \varepsilon) = \mathrm{e}^{O(\varepsilon)}$. ■

There are also (ε, δ) analogues of the above propositions, where we require that, with all but negligible probability (related to δ), the posterior probability distributions should be close to each other [65].

Interpretations of the Definition. We can now provide some more intuitive interpretations of (and cautions about) the definition of differential privacy:

- Whatever an adversary learns about you, she could have learned from the rest of the dataset (in particular, even if you did not participate). Note that this does *not* say that the adversary does not learn anything about you; indeed, learning about the population implies learning about individuals. For example, if an adversary learns that smoking correlates with lung cancer (the kind of fact that differential privacy is meant to allow learning) and knows that you smoke, it can deduce that you are more likely to get lung cancer. However, such a deduction is not because of the use of your data in the differentially private mechanism, and thus may not be considered a privacy violation.

- The mechanism will not leak a significant amount of information specific to an individual (or a small group, as we will see in the next section). Consequently, differential privacy is not an achievable privacy notion if the goal of the analysis is to take an action on a specific individual in the dataset (e.g., to identify a candidate for a drug trial, a potential terrorist, or a promising customer).

The above interpretations hold regardless of what auxiliary information or computational strategy the adversary uses. Indeed, the definition provides an information-theoretic form of security. In Section 7.10, we will consider a computational analogue of differential privacy, where we restrict to polynomial-time adversaries.

Variants of the definition and notation. In our treatment, the dataset is an *ordered* n-tuple $x \in \mathcal{X}^n$, where n is known and public (not sensitive information).

A common alternative treatment is to consider datasets x that are *multisets* of elements of $\mathcal{X}$, without a necessarily known or public size. Then, a convenient notation is to represent x as a histogram – that is, as an element of $\mathbb{N}^{\mathcal{X}}$. In the multiset definition, the distance between two datasets is the symmetric difference $|x \Delta x'|$, which corresponds to ℓ_1 distance in histogram notation. Thus, neighboring datasets (at distance 1) are ones that differ by addition or removal of one item. Differential privacy under this definition has a nice interpretation as hiding whether you participated in a dataset at all (without having to replace you by an alternate row to keep the dataset size the same).

There is not a big difference between the two notions, as one can estimate $n = |x|$ with differential privacy (it is just a counting query), the distance between two unordered datasets of the same size under addition/removal versus substitution differ by at most a factor of 2, and one can apply a differentially private mechanism designed for ordered tuples to an unordered dataset by randomly ordering the elements of the dataset.

7.1.7 Preview of the Later Sections

The primary goal of this tutorial is to illustrate connections of differential privacy to computational complexity and cryptography. Consequently, our treatment of the algorithmic foundations of differentially private is very incomplete, and we recommend the monograph of Dwork and Roth [36] for a thorough treatment, including more proofs and examples for the background material that is only sketched here. We also focus heavily on counting queries in this tutorial, because they suffice to bring out most of the connections we wish to illustrate. However, the algorithmic literature on differential privacy now covers a vast range of data-analysis tasks, and obtaining a thorough complexity-theoretic understanding of such tasks is an important direction for future work.

The topics that will be covered in the later sections are as follows:

Section 7.2: We will describe composition theorems that allow us to reason about the level of differential privacy provided when many differentially private algorithms are executed independently. In particular, this will give us algorithms to

answer nearly n^2 counting queries accurately while satisfying differential privacy.

Section 7.3: We will briefly survey some alternatives to using global sensitivity to calibrate the level of noise added for differentially private estimates; sometimes we can get away with adding noise that is proportional to the sensitivity of the query in a local neighborhood of our dataset x (but we need to be careful in doing so).

Section 7.4: We will present some remarkable algorithms that can answer many more than n^2 counting queries with differential privacy. These algorithms are inspired by ideas from computational learning theory, such as Occam's razor and the multiplicative weights method. Unfortunately, these algorithms are computationally quite expensive, requiring time that is polynomial in the size of the data universe $\mathcal{X}$ (which in turn is exponential in the bit-length of row elements).

Section 7.5: We will prove a number of information-theoretic lower bounds on differential privacy, showing that it is impossible to answer too many queries with too much accuracy. Some of the lower bounds will be based on combinatorial and geometric ideas (such as "discrepancy"), and others will be on fingerprinting codes, which were developed as a tool in cryptography (for secure digital content distribution).

Section 7.6: We will turn to computational hardness results for differential privacy, giving evidence that there is no way in general to make the algorithms of Section 7.4 computationally efficient. These hardness results will be based on cryptographic constructs (such as traitor-tracing schemes and digital signatures), and one result will also use probabilistically checkable proofs.

Section 7.7: Next, we will turn to some additional algorithms that bypass the hardness results of Section 7.6 by focusing on specific, structured families of counting queries (and use alternative output representations). The methods employed include low-degree approximations of Boolean functions (via Chebychev polynomials) and convex geometry and optimization (semidefinite programming, Gaussian width, Grothendieck's inequality).

Section 7.8: We will then look at PAC learning with differential privacy, showing both some very general but computationally inefficient positive results, as well as some efficient algorithms. We will then see how methods from communication complexity have been used to show that the sample complexity of differentially private PAC learning (with pure differential privacy) is inherently higher than that of nonprivate PAC learning.

Section 7.9: In this section, we will explore generalizations of differential privacy to the case where the data is distributed among multiple parties, rather than all being held by a single trusted curator. We will show, using connections to randomness extractors and to information complexity, that sometimes distributed differential privacy cannot achieve the same level of accuracy attained in the centralized model.

Section 7.10: The aforementioned limitations of multiparty differential privacy can be avoided by using cryptography (namely, secure multiparty computation) to implement the trusted curator. However, this requires a relaxation of differential

privacy to computationally bounded adversaries. We will present the definition of computational differential privacy, and point out its connection to the notion of "pseudodensity" studied in the theory of pseudorandomness.

7.2 Composition Theorems for Differential Privacy

7.2.1 Postprocessing and Group Privacy

One central property of differential privacy, which we will use throughout the tutorial, is that it is preserved under "postprocessing":

Lemma 7.2.1 (Postprocessing). *If $\mathcal{M} : \mathcal{X}^n \to \mathcal{Y}$ is (ε, δ)-differentially private and $\mathcal{F} : \mathcal{Y} \to \mathcal{Z}$ is any randomized function, then $\mathcal{F} \circ \mathcal{M} : \mathcal{X}^n \to \mathcal{Z}$ is (ε, δ)-differentially private.*

Proof: Consider $\mathcal{F}$ to be a distribution on deterministic functions $f : \mathcal{Y} \to \mathcal{Z}$. Then, for every $x \sim x' \in \mathcal{X}^n$ and every subset $T \subseteq \mathcal{Z}$, we have

$$\begin{aligned}\Pr[(\mathcal{F} \circ \mathcal{M})(x) \in T] &= \mathop{\mathrm{E}}_{f \leftarrow \mathcal{F}}[\Pr[\mathcal{M}(x) \in f^{-1}(T)]] \\ &\leq \mathop{\mathrm{E}}_{f \leftarrow \mathcal{F}}[\mathrm{e}^{\varepsilon} \cdot \Pr[\mathcal{M}(x') \in f^{-1}(T)] + \delta] \\ &= \mathrm{e}^{\varepsilon} \cdot \Pr[(\mathcal{F} \circ \mathcal{M})(x') \in T] + \delta.\end{aligned}$$

■

Another useful property, alluded to in Section 7.1.6, is that differential privacy provides protection for small groups of individuals. For $x, x' \in \mathcal{X}^n$, let $d(x, x')$ denote the Hamming distance between x and x', or in other words the number of rows that need to be changed to go from x to x' (so $x \sim x'$ iff $d(x, x') \leq 1$).

Then the "group privacy" lemma for differential privacy is as follows:

Lemma 7.2.2 (Group privacy). *If $\mathcal{M}$ is an (ε, δ)-differentially private mechanism, then for all pairs of datasets $x, x' \in \mathcal{X}^n$, $\mathcal{M}(x)$ and $\mathcal{M}(x')$ are $(k\varepsilon, k \cdot \mathrm{e}^{k\varepsilon} \cdot \delta)$-indistinguishable for $k = d(x, x')$.*

Proof: We use a hybrid argument. Let $x_0, x_1, x_2, \ldots, x_k$ be such that $x_0 = x$ and $x_k = x'$ and for each i such that $0 \leq i \leq k-1$, x_{i+1} is obtained from x_i by changing one row. Then, for all $T \subseteq \mathcal{Y}$, since $\mathcal{M}$ is (ε, δ)-differentially private,

$$\begin{aligned}\Pr[\mathcal{M}(x_0) \in T] &\leq \mathrm{e}^{\varepsilon} \Pr[\mathcal{M}(x_1) \in T] + \delta \\ &\leq \mathrm{e}^{\varepsilon} (\mathrm{e}^{\varepsilon} \Pr[\mathcal{M}(x_2) \in T] + \delta) + \delta \\ &\quad \vdots \\ &\leq \mathrm{e}^{k\varepsilon} \cdot \Pr[\mathcal{M}(x_k) \in T] + (1 + \mathrm{e}^{\varepsilon} + \mathrm{e}^{2\varepsilon} + \cdots + \mathrm{e}^{(k-1)\cdot\varepsilon}) \cdot \delta \\ &\leq \mathrm{e}^{k\varepsilon} \cdot \Pr[\mathcal{M}(x_k) \in T] + k \cdot \mathrm{e}^{k\varepsilon} \cdot \delta.\end{aligned}$$

■

Note that, when $\delta = 0$, ε-differential privacy provides nontrivial guarantees for datasets x, x' even at distance n, namely $(n\varepsilon, 0)$-indistinguishability, which in particular implies that $\mathcal{M}(x)$ and $\mathcal{M}(x')$ have the same support. In contrast, when $\delta > 0$, we only get nontrivial guarantees for datasets at distance $k \leq \ln(1/\delta)/\varepsilon$; when k is larger, $k \cdot e^{k\varepsilon} \cdot \delta$ is larger than 1. This gap is a source of the additional power of (ε, δ)-differential privacy (as we will see).

7.2.2 Answering Many Queries

Now we consider a different form of composition, where we independently execute several differentially private mechanisms. Let $\mathcal{M}_1, \mathcal{M}_2, \ldots, \mathcal{M}_k$ be differentially private mechanisms. Let

$$\mathcal{M}(x) = (\mathcal{M}_1(x), \mathcal{M}_2(x), \ldots, \mathcal{M}_k(x)),$$

where each $\mathcal{M}_i$ is run with independent coin tosses; for example, this is how we might obtain a mechanism answering a k-tuple of queries.

The basic composition lemma says that the privacy degrades at most linearly with the number of mechanisms executed.

Lemma 7.2.3 (Basic composition). *If $\mathcal{M}_1, \ldots, \mathcal{M}_k$ are each (ε, δ)-differentially private, then $\mathcal{M}$ is $(k\varepsilon, k\delta)$-differentially private.*

However, if we are willing to tolerate an increase in the δ term, the privacy parameter ε only needs to degrade proportionally to $\sqrt{k}$:

Lemma 7.2.4 (Advanced composition [42]). *If $\mathcal{M}_1, \ldots, \mathcal{M}_k$ are each (ε, δ)-differentially private and $k < 1/\varepsilon^2$, then for all $\delta' > 0$, $\mathcal{M}$ is $(O(\sqrt{k \log(1/\delta')}) \cdot \varepsilon, k\delta + \delta')$-differentially private.*

We now prove the above lemmas, starting with basic composition.

Proof of Lemma 7.2.3: We start with the case $\delta = 0$. Fix datasets x, x' such that $x \sim x'$. For an output $y \in \mathcal{Y}$, define the *privacy loss* to be

$$L_{\mathcal{M}}^{x \to x'}(y) = \ln\left(\frac{\Pr[\mathcal{M}(x) = y]}{\Pr[\mathcal{M}(x') = y]}\right) = -L_{\mathcal{M}}^{x' \to x}(y).$$

When $L_{\mathcal{M}}^{x \to x'}(y)$ is positive, the output y is "evidence" that the dataset is x rather than x'; and conversely when it is negative.

Notice that ε^*-differential privacy of $\mathcal{M}$ is equivalent to the statement that, for all $x \sim x'$ and all $y \in \mathrm{Supp}(\mathcal{M}(x)) \cup \mathrm{Supp}(\mathcal{M}(x'))$,

$$|L_{\mathcal{M}}^{x \to x'}(y)| \leq \varepsilon^*.$$

Now, for $\mathcal{M} = (\mathcal{M}_1, \mathcal{M}_2, \ldots, \mathcal{M}_k)$ and $y = (y_1, y_2, \ldots, y_k)$, we have

$$\begin{aligned} L_{\mathcal{M}}^{x\to x'}(y) &= \ln\left(\frac{\Pr[\mathcal{M}_1(x)=y_1 \wedge \mathcal{M}_2(x)=y_2 \wedge \cdots \wedge \mathcal{M}_k(x)=y_k]}{\Pr[\mathcal{M}_1(x')=y_1 \wedge \mathcal{M}_2(x')=y_2 \wedge \cdots \wedge \mathcal{M}_k(x')=y_k]}\right) \\ &= \ln\left(\frac{\prod_{i=1}^k \Pr[\mathcal{M}_i(x)=y_i]}{\prod_{i=1}^k \Pr[\mathcal{M}_i(x')=y_i]}\right) \\ &= \sum_{i=1}^k L_{\mathcal{M}_i}^{x\to x'}(y_i), \end{aligned}$$

so

$$\left|L_{\mathcal{M}}^{x\to x'}(y)\right| \le \sum_{i=1}^k \left|L_{\mathcal{M}_i}^{x\to x'}(y_i)\right| \le k\cdot\varepsilon.$$

For the case that $\delta > 0$, we use Lemma 7.1.5. Specifically, since $\mathcal{M}_i(x_i)$ and $\mathcal{M}_i(x_i')$ are (ε,δ)-indistinguishable, there are events E_i and E_i' of probability at least $1-\delta$ such that, for all y_i, we have

$$\left|\ln\left(\frac{\Pr[\mathcal{M}(x_i)=y_i|E_i]}{\Pr[\mathcal{M}(x_i')=y_i|E_i']}\right)\right| \le \varepsilon.$$

Thus, in the above analysis, we instead condition on the events $E = E_1 \wedge E_2 \wedge \cdots \wedge E_k$ and $E' = E_1' \wedge E_2' \wedge \cdots \wedge E_k'$, redefining our privacy losses as

$$\begin{aligned} L_{\mathcal{M}_i}^{x_i\to x_i'}(y_i) &= \ln\left(\frac{\Pr[\mathcal{M}_i(x_i)=y_i|E_i]}{\Pr[\mathcal{M}(x_i')=y_i|E_i']}\right), \\ L_{\mathcal{M}}^{x\to x'}(y) &= \ln\left(\frac{\Pr[\mathcal{M}(x)=y|E]}{\Pr[\mathcal{M}(x')=y|E']}\right). \end{aligned}$$

Then we still have

$$\left|L_{\mathcal{M}}^{x\to x'}(y)\right| \le \sum_{i=1}^k \left|L_{\mathcal{M}_i}^{x\to x'}(y_i)\right| \le k\cdot\varepsilon.$$

By a union bound, the probability of the events E and E' are at least $1-k\cdot\delta$, so by Lemma 7.1.5, $\mathcal{M}(x)$ and $\mathcal{M}(x')$ are $(k\varepsilon, k\delta)$-indistinguishable, as required. ■

We now move on to advanced composition.

Proof sketch of Lemma 7.2.4: We again focus on the $\delta = 0$ case; the extension to $\delta > 0$ is handled similarly to the proof of Lemma 7.2.3. The intuition for how we can do better than the linear growth in ε is that some of the y_i's will have positive privacy loss (i.e., give evidence for dataset x) while some will have negative privacy loss (i.e., give evidence for dataset x'), and the cancellations between these will lead to a smaller overall privacy loss.

To show this, we consider the *expected* privacy loss

$$\mathop{\mathrm{E}}_{y_i\leftarrow\mathcal{M}_i(x)}[L_{\mathcal{M}_i}^{x\to x'}(y_i)].$$

By definition, this equals the Kullback–Leibler divergence (a.k.a. relative entropy)

$$D(\mathcal{M}_i(x) \,\|\, \mathcal{M}_i(x')),$$

which is known to always be nonnegative.

We first prove the following claim, which shows that the expected privacy loss of a differentially private mechanism is quite a bit smaller than the upper bound on the maximum privacy loss of ε:

Claim 7.2.5. *If $\mathcal{M}_i$ is ε-differentially private, where $\varepsilon \leq 1$, then*

$$\mathop{\mathrm{E}}_{y_i \leftarrow \mathcal{M}_i(x)} [L^{x \to x'}_{\mathcal{M}_i}(y_i)] \leq 2\varepsilon^2.$$

Proof of claim: We will show that

$$D(\mathcal{M}_i(x) \| \mathcal{M}_i(x')) + D(\mathcal{M}_i(x') \| \mathcal{M}_i(x)) \leq 2\varepsilon^2,$$

and then the result follows by the nonnegativity of divergence. Now,

$$\begin{aligned} D(\mathcal{M}_i(x) \| \mathcal{M}_i(x')) + D(\mathcal{M}_i(x') \| \mathcal{M}_i(x)) &= \mathop{\mathrm{E}}_{y_i \leftarrow \mathcal{M}_i(x)} [L^{x \to x'}_{\mathcal{M}_i}(y_i)] + \mathop{\mathrm{E}}_{y_i \leftarrow \mathcal{M}_i(x')} [L^{x' \to x}_{\mathcal{M}_i}(y_i)] \\ &= \mathop{\mathrm{E}}_{y_i \leftarrow \mathcal{M}_i(x)} [L^{x \to x'}_{\mathcal{M}_i}(y_i)] - \mathop{\mathrm{E}}_{y_i \leftarrow \mathcal{M}_i(x')} [L^{x \to x'}_{\mathcal{M}_i}(y_i)], \end{aligned}$$

and using the upper bound of ε on privacy loss we get that

$$\begin{aligned} &\mathop{\mathrm{E}}_{y_i \leftarrow \mathcal{M}_i(x)} [L^{x \to x'}_{\mathcal{M}_i}(y_i)] - \mathop{\mathrm{E}}_{y_i \leftarrow \mathcal{M}_i(x')} [L^{x \to x'}_{\mathcal{M}_i}(y_i)] \\ &\leq 2 \cdot \left(\max_{y_i \in \mathrm{Supp}(\mathcal{M}_i(x)) \cup \mathrm{Supp}(\mathcal{M}_i(x'))} \left| L^{x \to x'}_{\mathcal{M}_i}(y_i) \right| \right) \cdot \mathrm{SD}(\mathcal{M}_i(x), \mathcal{M}_i(x')) \\ &\leq 2\varepsilon \cdot (1 - \mathrm{e}^{-\varepsilon}) \\ &\leq 2\varepsilon^2, \end{aligned}$$

where SD is statistical distance, and we use the fact that $(\varepsilon, 0)$-indistinguishability implies a statistical distance of at most $1 - \mathrm{e}^{-\varepsilon}$. ∎

Thus by linearity of expectation, for the overall expected privacy loss, we have

$$\mathop{\mathrm{E}}_{y \leftarrow \mathcal{M}(x)} [L^{x \to x'}_{\mathcal{M}}(y)] = k \cdot O(\varepsilon^2) \stackrel{\mathrm{def}}{=} \mu.$$

Applying the Hoeffding bound for random variables whose absolute value is bounded by ε, we get that, with probability at least $1 - \delta'$ over $y \leftarrow \mathcal{M}(x)$,

$$L^{x \to x'}_{\mathcal{M}}(y) \leq \mu + O\left(\sqrt{k \log(1/\delta')} \right) \cdot \varepsilon \leq O\left(\sqrt{k \log(1/\delta')} \right) \cdot \varepsilon \stackrel{\mathrm{def}}{=} \varepsilon',$$

where the second inequality uses the assumption that $k < 1/\varepsilon^2$ (so $k\varepsilon^2 \leq \sqrt{k\varepsilon^2}$ and hence $\mu \leq O(\sqrt{k}) \cdot \varepsilon$).

Now for any set T, we have

$$\begin{aligned}\Pr[\mathcal{M}(x) \in T] &\leq \Pr_{y \leftarrow \mathcal{M}(x)}\left[L_{\mathcal{M}}^{x \to x'}(y) > \varepsilon'\right] + \sum_{y \in T : L_{\mathcal{M}}^{x \to x'}(y) \leq \varepsilon'} \Pr[\mathcal{M}(x) = y] \\ &\leq \delta' + \sum_{y \in T : L_{\mathcal{M}}^{x \to x'}(y) \leq \varepsilon'} e^{\varepsilon'} \cdot \Pr[\mathcal{M}(x') = y] \\ &\leq \delta' + e^{\varepsilon'} \cdot \Pr[\mathcal{M}(x') \in T],\end{aligned}$$

so $\mathcal{M}$ is indeed (ε', δ')-differentially private. ∎

It should be noted that, although Lemma 7.2.4 is stated in terms of queries being asked *simultaneously* (in particular, *nonadaptively*), a nearly identical proof (appealing to Azuma's inequality, instead of Hoeffding) shows that an analogous conclusion holds even when the queries (i.e., mechanisms) are chosen *adaptively* (i.e., the choice of $\mathcal{M}_{i+1}$ depends on the outputs of $\mathcal{M}_1(x), \ldots, \mathcal{M}_i(x)$).

Observe that, if we have a set $\mathcal{Q}$ of $k = |\mathcal{Q}|$ counting queries and we wish to obtain a final privacy of (ε, δ'), then we can achieve this by first adding Laplace noise to achieve an initial privacy guarantee of ε_0 for each query and then use the composition theorems. To use the basic composition lemma, we would have to set

$$\varepsilon_0 = \frac{\varepsilon}{k},$$

so the Laplace noise added per query has scale

$$O\left(\frac{1}{\varepsilon_0 n}\right) = O\left(\frac{k}{\varepsilon n}\right).$$

To obtain a bound on the *maximum* noise added to any of the queries, we can do a union bound over the k queries. Setting $\beta = 1/O(k)$ in Theorem 7.1.3, with high probability, the maximum noise will be at most

$$\alpha = O\left(\frac{k \cdot \log k}{\varepsilon n}\right).$$

Steinke and Ullman [99] showed how to save the $\log k$ factor by carefully correlating the noise used for the k queries, and thus showed:

Theorem 7.2.6 (Arbitrary counting queries with pure differential privacy [99]). *For every set $\mathcal{Q}$ of counting queries and $\varepsilon > 0$, there is an ε-differentially private mechanism $\mathcal{M} : \mathcal{X}^n \to \mathbb{R}^{\mathcal{Q}}$ such that, on every dataset $x \in \mathcal{X}^n$, with high probability $\mathcal{M}(x)$ answers all the queries in $\mathcal{Q}$ to within additive error*

$$\alpha = O\left(\frac{|\mathcal{Q}|}{\varepsilon n}\right).$$

Thus, taking ε to be constant, we can answer any $|\mathcal{Q}| = o(n)$ counting queries with vanishingly small error, which we will see is optimal for pure differential privacy (in Section 7.5.2).

Similarly, to use the advanced composition theorem, we would have to set

$$\varepsilon_0 = \frac{\varepsilon}{c \cdot \sqrt{k \cdot \log(1/\delta)}},$$

yielding a maximum error of

$$\alpha = O\left(\frac{\log k}{\varepsilon_0 n}\right) = O\left(\frac{\sqrt{k \cdot \log(1/\delta)} \cdot \log k}{\varepsilon n}\right).$$

Again, it is known how to (mostly) remove the $\log k$ factor:

Theorem 7.2.7 (Arbitrary counting queries with approximate differential privacy [99]). *For every set $\mathcal{Q}$ of counting queries over data universe $\mathcal{X}$, and $\varepsilon, \delta > 0$, there is an (ε, δ)-differentially private mechanism $\mathcal{M} : \mathcal{X}^n \to \mathbb{R}^k$ such that, on every dataset $x \in \mathcal{X}^n$, with high probability $\mathcal{M}(x)$ answers all the queries to within error*

$$\alpha = O\left(\frac{\sqrt{|\mathcal{Q}| \cdot \log(1/\delta) \cdot \log\log|\mathcal{Q}|}}{\varepsilon n}\right).$$

Again taking ε to be constant and δ to be negligible (e.g., $\delta = 2^{-\log^2(n)}$), we can take $k = |\mathcal{Q}| = \tilde{\Omega}(n)$ and obtain error $o(1/\sqrt{n})$ (smaller than the sampling error!), which we will see is essentially optimal for any reasonable notion of privacy (in Section 7.5.1). If we want error $o(1)$, we can take $k = \tilde{\Omega}(n^2)$, which is known to be optimal for differential privacy if the answers are not coordinated based on the queries [43] or if the data universe is large (as we will see in Section 7.5). However, in Section 7.4, we will see some beautiful algorithms that can answer many more than n^2 queries if the data universe is not too large (forcing the queries to have some implicit relationships) by carefully coordinating the noise between the queries.

Optimal composition. Remarkably, Kairouz, Oh, and Viswanath [64] have given an *optimal* composition theorem for differential privacy, which provides an exact characterization of the best privacy parameters that can be guaranteed when composing a number of (ε, δ)-differentially private mechanisms. The key to the proof is showing that an (ε, δ) generalization of randomized response (as defined in Section 7.1.5) is the worst mechanism for composition. Unfortunately, the resulting optimal composition bound is quite complex, and indeed is even #P-complete to compute exactly when composing mechanisms with different $(\varepsilon_i, \delta_i)$ parameters [82]. Thus, for theoretical purposes, it is still most convenient to use Lemmas 7.2.3 and 7.2.4, which give the right asymptotic behavior for most settings of parameters that tend to arise in theoretical applications.

7.2.3 Histograms

The bounds of Theorems 7.2.6 and 7.2.7 are for arbitrary, worst-case families of counting queries. For specific families of counting queries, one may be able to do much better. A trivial example is when the same query is asked many times; then we can compute just one noisy answer, adding noise $\mathrm{Lap}(1/\varepsilon)$, and give the same answer for all the queries. A more interesting example is the family $\mathcal{Q}^{\mathrm{pt}}$ of

point functions on a data universe $\mathcal{X}$, as defined in Section 7.1.3. Answering all $|\mathcal{X}|$ queries in $\mathcal{Q}^{\text{pt}}$ (i.e., estimating the histogram of the dataset) using the above theorems would incur error at least $\sqrt{|\mathcal{X}|}/\varepsilon n$. However, it turns out that we can achieve error $O(\log|\mathcal{X}|)/\varepsilon n$.

Proposition 7.2.8 (Laplace histograms). *For every finite data universe $\mathcal{X}$, $n \in \mathbb{N}$, and $\varepsilon > 0$, there is an ε-differentially private mechanism $\mathcal{M} : \mathcal{X}^n \to \mathbb{R}^{\mathcal{X}}$ such that, on every dataset $x \in \mathcal{X}^n$, with high probability $\mathcal{M}(x)$ answers all of the counting queries in $\mathcal{Q}^{pt}(\mathcal{X})$ to within error*

$$O\left(\frac{\log|\mathcal{X}|}{\varepsilon n}\right).$$

Proof sketch: Recall that $\mathcal{Q}^{\text{pt}}(\mathcal{X})$ contains a query q_y for each $y \in \mathcal{X}$, where on a row $w \in \mathcal{X}$, $q_y(w)$ is 1 iff $w = y$. The mechanism $\mathcal{M}$ adds independent noise distributed according to $\text{Lap}(2/\varepsilon n)$ to the result of each query $q_y \in \mathcal{Q}^{\text{pt}}$. This ensures that each individual noisy answer is $\varepsilon/2$-differentially private. To show that we obtain ε-differential privacy overall, the key observation is that, for two neighboring datasets x, x', there are only two queries $q_y, q_{y'} \in \mathcal{Q}^{\text{pt}}$ on which x and x' differ (corresponding to the values that x and x' have in the row where they differ). Thus, the proof of basic composition lemma (Lemma 7.2.3) implies that $\mathcal{M}(x)$ and $\mathcal{M}(x')$ are $(2 \cdot (\varepsilon/2), 0)$-indistinguishable, as desired. ∎

We can also use the output of this mechanism to answer an arbitrary counting query $q : \mathcal{X} \to \{0,1\}$, noting that $q(x) = \sum_{y \in \mathcal{X}} q_y(x) \cdot q(y)$. The above mechanism gives us $a_y = q_y(x) + \text{Lap}(2/\varepsilon n)$ for every $y \in \mathcal{X}$, from which we can compute the quantity $a = \sum_{y \in \mathcal{X}} a_y \cdot q(y)$, which has expectation $q(x)$ and standard deviation $O(\sqrt{|\mathcal{X}|}/\varepsilon n)$. For answering multiple queries, we can apply Chernoff/Hoeffding and union bounds,[4] yielding the following:

Theorem 7.2.9 (Arbitrary counting queries via the Laplace histogram). *For every set $\mathcal{Q}$ of counting queries on data universe $\mathcal{X}$, $n \in \mathbb{N}$, and $\varepsilon > 0$, there is an ε-differentially private mechanism $\mathcal{M} : \mathcal{X}^n \to \mathbb{R}^{\mathcal{Q}}$ such that on every dataset $x \in \mathcal{X}^n$, with high probability $\mathcal{M}(x)$ answers all the queries to within error*

$$O\left(\frac{\sqrt{|\mathcal{X}| \cdot \log|\mathcal{Q}|}}{\varepsilon n}\right).$$

Note that the dependence on $k = |\mathcal{Q}|$ has improved from $\sqrt{k}$ obtained by advanced composition or Theorem 7.2.7 to $\sqrt{\log k}$, at the price of introducing a (rather large) dependence on $|\mathcal{X}|$. Thus, for a family $\mathcal{Q}$ of counting queries on data universe $\mathcal{X}$, it is

[4] A bit of care is needed since the $\text{Lap}(2/\varepsilon n)$ noise random variables are not bounded. This can be handled by first arguing that, with high probability, at most a $2^{-\Theta(t)}$ fraction of the noise random variables have magnitude in the range $[t/\varepsilon n, 2t/\varepsilon n)$. Then, conditioned on the magnitudes of the noise random variables (but not their signs), we can group the random variables according to their magnitudes (up to a factor of 2) and apply Hoeffding to each group separately.

better to use the Laplace histogram when $|\mathcal{X}| \ll |\mathcal{Q}|$ and it is better to use advanced composition or Theorem 7.2.7 when $|\mathcal{X}| > |\mathcal{Q}|$.

Let us summarize the best error bounds we have seen so far for the example families of counting queries given in Section 7.1.3.

Table 7.1: Error bounds for specific query families on a data universe $\mathcal{X}$ of size $D = 2^d$ (e.g., $\mathcal{X} = \{0,1\}^d$ or $\mathcal{X} = \{1, 2, \ldots, D\}$).

Query family $\mathcal{Q}$	$\lvert\mathcal{Q}\rvert$	$(\varepsilon, 0)$-dp	Ref.	(ε, δ)-dp	Ref.
$\mathcal{Q}^{\text{pt}}$	D	$O\left(\frac{d}{\varepsilon n}\right)$	Prop. 7.2.8	$O\left(\frac{d}{\varepsilon n}\right)$	Prop. 7.2.8
$\mathcal{Q}^{\text{thr}}$	D	$\frac{\tilde{O}(\sqrt{D})}{\varepsilon n}$	Thm. 7.2.9	$\frac{\tilde{O}(\sqrt{D})}{\varepsilon n}$	Thm. 7.2.9
$\mathcal{Q}^{\text{conj}}$	3^d	$\frac{\tilde{O}(\sqrt{D})}{\varepsilon n}$	Thm. 7.2.9	$\frac{\tilde{O}(\sqrt{D})}{\varepsilon n}$	Thm. 7.2.9
$\mathcal{Q}^{\text{means}}$	d	$O\left(\frac{d}{\varepsilon n}\right)$	Thm. 7.2.6	$O\left(\frac{\sqrt{d\log(1/\delta)\cdot\log\log d}}{\varepsilon n}\right)$	Thm. 7.2.7
$\mathcal{Q}_t^{\text{conj}}$ for $t \ll d$	$O(d^t)$	$O\left(\frac{d^t}{\varepsilon n}\right)$	Thm. 7.2.6	$O\left(\frac{d^{t/2}\cdot\sqrt{\log(1/\delta)\cdot\log\log d}}{\varepsilon n}\right)$	Thm. 7.2.7

We will see substantial improvements to most of these bounds in later sections.

7.3 Alternatives to Global Sensitivity

In this section, we consider the question of whether we can do better than adding noise $\text{Lap}(\text{GS}_q/\varepsilon)$, where GS_q denotes the global sensitivity of query q (cf. Theorem 7.1.3).

As a first attempt, let us define a notion of "local sensitivity" at x:

$$\text{LS}_q(x) = \max\{q(x) - q(x')| : x' \sim x\}.$$

The difference from global sensitivity is that we only take the maximum over datasets x' that are neighbors to our input dataset x, rather than taking the maximum over all neighboring pairs $x'\tilde{x}''$.

Naively, we might hope that $\mathcal{M}(x) = q(x) + \mathit{Noise}(O(\text{LS}_q(x)))$ might provide differential privacy. Indeed, the local sensitivity provides a lower bound on the error we need to introduce:

Proposition 7.3.1 (Local sensitivity lower bound). *Let $q : \mathcal{X}^n \to \mathbb{R}$ be a real-valued query and $\mathcal{M} : \mathcal{X}^n \to \mathcal{Y}$ be an (ε, δ)-differentially private mechanism. Then*

1. *For every $x_0 \sim x_1 \in \mathcal{X}^n$, there is a $b \in \{0,1\}$ such that*

$$\Pr\left[|\mathcal{M}(x_b) - q(x_b)| < \frac{|q(x_0) - q(x_1)|}{2}\right] \leq \frac{1+\delta}{1+\mathrm{e}^{-\varepsilon}} = \frac{1}{2} + O(\delta + \varepsilon).$$

2. *For every $x \in \mathcal{X}^n$, there is some x' at Hamming distance at most 1 from x such that*
$$\Pr\left[|\mathcal{M}(x') - q(x')| < \frac{\mathrm{LS}_q(x)}{2}\right] \leq \frac{1+\delta}{1+\mathrm{e}^{-\varepsilon}} = \frac{1}{2} + O(\delta + \varepsilon).$$

Proof:
1. Let $\mathcal{G}_b = \left\{y \in \mathbb{R} : |y - q(x_b)| < \frac{|q(x_0)-q(x_1)|}{2}\right\}$ and $p = \min\{\Pr[\mathcal{M}(x_0) \in \mathcal{G}_0], \Pr[\mathcal{M}(x_1) \in \mathcal{G}_1]\}$. Then:

$$\begin{aligned} 1 - p &\geq \Pr[\mathcal{M}(x_0) \notin \mathcal{G}_0] \\ &\geq \Pr[\mathcal{M}(x_0) \in \mathcal{G}_1] \\ &\geq \mathrm{e}^{-\varepsilon} \cdot \Pr[\mathcal{M}(x_1) \in \mathcal{G}_1] - \delta \\ &\geq \mathrm{e}^{-\varepsilon} \cdot p - \delta. \end{aligned}$$

Solving, we deduce that $p \leq (1+\delta)/(1+\mathrm{e}^{-\varepsilon})$.

2. Follows from part 1 by taking $x_0 = x$ and $x_1 \sim x$ such that $\mathrm{LS}_q(x) = |q(x) - q(x_1)|$. ∎

The problem with trying to use the local sensitivity to calibrate the noise is that we do not want the amount of noise to itself distinguish between neighboring x and x'. For instance, let x be such that $q(x) = q(x') = 0$ for all $x' \sim x$, but where there is one such neighbor $x' \sim x$ where x' has a neighbor x'' such that $q(x'') = 10^9$. $\mathrm{LS}_q(x) = 0$, but $\mathrm{LS}_q(x')$ is large, and answering queries noisily based on LS_q would violate privacy because it distinguishes between x and x'.

Still, perhaps one could hope to provide only a small amount of noise if LS_q is small everywhere "near" x. For example, consider the query that asks for the *median* of n points $\{x_1, x_2, \ldots x_n\} \subseteq [0,1]$. The global sensitivity for this query is high. Indeed, consider the instance x where $(n+1)/2$ entries are 1 and $(n-1)/2$ entries are 0 (and thus the median is 1), as compared with the neighboring instance x' where one entry is changed from 1 to 0 (and thus the median is 0).

On the other hand, if there are *many* data points near the median, then it would follow that the local sensitivity is small, not only at x but also at all datasets close to x. For such instances x, we could indeed get away with adding only a small amount of noise, while maintaining privacy. This is the type of situation that we will investigate. There are several related approaches that have been taken along these lines, which we will discuss:

1. Smooth sensitivity [86]
2. Propose–test–release [34]
3. Releasing stable values [96]
4. Privately bounding local sensitivity [68]

We remark that yet another approach, called *restricted sensitivity*, aims to add even *less* noise than the local sensitivity [12, 68, 27, 89]. The observation is that Proposition 7.3.1 does *not* say that the error on x must be at least $\mathrm{LS}_q(x)/2$; rather it says that the error must be at least $\mathrm{LS}_q(x)/2$ on x *or one of its neighbors*. Thus if we have a hypothesis that our dataset belongs to some set $H \subseteq \mathcal{X}^n$ (e.g. in the case of a social

network, we might believe that the graph is of bounded degree), it might suffice to add noise proportional to the *restricted sensitivity*, where we maximize $|q(x) - q(x')|$ over $x \sim x' \in H$, which can be much smaller than even the local sensitivity. The noise will still need to be at least $\mathrm{LS}_q(x)/2$ on some neighbors x' of x, but these can be neighbors outside of H.

7.3.1 Smooth Sensitivity

Define *smooth sensitivity* of query $q : \mathcal{X}^n \to \mathbb{R}$ at x as follows:

$$\mathrm{SS}_q^{\varepsilon}(x) = \max\{\mathrm{LS}_q(x') \cdot \mathrm{e}^{-\varepsilon d(x,x')} : x' \in \mathcal{X}^n\},$$

where $d(x, x')$ denotes Hamming distance. Intuitively, we are smoothing out the local sensitivity, so that it does not change much between neighboring datasets.

Nissim, Raskhodnikova, and Smith [86] introduced the notion of smooth sensitivity and showed that:

- Adding noise $O(\mathrm{SS}_q^{\varepsilon}(x)/\varepsilon)$ (according to a Cauchy distribution) is sufficient for ε-differential privacy.
- SS_q can be computed efficiently when q is the median query (despite the fact that it is defined as the maximum over a set of size $|\mathcal{X}|^n$), as well as for a variety of graph statistics (under edge-level differential privacy, cf. Section 7.3.4).

Zhang et al. [111] gave an alternative approach to "smoothing out" local sensitivity, which empirically provides improvements in accuracy.

7.3.2 Propose–Test–Release

A different way to provide less noise is to simply not allow certain queries. That is: rather than using Laplace noise at a level that is high enough no matter what possible dataset might be queried, an alternative is to initially *propose* an amount of noise that seems tolerable, and then *test* whether answering a query with this amount of noise would violate privacy (namely, if the noise magnitude is less than the local sensitivity in a neighborhood of the current dataset). If the test passes, then we *release* a noisy answer. But perhaps we detect that adding this (small) amount of noise *would* violate privacy. In that case, we simply refuse to answer. Of course, we should carry out the test in a differentially private manner.

More precisely, propose–test–release consists of the following three steps (parameterized by a query $q : \mathcal{X}^n \to \mathbb{R}$ and $\varepsilon, \delta, \beta \geq 0$), yielding a mechanism $\mathcal{M} : \mathcal{X}^n \to \mathbb{R} \cup \{\perp\}$ that does the following on a dataset $x \in \mathcal{X}^n$:

1. Propose a target bound β on local sensitivity.
2. Let $\hat{d} = d(x, \{x' : \mathrm{LS}_q(x') > \beta\}) + \mathrm{Lap}(1/\varepsilon)$, where d denotes Hamming distance.
3. If $\hat{d} \leq \ln(1/\delta)/\varepsilon$, output $\perp$.
4. If $\hat{d} > \ln(1/\delta)/\varepsilon$, output $q(x) + \mathrm{Lap}(\beta/\varepsilon)$.

Proposition 7.3.2 (Propose–test–release [34]). *For every query $q : \mathcal{X}^n \to \mathbb{R}$ and $\varepsilon, \delta, \beta \geq 0$, the above algorithm is $(2\varepsilon, \delta)$-differentially private.*

Proof: Consider any two neighboring datasets $x \sim x'$. Because of the Laplacian noise in the definition of $\hat{d}$ and the fact that Hamming distance has global sensitivity at most 1, it follows that

$$\Pr[\mathcal{M}(x) = \perp] \in [e^{-\varepsilon} \cdot \Pr[\mathcal{M}(x') = \perp], e^{\varepsilon} \cdot \Pr[\mathcal{M}(x') = \perp]]. \tag{7.3}$$

Also, for those outputs that are not $\perp$, we have two cases:

Case 1: $\mathrm{LS}_q(x) > \beta$. In this case, $d(x, \{x'' : \mathrm{LS}_q(x'') > \beta\}) = 0$, so the probability that $\hat{d}$ will exceed $\ln(1/\delta)/\varepsilon$ is at most δ. Thus, for every set $T \subseteq \mathbb{R} \cup \{\perp\}$, we have

$$\begin{aligned}\Pr[\mathcal{M}(x) \in T] &\leq \Pr[\mathcal{M}(x) \in T \cap \{\perp\}] + \Pr[\mathcal{M}(x) \neq \perp] \\ &\leq e^{\varepsilon} \cdot \Pr[\mathcal{M}(x') \in T \cap \{\perp\}] + \delta \\ &\leq e^{\varepsilon} \cdot \Pr[\mathcal{M}(x') \in T] + \delta,\end{aligned}$$

where the second inequality follows from (7.3), noting that $T \cap \{\perp\}$ equals either $\{\perp\}$ or $\emptyset$.

Case 2: $\mathrm{LS}_q(x) \leq \beta$. In this case, $|q(x) - q(x')| \leq \beta$, which in turn implies the $(\varepsilon, 0)$-indistinguishability of $q(x)+\mathrm{Lap}(\beta/\varepsilon)$ and $q(x')+\mathrm{Lap}(\beta/\varepsilon)$. Thus, by (7.3) and basic composition, we have $(2\varepsilon, 0)$-indistinguishability overall. ■

Notice that, like smooth sensitivity, the naive algorithm for computing $d(x, \{x' : \mathrm{LS}_q(x') > \beta\})$ enumerates over all datasets $x' \in \mathcal{X}^n$. Nevertheless, for the median function, it can again be computed efficiently.

7.3.3 Releasing Stable Values

A special case of interest in propose–test–release is when $\beta = 0$. Then it can be verified that $d(x, \{x' : \mathrm{LS}_q(x') > \beta\}) = d(x, \{x' : q(x') \neq q(x)\}) - 1$, so the algorithm is testing whether the function q is constant in a neighborhood of x (of radius roughly $\ln(1/\delta)/\varepsilon$), and if so, it outputs q with no noise; that is, if q is stable around x, then we can safely release the value $q(x)$ (*exactly*, with no noise!), provided our test of stability is differentially private. This also applies to, and indeed makes the most sense for, discrete-valued functions $q : \mathcal{X}^n \to \mathcal{Y}$. In more detail, the mechanism works as follows on $x \in \mathcal{X}^n$:

1. Let $\hat{d} = d(x, \{x' : q(x') \neq q(x)\})+\mathrm{Lap}(1/\varepsilon)$, where d denotes Hamming distance.
2. If $\hat{d} \leq 1 + \ln(1/\delta)/\varepsilon$, output $\perp$.
3. Otherwise output $q(x)$.

Similarly to Proposition 7.3.2, we have:

Proposition 7.3.3 (Releasing stable values). *For every query $q : \mathcal{X}^n \to \mathcal{Y}$ and $\varepsilon, \delta > 0$, the above algorithm is (ε, δ)-differentially private.*

Consider, for example, the *mode* function $q : \mathcal{X}^n \to \mathcal{X}$, where $q(x)$ is defined to be the most frequently occurring data item in x (breaking ties arbitrarily). Then $d(x, \{x' : q(x') \neq q(x)\})$ equals half of the gap in the number of occurrences between the mode and the second most frequently occurring item (rounded up). So we have:

Proposition 7.3.4 (Stability-based mode). *For every data universe $\mathcal{X}$, $n \in \mathbb{N}$, and $\varepsilon, \delta \geq 0$, there is an (ε, δ)-differentially private algorithm $\mathcal{M} : \mathcal{X}^n \to \mathcal{X}$ such that, for every dataset $x \in \mathcal{X}^n$ where the difference between the number of occurrences of the mode and the second most frequently occurring item is larger than $4\lceil \ln(1/\delta)/\varepsilon \rceil$, $\mathcal{M}(x)$ outputs the mode of x with probability at least $1 - \delta$.*

If instead we had used the Laplace Histogram of Proposition 7.2.8 (outputting the bin $y \in \mathcal{X}$ with the largest noisy count), we would require a gap of $\Theta(\log |\mathcal{X}|)/\varepsilon$ in the worst case, so the stability-based method is better when $|\mathcal{X}|$ is large compared with $1/\delta$. Indeed, let us now show how stability-based ideas can in fact produce noisy histograms with an error bound of $O(\log(1/\delta))/\varepsilon n$.

Theorem 7.3.5 (Stability-based histograms [24]). *For every finite data universe $\mathcal{X}$, $n \in \mathbb{N}$, $\varepsilon \in (0, \ln n)$, and $\delta \in (0, 1/n)$, there is an (ε, δ)-differentially private mechanism $\mathcal{M} : \mathcal{X}^n \to \mathbb{R}^{\mathcal{X}}$ such that, on every dataset $x \in \mathcal{X}^n$, with high probability $\mathcal{M}(x)$ answers all of the counting queries in $\mathcal{Q}^{pt}(\mathcal{X})$ to within error*

$$O\left(\frac{\log(1/\delta)}{\varepsilon n}\right).$$

The intuition for the algorithm is that, if we only released noisy answers for point functions q_y that are nonzero on the dataset x, the error bound in Proposition 7.2.8 would improve from $O(\log |\mathcal{X}|)/\varepsilon n$ to $O(\log n)/\varepsilon n \leq O(\log(1/\delta))/\varepsilon n$, since at most n point functions can be nonzero on any dataset (namely those corresponding to the rows of the dataset). However, revealing which point functions are nonzero would not be differentially private. Thus, we only release the point functions that are *far* from being zero (i.e., ones where the query is nonzero on all datasets at noisy distance at most $O(\log(1/\delta)/\varepsilon)$ from the given dataset, analogously to Proposition 7.3.3).

Proof: The algorithm is the same as the Laplace histogram of Proposition 7.2.8, except that we do not add noise to counts that are zero, and reduce all noisy counts that are smaller than $O(\log(1/\delta)/\varepsilon n$ to zero.

Specifically, given a dataset $x \in \mathcal{X}^n$, the algorithm works as follows:

1. For every point $y \in \mathcal{X}$:
   a. If $q_y(x) = 0$, then set $a_y = 0$.
   b. If $q_y(x) > 0$, then:
      i. Set $a_y \leftarrow q_y(x) + \mathrm{Lap}(2/\varepsilon n)$.
      ii. If $a_y < 2\ln(2/\delta)/\varepsilon n + 1/n$, then set $a_y \leftarrow 0$.
2. Output $(a_y)_{y \in \mathcal{X}}$.

Now let us analyze this algorithm.

Utility: The algorithm gives exact answers for queries q_y where $q_y(x) = 0$. There are at most n queries q_y with $q_y(x) > 0$ (namely, ones where $y \in \{x_1, \ldots, x_n\}$). By the tails of the Laplace distribution and a union bound, with high probability, all of the noisy answers $q_y(x) + \mathrm{Lap}(2/\varepsilon n)$ computed in step 1(b)i have error at most

$O((\log n)/\varepsilon n) \leq O(\log(1/\delta)/\varepsilon n)$. Truncating the small values to zero in step 1(b)ii introduces an additional error of up to $2\ln(1/\delta)/\varepsilon n + 1/n = O(\log(1/\delta)/\varepsilon n)$.

Privacy: Consider two neighboring datasets $x \sim x'$, where dataset x' is obtained by replacing row x_i with x'_i. Then the only point queries that differ on x and x' are q_{x_i} and $q_{x'_i}$. Since the answers to different queries q_y are independent, we can analyze the answer to each query separately and then apply composition. Consider the answers $a_{x_i}(x)$ and $a_{x_i}(x')$ to query q_{x_i} on datasets x and x', respectively. We know that $q_{x_i}(x) > 0$ (since row x_i is in x). If we also have $q_{x_i}(x') > 0$, then $a_{x_i}(x)$ and $a_{x_i}(x')$ are $(\varepsilon/2, 0)$-indistinguishable by the differential privacy of the Laplace mechanism. (We can view the truncation step as postprocessing.) If $q_{x_i}(x') = 0$, then $a_{x_i}(x')$ is always 0, and $q_{x_i}(x) = 1/n$ (since x and x' agree on all other rows), which means that $\Pr[a_{x_i}(x) \neq 0] = \Pr[\mathrm{Lap}(2/\varepsilon n) \geq 2\ln(2/\delta)/\varepsilon n] \leq \delta/2$ and we have $(0, \delta/2)$-indistinguishability. Thus, in all cases, $a_{x_i}(x)$ and $a_{x_i}(x')$ are $(\varepsilon/2, \delta/2)$-indistinguishable. By symmetry the same holds for the answers $a_{x'_i}(x)$ and $a_{x'_i}(x')$. On all other queries y, $a_y(x)$ and $a_y(x')$ are identically distributed. By basic composition, the joint distributions of all answers are (ε, δ)-indistinguishable. ■

7.3.4 Privately Bounding Local Sensitivity

Rather than *proposing* (arbitrarily) a threshold β as in propose–test–release, more generally we might try to *compute* a differentially private upper bound on the local sensitivity. That is, we will try to compute a differentially private estimate $\hat{\beta} = \hat{\beta}(x)$ such that, with probability at least $1-\delta$, $\mathrm{LS}_q(x) \leq \hat{\beta}$. If we can do this, then outputting $q(x) + \mathrm{Lap}(\hat{\beta}/\varepsilon)$ will give an (ε, δ)-differentially private algorithm, by an analysis as in the previous section.

The setting in which we will explore this possibility is where our dataset is a graph and we want to estimate the number of triangles in the graph.

There are (at least) two notions of privacy that one might wish to consider for graph algorithms:

- *Edge-level privacy*. In this setting, we say that $G \sim G'$ if the graphs G and G' differ on one edge. This is a special case of the setting we have been studying, where we think of an n-vertex graph as a dataset consisting of $\binom{n}{2}$ rows from universe $\mathcal{X} = \{0, 1\}$.
- *Node-level privacy*. In this setting, we say that $G \sim G'$ if the graphs G and G' differ only on edges that are adjacent to one vertex. This does not quite fit in the tuple-dataset setting we have been studying, but the concept of differential privacy naturally generalizes to this (as well as any other family of "datasets" with some notion of "neighbors").

In applications (e.g., to social networks), node-level privacy is a preferable notion of privacy, since it simultaneously protects all of the relationships associated with a vertex (which typically represents an individual person), rather than just a single relationship at a time. However, since our goal is only to illustrate the method of privately bounding local sensitivity, we will consider only edge-level privacy. Let

$q_\Delta(G)$ be the number of triangles in G (where the Δ is meant to be evocative of a triangle). It can be verified that

$$\mathrm{LS}_{q_\Delta}(G) = \max\{j : \exists u \exists v \; u \text{ and } v \text{ have } j \text{ common neighbors}\}.$$

This, in turn, is no more than the maximum degree of G. In contrast the global sensitivity is $\mathrm{GS}_{q_\Delta} = n - 2$. However, if we consider the global sensitivity of the local sensitivity, we have $\mathrm{GS}_{\mathrm{LS}_{q_\Delta}} = 1$. (If we think of the local sensitivity as a discrete analogue of a derivative, then this is the analogue of having a bounded second derivative, despite the derivative sometimes being large.)

Consider the following mechanism $\mathcal{M}(G)$:

- Compute $\hat{\beta} = \mathrm{LS}_{q_\Delta}(G) + \mathrm{Lap}(1/\varepsilon) + \ln(1/\delta)/\varepsilon$.
- Output $q_\Delta(G) + \mathrm{Lap}(\hat{\beta}/\varepsilon)$.

This mechanism can be shown to be $(2\varepsilon, \delta)$-differentially private, and the total noise is of magnitude

$$O\left(\frac{\mathrm{LS}_{q_\Delta}(G) + (1 + \log(1/\delta))/\varepsilon}{\varepsilon}\right).$$

Note that this approach is computationally efficient if we can efficiently evaluate the query q, can efficiently calculate LS_q (which can be done using $m \cdot (|\mathcal{X}| - 1)$ evaluations of q when the dataset is in $\mathcal{X}^m$), and have an upper bound on $\mathrm{GS}_{\mathrm{LS}_q}$.

7.4 Releasing Many Counting Queries with Correlated Noise

We have seen (in Theorems 7.2.6, 7.2.7, and 7.2.9) that any set $\mathcal{Q}$ of counting queries over data universe $\mathcal{X}$ can be answered with differential privacy and an error of at most

$$\alpha \leq O\left(\min\left\{\frac{|\mathcal{Q}|}{\varepsilon n}, \frac{\sqrt{|\mathcal{Q}| \cdot \log(1/\delta) \cdot \log\log|\mathcal{Q}|}}{\varepsilon n}, \frac{\sqrt{|\mathcal{X}| \cdot \log|\mathcal{Q}|}}{\varepsilon n}\right\}\right)$$

on each of the queries (with high probability). When both $|\mathcal{Q}|$ and $|\mathcal{X}|$ are larger than n^2, the amount of error is larger than 1, and hence these approaches provide nothing useful (recall that the true answers lie in $[0, 1]$).

In this section, we will see two methods that can answer many more than n^2 counting queries on a data universe of size much larger than n^2. Both use ideas from learning theory.

7.4.1 The SmallDB Algorithm

Theorem 7.4.1 (The smallDB algorithm, Blum et al. [14]). *For every set $\mathcal{Q}$ of counting queries on a data universe $\mathcal{X}$ and every $\varepsilon > 0$, there exists an ε-*

differentially private mechanism $\mathcal{M}$ such that, for all datasets $x \in \mathcal{X}^n$, with high probability $\mathcal{M}(x)$ answers all queries in $\mathcal{Q}$ to within error at most

$$\alpha = O\left(\frac{\log|\mathcal{Q}| \log|\mathcal{X}|}{\varepsilon n}\right)^{1/3}.$$

Moreover, $M(x)$ outputs a "synthetic dataset" $y \in \mathcal{X}^m$ with $m = O(\log|\mathcal{Q}|/\alpha^2)$ such that, with high probability, we have $|q(y) - q(x)| \leq \alpha$ for all $q \in \mathcal{Q}$, i.e., we can calculate all the answers using the (smaller) synthetic dataset.

In fact, the bounds can be improved to $\alpha = \tilde{O}(\mathrm{VC}(\mathcal{Q}) \cdot \log|\mathcal{X}|/\varepsilon n)^{1/3}$ and $m = \mathrm{VC}(\mathcal{Q}) \cdot \tilde{O}(1/\alpha^2)$, where $\mathrm{VC}(\mathcal{Q})$ is the Vapnik–Chervonenkis dimension of the class $\mathcal{Q}$.[5]

The key point is that the error grows (less than) logarithmically with the number $|\mathcal{Q}|$ of queries and the size $|\mathcal{X}|$ of the data universe; this allows us to handle even exponentially many queries. (On the other hand, the error vanishes more slowly with n than the earlier results we have seen — like $1/n^{1/3}$ rather than $1/n$.) Let us compare the implications of the smallDB algorithm for concrete query families with the bounds we saw in Section 7.2 for pure differential privacy (Table 7.1):

Table 7.2: Error bounds for specific query families under $(\varepsilon, 0)$-differential privacy on a data universe $\mathcal{X}$ of size $D = 2^d$ (e.g. $\mathcal{X} = \{0,1\}^d$ or $\mathcal{X} = \{1, 2, \ldots, D\}$). Highlighted cells indicate the best bounds in the regime where $n \leq D^{o(1)}$ or $n \leq d^{o(t)}$.

Query family $\mathcal{Q}$	$\lvert\mathcal{Q}\rvert$	$\mathrm{VC}(\mathcal{Q})$	Previous bound	Ref.	Theorem 7.4.1
$\mathcal{Q}^{\mathrm{pt}}$	D	1	$O\left(\frac{d}{\varepsilon n}\right)$	Prop. 7.2.8	$\tilde{O}\left(\frac{d}{\varepsilon n}\right)^{1/3}$
$\mathcal{Q}^{\mathrm{thr}}$	D	1	$\frac{\tilde{O}(\sqrt{D})}{\varepsilon n}$	Thm. 7.2.9	$\tilde{O}\left(\frac{d}{\varepsilon n}\right)^{1/3}$
$\mathcal{Q}^{\mathrm{conj}}$	3^d	d	$\frac{\tilde{O}(\sqrt{D})}{\varepsilon n}$	Thm. 7.2.9	$O\left(\frac{d^2}{\varepsilon n}\right)^{1/3}$
$\mathcal{Q}^{\mathrm{means}}$	d	$\lfloor\log_2 d\rfloor$	$O\left(\frac{d}{\varepsilon n}\right)$	Thm. 7.2.6	$O\left(\frac{d \log d}{\varepsilon n}\right)^{1/3}$
$\mathcal{Q}_t^{\mathrm{conj}}$ for $t \ll d$	$O(d^t)$	$O(t \log d)$	$O\left(\frac{d^t}{\varepsilon n}\right)$	Thm. 7.2.6	$O\left(\frac{t \cdot d \log d}{\varepsilon n}\right)^{1/3}$

We see that there is an exponential improvement in the dependence on $D = 2^d = |\mathcal{X}|$ for the case of threshold functions and conjunctions (and similarly in the dependence on t for t-way conjunctions). In particular, we only need n to be polynomially large in the bit-length d of the rows to have vanishingly small error; in such a case, we can produce and publish a differentially private synthetic dataset that accurately summarizes exponentially many ($2^{\Theta(d)}$) statistics about the original dataset (e.g., the fractions of individuals with every combination of attributes, as in $\mathcal{Q}^{\mathrm{conj}}(d)$). It is amazing that such a rich release of statistics is compatible with strong privacy protections.

[5] $\mathrm{VC}(\mathcal{Q})$ is defined to be the largest number k such that there exist $x_1, \ldots, x_k \in \mathcal{X}$ for which $\{(q(x_1), \ldots, q(x_k)) : q \in \mathcal{Q}\} = \{0,1\}^k$. Clearly, $\mathrm{VC}(\mathcal{Q}) \leq \log|\mathcal{Q}|$.

These improvements also hold compared with the bounds we had for (ε, δ)-differential privacy (where the dependence on $|\mathcal{Q}|$ was only quadratically better than for pure differential privacy). On the other hand, for point functions and attribute means, our earlier bounds (even for pure differential privacy) are better than what is given by Theorem 7.4.1.

Proof of Theorem 7.4.1: We begin by establishing the existence of at least one accurate m-row synthetic dataset y^*: Let y^* be a random sample of m rows from x, say with replacement for simplicity. By the Chernoff bound,

$$\Pr[\ \exists q \in \mathcal{Q} \text{ s.t. } |q(y^*) - q(x)| > \alpha\ )] \ \leq\ 2^{-\Omega(m\alpha^2)} \cdot |\mathcal{Q}| \ <\ 1\,,$$

for an appropriate choice of $m = O(\log|\mathcal{Q}|/\alpha^2)$. This is similar to "Occam's razor" arguments in computational learning theory (cf. [70]). In fact, it is known that $m = O(\mathrm{VC}(\mathcal{Q}) \cdot \log(1/\alpha)/\alpha^2)$ suffices.

Of course, outputting a random subsample of the dataset will not be differentially private. Instead, we use (a special case of) the *exponential mechanism* of McSherry and Talwar [79]. Specifically, consider the following mechanism $\mathcal{M}(x)$:

1. For each $y \in \mathcal{X}^m$, define $\mathrm{weight}_x(y) = \exp\left(-\varepsilon n \cdot \max_{q\in\mathcal{Q}} |q(y) - q(x)|\right)$.
2. Output y with probability proportional to $\mathrm{weight}_x(y)$. That is,

$$\Pr[\mathcal{M}(x) = y] = \frac{\mathrm{weight}_x(y)}{\sum_{z\in\mathcal{X}^m} \mathrm{weight}_x(z)}.$$

Notice that, if $x \sim x'$, then $\mathrm{weight}_x(y)$ and $\mathrm{weight}_{x'}(y)$ differ by a multiplicative factor of at most e^{ε}. That is, we smoothly vary the weight put on different synthetic datasets according to the amount of error they will give us, with low-error synthetic datasets receiving the highest weight.

Let us now formally analyze this algorithm.

Privacy: Fix $x \sim x' \in \mathcal{X}^n$, $y \in \mathcal{X}^m$. Then,

$$\Pr[\mathcal{M}(x) = y] = \frac{\mathrm{weight}_x(y)}{\sum_{y'} \mathrm{weight}_x(y')} \leq \frac{\mathrm{e}^{\varepsilon} \cdot \mathrm{weight}_{x'}(y)}{\sum_{y'} \mathrm{e}^{-\varepsilon} \cdot \mathrm{weight}_{x'}(y')} \leq \mathrm{e}^{2\varepsilon} \cdot \Pr[M(x') = y].$$

Thus, we have 2ε-differential privacy.

Accuracy: Define an output $y \in \mathcal{X}^m$ to be *β-accurate* if $\max_{q\in\mathcal{Q}} |q(y) - q(x)| \leq \beta$. Our goal is to show that, with high probability, $\mathcal{M}(x)$ is 2α-accurate. Recall that earlier we showed that there *exists* an α-accurate output y^*. We have

$$\begin{aligned}\Pr[\mathcal{M}(x) \text{ is not } 2\alpha\text{-accurate}] &= \sum_{\substack{y\in\mathcal{X}^m,\\ y \text{ not } 2\alpha\text{-accurate}}} \frac{\text{weight}_x(y)}{\sum_z \text{weight}_x(z)}\\ &\le \sum_{\substack{y\in\mathcal{X}^m,\\ y \text{ not } 2\alpha\text{-accurate}}} \frac{\text{weight}_x(y)}{\text{weight}_x(y^*)}\\ &\le |\mathcal{X}|^m \cdot \frac{\exp(-\varepsilon n \cdot 2\alpha)}{\exp(-\varepsilon n \cdot \alpha)}\\ &\ll 1 \qquad (\text{if } \alpha\varepsilon n > 2m \log|\mathcal{X}|).\end{aligned}$$

Recall that $m = O(\log|\mathcal{Q}|)/\alpha^2$. Solving for α gives the theorem. ∎

The exponential mechanism is quite general and powerful, and can be used to design differentially private mechanisms for sampling "good" outputs from any output space $\mathcal{Y}$. Specifically, we can replace the expression

$$-\max_{q\in Q} |q(y) - q(x)|$$

with an arbitrary "score function" $\text{score}(x, y)$ indicating how good y is as an output on dataset x, and replace the factor of n in the exponent with a bound B on the reciprocal of $\max_z \text{GS}_{\text{score}(\cdot,z)}$. That is, we obtain the following mechanism $\mathcal{M}_{\text{score},B}(x)$:

1. For each $y \in \mathcal{Y}$, define $\text{weight}_x(y) = \exp(\varepsilon \cdot \text{score}(x, y)/B)$.
2. Output y with probability proportional to $\text{weight}_x(y)$. That is,

$$\Pr[\mathcal{M}(x) = y] = \frac{\text{weight}_x(y)}{\sum_{z\in\mathcal{Y}} \text{weight}_x(z)}.$$

Similarly to the proof of Theorem 7.4.1, it can be shown that:

Proposition 7.4.2 (The exponential mechanism, McSherry and Talwar [79]). *For every function* $\text{score} : \mathcal{X}^n \times \mathcal{Y} \to \mathbb{R}$ *such that* $\mathcal{Y}$ *is finite,* $\varepsilon \ge 0$*, and* $B > 0$*,*

1. *If* $B \ge \max_z \text{GS}_{\text{score}(\cdot,z)}$*, then the mechanism* $\mathcal{M}_{\text{score},B}$ *is* 2ε*-differentially private, and*
2. *For every dataset* $x \in \mathcal{X}^n$*, with high probability,* $\mathcal{M}_{\text{score},B}(x)$ *outputs* y *such that*

$$\text{score}(x, y) \ge \text{argmax}_{y^*} \text{score}(x, y^*) - O(\log|\mathcal{Y}|) \cdot B/\varepsilon.$$

The downside. While the exponential mechanism is very powerful, it can be computationally very expensive, as a direct implementation requires enumerating over all $y \in \mathcal{Y}$. Indeed, in the application of Theorem 7.4.1, the computation time is roughly

$$|\mathcal{Y}| = |\mathcal{X}|^m = \exp\left(\frac{\log|\mathcal{Q}| \log|\mathcal{X}|}{\alpha^2}\right),$$

so it is very slow. For example, we get runtime $\exp(d^2/\alpha^2)$ for the query family $\mathcal{Q}^{\text{conj}}$ of conjunctions on $\{0, 1\}^d$.

7.4.2 Private Multiplicative Weights

We now present a state-of-the-art algorithm for general queries:

Theorem 7.4.3 (Private multiplicative weights, Hardt and Rothblum [58]). *For every set $\mathcal{Q}$ of counting queries on a data universe $\mathcal{X}$ and every $\varepsilon, \delta > 0$, there exists an (ε, δ)-differentially private mechanism $\mathcal{M}$ such that, for all datasets $x \in \mathcal{X}^n$, with high probability $\mathcal{M}(x)$ answers all queries in $\mathcal{Q}$ to within error at most*

$$\alpha = O\left(\frac{\sqrt{\log|\mathcal{X}| \cdot \log(1/\delta)} \cdot \log|\mathcal{Q}|}{\varepsilon n}\right)^{1/2}.$$

Moreover, $\mathcal{M}(x)$ can answer the queries in an online fashion (answering each query as it arrives) and runs in time $\mathrm{poly}(n, |\mathcal{X}|)$ *per query.*

The algorithm can also be modified to produce a synthetic dataset, though we will not show this here.

Note that the error vanishes more quickly with n than in Theorem 7.4.1 (as $1/n^{1/2}$ rather than $1/n^{1/3}$), and the $\log|\mathcal{X}|$ has been replaced by $\sqrt{\log|\mathcal{X}| \cdot \log(1/\delta)}$. Comparing with the results we have seen for our example query families, we have

Table 7.3: Error bounds for specific query families under (ε, δ)-differential privacy on a data universe $\mathcal{X}$ of size $D = 2^d$ (e.g., $\mathcal{X} = \{0,1\}^d$ or $\mathcal{X} = \{1, 2, \ldots, D\}$). Highlighted cells indicate the best bounds in the regime where $n \le D^{o(1)}$ or $n \le d^{o(t)}$ and $\delta \ge 2^{-\mathrm{polylog}(n)}$. In the case of incomparable bounds, both are highlighted.

Query family $\mathcal{Q}$	Sect. 7.2	Ref.	Thm. 7.4.1	Thm. 7.4.3
$\mathcal{Q}^{\mathrm{pt}}$	$O\left(\frac{d}{\varepsilon n}\right)$	Prop. 7.2.8		$O\left(\frac{d^{3/2} \cdot \sqrt{\log(1/\delta)}}{\varepsilon n}\right)^{1/2}$
$\mathcal{Q}^{\mathrm{thr}}$	$\frac{\tilde{O}(\sqrt{D})}{\varepsilon n}$	Thm. 7.2.9	$\tilde{O}\left(\frac{d}{\varepsilon n}\right)^{1/3}$	$O\left(\frac{d^{3/2} \cdot \sqrt{\log(1/\delta)}}{\varepsilon n}\right)^{1/2}$
$\mathcal{Q}^{\mathrm{conj}}$	$\frac{\tilde{O}(2^{d/2})}{\varepsilon n}$	Thm. 7.2.9	$O\left(\frac{d^2}{\varepsilon n}\right)^{1/3}$	$O\left(\frac{d^{3/2} \cdot \sqrt{\log(1/\delta)}}{\varepsilon n}\right)^{1/2}$
$\mathcal{Q}^{\mathrm{means}}$	$O\left(\frac{\sqrt{d \log(1/\delta) \cdot \log\log d}}{\varepsilon n}\right)$	Thm. 7.2.7		$O\left(\frac{\sqrt{d \log(1/\delta) \cdot \log d}}{\varepsilon n}\right)^{1/2}$
$\mathcal{Q}_t^{\mathrm{conj}}$ for $t \ll d$	$O\left(\frac{d^{t/2} \cdot \sqrt{\log(1/\delta) \cdot \log\log d}}{\varepsilon n}\right)$	Thm. 7.2.7	$O\left(\frac{t \cdot d \log d}{\varepsilon n}\right)^{1/3}$	$O\left(\frac{t \log d \sqrt{d \log(1/\delta)}}{\varepsilon n}\right)^{1/2}$

For $\mathcal{Q}^{\mathrm{conj}}$ and $\mathcal{Q}_t^{\mathrm{conj}}$, we obtain a saving in the dependence on $|\mathcal{X}| = 2^d$. In particular, for answering all conjunctions on $\{0,1\}^d$ with error tending to zero, we only need $n = \omega(d^{3/2} \cdot \sqrt{\log(1/\delta)}/\varepsilon)$ rather than $n = \omega(d^2/\varepsilon)$ as in Theorem 7.4.1. The running time has improved too, but is still at least $|\mathcal{X}| \cdot |\mathcal{Q}|$, which is exponential in d. (Of course, in this generality, one needs $|\mathcal{X}| \cdot |\mathcal{Q}|$ bits to specify an arbitrary set of counting queries on $\{0,1\}^d$.)

Proof: The algorithm views the dataset x as a distribution on types $r \in \mathcal{X}$:

$$x(r) = \frac{\#\{i \in [n] : x_i = r\}}{n}.$$

Then,

$$q(x) = \mathop{\mathrm{E}}_{r \leftarrow x}[q(r)].$$

The algorithm will maintain a distribution h on $\mathcal{X}$, some hypothesis for what the data distribution is. It will try to answer queries with h, and update h when it leads to too much error. It will turn out that only a small number of updates are needed, and this will imply that the overall privacy loss is small. Here are the details:

1. INITIALIZE the hypothesis h to the uniform distribution on $\mathcal{X}$.
2. REPEAT at most $O(\log|\mathcal{X}|)/\alpha^2$ times (outer loop)
   a. RANDOMIZE the accuracy threshold: $\hat{\alpha} = \alpha/2 + \mathrm{Lap}(1/\varepsilon_0 n)$, where ε_0 is a parameter that will be set later in the proof.
   b. REPEAT (inner loop)
      i. Receive next query q.
      ii. If $|q(x) - q(h)| + \mathrm{Lap}(1/\varepsilon_0 n) < \hat{\alpha}$, then output $a = q(h)$ and CONTINUE inner loop. Otherwise, output $a = q(x) + \mathrm{Lap}(1/\varepsilon_0 n)$ (with fresh noise) and EXIT inner loop.
   c. UPDATE the hypothesis h:
      i. Reweight using query q: $\forall w \in \mathcal{X}\ g(w) = \begin{cases} h(w)\mathrm{e}^{(\alpha/8)\cdot q(w)} & \text{if } a > q(h), \\ h(w)\mathrm{e}^{-(\alpha/8)\cdot q(w)} & \text{if } a < q(h). \end{cases}$
      ii. Renormalize: $\forall w \in \mathcal{X}\ h(w) = \dfrac{g(w)}{\sum_{v \in \mathcal{X}} g(v)}$.
   d. CONTINUE outer loop.

Utility analysis: By the exponentially vanishing tails of the Laplace distribution, with high probability none of the (at most $3|\mathcal{Q}|$) samples from $\mathrm{Lap}(1/\varepsilon_0 n)$ used in steps 2a and 2(b)ii has magnitude larger than

$$O\left(\frac{\log|\mathcal{Q}|}{\varepsilon_0 n}\right) \le \frac{\alpha}{8},$$

provided we set $\varepsilon_0 \ge c \log|\mathcal{Q}|/\alpha n$ for a sufficiently large constant c. By the triangle inequality, this implies that all answers that we provide are within $\pm 3\alpha/4$ of $q(x)$.

Now, we must show that the mechanism will not stop early.

Claim 7.4.4. *Assuming all the samples from* $\mathrm{Lap}(1/\varepsilon_0 n)$ *have magnitude at most* $\alpha/8$*, the outer loop cannot exceed its budget of* $O(\log|\mathcal{X}|)/\alpha^2$ *iterations.*

Proof sketch: We use the Kullback–Leibler divergence $D(x\|h)$ as a potential function. At the start, h is the uniform distribution on $|\mathcal{X}|$, so

$$D(x\|h) = \log|\mathcal{X}| - H(x) \le \log|\mathcal{X}|,$$

where $H(x)$ is the Shannon entropy of the distribution x. Suppose that, in some iteration, we do an update (i.e., reweight and renormalize) to go from hypothesis h

to hypothesis h'. Since all the noise samples have magnitude at most $\alpha/8$, we must have $|q(x) - q(h)| \geq \alpha/4$ in order to do an update, and in this case $b - q(h)$ has the same sign as $q(x) - q(h)$. By a tedious but standard calculation (used in typical analyses of the multiplicative weights method), this implies that

$$D(x\|h') \leq D(x\|h) - \Omega(\alpha^2).$$

Since divergence is always nonnegative, we can have at most $\log|\mathcal{X}|/\Omega(\alpha^2)$ updates. ■

Privacy analysis: The mechanism takes a dataset x and outputs a sequence $(a_1, \ldots, a_k)$ of noisy answers to a sequence of queries $(q_1, \ldots, q_k)$ (which we will treat as fixed in this analysis). Note that the output $(a_1, \ldots, a_k)$ is determined by the sequence $(b_1, \ldots, b_k)$ where $b_i = \perp$ if there is no update on query q_i and $b_i = a_i$ otherwise. (This information suffices to maintain the hypothesis h used by the algorithm, as the update to h done in step 2c depends only on the current query q_i and the noisy answer $a_i = b_i$.) Thus, by closure under postprocessing (Lemma 7.2.1), it suffices to show that the mechanism that outputs the sequence $(b_1, \ldots, b_k)$ is (ε, δ)-differentially private. This mechanism, in turn, is obtained by (adaptively) composing $O(\log|\mathcal{X}|)/\alpha^2$ submechanisms, each corresponding to one execution of the outer loop. Specifically, each such submechanism is parameterized by the output of the previous submechanisms, which is of the form $(b_1, \ldots, b_{i-1})$ with $b_{i-1} \neq \perp$, and produces the output $(b_i, \ldots, b_j)$ corresponding to one more execution of the outer loop — so $b_i = b_{i+1} = \cdots = b_{j-1} = \perp$ and $b_j \neq \perp$ (unless $j = k$, in which case we may also have $b_j = \perp$).

We will argue below that each such submechanism is $4\varepsilon_0$-differentially private (even though the number of queries it answers can be unbounded). Given this claim, we can apply advanced composition to deduce that the overall mechanism satisfies (ε, δ)-differential privacy for

$$\varepsilon = O\left(\sqrt{\frac{\log|\mathcal{X}|\log(1/\delta)}{\alpha^2}} \cdot \varepsilon_0\right).$$

Substituting $\varepsilon_0 = c\log|\mathcal{Q}|/\alpha n$ (as needed in the utility analysis above) and solving for α yields the theorem.

So now we turn to analyzing a submechanism $\mathcal{M}$ corresponding to a single execution of the outer loop (after a fixed prior history $(b_1, \ldots, b_{i-1})$). Since it suffices to verify pure differential privacy with respect to singleton outputs, it suffices to show that, for every hypothesis h (determined by the prior history $(b_1, \ldots, b_{i-1})$) and every possible output sequence $b = (b_i, \ldots, b_j)$ with $b_i = b_{i+1} = \cdots = b_{j-1} = \perp$, the following mechanism $\mathcal{M}_{h,b}(x)$, which tests whether the output of the next iteration of the outer loop is b, is $4\varepsilon_0$-differentially private:

1. SAMPLE $\nu_\alpha, \nu_i, \nu_{i+1}, \ldots, \nu_j, \nu_a \leftarrow \mathrm{Lap}(1/\varepsilon_0 n)$. (Making all random choices at start.)
2. RANDOMIZE the accuracy threshold: $\hat{\alpha} = \alpha/2 + \nu_\alpha$.

3. REPEAT for $t = i$ to j (inner loop)
   a. Receive next query q_t.
   b. If $b_t = \perp$ and $|q_t(x) - q_t(h)| + \nu_t \geq \hat{\alpha}$, then HALT and OUTPUT 0.
   c. If $b_t \neq \perp$ (which implies $t = j$), then:
      i. If $|q_t(x) - q_t(h)| + \nu_t < \hat{\alpha}$, HALT and OUTPUT 0.
      ii. If $q_t(x) + \nu_a \neq b_j$, HALT and OUTPUT 0.
4. OUTPUT 1 (if we have not halted with output 0 so far).

Let us consider the case when $b_j \neq \perp$; the case when $b_j = \perp$ is similar but simpler. We will argue $4\varepsilon_0$-differential privacy even when $\nu_i, \nu_{i+1}, \ldots, \nu_{j-1}$ are fixed to arbitrary values (so the only randomness is from ν_α, ν_j, ν_a); averaging over these independent random variables will preserve differential privacy.

To show this, we will show that we can compute the output of $\mathcal{M}_{h,b}$ from the composition of three algorithms, which are ε_0-, $2\varepsilon_0$-, and ε_0-differentially private, respectively.

To determine whether we ever halt and output 0 in step 3b it suffices to calculate

$$\beta = \hat{\alpha} - \max_{i \leq t < j}(|q_t(x) - q_t(h)| + \nu_t) = \alpha/2 + \nu_\alpha - \max_{i \leq t < j}(|q_t(x) - q_t(h)| + \nu_t).$$

We halt and output 0 in one of the executions of step 3b iff $\beta \leq 0$. The calculation of β is ε_0-differentially private by the Laplace mechanism because $\alpha/2 - \max_{i \leq t < j}(|q_t(x) - q_t(h)| + \nu_t)$ has sensitivity at most $1/n$ as a function of the dataset x (recalling that h and the ν_t's for $i \leq t < j$ are all fixed) and ν_α is distributed according to $\mathrm{Lap}(1/\varepsilon_0 n)$. This argument is the key to why the private multiplicative weights can answer so many queries—we are only paying once for privacy despite the fact that this condition involves an unbounded number of queries.

Given β, to determine whether or not we halt and output 0 in step 3(c)i, it suffices to test whether $|q_j(x) - q_j(h)| + \nu_j \geq \hat{\alpha} = \beta + \max_{i \leq t < j}(|q_t(x) - q_t(h)| + \nu_t)$. This is $2\varepsilon_0$-differentially private by the Laplace mechanism because $|q_j(x) - q_j(h)| - \beta - \max_{i \leq t < j}(|q_t(x) - q_t(h)| + \nu_t)$ has sensitivity at most $2/n$ as a function of x and ν_j is independently distributed according to $\mathrm{Lap}(1/\varepsilon_0 n)$.

Finally, step 3(c)ii is ε_0-differentially private by the Laplace mechanism (with fresh randomness ν_a). ∎

Remark 7.4.5.

- *The hypothesis h maintained by the private multiplicative weights algorithm can be thought of as a fractional version of a synthetic dataset. Indeed, with a bit more work it can be ensured that at the end of the algorithm, we have $|q(h) - q(x)| \leq \alpha$ for all $q \in \mathcal{Q}$. Finally, random sampling from the distribution h can be used to obtain a true, integral synthetic dataset $y \in \mathcal{X}^m$ of size $m = O(\log|\mathcal{Q}|/\alpha^2)$ just like in Theorem 7.4.1.*
- *The algorithm works in an online fashion, meaning that it can answer query q_i without knowing the future queries $q_{i+1}, q_{i+2}, \ldots$. However, if all queries are given simultaneously, the algorithm can be sped up by using the exponential*

mechanism (Proposition 7.4.2) to identify queries that will generate an update (rather than wasting time on queries that do not generate an update) [61].

7.5 Information-Theoretic Lower Bounds

In the previous section, we have seen differentially private algorithms that can answer many counting queries with good accuracy. Now we turn to lower bounds, with the goal of showing that these algorithms are nearly optimal in terms of the number of queries and accuracy they can achieve. These lower bounds will be information-theoretic, meaning that they apply regardless of the computational resources of the mechanism $\mathcal{M}$.

7.5.1 Reconstruction Attacks and Discrepancy

7.5.1.1 Reconstruction

We begin by defining a very weak standard for privacy, namely avoiding an attack that reconstructs almost all of the dataset:

Definition 7.5.1 (Blatant nonprivacy, Dinur and Nissim [31]). *A mechanism $\mathcal{M} : \mathcal{X}^n \to \mathcal{Y}$ is called* blatantly nonprivate *if, for every $x \in \mathcal{X}^n$, one can use $\mathcal{M}(x)$ to compute an $x' \in \mathcal{X}^n$, such that x' and x differ in at most $n/10$ coordinates (with high probability over the randomness of $\mathcal{M}$).*

It can be shown that a mechanism that is $(1, 0.1)$-differentially private cannot be blatantly nonprivate (if $|\mathcal{X}| > 1$). Indeed, if we run an (ε, δ)-differentially private mechanism $\mathcal{M}$ on a uniformly random dataset $X \leftarrow \mathcal{X}^n$, then the expected fraction of rows that any adversary can reconstruct is at most $e^{\varepsilon}/|\mathcal{X}| + \delta$ (since if we replace any row X_i with an independent row X'_i, $\mathcal{M}(X_{-i}, X'_i)$ reveals no information about X_i and thus does not allow for reconstructing X_i with probability larger than $1/|\mathcal{X}|$).

We now give some fundamental lower bounds, due to Dinur and Nissim [31], on the tradeoff between the error and the number of counting queries that can be answered while avoiding blatant nonprivacy. These lower bounds predate, and indeed inspired, the development of differential privacy.

Let $\mathcal{X} = \{0, 1\}$. Then a dataset of n people is simply a vector $x \in \{0, 1\}^n$. We will consider (normalized) *inner-product queries* specified by a vector $q \in \{0, 1\}^n$: the intended answer to the query q is $\langle q, x\rangle/n \in [0, 1]$. Think of the bits in x as specifying a sensitive attribute of the n members of the dataset and q as specifying a subset of the population according to some publicly known demographics. Then $\langle q, x\rangle/n$ measures the correlation between the specified demographic traits and the sensitive attribute.

These are not exactly counting queries, but they can be transformed into counting queries as follows: Let $\tilde{\mathcal{X}} = [n] \times \{0, 1\}$ be our data universe, map an inner-product query $q \in \{0, 1\}^n$ to the counting query $\tilde{q}((i, b)) = q_i \cdot b$, and consider datasets of the form $\tilde{x} = ((1, x_1), (2, x_2), \ldots, (n, x_n))$, $\tilde{q}((i, b)) = q_i \cdot b$. Then $\tilde{q}(\tilde{x}) = \langle q, x\rangle/n$, and

reconstructing x is equivalent to reconstructing $\tilde{x}$, which again contradicts $(1, 0.1)$-differential privacy.

Theorem 7.5.2 (Reconstruction from many queries with large error [31]). *Let $x \in \{0,1\}^n$. If we are given, for each $q \in \{0,1\}^n$, a value $y_q \in \mathbb{R}$ such that*

$$\left| y_q - \frac{\langle q, x\rangle}{n} \right| \leq \alpha,$$

then one can use the y_q's to compute $x' \in \{0,1\}^n$ such that x and x' differ in at most 4α fraction of coordinates.

Corollary 7.5.3. *If $\mathcal{M}(x)$ is a mechanism that outputs values y_q as above with $\alpha \leq 1/40$, then $\mathcal{M}$ is blatantly nonprivate.*

Thus at least $\Omega(1)$ additive error is needed for privately answering all 2^n normalized inner-product queries, which as noted correspond to 2^n counting queries on a data universe of size $2n$.

The smallDB mechanism (Theorem 7.4.1) can answer $\exp(\tilde{\Omega}(n))$ counting queries over a data universe $\mathcal{X}$ with ε-differential privacy and error α provided $|\mathcal{X}| \leq \exp(\text{polylog}(n))$ and $\varepsilon, \alpha \geq 1/\text{polylog}(n)$. Corollary 7.5.3 says that we cannot push this further to answer 2^n queries.

Proof of Theorem 7.5.2: Pick any $x' \in \{0,1\}^n$ such that, for all $q \in \{0,1\}^n$,

$$\left| y_q - \frac{\langle q, x'\rangle}{n} \right| \leq \alpha.$$

(We know that at least one such x' exists, namely x.)

We need to prove that x and x' differ on at most a 4α fraction of coordinates. Let $q_1 = x$ and let q_0 be the bitwise complement of x. Then, the relative Hamming distance between x and x' equals

$$\begin{aligned}
\frac{d(x,x')}{n} &= \frac{|\langle q_0, x\rangle - \langle q_0, x'\rangle| + |\langle q_1, x\rangle - \langle q_1, x'\rangle|}{n} \\
&\leq \left| \frac{\langle q_0, x\rangle}{n} - y_{q_0} \right| + \left| y_{q_0} - \frac{\langle q_0, x'\rangle}{n} \right| + \left| \frac{\langle q_1, x\rangle}{n} - y_{q_1} \right| + \left| y_{q_1} - \frac{\langle q_1, x'\rangle}{n} \right| \\
&\leq 4 \cdot \alpha.
\end{aligned}$$

■

Of course we can avoid the above attack by restricting the adversary to fewer than 2^n queries. The next theorem will say that, even for much fewer queries (indeed $O(n)$ queries), we must incur a significant amount of error, $\alpha \geq \Omega(1/\sqrt{n})$. This is tight, matching Theorem 7.2.7 up to a factor of $O(\sqrt{\log(1/\delta) \cdot \log\log n})$. We will in fact study the more general question of what additive error is needed for privately answering any set $\mathcal{Q}$ of counting queries.

Let $q_1, \ldots, q_k \in \{0,1\}^n$ be a collection of vectors, which we view as specifying inner-product queries $\langle q, x\rangle/n$ as above. Suppose we have a mechanism $\mathcal{M}$

that answers these queries to within error α, i.e., with high probability outputs $y_1, \ldots, y_k \in [0,1]$ with

$$\left| y_j - \frac{\langle q_j, x \rangle}{n} \right| \leq \alpha.$$

Let us try to show that $\mathcal{M}$ is blatantly nonprivate. Our privacy-breaking strategy is the same: take any $x' \in \{0,1\}^n$ with

$$\left| y_j - \frac{\langle q_j, x' \rangle}{n} \right| \leq \alpha$$

for each j.

Then, by the triangle inequality, we have $|\langle q_j, x - x' \rangle|/n \leq 2\alpha$ for all $j = 1, \ldots, k$. For blatant nonprivacy, we want to use this to deduce that x and x' have Hamming distance at most $n/10$, i.e., $\|x - x'\|_1 \leq n/10$. Suppose not. Let $z = x - x'$. Let $\mathcal{Q}$ denote the $k \times n$ matrix whose rows are the q_j. Thus, we have

1. z is a $\{0, +1, -1\}$ vector with $\|z\|_1 > n/10$,
2. $\|\mathcal{Q}z\|_\infty \leq 2\alpha n$.

Thus, we have a contradiction (and hence can conclude that $\mathcal{M}$ is blatantly nonprivate) if the partial discrepancy of $\mathcal{Q}$, defined as follows, is larger than $2\alpha n$:

Definition 7.5.4 ((Partial) discrepancy). *For a $k \times n$ matrix $\mathcal{Q}$, we define its* discrepancy $\mathsf{Disc}(\mathcal{Q})$ *and its* partial discrepancy $\mathsf{PDisc}(\mathcal{Q})$ *as*

$$\mathsf{Disc}(\mathcal{Q}) = \min_{z \in \{\pm 1\}^n} \|\mathcal{Q}z\|_\infty, \text{ and}$$
$$\mathsf{PDisc}(\mathcal{Q}) = \min_{\substack{z \in \{0,+1,-1\}^n, \\ \|z\|_1 > n/10}} \|\mathcal{Q}z\|_\infty.$$

The qualifier "partial" refers to the fact that we allow up to 90% of z's coordinates to be zero, in contrast to ordinary discrepancy which only considers vectors $z \in \{\pm 1\}^n$. A more combinatorial perspective comes if we think of the rows of $\mathcal{Q}$ as characteristic vectors of subsets of $\mathcal{X}$, and z as a partial ± 1-coloring of the elements of $\mathcal{X}$. Then $\|\mathcal{Q}z\|_\infty$ measures the largest imbalance in coloring over all the sets in $\mathcal{Q}$, and $\mathsf{PDisc}(\mathcal{Q})$ refers to minimizing this maximum imbalance over all partial colorings z.

Summarizing the discussion before Definition 7.5.4, we have:

Theorem 7.5.5 (Reconstruction via partial discrepancy). *Let $q_1, \ldots, q_k \in \{0,1\}^n$ and $\mathcal{Q}$ be the $k \times n$ matrix whose rows are the q_j's. Then any mechanism $\mathcal{M} : \{0,1\}^n \to \mathbb{R}^k$ that answers all of the normalized inner-product queries specified by $q_1, \ldots, q_k$ to within additive error α smaller than $\mathsf{PDisc}(\mathcal{Q})/2n$ is blatantly nonprivate.*

We note that Theorem 7.5.5 is a generalization of Theorem 7.5.2. Indeed, if $\mathcal{Q}$ is the $2^n \times n$ matrix whose rows are all bitstrings of length n (i.e., the family of all subsets of $[n]$), then the partial discrepancy of $\mathcal{Q}$ is greater than $n/20$. (For a partial

coloring z with greater than $n/10$ nonzero entries, either the set of coordinates on which z is 1 or the set of coordinates on which z is -1 will have imbalance greater than $n/20$.)

Let us now use Theorem 7.5.5 to deduce the second theorem of Dinur and Nissim [31].

Theorem 7.5.6 (Reconstruction from few queries with small error [31]). *There exists $c > 0$ and $q_1, \ldots, q_n \in \{0,1\}^n$ such that any mechanism that answers the normalized inner-product queries specified by $q_1, \ldots, q_n$ to within error at most $c/\sqrt{n}$ is blatantly nonprivate.*

In fact, the theorem holds for a random set of queries, as follows from combining the following lemma (setting $k = s = n$) with Theorem 7.5.5:

Lemma 7.5.7 (Discrepancy of a random matrix). *For all integers $k \geq s \geq 0$, with high probability, a $k \times s$ matrix $\mathcal{Q}$ with uniform and independent entries from $\{0,1\}$ has partial discrepancy at least*

$$\Omega\left(\min\left\{\sqrt{s \cdot (1 + \log(k/s))}, s\right\}\right).$$

Up to the hidden constant, this is the largest possible discrepancy for a $k \times s$ matrix. Indeed, a random coloring achieves discrepancy at most $O(\sqrt{s \cdot \log k})$ (by a Chernoff bound and union bound). The celebrated "six standard deviations suffice" result of Spencer [97] improves the $\log k$ to $\log(k/s)$.

Proof sketch: Pick the rows $q_1, \ldots, q_k \in \{0,1\}^s$ uniformly at random. Fix $z \in \{0, +1, -1\}^s$ with $\|z\|_1 > s/10$. Then for each j, $\langle q_j, z\rangle$ is a difference of two binomial distributions, at least one of which is the sum of more than $s/20$ independent, unbiased $\{0,1\}$ random variables (since z has more than $s/20$ coordinates that are all 1 or all -1). By anticoncentration of the binomial distribution (cf. [76, Prop. 7.3.2]), we have for every $t \geq 0$

$$\Pr_{q_j}\left[\left|\langle q_j, z\rangle\right| \geq \min\{t\sqrt{s}, s/20\}\right] \geq \max\left\{1 - O(t), \Omega\left(e^{-O(t^2)}\right)\right\}.$$

Thus, for each z we have

$$\Pr\left[\forall j \in [k], \left|\langle q_j, z\rangle\right| < \min\{t\sqrt{s}, s/20\}\right] \leq \min\left\{O(t), 1 - \Omega\left(e^{-O(t^2)}\right)\right\}^k.$$

By a union bound, we have

$$\begin{aligned}\Pr\Big[\exists z \in \{-1, 0, +1\}^s : \|z\|_1 > s/10 \text{ and } \forall j \in [k], \left|\langle q_j, z\rangle\right| < \min\{t\sqrt{s}, s/20\}\Big] \\ < 3^s \cdot \min\left\{O(t), 1 - \Omega\left(e^{-O(t^2)}\right)\right\}^k.\end{aligned}$$

We now choose t to ensure that this probability is small. For every $k \geq s$, taking t to be a small enough constant suffices to ensure that $3^s \cdot O(t)^k \ll 1$. However, once k/s is sufficiently large, we can take a larger value of t (corresponding to higher discrepancy) if we use the other term in the min. Specifically, we can take $t = c\sqrt{\log(ck/s)}$ for a sufficiently small constant c, and obtain

$$3^s \cdot \left(1 - \Omega\left(e^{-O(t^2)}\right)\right)^k \leq 3^s \cdot \left(1 - \Omega\left(\frac{s}{ck}\right)\right)^k = 3^s \cdot e^{-\Omega(s/c)} \ll 1.$$

In all cases, we can take $t = \Omega\left(\sqrt{1 + \log(k/s)}\right)$, as needed for the lemma. ∎

The reconstruction attacks we gave in the proof of the above theorems take time more than 2^n, because they require searching for a vector $x' \in \{0,1\}^n$ such that

$$\forall j \left| y_j - \frac{\langle q_j, x' \rangle}{n} \right| \leq \alpha. \tag{7.4}$$

However, it is known how to obtain a polynomial-time reconstruction attack for certain query families. In particular, a polynomial-time analogue of Theorem 7.5.6 can be obtained by using a linear program to efficiently find a *fractional* vector $x' \in [0,1]^n$ satisfying Condition (7.4) and then rounding x' to an integer vector. To show that this attack works, we need to lower-bound the fractional analogue of partial discrepancy, namely

$$\inf_{\substack{z \in [-1,1]^n, \\ \|z\|_1 > n/10}} \|\mathcal{Q}z\|_\infty,$$

which again can be shown to be $\Omega(\sqrt{n})$ for a random $n \times n$ matrix $\mathcal{Q}$, as well as for some explicit constructions [37].

One can consider a relaxed notion of accuracy, where the mechanism is only required to give answers with at most $c/\sqrt{n}$ additive error for 51% of the queries, and for the remaining 49% it is free to make arbitrary error. Even such a mechanism can be shown to be blatantly nonprivate. If one wants this theorem with a polynomial-time privacy-breaking algorithm, then this can also be done with the 51% replaced by about 77%. (This is a theorem of Dwork, McSherry, and Talwar [39], and is based on connections to compressed sensing.)

7.5.1.2 Discrepancy Characterizations of Error for Counting Queries

We now work towards characterizing the error required for differential privacy for answering a given set of counting queries. Let $q_1, \ldots, q_k \in \{0,1\}^{\mathcal{X}}$ be a given set of *counting queries* over a data universe $\mathcal{X}$ (viewed as vectors of length $|\mathcal{X}|$). We will abuse notation and use $\mathcal{Q}$ to denote both the set $\{q_1, \ldots, q_k\}$ of counting queries as well as the $k \times |\mathcal{X}|$ matrix whose rows are the q_j. For a set $S \subseteq \mathcal{X}$, we let $\mathcal{Q}_S$ denote the restriction of $\mathcal{Q}$ to the columns of S.

Then we have:

Theorem 7.5.8 (Partial discrepancy lower bound). *Let $\mathcal{Q} = \{q : \mathcal{X} \to \{0,1\}\}$ be a set of counting queries over data universe $\mathcal{X}$, and let $\mathcal{M} : \mathcal{X}^n \to \mathbb{R}^{\mathcal{Q}}$ be a $(1, 0.1)$-differentially private mechanism that with high probability answers every query in $\mathcal{Q}$ with error at most α. Then*

$$\alpha \geq \max_{\substack{S \subseteq \mathcal{X}, |S| \leq 2n \\ |S| \text{ even}}} \mathrm{PDisc}(\mathcal{Q}_S)/2n.$$

Proof sketch: Suppose for contradiction that $\alpha < \mathsf{PDisc}(\mathcal{Q}_S)/2n$ for some set S of size at most $2n$. Let us restrict attention to datasets x of the following form: the first $|S|/2$ rows of x, denoted y, consist of $|S|/2$ distinct elements of S, and the rest are fixed to an arbitrary value $w \in \mathcal{X}$. Then for a counting query $q : \mathcal{X} \to \{0,1\}$, we have

$$q(x) = \frac{\langle q_S, \chi(y)\rangle + (n - |S|/2) \cdot q(w)}{n},$$

where $q_S \in \{0,1\}^S$ is the vector $(q(s))_{s\in S}$ (one of the rows in $\mathcal{Q}_S$) and $\chi(y) \in \{0,1\}^S$ is the characteristic vector of y (i.e., the indicator of which elements of S are in y). Thus, an estimate of $q(x)$ to within additive error at most α yields an estimate of the normalized inner product $\langle q_S, \chi(y)\rangle/|S|$ to within additive error $\alpha n/|S| < \mathsf{PDisc}(\mathcal{Q}_S)/2$. If we have such estimates for every query $q \in \mathcal{Q}$, then by Theorem 7.5.5, we can reconstruct at least 90% of the coordinates of the characteristic vector $\chi(y)$, which can be shown to contradict $(1, 0.1)$-differential privacy. ∎

If we do not fix n but require the error to scale linearly with n, then this lower bound can be phrased in terms of *hereditary partial discrepancy*, which is defined to be

$$\mathsf{HerPDisc}(\mathcal{Q}) \stackrel{\text{def}}{=} \max_{S\subseteq\mathcal{X}} \mathsf{PDisc}(\mathcal{Q}_S).$$

In this language, we have the theorem of Muthukrishnan and Nikolov [83]:

Theorem 7.5.9 (Hereditary discrepancy lower bound [83]). *For every set $\mathcal{Q} = \{q : \mathcal{X} \to \{0,1\}\}$ of counting queries over data universe $\mathcal{X}$, the following holds for all sufficiently large n (in particular for all $n \geq |\mathcal{X}|/2$): Let $\mathcal{M} : \mathcal{X}^n \to \mathbb{R}^{\mathcal{Q}}$ be a $(1, 0.1)$-differentially private mechanism that with high probability answers every query in $\mathcal{Q}$ with error at most α. Then*

$$\alpha \geq (\mathsf{HerPDisc}(\mathcal{Q}) - 1)/2n.$$

(We subtract 1 from the hereditary partial discrepancy to compensate for the fact it removes the constraint that $|S|$ is even from Theorem 7.5.8.) Put differently, the hereditary partial discrepancy is a lower bound on the non-normalized error (αn) needed to answer the queries with differential privacy (for sufficiently large n). Remarkably, Nikolov, Talwar, and Zhang [85] showed that this bound is nearly tight:

Theorem 7.5.10 (Hereditary discrepancy upper bound [85]). *For every set $\mathcal{Q} = \{q : \mathcal{X} \to \{0,1\}\}$ of counting queries over data universe $\mathcal{X}$, every $\varepsilon, \delta > 0$, and $n \in \mathbb{N}$, there is an (ε, δ)-differentially private mechanism $\mathcal{M} : \mathcal{X}^n \to \mathbb{R}^{\mathcal{Q}}$ that answers every query in $\mathcal{Q}$ with error*

$$\alpha \leq \frac{\mathsf{HerPDisc}(\mathcal{Q}) \cdot \mathrm{polylog}(|\mathcal{Q}|) \cdot \sqrt{\log(1/\delta)}}{\varepsilon n}$$

with high probability.

We will not prove the latter theorem, but will get a taste of its techniques in Section 7.7.3. We note that the distinction between partial discrepancy and ordinary

discrepancy becomes less significant once we move to the hereditary versions. Indeed, if we define $\mathsf{HerDisc}(\mathcal{Q}) \stackrel{\text{def}}{=} \max_{S \subseteq \mathcal{X}} \mathsf{Disc}(\mathcal{Q}_S)$, then it is known that

$$\mathsf{HerPDisc}(\mathcal{Q}) \leq \mathsf{HerDisc}(\mathcal{Q}) \leq \mathsf{HerPDisc}(\mathcal{Q}) \cdot O(\min\{\log|\mathcal{X}|, \log|\mathcal{Q}|\}). \quad (7.5)$$

(See the book by Matoušek [75] for proofs.) Hereditary discrepancy is a well-studied concept in combinatorics, and a remarkable byproduct of the aforementioned work on differential privacy was a polylogarithmic approximation algorithm for hereditary discrepancy, solving a long-standing open problem [85].

7.5.1.3 Discrepancy Lower Bounds for Specific Query Families

Note that Theorems 7.5.9 and 7.5.10 only provide a nearly tight characterization in case we look for error bounds of the form $f(\mathcal{Q})/n$, which scale linearly with n (ignoring the dependence on ε and $\log(1/\delta)$ for this discussion). In particular, the lower bound of Theorem 7.5.9 only says that $\mathsf{HerPDisc}(\mathcal{Q})$ is a lower bound on the function $f(\mathcal{Q})$ *for sufficiently large* n. If our dataset size n is below the point at which this lower bound kicks in, we may be able to achieve significantly smaller error.

For finite dataset sizes n, we can use the lower bound of Theorem 7.5.8:

$$\alpha \geq \max_{\substack{S \subseteq \mathcal{X}, |S| \leq 2n \\ |S| \text{ even}}} \mathsf{PDisc}(\mathcal{Q}_S)/2n.$$

Unfortunately, partial discrepancy is a combinatorially complex quantity, and can be hard to estimate. Fortunately, there are several relaxations of it that can be easier to estimate and thereby prove lower bounds:

Proposition 7.5.11. *Let $\mathcal{Q}$ be a $k \times |\mathcal{X}|$ query matrix (with $\{0,1\}$ entries). Then:*

1. *For every $S \subseteq \mathcal{X}$ and $T \subseteq [k]$, we have*

$$\mathsf{PDisc}(\mathcal{Q}_S)) > \frac{1}{10}\sqrt{\frac{|S|}{|T|}} \cdot \sigma_{\min}(\mathcal{Q}_S^T),$$

   where $\mathcal{Q}_S^T$ denotes the $|T| \times |S|$ submatrix of $\mathcal{Q}_S$ with rows indexed by T, and $\sigma_{\min}(\mathcal{Q}_S^T)$ denotes the smallest singular value of $\mathcal{Q}_S^T$.

2.
$$\max_{\substack{S \subseteq \mathcal{X}, |S| \leq 2n \\ |S| \text{ even}}} \mathsf{PDisc}(\mathcal{Q}_S) > \frac{\min\{\mathrm{VC}(\mathcal{Q}) - 1, 2n\}}{20}.$$

Proof:

1. We have

$$\begin{aligned}
\mathsf{PDisc}(\mathcal{Q}_S) &\geq \mathsf{PDisc}(\mathcal{Q}_S^T) \\
&= \min_{\substack{z\in\{-1,1\}^{|S|},\\ \|z\|_1>|S|/10}} \|\mathcal{Q}_S^T z\|_\infty \\
&> \inf_{z\neq 0} \frac{\|\mathcal{Q}_S^T z\|_\infty}{\|z\|_1 \cdot 10/|S|} \\
&\geq \inf_{z\neq 0} \frac{\|\mathcal{Q}_S^T z\|_2 / \sqrt{|T|}}{(\|z\|_2 \cdot \sqrt{|S|}) \cdot 10/|S|} \\
&= \frac{1}{10}\sqrt{\frac{|S|}{|T|}} \cdot \sigma_{\min}(\mathcal{Q}_S^T).
\end{aligned}$$

2. By definition of VC dimension, there is an even-sized set S of at least $\min\{(\mathrm{VC}(\mathcal{Q}) - 1, 2n\}$ columns for which the rows of $\mathcal{Q}_S$ contain all 2^k binary strings of length k. The partial discrepancy of this set of vectors is thus greater than $k/20$.

■

Combining Proposition 7.5.11 with Theorem 7.5.8, we obtain lower bounds on the error α needed by differentially private mechanisms in terms of least singular values of submatrices $\mathcal{Q}_S^T$ and in terms of the VC dimension $\mathrm{VC}(\mathcal{Q})$. The lower bound on error in terms of least singular values is due to Kasiviswanathan et al. [66], and the lower bound on error in terms of VC dimension is due to Blum et al. [14]. An advantage of using the singular-value relaxation in place of partial discrepancy is that it allows for a polynomial-time reconstruction attack, similarly to the discussion after the proof of Theorem 7.5.6. The attack based on VC dimension is based on brute-force enumeration, just like Theorem 7.5.2, but the search space is of size $2^{\mathrm{VC}(\mathcal{Q})} \leq |\mathcal{Q}|$.

Recall that the largest possible discrepancy among $k \times s$ matrices (with $k \geq s$) is achieved (up to constant factors) by a random matrix, with the bound stated in Lemma 7.5.7. To apply this for lower bounds on differentially private release of counting queries, we can take $\mathcal{Q}$ to be a family of k random counting queries over a data universe $\mathcal{X}$, and $S \subseteq \mathcal{X}$ to be an arbitrary subset of size $s = \min\{|\mathcal{Q}|, |\mathcal{X}|, n\}$. Then $\mathcal{Q}_S$ is a random matrix, and combining Lemma 7.5.7 and Theorem 7.5.8, we obtain:

Theorem 7.5.12 (Largest possible discrepancy lower bound). *For every data universe $\mathcal{X}$ and $n, k \in \mathbb{N}$, there is a family of k counting queries $\mathcal{Q}$ over $\mathcal{X}$ such that, if $\mathcal{M} : \mathcal{X}^n \to \mathbb{R}^{\mathcal{Q}}$ is a $(1, 0.1)$-differentially private mechanism that with high probability answers every query in $\mathcal{Q}$ with error at most α, we have*

$$\alpha \geq \Omega\left(\min\left\{\frac{\sqrt{|\mathcal{Q}|}}{n}, \frac{\sqrt{|\mathcal{X}| \cdot (1 + \log(|\mathcal{Q}|/|\mathcal{X}|))}}{n}, \sqrt{\frac{\log(|\mathcal{Q}|/n)}{n}}, 1\right\}\right).$$

Let us compare this with the upper bounds that we have for (ε, δ)-differential privacy given by Theorems 7.2.7, 7.2.9, and 7.4.3. For every family of counting queries, choosing the best of these algorithms will give an error bound of

$$\alpha \le O\left(\min\left\{\frac{\sqrt{|\mathcal{Q}| \cdot \log(1/\delta) \cdot \log\log|\mathcal{Q}|}}{\varepsilon n}, \frac{\sqrt{|\mathcal{X}| \cdot \log|\mathcal{Q}|}}{\varepsilon n}, \sqrt{\frac{\sqrt{\log|\mathcal{X}| \cdot \log(1/\delta) \cdot \log|\mathcal{Q}|}}{\varepsilon n}}, 1\right\}\right).$$

Ignoring factors of $\log(1/\delta)$ and $1/\varepsilon$, the first two bounds nearly match the first two lower bounds of Theorem 7.5.12. The third bound, however, differs by the $\sqrt{\log|\mathcal{X}|}$ factor that appears in the error bound of private multiplicative weights but does not appear in the lower bound (which leaves open the possibility of having vanishingly small error whenever $|\mathcal{Q}| \le f(n)$ for some $f(n) = \exp(\tilde{\Omega}(n))$, independent of the size of the data universe). In Section 7.5.3, we will see different lower-bound techniques that can yield this $\sqrt{\log|\mathcal{X}|}$ factor.

Let us now turn to the concrete families of counting queries from Section 7.1.3:

- **Point functions ($\mathcal{Q}^{\mathbf{pt}}$):** Here $\mathsf{PDisc}(\mathcal{Q}_S) = 1$ for every S (since all the sets are of size 1), so we do not obtain any interesting lower bound.
- **Threshold functions ($\mathcal{Q}^{\mathbf{thr}}$):** Here also $\mathsf{PDisc}(\mathcal{Q}_S) = 1$ for every S, because if we write $S = \{s_1 < s_2 < \cdots < s_t\}$ and color s_j according to the parity of j, every subset of S defined by a threshold function (i.e., every prefix of S) has imbalance at most 1.
- **Attribute means on $\{0,1\}^d$ ($\mathcal{Q}^{\mathbf{means}}(d)$):** Here we can analyze $\mathsf{PDisc}(\mathcal{Q}_S)$ for a uniformly random subset $S \subseteq \{0,1\}^d$ of size $s = \min\{n, d\}$. Then $\mathcal{Q}_S$ is statistically close to a uniformly random $\{0,1\}$ matrix of size $d \times s$, which by Lemma 7.5.7, has partial discrepancy $\Omega\left(\sqrt{s \cdot (1 + \log(d/s))}\right)$ with high probability. So when $d < n$, we have an error lower bound of $\Omega\left(\sqrt{d}/n\right)$, which is nearly tight, matching the upper bound of Theorem 7.2.7 up to a factor of $\sqrt{\log(1/\delta) \cdot \log\log d}/\varepsilon$. But when $d > n$, the lower bound is no better than $\Omega\left(\sqrt{(\log d)/n}\right)$, which leaves quite a large gap from the upper bound, which remains $O\left(\sqrt{d \cdot \log(1/\delta)\log\log d}/\varepsilon\right)$. In particular, the upper bound is useless when $d = \omega(n^2)$, but the lower bound leaves open the possibility of having vanishingly small error for any $d = 2^{o(n)}$.
- **t-way conjunctions on $\{0,1\}^d$ ($\mathcal{Q}_t^{\mathbf{conj}}(d)$):** The VC dimension of this class is at least $t \cdot \lfloor \log(d/t) \rfloor$, so we have an error lower bound of $\Omega(\min\{t \log(d/t)/n, 1\})$. For $t = O(1)$, Kasiviswanathan et al. [66] showed that, for the subset $T \subset \mathcal{Q}_t^{\mathrm{conj}}(d)$ consisting of the $\binom{d}{t}$ monotone conjunctions (without negations), if we pick a random set S of size $\min\{n, d^t/\operatorname{polylog}(d)\}$, we have $\sigma_{\min}(\mathcal{Q}_S^T) \ge \Omega(d^{t/2}/\operatorname{polylog}(n))$ with high probability. Consequently, we have

$$\mathsf{PDisc}(\mathcal{Q}_S) \ge \frac{1}{10} \cdot \sqrt{\frac{|S|}{\binom{d}{t}}} \cdot \Omega\left(\frac{d^{t/2}}{\operatorname{polylog}(n)}\right) = \tilde{\Omega}\left(\sqrt{\min\{n, d^t\}}\right).$$

When $n > d^t$, we get an error bound of $\alpha \geq \tilde{\Omega}(d^{t/2})/n$, which is tight up to polylogarithmic factors, but when $n = o(d^t)$, we are again quite far from the upper bounds of Theorem 7.2.7.

- **All conjunctions on $\{0,1\}^d$ ($\mathcal{Q}^{\mathbf{conj}}(d)$):** The VC dimension of this class is at least d, yielding an error lower bound of $\Omega(\min\{d/n, 1\})$. Matoušek et al. [77] showed that the hereditary discrepancy of $\mathcal{Q} = \mathcal{Q}^{\text{conj}}(d)$ is $\tilde{\Theta}((2/\sqrt{3})^d)$ and thus the same is also true for the partial hereditary discrepancy (by Inequality (7.5)). To use Theorem 7.5.8 when $n < 2^{d-1}$, we can restrict attention to the first $d' = \lfloor \log_2 n \rfloor$ variables, and obtain

$$\max_{\substack{S \subseteq \mathcal{X}, |S| \leq 2n \\ |S| \text{ even}}} \mathsf{PDisc}(\mathcal{Q}_S) \geq \tilde{\Omega}\left(\min\left\{\left(\frac{2}{\sqrt{3}}\right)^d, \left(\frac{2}{\sqrt{3}}\right)^{d'}\right\}\right) \geq \tilde{\Omega}\left(\min\left\{2^{0.21d}, n^{0.21}\right\}\right).$$

This yields an error lower bound of

$$\alpha \geq \tilde{\Omega}\left(\min\left\{\frac{2^{0.21d}}{n}, \frac{1}{n^{0.79}}\right\}\right).$$

By the hereditary discrepancy upper bound (Theorem 7.5.10), there is an algorithm that achieves error $\alpha \leq \frac{\tilde{O}((2/\sqrt{3})^d) \cdot \sqrt{\log(1/\delta)}}{\varepsilon n} \approx \frac{2^{0.21d} \cdot \sqrt{\log(1/\delta)}}{\varepsilon n}$, so the bounds are nearly matching when $n \gg 2^{0.21d}$. But when $n = 2^{o(d)}$, the lower bound of $1/n^{0.79}$ is quite far from the upper bound of $O(d^{3/2}\sqrt{\log(1/\delta)}/\varepsilon n)^{1/2}$ given by private multiplicative weights (Theorem 7.4.3).

Table 7.4 summarizes these lower bounds and compares them with the upper bounds we have seen.

Table 7.4: Error bounds for specific query families under (ε, δ)-differential privacy on a data universe $\mathcal{X}$ of size $D = 2^d$ (e.g., $\mathcal{X} = \{0,1\}^d$ or $\mathcal{X} = \{1, 2, \ldots, D\}$). Lower bounds apply for $(1, 0.1)$-differential privacy.

Query family $\mathcal{Q}$	Upper bounds	Ref.	Lower bounds from Thm. 7.5.8
$\mathcal{Q}^{\text{means}}$	$O\left(\frac{\sqrt{d\log(1/\delta)\cdot\log\log d}}{\varepsilon n}\right)$	Thm. 7.2.7	$\Omega\left(\frac{\sqrt{d}}{n}\right)$ if $d \leq n$ $\Omega\left(\sqrt{\frac{1+\log(d/n)}{n}}\right)$ if $d > n$
$\mathcal{Q}_t^{\text{conj}}$, $t \ll d$	$O\left(\frac{d^{t/2}\cdot\sqrt{\log(1/\delta)\cdot\log\log d}}{\varepsilon n}\right)$ $O\left(\frac{t\log d\sqrt{d\log(1/\delta)}}{\varepsilon n}\right)^{1/2}$	Thm. 7.2.7 Thm. 7.4.3	$\min\left\{\frac{\tilde{\Omega}(d^{t/2})}{n}, \tilde{\Omega}\left(\frac{1}{\sqrt{n}}\right)\right\}$ if $t = O(1)$ $\Omega\left(\min\left\{\frac{t\log(d/t)}{n}, 1\right\}\right)$
$\mathcal{Q}^{\text{conj}}$	$\frac{\tilde{O}(2^{0.21d})}{n}$ $O\left(\frac{d^{3/2}\cdot\sqrt{\log(1/\delta)}}{\varepsilon n}\right)^{1/2}$	Thm. 7.5.10 Thm. 7.4.3	$\min\left\{\frac{\tilde{\Omega}(2^{0.21d})}{\varepsilon n}, \tilde{\Omega}\left(\frac{1}{n^{0.79}}\right)\right\}$ $\Omega\left(\min\left\{\frac{d}{n}, 1\right\}\right)$

7.5.2 Packing Lower Bounds

We will now see a geometric approach to lower bounds that often gives tight lower bounds on $(\varepsilon, 0)$-differential privacy, and can separate it from (ε, δ)-differential privacy. In particular, we will prove that answering k arbitrary counting queries with $(\varepsilon, 0)$-differential privacy requires an error of $\alpha \geq \Omega(k/\varepsilon n)$, whereas we saw in Theorem 7.2.7 that we can achieve error $O(\sqrt{k \cdot \log(1/\delta)}/\varepsilon n)$ with (ε, δ)-differential privacy.

The approach is not specific to counting queries, and can be applied to virtually any computational problem that we might try to solve with differential privacy. Suppose that, for every dataset $x \in \mathcal{X}^n$, we have a set $\mathcal{G}_x \subseteq \mathcal{Y}$ of outputs that are "good" for x. Then the lower bound says that, if we have a "large" collection of datasets x such that the sets $\mathcal{G}_x$ are disjoint, then any $(\varepsilon, 0)$-differentially private mechanism must fail to produce a good output with high probability on at least one of the datasets in this collection.

Theorem 7.5.13 (Packing lower bound [59, 10]). *Let $\mathcal{C} \subseteq \mathcal{X}^n$ be a collection of datasets all at Hamming distance at most m from some fixed dataset $x_0 \in \mathcal{X}^n$, and let $\{\mathcal{G}_x\}_{x \in \mathcal{C}}$ be a collection of* disjoint *subsets of $\mathcal{Y}$. If there is an (ε, δ)-differentially private mechanism $\mathcal{M} : \mathcal{X}^n \to \mathcal{Y}$ such that $\Pr[\mathcal{M}(x) \in \mathcal{G}_x] \geq p$ for every $x \in \mathcal{C}$, then*

$$\frac{1}{|\mathcal{C}|} \geq p \cdot \mathrm{e}^{-m \cdot \varepsilon} - \delta.$$

In particular, when $p = 1/2$ and $\delta = 0$, we have $|\mathcal{C}| \leq 2 \cdot \mathrm{e}^{m\varepsilon}$.

Proof: By group privacy (Lemma 7.2.2), for every $x \in \mathcal{C}$, we have

$$\Pr[\mathcal{M}(x_0) \in \mathcal{G}_x] \geq p \cdot \mathrm{e}^{-m\varepsilon} - m\delta.$$

Since the sets $\mathcal{G}_x$ are disjoint, we have

$$\begin{aligned} 1 &\geq \Pr\left[\mathcal{M}(x_0) \in \bigcup_{x \in \mathcal{C}} \mathcal{G}_x\right] \\ &= \sum_{x \in \mathcal{C}} \Pr\left[\mathcal{M}(x_0) \in \mathcal{G}_x\right] \\ &\geq |\mathcal{C}| \cdot (p \cdot \mathrm{e}^{-m\varepsilon} - m\delta). \end{aligned}$$

■

Note that, when $\delta = 0$, the theorem (setting $m = n$) says that we can only have roughly $\mathrm{e}^{\varepsilon n} \ll |\mathcal{X}|^n$ datasets on which a differentially private mechanism's behavior is really distinct.

But for $\delta > 0$, the theorem says nothing when $m > \ln(1/\delta)/\varepsilon$ (because $p \cdot \mathrm{e}^{-m\varepsilon} - m\delta < 0$). The reason is the use of group privacy (Lemma 7.2.2), which tells us nothing when considering datasets that are at distance larger than $\ln(1/\delta)/\varepsilon$.

Let us now see how packing implies a lower bound of $\Omega(\min\{\log|\mathcal{X}|, \log(1/\delta)\}/\varepsilon n)$ for nonredundant classes of counting queries, namely ones where all elements of the data universe are distinguishable by the queries.

Theorem 7.5.14 (Packing lower bound for nonredundant queries). *Let $\mathcal{Q} = \{q : \mathcal{X} \to \{0,1\}\}$ be any class of counting queries that distinguish all the elements of $\mathcal{X}$. That is, for all $w \neq w' \in \mathcal{X}$, there is a query $q \in \mathcal{Q}$ such that $q(w) \neq q(w')$. Suppose $\mathcal{M} : \mathcal{X}^n \to \mathbb{R}^{\mathcal{Q}}$ is an (ε, δ)-differentially private mechanism that with high probability answers every query in $\mathcal{Q}$ with error at most α. Then*

$$\alpha \geq \min\left\{\Omega\left(\frac{\log|\mathcal{X}|}{\varepsilon n}\right), \Omega\left(\frac{\log(1/\delta)}{\varepsilon n}\right), \frac{1}{2}\right\}.$$

Note that an error bound of $1/2$ is achievable by the trivial $(0,0)$-differentially private algorithm that answers $1/2$ for all queries.

The hypothesis holds for all of the concrete query families we have considered (point functions, threshold functions, attribute means, and t-way conjunctions). In particular, for the class of point functions $\mathcal{Q}^{\text{pt}}(\{0,1\}^d)$, the lower bound of $\alpha \geq \Omega(\min\{d/\varepsilon n, \log(1/\delta)/\varepsilon n\})$ is tight, matched by Proposition 7.2.8 and Theorem 7.3.5 (which algorithm is better depends on whether d or $\log(1/\delta)$ is larger). In particular, this shows that approximate differential privacy can achieve smaller error (namely $\tilde{O}(\sqrt{d}) \cdot \sqrt{\log(1/\delta)}/\varepsilon n$) than is possible with pure differential privacy when $\log(1/\delta) < d/\operatorname{polylog}(d)$.

For attribute means over $\{0,1\}^d$ (i.e., $\mathcal{Q}^{\text{means}}(d)$), we obtain a tight lower bound of $\Omega(d/\varepsilon n)$ when $\delta = 0$, which matches the upper bound for arbitrary sets of $k = d$ counting queries given by Theorem 7.2.6. By Theorem 7.2.7, approximate differential privacy can achieve asymptotically smaller error when $k > \log(1/\delta)$.

Proof: For a dataset $x \in \mathcal{X}^n$, let $\mathcal{G}_x$ be the closed ℓ_∞ ball of radius α around the vector $(q(x))_{q\in\mathcal{Q}}$. The assumption about $\mathcal{M}$ implies that, for every dataset $x \in \mathcal{X}^n$, we have $\Pr[\mathcal{M}(x) \in \mathcal{G}_x] \geq 1/2$.

We will now construct a set $\mathcal{C}$ of $|\mathcal{X}|$ datasets for which the $\mathcal{G}_x$'s are disjoint. Specifically, for each $w \in \mathcal{X}$, let $x(w) \in \mathcal{X}^n$ be the dataset whose first $m = \lfloor 2\alpha n + 1 \rfloor$ rows are all equal to w, and whose remaining $n - m$ rows are all equal to w_0 for a fixed element $w_0 \in \mathcal{X}$. We will take $\mathcal{C} = \{x(w) : w \in \mathcal{X}\}$. To see that $\mathcal{G}_{x(w)}$ and $\mathcal{G}_{x(w')}$ are disjoint for every $w \neq w'$, let q be a query such that $q(w) \neq q(w')$ (which exists by hypothesis). Then $|q(x(w)) - q(x(w'))| = m/n > 2\alpha$. The datasets in $\mathcal{C}$ are all at distance at most m from the dataset $x(w_0)$. Thus by Theorem 7.5.13, we deduce that

$$\frac{1}{|\mathcal{X}|} \geq e^{-\varepsilon m}/2 - \delta,$$

which implies that either $\delta \geq e^{-\varepsilon m}/4$, in which case $\alpha \geq \Omega(\ln(1/\delta)/\varepsilon n)$, or $1/|\mathcal{X}| \geq e^{-\varepsilon m}/4$, in which case $\alpha \geq \Omega(\log|\mathcal{X}|/\varepsilon n)$. ∎

Now, let us see how the packing lower bound can be applied to arbitrary sets $\mathcal{Q}$ of counting queries to obtain tight bounds on the *sample complexity*—how large n

needs to be to achieve an arbitrarily small, but constant error α—with the matching upper bound coming from an instantiation of the exponential mechanism.

To formalize this, let $\mathcal{X}$ be our data universe, and consider the $|\mathcal{X}|$ vectors in $\mathbb{R}^{\mathcal{Q}}$ corresponding to the tuples of answers that can be achieved on individual elements on $\mathcal{X}$; that is, for each $w \in \mathcal{X}$, let $a_w = (q(w))_{q \in \mathcal{Q}}$. Now, following Hardt and Talwar [59], we consider the convex body $K = \text{ConvexHull}(\{a_w : w \in \mathcal{X}\})$ that is the convex hull of all of these vectors. Notice that, for any dataset $x \in \mathcal{X}$, the tuple of answers on x is $a_x = (1/n) \sum_{i=1}^{n} a_{x_i} \in K$.

Define the *packing number* $P_\alpha(K)$ to be the largest number of points we can fit in K such that all the pairwise ℓ_∞ distances are greater than α. (That is, the closed ℓ_∞ balls of radius $\alpha/2$ centered at the points are disjoint. But we do not require that the balls themselves are entirely contained within K; this notion of packing is sometimes referred to as *metric entropy*.)

Theorem 7.5.15 (Packing characterization of sample complexity).

1. *For all sufficiently small $\beta > 0$, there is an $\alpha > 0$ such that the following holds for all sets $\mathcal{Q} = \{q : \mathcal{X} \to \{0,1\}\}$ of counting queries, $n \in \mathbb{N}$, and $\varepsilon \in (0,1)$: If $\mathcal{M} : \mathcal{X}^n \to \mathbb{R}^{\mathcal{Q}}$ is an $(\varepsilon, 0)$-differentially private mechanism that, on every dataset $x \in \mathcal{X}^n$, answers all of the queries in $\mathcal{Q}$ to within error at most α with high probability, then*
$$n \geq \frac{\log(P_\beta(K))}{\beta\varepsilon},$$
where K is the convex body corresponding to $\mathcal{Q}$ as defined above.
2. *For every $\alpha > 0$, there is a $\beta > 0$ such that the following holds for all sets $\mathcal{Q} = \{q : \mathcal{X} \to \{0,1\}\}$ of counting queries, $n \in \mathbb{N}$, and $\varepsilon \in (0,1)$: If*
$$n \geq \frac{\log(P_\beta(K))}{\beta\varepsilon},$$
where K is the convex body corresponding to $\mathcal{Q}$, then there is an $(\varepsilon, 0)$-differentially private mechanism that, on every dataset $x \in \mathcal{X}^n$, answers all of the queries in $\mathcal{Q}$ to within error at most α with high probability.

Thus, to achieve error $\alpha = o(1)$, it is necessary and sufficient to have $n = \omega(P_{o(1)}(K))$. The above theorem is based on ideas from [93, Lecture 6].[6]

Proof:

1. Let $M = P_\beta(K)$ and let $a_1, \ldots, a_M$ be the corresponding points in K, all at pairwise ℓ_∞ distance greater than β.
   Our first step will be to approximate the points a_j by points $a_{y^{(j)}}$ for datasets of size $m = \beta n/2$, so that $\|a_j - a_{y^{(j)}}\|_\infty \leq \beta/3$. The definition of K tells us that, for

[6] In [93, Lecture 6], the bounds are stated in terms of the discrete set of points $K_n = \{a_x : x \in \mathcal{X}^n\}$ rather than the convex body K. An advantage of Theorem 7.5.15 is that the set K does not depend on n (since we are trying to characterize n in terms of it), but the formulation in [93] has the advantage of applying even to arbitrary low-sensitivity families (rather than just counting or statistical queries).

each point a_j there is a distribution D_j on $\mathcal{X}$ such that $a_j = \mathrm{E}_{w \leftarrow D_j}[a_w]$, where $a_w = (q(w))_{q \in \mathcal{Q}}$ is the vertex of K corresponding to the answers on $w \in \mathcal{X}$. We will probabilistically construct the dataset $y^{(j)} \in \mathcal{X}^m$ by randomly sampling m rows according to D_j. As mentioned in the proof of Theorem 7.4.1, if $m \geq O(\mathrm{VC}(\mathcal{Q}) \cdot \log(1/\beta)/\beta^2)$, then standard results in learning theory show that with high probability we have $\|a_j - a_{y^{(j)}}\|_\infty \leq \beta/3$, as desired. By Proposition 7.5.11 and Theorem 7.5.8, we know that $n \geq \Omega(\mathrm{VC}(\mathcal{Q})/\alpha)$ (for sufficiently small α), and thus $m = \beta n/2 \geq \Omega(\beta\,\mathrm{VC}(\mathcal{Q})/\alpha)$. Thus we can take α small enough (depending on β), to ensure that we have $m \geq O(\mathrm{VC}(\mathcal{Q}) \cdot \log(1/\beta)/\beta^2)$ as needed.

Given the datasets $y^{(j)} \in \mathcal{X}^m$, observe that the points $a_{y^{(j)}}$ are at pairwise distance greater than $\beta - 2\beta/3 = \beta/3$ (by the triangle inequality). Now we construct datasets $x^{(j)} \in \mathcal{X}^n$ of size n by padding the $y^{(j)}$'s with $n - m$ copies of a fixed row w from $\mathcal{X}$; the points $a_{x^{(j)}}$ are now at pairwise distance greater than $(m/n) \cdot (\beta/3) = \beta^2/6$. So if for every $x \in \mathcal{X}^n$, we take the set $\mathcal{G}_x$ to be a closed ℓ_∞ ball of radius $\beta^2/12$, then the sets $\{\mathcal{G}_{x^{(j)}}\}_{1 \leq j \leq M}$ are disjoint. Moreover we can take $\alpha \leq \beta^2/12$, and then the α-accuracy hypothesis on $\mathcal{M}$ says that, for every $x \in \mathcal{X}^n$, $\Pr[\mathcal{M}(x) \in \mathcal{G}_x] \geq 1/2$.

So all the conditions of Theorem 7.5.13 are satisfied (with $p = 1/2$, $\delta = 0$) and we obtain

$$2^{(\log \mathrm{e}) \cdot (\beta n/2) \cdot \varepsilon} = \mathrm{e}^{m \cdot \varepsilon} \geq \frac{M}{2} \geq M^{(\log \mathrm{e})/2},$$

where the latter inequality uses $M \geq 1/(2\beta) \geq 2^{3.6} \geq 2^{1/(1-(\log \mathrm{e})/2)}$ for any $\mathcal{Q}$ containing a nonconstant query and sufficiently small β. This implies that $n \geq \log(P_\beta(K)/\beta\varepsilon$, as desired.

2. Let $M = P_\beta(K)$, and let $a_1, \ldots, a_M$ be the corresponding points in K all at pairwise distance greater than β from each other. By the maximality of the packing, every point in K is at ℓ_∞ distance at most β from at least one of the a_i's (otherwise we could add the point to obtain a larger packing).[7] On a dataset $x \in \mathcal{X}^n$, we will use the exponential mechanism (Proposition 7.4.2) to sample a point a_j that is close to a_x in ℓ_∞ distance, in a manner similar to Theorem 7.4.1. Specifically,

$$\mathcal{M}(x) : \text{output } a_j \text{ with probability } \propto \mathrm{e}^{-\varepsilon n \cdot \|a_j - a_x\|_\infty}.$$

Indeed, Theorem 7.4.1 is a special case of this mechanism where we take the a_j's to be the answer vectors a_y that we get from small datasets $y \in \mathcal{X}^m$. By Proposition 7.4.2 (with $\mathrm{score}(x, a_j) = -\|a_j - a_x\|_\infty$), this mechanism is 2ε-differentially private, and achieves error at most $\beta + O(\log M)/\varepsilon n$ with high probability. Thus, if $n \geq (\log M)/\beta(2\varepsilon)$ and β is sufficiently small (depending on α), we obtain error at most α with high probability.

■

Note that there is a significant loss in the dependence on the error α in the proofs, so this theorem does not determine the rate at which we can get the error to decay

[7] In other words $\{a_1, \ldots, a_M\}$ form a *β-net* of K with respect to ℓ_∞ norm.

as a function of the other parameters (for example, whether we can get it to decay linearly in n or $\sqrt{n}$). If we work with ℓ_2 rather than ℓ_∞ error, then tighter characterizations of the rate of error decay are known (up to factors polylog($|\mathcal{Q}|, |\mathcal{X}|$)), by applying more sophisticated geometric methods to the convex body K [59, 11, 85].

7.5.3 Fingerprinting Lower Bounds

The lower bounds from Sections 7.5.1 and 7.5.2 above address two extreme ranges of δ. Reconstruction attacks prove lower bounds even for constant δ (e.g., $\delta = .1$), and packing (mainly) proves lower bounds for $\delta = 0$. Recall that, for satisfactory privacy guarantees, the desired range of δ is that it should be cryptographically negligible, i.e., $\delta = n^{-\omega(1)}$, as (ε, δ)-differential privacy allows for leaking each row with probability δ. In particular, when $\delta \geq 1/n$, we can output a subsample consisting of a δ fraction of the rows of the dataset, which in turns allows for answering any family $\mathcal{Q}$ of counting queries to within accuracy $\alpha = O\left(\sqrt{(\log |\mathcal{Q}|)/\delta n}\right)$ (by a Chernoff Bound). (When δ is constant, this matches the best lower bound we can get from discrepancy in the regime where $n \ll \min\{|\mathcal{Q}|, |\mathcal{X}|\}$, cf. Theorem 7.5.12.) Thus, to prove lower bounds of the form $\alpha = \Omega(1)$, we need to focus on the regime $\delta \leq O(\log |\mathcal{Q}|)/n$.

It turns out that a very well-suited tool for this task is *fingerprinting codes*, which were developed in the cryptography literature by Boneh and Shaw [15] for a completely different task. Specifically, they were designed for preventing piracy of digital content. Imagine a digital movie distribution company that wants to deliver copies of a movie to n different customers, and the company wants to mark each copy so that, if one of the customers or a coalition S of the customers released a pirated copy of the movie created from their own copies, the distribution company would be able to point a finger at one of the pirates in S. There are d scenes in the movie, and each of the scenes can be watermarked by either 0 or 1 (say by choosing one of two slightly different angles from which the movie was shot). The colluding pirates may splice their copies to evade detection. The fingerprinting code should help protect the movie by specifying for each scene and each customer whether it should be watermarked by 0 or 1. An associated tracing algorithm should determine one of the colluding pirates with high probability from the code and a pirated copy.

Definition 7.5.16 (Fingerprinting codes, syntax). *A* fingerprinting code *of length $d = d(n)$ for n users consists of two randomized algorithms:*

1. *A* generating algorithm Gen *that takes the number n of users and produces an $n \times d$ binary fingerprinting matrix C where $C_{i,j} \in \{0, 1\}$ determines the watermark of customer i in scene j along with a* tracing key *tk. (It turns out that without loss of generality we can take tk = C.)*
2. *A* tracing algorithm Trace *that takes as input the tracing key tk and watermarks $w \in \{0, 1\}^d$ from a potentially pirated movie and outputs an element of $[n] \cup \{\perp\}$ (which we interpret as an accused customer or "fail").*

For a generating matrix C and a coalition $S \subseteq \{1, \ldots, n\}$, we say that $w \in \{0, 1\}^d$ is *feasible for S* if, for every $j \in \{1, \ldots, d\}$, w_j equals to $c_{i,j}$ for some $i \in S$. Put

differently, if C_S, the submatrix of C consisting of the rows in S, is constant on value b_j on some column j, then we require that $w_j = b_j$. This captures the constraint that the coalition produces its pirated movie by splicing its copies together.

That is, a coalition S can deploy an arbitrary (randomized) pirating algorithm $\mathcal{P} : \{0,1\}^{|S|\times d} \to \{0,1\}^d$ that takes as its input C_S for a generating matrix C and produces a watermark sequence w that is feasible for S. (So we will require security even against pirates who are able to determine the watermarks in their movie copies.)

Definition 7.5.17 (Fingerprinting codes, security). *A fingerprinting code* $(\mathsf{Gen}, \mathsf{Trace})$ *is* secure *if, for every n, every $S \subseteq \{1,\ldots,n\}$ and every randomized pirating algorithm $\mathcal{P} : \{0,1\}^{|S|\times d} \to \{0,1\}^d$, we have*

$$\Pr_{\substack{C\leftarrow\mathsf{Gen}(1^n)\\ w\leftarrow P(C_S)}}[w \text{ is feasible for } C \text{ and } S, \text{ and } \mathsf{Trace}(C,w) \notin S] \le \mathrm{neg}(n).$$

(Recall that $\mathrm{neg}(n)$ *denotes a negligible probability, i.e., $n^{-\omega(1)}$.)*

An optimal construction of fingerprinting codes was given by Tardos [101]:

Theorem 7.5.18 (Optimal fingerprinting codes [101]). *For every n, there is a fingerprinting code of length $d = \tilde{O}(n^2)$ for n users.*

We will not prove this theorem, but will instead show a simpler but suboptimal construction from the original paper of Boneh and Shaw [15].

A fingerprinting code of length $\tilde{O}(n^3)$: $\mathsf{Gen}(1^n)$ outputs a matrix obtained by *randomly permuting* columns of the matrix

$$\begin{array}{ccccccc}
\text{0 block} & & \text{1st block} & \text{2nd block} & & \ldots & n\text{-th block}\\
\left(\begin{array}{c} \\ \\ \\ 0 \\ \\ \\ \end{array}\right. & & \begin{array}{c} 111\ldots111 \\ 000\ldots000 \\ \\ 0 \\ \\ \\ \end{array} & \begin{array}{c} 111\ldots111 \\ 111\ldots111 \\ 000\ldots000 \\ 0 \\ \\ \\ \end{array} & & \begin{array}{c} 111\ldots111 \\ 111\ldots111 \\ 111\ldots111 \\ \ldots \\ 000\ldots000 \\ \\ \end{array} & \left.\begin{array}{c} \\ \\ \\ 1 \\ \\ \\ \end{array}\right)
\end{array}$$

Each block spans $\tilde{O}(n^2)$ identical columns. For such a randomly generated matrix, a coalition S that does not include the i-th user cannot distinguish columns that come from the $(i-1)$-th and the i-th blocks of the matrix, as these columns are identical in the submatrix C_S. The tracing algorithm takes advantage of this observation. The tracing algorithm $\mathsf{Trace}(C,w)$ outputs the first i such that

$$\operatorname*{Avg}_{j \text{ in block } i}[w_j] - \operatorname*{Avg}_{j \text{ in block } i-1}[w_j] \ge \frac{1}{n},$$

where $\mathrm{Avg}_{j\in T} f(j)$ denotes the average of $f(j)$ over j in set T. For a feasible codeword w, such an index i is guaranteed to exist since $\mathrm{Avg}_{j \text{ in block } 0}[w_j] = 0$ and

$\text{Avg}_{j \text{ in block } n}[w_j] = 1$. The correctness of the tracing algorithm follows from the following claim, which ensures that the probability we falsely accuse a user outside the coalition S is negligible:

Claim 7.5.19. *For a given coalition S and pirate $\mathcal{P}$, a randomly generated $C \leftarrow \mathsf{Gen}(1^n)$ and $w \leftarrow \mathcal{P}(C_S)$, with probability greater than $1 - \text{neg}(n)$, for all $i \notin S$, we have*

$$\underset{j \text{ in block } i}{\text{Avg}}[w_j] - \underset{j \text{ in block } i-1}{\text{Avg}}[w_j] < \frac{1}{n}.$$

Proof: Fix $i \notin S$, and condition on the codeword $w \leftarrow \mathcal{P}(C_S)$. Since columns from block i and $i-1$ are identical in C_S, it is still not determined which permuted columns are from block i and which are from block $i-1$. More precisely, if we condition additionally on the entire submatrix C_S of the (permuted) codebook C as well as the permuted locations of all columns other than those from blocks i and $i-1$, then the blocks i and $i-1$ are still a uniformly random partition of their union into two equal-sized sets. The averages $\text{Avg}_{j \text{ in block } i}[w_j]$ and $\text{Avg}_{j \text{ in block } i-1}[w_j]$ have the same expectation over the choice of the partition (namely $\text{Avg}_{j \text{ in block } i \text{ or } i-1}[w_j]$). Since each is the average over $\tilde{O}(n^2)$ coordinates (selected without replacement from the union), Chernoff-type bounds imply that, with all but negligible probability (depending on the choice of the polylog(n) factor in the $\tilde{O}(\cdot)$), they will each deviate from the expectation by less than $1/2n$ (and hence will differ from each other by less than $1/n$). ∎

While the analysis of optimal fingerprinting codes, with $d = \tilde{O}(n^2)$, is more involved, the description of the codes is very simple. Following generalizations and simplifications given in Dwork et al. [47], for every $j \in [d]$, we can pick a bias $p_j \leftarrow [0,1]$ uniformly at random, and then generate the j-th column as n independent samples from the Bernoulli distribution with expectation p_j. In fact, any sufficiently "smooth" and "spread out" distribution on the p_j's can be used.

Now, we will use fingerprinting codes to derive lower bounds on differential privacy, following Bun et al. [21]:

Theorem 7.5.20 (Fingerprinting codes ⇒ for attribute means [21]). *If there is a fingerprinting code with codewords of length d for $n+1$ users then there is no $(1, 1/10n)$-differentially private mechanism $\mathcal{M} : (\{0,1\}^d)^n \to [0,1]^d$ for answering all d attribute means (i.e., the counting queries $\mathcal{Q}^{means}(d)$) with error $\alpha < 1/2$.*

Proof: Suppose for contradiction that there exists a $(1, 1/10n)$-differentially private mechanism $\mathcal{M}$ for answering attribute means with error $\alpha < 1/2$. Without loss of generality, we may assume that, for every dataset x, the output distribution of $\mathcal{M}(x)$ does not depend on the order of the rows of x (else $\mathcal{M}$ can randomly permute them first).

Use the hypothesized fingerprinting code to generate a (random) codebook C for $n+1$ users. Let $S = \{1, \ldots, n\}$ (i.e., the coalition consisting of all users except user $n+1$). Let $(a_1, \ldots, a_d)$ be attribute means obtained from $\mathcal{M}$ on the data set C_S. Define a vector $w \in \{0,1\}^d$ by rounding vector $(a_1, \ldots, a_d)$ to the nearest integer. Since $\mathcal{M}$ makes error less than $1/2$ (with high probability), w is a feasible pirated

codeword for C_S. That is, we think of $\mathcal{P}(\cdot) = \text{Round}(\mathcal{M}(\cdot))$ as the pirate for the fingerprinting code. Since $\mathcal{M}$ is differentially private, so is $\mathcal{P}$.

By the properties of the fingerprinting code

$$\Pr[\mathsf{Trace}(tk, \mathcal{P}(C_S)) \in \{1, \ldots, n\}] \geq 1 - \text{neg}(n),$$

where the probability is taken over $(C, tk) \leftarrow \mathsf{Gen}(1^{n+1})$ and the coin tosses of $\mathcal{P}$.

Hence, for n large enough, there is an i^* such that

$$\Pr[\mathsf{Trace}(tk, \mathcal{P}(C_S)) = i^*] \geq \frac{1}{2n}.$$

Let $S' = \{1, \ldots, n+1\} - \{i^*\}$. Since C_S and $C_{S'}$ are neighboring datasets (after an appropriate permutation of the rows), the differential privacy of $\mathcal{P}$ tells us that

$$\Pr[\mathsf{Trace}(tk, \mathcal{P}(C_S)) = i^*] \leq \mathrm{e}^1 \cdot \Pr[\mathsf{Trace}(tk, \mathcal{P}(C_{S'})) = i^*] + \frac{1}{10n}.$$

Thus, we have

$$\Pr[\mathsf{Trace}(tk, \mathcal{P}(C_{S'})) = i^*] \geq \frac{1}{2en} - \frac{1}{10en} \geq \Omega(1/n),$$

which contradicts the security of the fingerprinting code, as with nonnegligible probability we are accusing someone not in the coalition S'. ∎

Notice that the "good guys" and "bad guys" have switched roles in this relation between fingerprinting codes and differential privacy. The mechanism $\mathcal{M}$, which is supposed to protect privacy, plays the role of the adversarial pirate $\mathcal{P}$ for the fingerprinting code. And the Trace algorithm from the fingerprinting code (corresponding to the "authorities") plays the role of the privacy adversary. Tracing attacks (determining whether an individual was in the dataset or not) are not quite as devastating as the reconstruction attacks, but they still can be quite significant—for example, if the dataset consists of a collection of individuals who were all diagnosed with a particular disease. Indeed such tracing attacks (on releases of exact rather than approximate statistics) led the US National Institutes of Health to remove online access to summary statistics of certain genomic datasets [63, 110]. For a fingerprinting code to give a "realistic" attack, the tracing should not require extensive auxiliary information (captured by the tracing key tk) and should be fairly robust to the distribution according to which the codebook was generated. These issues are explored in [47].

Combining Theorems 7.5.18 and 7.5.20, we see that estimating d attribute means on a dataset of size $n = \tilde{\Omega}(\sqrt{d})$ requires an error of $\alpha \geq 1/2$ for $(1, 1/10n)$-differential privacy. Simple reductions imply that, in general, we need error $\alpha > \tilde{\Omega}(\sqrt{d})/\varepsilon n$. Steinke and Ullman [99] have tightened the lower bound to nearly match Theorem 7.2.7 (up to a factor of $O\left(\sqrt{\log \log d}\right)$):

Theorem 7.5.21 (Fingerprinting lower bound for attribute means [99]). *The following holds for every $d \in \mathbb{N}$, $\varepsilon \in (0, 1)$, and $\delta \in (2^{-d}, 1/n^{1.1})$. Suppose*

$\mathcal{M} : (\{0,1\}^d)^n \to [0,1]^d$ is an (ε, δ)-differentially private mechanism that with high probability answers every attribute mean query in $\mathcal{Q}^{means}(d)$ with error at most α. Then

$$\alpha \geq \Omega\left(\min\left\{\frac{\sqrt{d\log(1/\delta)}}{\varepsilon n}, 1\right\}\right).$$

Recall from Table 7.4 that partial discrepancy gave a lower bound of $\Omega(\sqrt{d}/n)$ when $d < n$, and otherwise gave a lower bound no better than $\sqrt{(\log d)/n}$. Packing (Theorem 7.5.14) gave a lower bound of $\Omega(\min\{d, \log(1/\delta)\}/\varepsilon n)$. Theorem 7.5.21 subsumes all of these bounds.

The fingerprinting lower bound above is for a particular family of counting queries—attribute means—in which the number of queries ($|\mathcal{Q}^{\text{means}}(d)| = d$) is logarithmic in the size of the data universe ($\mathcal{X} = \{0,1\}^d$), but it can be composed with reconstruction attacks of Section 7.5.1 to also yield nearly tight lower bounds for the case in which the number $|\mathcal{Q}|$ of queries is much larger:

Theorem 7.5.22 (Lower bounds for arbitrary counting queries [21]). *For every $d, n, k \in \mathbb{N}$ such that $n^{2.1} \leq k \leq 2^{2^{d/3}}$, there is a family $\mathcal{Q}$ of k counting queries on data universe $\mathcal{X} = \{0,1\}^d$ such that the following holds: If $\mathcal{M} : (\mathcal{X})^n \to \mathbb{R}^{\mathcal{Q}}$ is an $(\varepsilon, 1/10n)$ differentially private mechanism that with high probability answers all queries in $\mathcal{Q}$ within error at most α, then*

$$\alpha \geq \tilde{\Omega}\left(\frac{\sqrt{\log|\mathcal{X}|} \cdot \log(|\mathcal{Q}|)}{\varepsilon n}\right)^{1/2}.$$

This theorem mostly closes the gap between the largest discrepancy-based lower bounds (Theorem 7.5.12) and the upper bound given by private multiplicative weights (Theorem 7.4.3). So, we have a nearly tight understanding of the accuracy with which we can answer a worst-case set $\mathcal{Q}$ of counting queries, as a function of $|\mathcal{X}|$, $|\mathcal{Q}|$, n, and the privacy parameters. In fact, a similar lower bound is also known for the special case of t-way marginals, by composing the fingerprinting lower bound for attribute means with reconstruction lower bounds for marginals [14, 66, 29]:

Theorem 7.5.23 (Lower bound for t-way marginals [21]). *For every constant $\ell \in \mathbb{N}$, the following holds for all $d, n, t \in \mathbb{N}$ such that $n \leq d^{2\ell/3}/\varepsilon$ and $\ell + 1 \leq t \leq d$: If $\mathcal{M} : (\{0,1\}^d)^n \to \mathbb{R}^{\mathcal{Q}_t^{conj}(d)}$ is an $(\varepsilon, 1/10n)$-differentially private mechanism that with high probability answers all queries in $\mathcal{Q}_t^{conj}(d)$ to within error at most α, then*

$$\alpha \geq \min\left\{\tilde{\Omega}\left(\frac{t\sqrt{d}}{\varepsilon n}\right)^{1/2}, \Omega(1)\right\}.$$

However, as we have seen for point functions (Proposition 7.2.8 and Theorem 7.3.5), for some families of queries $\mathcal{Q}$, one can do much better than these bounds. Ideally, we would understand the best accuracy achievable in terms of the

combinatorial structure of the query family, similarly to what the hereditary dfiscrepancy bounds (Theorems 7.5.9 and 7.5.10) give, but for a given value of n and ideally without extra polylog($|\mathcal{Q}|$) factors.

Open Problem 7.5.24. For an arbitrary family $\mathcal{Q} = \{q : \mathcal{X} \to \{0,1\}\}$ of counting queries, $n \in \mathbb{N}$, $\varepsilon > 0$, and $\delta = o(1/n)$, characterize (to within "small" approximation factors) the smallest achievable error by (ε, δ)-differentially private mechanisms $\mathcal{M} : \mathcal{X}^n \to \mathbb{R}^{\mathcal{Q}}$.

A potentially easier task, advocated by Beimel et al. [10], is to characterize the "sample complexity" for constant error, as we did for pure differential privacy in Theorem 7.5.15:

Open Problem 7.5.25. For an arbitrary family $\mathcal{Q} = \{q : \mathcal{X} \to \{0,1\}\}$ of counting queries, $\varepsilon > 0$, and $\delta = o(1/n)$, characterize (to within "small" approximation factors) the sample complexity (i.e., smallest value of n) needed by (ε, δ)-differentially private mechanisms $\mathcal{M} : \mathcal{X}^n \to \mathbb{R}^{\mathcal{Q}}$ to answer all the queries in $\mathcal{Q}$ to within an arbitrarily small constant error $\alpha > 0$.

We note that there is a partial converse to the connections between fingerprinting codes and differential privacy [21]; that is, if answering a set $\mathcal{Q}$ of counting queries is impossible with differential privacy for a given set of parameters $(\alpha, n, \varepsilon, \delta)$, this implies a weak form of a fingerprinting code that is defined with respect to the query family $\mathcal{Q}$ and the given parameters. It would be very interesting to tighten this relationship; this would be one approach to Open Problems 7.5.24 and 7.5.25.

Open Problem 7.5.26. Identify a variant of fingerprinting codes whose existence is *equivalent* to the impossibility of answering a family $\mathcal{Q}$ accurately with differential privacy (up to some loss in parameters).

7.6 Computational Lower Bounds

Now we turn to *computational* lower bounds, giving evidence that some tasks that are information-theoretically possible with differential privacy are nevertheless computationally intractable. Specifically, recall that both the smallDB and private multiplicative weights algorithms of Section 7.4 can accurately answer (many) more than n^2 counting queries over data universe $\mathcal{X} = \{0,1\}^d$ with differential privacy, provided that n is large enough compared with d (e.g., $n \geq d^2$), but use computation time exponential in d. Below we will see evidence that this exponential computation is necessary in the worst case.

7.6.1 Traitor-Tracing Lower Bounds

Our first hardness results will be based on *traitor-tracing schemes*, which were introduced by Chor et al. [28] as a cryptographic tool for preventing piracy of digital content, like fingerprinting codes. Their benefit over fingerprinting codes is that they

allow for distributing an unbounded amount of content over a broadcast channel (after a setup phase where private keys are sent to the users). The price is having computational rather than information-theoretic security. The notion of traitor-tracing schemes predated the notion of fingerprinting codes, and their application to lower bounds for differential privacy also came first, in Dwork et al. [40].

To motivate the definition of traitor-tracing schemes, imagine a video-streaming company that distributes software or hardware that is capable of decoding their (encrypted) streaming signal. Each customer gets his own decryption program that has a unique decryption key, so that copying can be detected. However, we are also concerned that S customers might collude to create (and sell) unauthorized pirate decryption programs. They can build their pirate program using the decryption keys found in their own decryption program in an arbitrary way, so we may not be able to explicitly read off any of the keys from the pirate program. The goal of the traitor-tracing scheme is to be able to identify at least one of the colluding customers who contributed his decryption key. We can formalize this setup as follows:

Definition 7.6.1. *A* traitor-tracing scheme *consists of four algorithms (*$\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{Trace}$*) as follows:*

1. *The (randomized)* key generation algorithm $\mathsf{Gen}(1^d, 1^n)$ *takes as input* $1^d, 1^n$*, where d is a security parameter and n is a number of customers, and outputs* $(k_1, \ldots, k_n, bk, tk)$*, where* $k_i \in \{0,1\}^d$ *is the* decryption key *for user i, bk is the* broadcast key*, and tk is the* tracing key.
2. *The (randomized)* encryption algorithm $\mathsf{Enc}_{bk}(m)$ *takes as input the broadcast key bk and a message* $m \in \{0,1\}$ *and outputs a ciphertext c.*
3. *The* decryption algorithm $\mathsf{Dec}_{k_i}(c)$ *takes as input a user key* k_i *and a ciphertext c and outputs a message* $m \in \{0,1\}$*. We require that it always holds that* $\mathsf{Dec}_{k_i}(\mathsf{Enc}_{bk}(m)) = m$ *for keys* (k_i, bk) *that are output by* Gen*.*
4. *The syntax of the (randomized)* tracing algorithm Trace *will be described below (as there are two variants).*

We will consider two different scenarios for tracing, depending on the type of pirates that we wish to trace and the access that Trace has to those pirates. Each will give us different types of lower bounds for differential privacy.

Stateless pirates Here the tracer can run the pirate decryption program many times from its same initial state, but on different ciphertexts as input. For example, this models the scenario where the pirate decryption program is a piece of software whose code is given to the tracer. We want to be able to trace given any pirate program that is correctly able to decrypt proper encryptions with high probability (though the tracer will feed the pirate malformed ciphertexts that are neither encryptions of 0 or 1 to help in identifying one of the colluders). This is the original and most standard notion of traitor tracing in the literature.

Stateful but cooperative pirates Here the tracer can submit a sequence of ciphertexts to the pirate, but the pirate may answer them in a correlated fashion, for example, changing its behavior to evade tracing if it receives and detects

a malformed ciphertext. However, we will only require tracing for "cooperative" pirates, which still correctly distinguish encryptions of 0 from 1 even if they receive some other malformed ciphertexts. Tracing stateful pirates is well-motivated for traitor tracing; the "cooperativeness" condition is less natural in that context, but arises naturally in our application to differential privacy lower bounds.

We now formalize these two requirements.

Definition 7.6.2 (Tracing stateless pirates). *A traitor-tracing scheme (*$\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{Trace}$*) is* secure against stateless pirates *if the following holds for every* $n = \mathrm{poly}(d)$ *and every* $S \subseteq [n]$*: let* $\mathcal{P}$ *be a probabilistic* $\mathrm{poly}(d)$*-time algorithm that given the keys* $(k_i)_{i \in S}$ *outputs a Boolean circuit* $\tilde{P}$*. Then,*

$$\Pr[\mathsf{Trace}(\tilde{P}, tk) \notin S \text{ and } \tilde{P} \text{ is a useful decryptor}] \leq \mathrm{neg}(d),$$

where the probabilities are taken over $(k_1, \ldots, k_n, bk, tk) \leftarrow \mathsf{Gen}(1^d, 1^n)$*,* $\tilde{P} \leftarrow \mathcal{P}((k_i)_{i \in S})$*, and the coins of* Trace *and* $\mathcal{P}$*. The condition that* $\tilde{P}$ *is a useful decryptor means that, for every* $m \in \{0,1\}$*,* $\Pr[\tilde{P}(\mathsf{Enc}_{bk}(m)) = m] = 1$*, where the probability is taken over the coin tosses of* Enc*. (In the literature, tracing is typically required even for pirates that have just a nonnegligible advantage in distinguishing encryptions of 0 from encryptions of 1, but tracing pirate decoders that always decrypt correctly will suffice for our purposes.)*

Definition 7.6.3 (Tracing stateful pirates). *A traitor-tracing scheme (*$\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{Trace}$*) is* secure against stateful but cooperative pirates *if there is a polynomial function* $k(\cdot,\cdot)$ *(called the* tracing query complexity*) such that, for every* $n = \mathrm{poly}(d)$ *and every* $S \subseteq [n]$*, the following holds for* $k = k(d,n)$*: Let* $\mathcal{P}$ *be any probabilistic* $\mathrm{poly}(d)$*-time algorithm that, given the keys* $(k_i)_{i \in S}$ *and a sequence* $(c_1, \ldots, c_k)$ *of ciphertexts, outputs a sequence* $(m_1, \ldots, m_k) \in \{0,1\}^k$*. Then,*

$$\Pr[\mathsf{Trace}^{\mathcal{P}((k_i)_{i \in S}, \cdot)}(tk) \notin S \text{ and } \mathcal{P} \text{ cooperates}] \leq \mathrm{neg}(d),$$

where the probabilities are taken over $(k_1, \ldots, k_n, bk, tk) \leftarrow \mathsf{Gen}(1^d, 1^n)$ *and the coins of* Trace*. We require that* Trace *makes only one query* $(c_1, \ldots, c_k)$ *to* $\mathcal{P}$ *(amounting to feeding* $k = k(d,n)$ *nonadaptively chosen ciphertexts to* $\mathcal{P}$*), and say that* $\mathcal{P}$ cooperates *if, for every coordinate* j *where* c_j *is in the support of* $\mathsf{Enc}_{bk}(b_j)$ *for some* $b_j \in \{0,1\}$*, we have* $b_j = m_j$*.*

We note that tracing stateless pirates is easier than tracing stateful but cooperative pirates, because whenever $\tilde{P}$ is a useful decryptor, using it to decrypt each ciphertext will qualify as cooperating.

Theorem 7.6.4 (Traitor-tracing schemes against stateful pirates [28, 103]). *Assuming one-way functions exist, there exists a traitor-tracing scheme secure against stateful but cooperative pirates with tracing query complexity* $k(n,d) = \tilde{O}(n^2)$*.*

Proof sketch: The key generation, encryption, and decryption are as in the original construction of Chor et al. [28] (which was for stateless pirates). Fix a secure private-key encryption system $(\mathsf{Enc}^0, \mathsf{Dec}^0)$ (which exists if one-way functions exist). $\mathsf{Gen}(1^d, 1^n)$ generates independently keys $k_1, \ldots, k_n$ for the encryption system $(\mathsf{Enc}^0, \mathsf{Dec}^0)$ and sets $tk = bk = (k_1, k_2, \ldots, k_n)$. Encoding is given by

$$\mathsf{Enc}_{bk}(b) = (\mathsf{Enc}^0_{k_1}(b), \mathsf{Enc}^0_{k_2}(b), \ldots, \mathsf{Enc}^0_{k_n}(b))$$

and decryption for user i by

$$\mathsf{Dec}_{k_i}(c_1, \ldots, c_n) = \mathsf{Dec}^0_{k_i}(c_i).$$

The tracing algorithm is from Ullman [103], and utilizes fingerprinting codes in order to minimize the tracing query complexity and handle stateful but cooperative pirates. $\mathsf{Trace}^P(tk, bk)$ first generates a fingerprinting codebook, namely an $n \times k$ matrix $C \leftarrow \mathsf{Gen}_{\text{f.p.}}(1^n)$. (Recall from Theorem 7.5.18 that we can take $k = \tilde{O}(n^2)$.) It then creates ciphertexts $c^{(1)}, c^{(2)}, \ldots, c^{(k)}$ by

$$c_i^{(j)} = \mathsf{Enc}^0_{k_i}(C_{i,j}).$$

The tracing algorithm queries its oracle $P((k_i)_{i \in S}, c^{(1)}, c^{(2)}, \ldots, c^{(k)})$ to get answers $w = (w_1, \ldots, w_k)$, and runs the tracing algorithm of the fingerprinting code $\mathsf{Trace}_{\text{f.p.}}(C, w)$ to get a suspect i. It outputs this i.

We sketch the correctness of this tracing scheme: if the pirate algorithm is computationally bounded, then it cannot learn any information about the messages encrypted by private keys of users not participating in S, so w essentially depends only on the rows of C in S. We now observe that w is feasible when $\mathcal{P}$ is cooperative, except with negligible probability. Indeed, if all entries of column j of C_S agree on value b_j, then to $\mathcal{P}$, $c^{(j)}$ is indistinguishable from a valid encryption of b_j, and hence $w_j = b_j$ with all but negligible probability. ∎

We now show that such traitor-tracing schemes imply the hardness of answering many counting queries with differential privacy, a result due to Ullman [103].

Theorem 7.6.5. (Tracing stateful pirates⇒hardness of answering many queries [103]). *If there exists a traitor-tracing scheme secure against stateful but cooperative pirates with tracing query complexity $k(d, n)$, then every $(1, 1/10n)$-differentially private mechanism for answering $k = k(n + 1, d)$ efficiently computable counting queries with error $\alpha < 1/2$ on datasets with n individuals from $\mathcal{X} = \{0, 1\}^d$ must run in time superpolynomial in d. Here the queries are given as* input *to the mechanism, as Boolean circuits of size* $\mathrm{poly}(n, d)$.

Proof sketch: Suppose $\mathcal{M}$ is a differentially private mechanism like in the statement of the theorem. We will show how to construct a pirate for the traitor-tracing scheme using $\mathcal{M}$ and conclude from the security of the scheme that $\mathcal{M}$ must have a runtime big enough to break the scheme.

Start by setting up the traitor-tracing scheme with $n + 1$ users and take a dataset x containing the keys of a coalition of n users obtained by removing one user at ran-

dom. We consider counting queries on this dataset given by ciphertext decryption: for a ciphertext c, the query q_c evaluates to $q_c(k_i) = \mathsf{Dec}_{k_i}(c)$, where we identify the row corresponding to the i-th user with its key k_i. Therefore, when query q_c is answered accurately by $\mathcal{M}$ on the dataset x we obtain an $\pm\alpha$-approximation a to the number of users in x whose key decrypts c to 1. If c is a valid encryption of a message $m \in \{0,1\}$, then $|a - m| \leq \alpha < 1/2$, so rounding a will equal m. With this notation, we define our pirate as follows:

$$\mathcal{P}((k_i)_{i\in S}, c^{(1)}, \ldots, c^{(k)}) = \mathrm{Round}(\mathcal{M}(x = (k_i)_{i\in S}, q_{c^{(1)}}, \ldots, q_{c^{(k)}})),$$

where $\mathrm{Round} : [0,1]^k \to \{0,1\}^k$ denotes componentwise rounding.

As discussed above, the accuracy of $\mathcal{M}$ implies that $\mathcal{P}$ is cooperative. On the other hand, the fact that $\mathcal{M}$ is differentially private implies that $\mathcal{P}$ is also differentially private. As in the proof of Theorem 7.5.20, tracing contradicts differential privacy. Thus, $\mathcal{P}$ must not be traceable, and hence must have superpolynomial running time. ∎

Combining the above two theorems we get:

Corollary 7.6.6 (Hardness of answering many counting queries). *Assume one-way functions exist. Then for every $n = \mathrm{poly}(d)$, there is no polynomial-time $(1, 1/10n)$-differentially private algorithm for answering more than $\tilde{O}(n^2)$ efficiently computable counting queries with error $\alpha < 1/2$ (given as Boolean circuits input to the mechanism) over data universe $\mathcal{X} = \{0,1\}^d$.*

This lower bound is nearly tight, in that we can answer $k = \tilde{\Omega}(n^2)$ efficiently computable counting queries in polynomial time with differential privacy using the Laplace mechanism and advanced composition (or Theorem 7.2.7).

Let us review the above proof's translation between objects in the traitor-tracing scheme and those in differential privacy:

$$\begin{aligned} \text{user keyspace } \{0,1\}^d &\mapsto \text{data universe } \mathcal{X} = \{0,1\}^d \\ \text{ciphertext } c &\mapsto \text{counting query } q_c(k) = \mathsf{Dec}_k(c) \\ \text{pirate } \mathcal{P} &\hookleftarrow \text{mechanism } \mathcal{M} \\ \text{tracing algorithm } \mathsf{Trace} &\mapsto \text{privacy adversary} \end{aligned}$$

In particular, mechanisms that take a sequence of counting queries as input and produce a vector of answers correspond very naturally to stateful but cooperative pirates. On the other hand, a common application of the algorithms of Section 7.4 is not to specify the queries as input, but rather to fix some large family of counting queries over data universe $\{0,1\}^d$ (for example, the family of 3^d conjunction queries) and then take n large enough so that we can produce a compact representation of the answers to all of these queries (e.g., a synthetic dataset). What does this translate to in the traitor-tracing world? Since we are interested in a family $\mathcal{Q}$ of efficiently computable counting queries, we ideally should have ciphertexts that are of length $\mathrm{poly}(d)$ (so that the queries have polynomial description length), not growing linearly with n as in Theorem 7.6.4. Second, the pirate P should no longer directly

produce answers to the queries (i.e., decrypt ciphertexts), but rather it should use its keys $(k_i)_{i \in S}$ to produce a summary (which we can view as an algorithm or data structure) $\tilde{P}$ that can then be used to estimate the answer to any query in the class (i.e., decrypt any properly generated ciphertext). This leads us naturally to traitor tracing with stateless pirates, as used in the original connection of Dwork et al. [40]:

Theorem 7.6.7 (Tracing stateless pirates ⇒ hardness of differentially private summaries [40]). *If there is a traitor-tracing scheme secure against stateful pirates with ciphertexts of length $\ell(n, d)$, then for every d and $n = \text{poly}(d)$, there is a family $\mathcal{Q}$ of efficiently computable counting queries of description length $\ell(n + 1, d)$ (and size $2^{\ell(n+1,d)}$) over data universe $\{0,1\}^d$, such that no polynomial-time $(1, 1/10n)$-differentially private mechanism can accurately summarize the answers to all of the queries in $\mathcal{Q}$ on datasets of size n.*

We note that this theorem is only interesting if $\ell \ll n$. Indeed, Theorem 7.5.2 shows that there is a family of 2^n efficiently computable counting queries over a data universe of size $2n$ that is information-theoretically impossible to answer accurately with differential privacy. So we need traitor-tracing schemes with ciphertext length that is smaller than n, the number of users, unlike in the construction of Theorem 7.6.4. At the time that Theorem 7.6.7 was proven, the best known construction of traitor-tracing schemes against stateless pirates had ciphertext length $\ell(n, d) = \sqrt{n} \cdot \text{poly}(d)$ [17] (under hardness assumptions about bilinear maps on certain groups), and this already implied an interesting hardness result for differential privacy. But it left open the possibility that producing differentially private summaries is possible for any efficiently computable family $\mathcal{Q}$ of counting queries provided that $n \geq (\log |\mathcal{X}|) \cdot (\log |\mathcal{Q}|)^2$.

Recently, however, there are candidate constructions of traitor-tracing schemes with ciphertext length $\ell = \text{poly}(d)$, independent of n, assuming the existence of one-way functions and either "secure multilinear maps" or "indistinguishability obfuscation" [51, 16]. This yields a family $\mathcal{Q}$ of $2^\ell = 2^{\text{poly}(d)}$ counting queries over a data universe $\mathcal{X}$ of size 2^d for which no $\text{poly}(d)$-time algorithm can produce an accurate differentially private summary (for any $n = \text{poly}(d)$). More recently, Kowalczyk et al. [72] showed that the same hardness result holds when either $|\mathcal{Q}|$ or $|\mathcal{X}|$ is $\text{poly}(n)$, by constructing traitor-tracing schemes where either the ciphertexts or the keys are of length $O(\log n)$, albeit with a weaker security property that still suffices to show hardness of differential privacy. Specifically, the theorem says:

Theorem 7.6.8 (iO ⇒ hardness of differential privacy [72]). *Assuming the existence of indistinguishability obfuscation and one-way functions:*

1. *For every $d \in \mathcal{N}$ and every $n = \text{poly}(d)$, there is a family $\mathcal{Q}$ of $O(n^7)$ efficiently computable counting queries over data universe $\mathcal{X} = \{0,1\}^d$ (specified by a uniform $\text{poly}(d)$-time evaluation algorithm that takes an ℓ-bit description of a query q, for $\ell = 7 \log n + O(1)$, and an input $y \in \{0,1\}^d$ and outputs $q(y)$) such that no polynomial-time differentially private mechanism can accurately answer all of the queries in $\mathcal{Q}$ on datasets of size n.*

2. *For every $\ell \in \mathcal{N}$ and every $n = \text{poly}(\ell)$, there is a family $\mathcal{Q}$ of 2^ℓ efficiently computable counting queries over data universe $\mathcal{X} = \{0,1\}^d$ for $d = 7 \log n + O(1)$ (specified by a uniform $\text{poly}(\ell)$-time evaluation algorithm that takes an ℓ-bit description of a query q and an input $y \in \{0,1\}^d$ and outputs $q(y)$) such that no polynomial-time differentially private mechanism can accurately summarize the answers to all of the queries in $\mathcal{Q}$ on datasets of size n.*

We note that, when $|\mathcal{Q}|$ and $|\mathcal{X}|$ are *both* of size $\text{poly}(n)$, the algorithm of Theorem 7.4.3 can answer all of the queries in polynomial time (so we cannot hope to prove hardness in this case). If, in part 1, the $|\mathcal{Q}|$ could be reduced to $n^{2+o(1)}$, then the hardness result would be stronger than that of Corollary 7.6.6 (albeit under a stronger complexity assumption). Indeed, here the set of queries is fixed and each query is described by $O(\log n)$ bits, whereas in Corollary 7.6.6, the queries have description length larger than n and need to be provided as input to the mechanism. It would also be interesting to reduce $|\mathcal{X}|$ to $n^{2+o(1)}$ in part 2; this too would be optimal because, when $|\mathcal{X}| \leq n^{2-\Omega(1)}$, the Laplace histogram is a $\text{poly}(n)$-time computable summary that is simultaneously accurate for up to $2^{n^{\Omega(1)}}$ queries (Theorem 7.2.9).

Open Problem 7.6.9. Can either $|\mathcal{Q}|$ or $|\mathcal{X}|$ in Theorem 7.6.8 be reduced to $n^{2+o(1)}$?

The existence of "indistinguishability obfuscation", as assumed in Theorem 7.6.8, is still very unclear, and thus it would be significant to replace it with a more well-understood complexity assumption:

Open Problem 7.6.10. Can a hardness result like Theorem 7.6.8 be established under a more standard and widely believed complexity assumption? This is open even for the case where we do not require either $|\mathcal{Q}|$ or $|\mathcal{X}|$ to be of size $\text{poly}(n)$, but rather we allow n and the mechanism running time to be $\text{poly}(d, \ell)$.

Similarly to (but earlier than) the case with fingerprinting codes, there is a partial converse to the connection between traitor-tracing schemes and the hardness of differential privacy [40], and it would be very interesting to tighten this relationship.

Open Problem 7.6.11. Identify a variant of traitor-tracing schemes whose existence is *equivalent* to the hardness of answering (or summarizing) counting queries with differential privacy (up to some loss in parameters, but ideally having a relationship holding per-query family $\mathcal{Q}$).

7.6.2 Lower Bounds for Synthetic Data

The lower bounds of the previous section provide families of efficiently computable counting queries that are hard to answer with differential privacy. However, these families consist of rather complicated functions that evaluate cryptographic algorithms (namely, the decryption algorithm for traitor-tracing schemes). We do not know similar results for simple/natural function classes of interest, such as the set of all 3^d conjunctions on data universe $\{0,1\}^d$.

However, we can prove a hardness result for differentially private algorithms that work by producing a synthetic dataset, as do the algorithms of Section 7.4. (This is explicitly stated for the smallDB algorithm, and the private multiplicative weights algorithm can be modified to produce synthetic data.) In fact, the result will hold even for the family $\mathcal{Q}_2^{\text{conj}}$ of 2-way marginals.

Theorem 7.6.12 (Hardness of synthetic data for simple queries [104]). *Assuming one-way functions exist, there exists a constant $\alpha > 0$ such that there is no $n = \text{poly}(d)$ and polynomial-time $(1, 1/10n)$-differentially private mechanism that given a dataset with n individuals over $\mathcal{X} = \{0,1\}^d$ outputs a synthetic dataset approximating all the counting queries in $\mathcal{Q}_2^{conj}(d)$ (i.e., all the 2-way marginals) to within additive error at most α.*

We note that the requirement that the mechanism produces a synthetic dataset cannot be removed from the theorem. Indeed, recall that the Laplace mechanism and advanced composition will approximate all $k = \Theta(d^2)$ 2-way conjunctions within error $\alpha = \tilde{O}(\sqrt{k})/\varepsilon n = \tilde{O}(d)/\varepsilon n$ in time $\text{poly}(n, d)$. So for $n = \text{poly}(d)$, we get vanishingly small error in polynomial time.

Proof: The main ingredients in the proof are digital signature schemes and probabilistically checkable proofs (PCPs). We will use digital signatures to construct datasets for which it is hard to generate synthetic data that preserves the answer to a cryptographically defined query, and then we will use PCPs to transform this cryptographic query into a collection of 2-way conjunctions.

Recall that a *digital signature scheme* is given by a triple of polynomial-time algorithms as follows:

1. A randomized *key generation* algorithm $\mathsf{Gen}(1^d) = (pk, sk)$ that produces a public key pk and a private key sk given a security parameter d as input.
2. A randomized *signing* algorithm that, given a message $m \in \{0,1\}^d$ and a secret key sk, produces a signature $\sigma = \mathsf{Sign}_{sk}(m) \in \{0,1\}^d$.
3. A deterministic *verification* algorithm $\mathsf{Ver}_{pk}(m, \sigma)$ that always accepts a signature for m generated using the secret key sk corresponding to pk.

Informally, we say that the scheme is *secure* if, given access to examples $(m_i, \sigma_i = \mathsf{Sign}_{sk}(m_i))$ signed with the same secret key, any algorithm running in time $\text{poly}(d)$ cannot generate a new message $m' \notin \{m_i\}$ and a signature σ' such that $\mathsf{Ver}_{pk}(m', \sigma') = 1$.

We now describe how to use digital signatures to construct datasets for which it is hard to generate synthetic data preserving the answer to a cryptographically defined counting query. This construction is due to Dwork et al. [40]:

The dataset: Generate $(pk, sk) \leftarrow \mathsf{Gen}(1^d)$ and construct a dataset x with n individuals, where each row contains a pair (m_i, σ_i) with m_i selected uniformly at random from $\{0,1\}^d$ and $\sigma_i \leftarrow \mathsf{Sign}_{sk}(m_i)$.

The query: Consider the counting query $q(\cdot) = \mathsf{Ver}_{pk}(\cdot)$. This query is efficiently computable and evaluates to 1 on the whole dataset.

The hardness: Now suppose for contradiction that there exists a polynomial-time differentially private mechanism $\mathcal{M}$ that given x produces a synthetic dataset $\hat{x} \in (\{0,1\}^d)^{\hat{n}}$ which is accurate with respect to q with high probability. By accuracy, $\hat{x}$ must contain at least one row $\hat{x}_j = (\hat{m}_j, \hat{\sigma}_j)$ such that $\mathsf{Ver}_{pk}(\hat{m}_j, \hat{\sigma}_j) = q(\hat{x}_j) = 1$. To derive a contradiction, we consider two cases:

- If $\hat{m}_j \notin x$, then $\mathcal{M}$ succeeded in creating a forgery for the signature scheme in polynomial time, contradicting its security.
- If $\hat{m}_j \in x$, then $\mathcal{M}$ intuitively has violated privacy, as it has copied part of a row (which is independent from all other rows) entirely in the output. More precisely, for every $i \in [n]$, the probability that an (ε, δ)-differentially private mechanism $\mathcal{M}$ outputs m_i is at most $e^{\varepsilon}/2^d + \delta$, since it could output m_i with probability at most $1/2^d$ if we replaced the i-th row with all zeroes. Thus, the probability $\mathcal{M}$ outputs any m_i is at most $n \cdot (e^{\varepsilon}/2^d + \delta) < 1/20$ for $\varepsilon = 1$ and $\delta = 1/10n$.

We now describe how to use PCPs to replace the cryptographic query Ver_{pk} with 2-way conjunctions. Actually, we will only describe how to get a result for 3-way conjunctions, as it uses a more familiar type of PCP theorem.

Recall that *Circuit SAT* is an NP-hard problem. Then, by a strong form of the PCP theorem there exist a constant $\alpha > 0$ and three polynomial time algorithms Red, Enc, Dec satisfying the following:

1. Red is a randomized reduction that, given a circuit C, outputs a 3-CNF formula $\mathsf{Red}(C) = \phi = \phi_1 \wedge \ldots \wedge \phi_m$ such that if C is satisfiable then ϕ is satisfiable, and otherwise there is no assignment satisfying more than $(1-\alpha)m$ clauses of ϕ.
2. If w is a satisfying assignment for C, then $z = \mathsf{Enc}(C, w)$ is a satisfying assignment for ϕ.
3. If z is an assignment for ϕ satisfying more than $(1-\alpha)m$ clauses, then $w = \mathsf{Dec}(C, z)$ is a satisfying assignment for C.

Item 1 is the standard formulation of the PCP theorem in terms of the hardness of approximating MAX-3SAT; it asserts a Karp reduction from Circuit SAT to the promise problem Gap-MAX-3SAT. Items 2 and 3 are saying that this reduction is actually a Levin reduction, meaning we can efficiently transform witnesses between the Circuit SAT instance and the corresponding Gap-MAX-3SAT instance.

Here is our modified construction:

The dataset: Let x be the dataset constructed above using digital signatures. We write z for the dataset with n individuals obtained by encoding each row x_i of x with the encoding algorithm given by the PCP theorem, relative to the circuit $C = \mathsf{Ver}_{pk}$. That is, $z_i = \mathsf{Enc}(\mathsf{Ver}_{pk}, x_i)$.

The queries: Our set of queries is all 3-way conjunctions, but we will only exploit accuracy with respect to the clauses of the 3-CNF formula $\phi = \phi_1 \wedge \cdots \wedge \phi_m$ output by $\mathsf{Red}(\mathsf{Ver}_{pk})$. Note that for every row z_i in z we have $\phi(z_i) = 1$ (since $\mathsf{Ver}_{pk}(x_i) = 1$), so for every clause ϕ_j in ϕ we have $\phi_j(z) = n^{-1} \sum_{i \in [n]} \phi_j(z_i) = 1$.

The hardness: Suppose for contradiction that $\mathcal{M}$ is a polynomial-time differentially private mechanism that produces synthetic datasets that are α-accurate

with respect to 3-way conjunctions and let $\hat{z} = \mathcal{M}(z)$. Then for every $j \in [m]$ we have $\phi_j(\hat{z}) \geq 1 - \alpha$. By averaging, this implies that there exists some row $\hat{z}_i$ of $\hat{z}$ that satisfies at least $(1 - \alpha) \cdot m$ clauses from ϕ. Therefore, using this row from the sanitized dataset we can obtain $(\hat{m}, \hat{\sigma}) = \mathsf{Dec}(\mathsf{Ver}_{pk}, \hat{z})$ such that $\mathsf{Ver}_{pk}(\hat{m}, \hat{\sigma}) = 1$. Now the same argument used earlier shows that either $(\hat{m}, \hat{\sigma})$ is a forgery (in case $\hat{m} \notin x$) or a violation of privacy (in case $\hat{m} \in x$).

■

The hardness results we have seen either apply to contrived (cryptographic) queries (Corollary 7.6.6 and Theorem 7.6.8) or constrain the form of the mechanism's output to synthetic data (Theorem 7.6.12). Obtaining a hardness result for *any* "natural" family of queries without restricting the form of the mechanism's output remains an intriguing open problem.

Open Problem 7.6.13. Give evidence of hardness of accurately answering any "natural" family of counting queries under differential privacy, without constraining the form of the mechanism's output.

At the same time, the lack of such a hardness result should provide some hope in looking for algorithms, and suggests that we should look for output representations other than synthetic data. We can gain hope from computational learning theory, where proper learning (where the learner's output is constrained to come from the same representation class as the concept it is learning) is often computationally harder than unconstrained, improper learning. Indeed, we will see the benefits of moving beyond synthetic data for conjunctions in the next section.

7.7 Efficient Algorithms for Specific Query Families

In this section, we will see that, for some specific, natural families of queries, one can in fact obtain efficient algorithms for answering more than n^2 queries.

7.7.1 Point Functions (Histograms)

We have already seen that, for the class $\mathcal{Q}^{\text{pt}} = \mathcal{Q}^{\text{pt}}(\mathcal{X})$ of point functions on $\mathcal{X}$, we can achieve a better accuracy–privacy tradeoff than is possible with an arbitrary class $\mathcal{Q}$ of efficiently computable queries. Indeed, Proposition 7.2.8 and Theorems 7.3.5 and 7.5.14 show that the optimal error achievable for $\mathcal{Q}^{\text{pt}}(\mathcal{X})$ is $\Theta(\min\{\log|\mathcal{X}|, \log(1/\delta), \varepsilon n\}/\varepsilon n)$, whereas for an arbitrary query family with $|\mathcal{Q}| = |\mathcal{X}|$, there is a lower bound of $\Omega((\log|\mathcal{X}|)^{3/2}/\varepsilon n)^{1/2}$ for a wide range of parameters (Theorem 7.5.22).

Now we will see that in fact the optimal algorithms for point functions can be implemented in polynomial time, and can be modified to generate synthetic data.

Theorem 7.7.1 (Point functions with differential privacy [2]). *For every data universe $\mathcal{X}$, $n \in \mathbb{N}$, and $\varepsilon, \delta > 0$ such that $\delta < 1/n$, there is a* $\text{poly}(n, \log|\mathcal{X}|)$*-time*

(ε, δ)-differentially private algorithm that takes a dataset of n rows from data universe $\mathcal{X} = \{0,1\}^d$ and outputs a synthetic dataset approximating the value of all counting queries in $\mathcal{Q}^{pt}(\mathcal{X})$ up to an additive error of

$$\alpha = O\left(\min\left\{\frac{\log|\mathcal{X}|}{\varepsilon n}, \frac{\log(1/\delta)}{\varepsilon n}, 1\right\}\right)$$

with high probability.

Proof sketch: The stability-based histogram of Theorem 7.3.5 with error $O(\log(1/\delta)/\varepsilon n)$ already runs in polynomial time, as it outputs nonzero values only for points that occur in the dataset. However, the basic Laplace-based histogram of Proposition 7.2.8 adds noise $\mathrm{Lap}(2/\varepsilon)$ to the value of all $|\mathcal{X}| = 2^d$ point functions, and thus does not run in polynomial time. Thus, to obtain a polynomial-time algorithm with error $\alpha = O(\log|\mathcal{X}|/\varepsilon n)$, first we consider a modification of the Laplace-based histogram algorithm that only uses the largest $O(1/\alpha)$ noisy fractional counts and treats the rest as zero. This modification maintains differential privacy by closure under postprocessing, and can be shown to maintain error $O(\log|\mathcal{X}|/\varepsilon n)$. (Note that there can only be at most $1/\beta$ points whose exact fractional counts are at least $\beta = \Omega(\alpha)$, and outputting zero for the remaining points introduces an error of at most β.) With this modification, to implement the mechanism efficiently, we can first add (discrete) Laplace noise to the $m \le n$ point functions q_y for the points y that occur at least once in the dataset, and then sample the distribution of the top $\lceil 1/\alpha \rceil$ values of $|\mathcal{X}| - m$ discrete $\mathrm{Lap}(2/\varepsilon)$ random variables. Sampling the latter distribution to within sufficient accuracy to maintain differential privacy (with some additional modifications to the mechanism) can be done in time $\mathrm{poly}(\log|\mathcal{X}|, 1/\varepsilon, \lceil 1/\alpha \rceil) = \mathrm{poly}(n, \log|\mathcal{X}|)$.

To obtain synthetic data in both cases, we can simply use the noisy answers to determine how many copies of each point to put in the synthetic dataset. With a synthetic dataset of size $O(1/\alpha)$, the errors due to rounding will only increase the error by a constant factor. ∎

7.7.2 Threshold Functions (CDFs)

For the class of threshold functions $\mathcal{Q}^{\mathrm{thr}}([2^d])$ on domain $[2^d]$, for pure differential privacy ($\delta = 0$), again the best possible accuracy is $\Theta(d/\varepsilon n)$, matching the lower bound of Theorem 7.5.14, and it can be achieved in polynomial time:

Theorem 7.7.2 (Thresholds with pure differential privacy [41, 45]). *For every $n, d \in \mathbb{N}$, $\varepsilon > 0$, there is a $\mathrm{poly}(n,d)$-time $(\varepsilon, 0)$-differentially private algorithm that takes a dataset of n rows from data universe $\mathcal{X} = [2^d]$ and outputs a synthetic dataset maintaining the value of all threshold-function counting queries up to an error of*

$$\alpha = \max\left\{\frac{O(d)}{\varepsilon n}, \tilde{O}\left(\frac{1}{\varepsilon n}\right)\right\}$$

with high probability.

Interestingly, in the case of approximate differential privacy, there is an inherent dependence on $\log^* d$ in the error.

Theorem 7.7.3 (Thresholds with approximate differential privacy [9, 22]). *For every $n, d \in \mathbb{N}$, $\varepsilon, \delta > 0$ such that $\exp(-\varepsilon n/\log^* n) \leq \delta \leq 1/n^2$:*

1. *There is a $\mathrm{poly}(n, d)$-time (ε, δ)-differentially private algorithm that takes a dataset of n rows from data universe $\mathcal{X} = [2^d]$ and outputs a synthetic dataset maintaining the value of all threshold-function counting queries up to an error of*
$$\alpha = \max\left\{\frac{2^{(1+o(1))\log^* d} \cdot \log(1/\delta)}{\varepsilon n}, \tilde{O}\left(\frac{1}{\varepsilon n}\right)\right\}.$$
2. *Every (ε, δ)-differentially private algorithm for answering all threshold functions on datasets of n rows from data universe $\mathcal{X} = [2^d]$ must incur an error of at least*
$$\alpha = \Omega\left(\min\left\{\frac{(\log^* d) \cdot \log(1/\delta)}{\varepsilon n}, 1\right\}\right).$$

We will not cover the proofs of these results, except to note that the $\log^* d$ lower bound has a Ramsey-theoretic proof [18], raising the possibility that there is a more general Ramsey-theoretic combinatorial quantity that can help in characterizing the optimal accuracy or sample complexity for differentially private algorithms (Open Problems 7.5.24 and 7.5.25).

Note that our understanding of threshold functions is not as tight as for point functions, and it would be interesting to close the gap between the upper and lower bounds. In particular:

Open Problem 7.7.4. Does the optimal error for releasing threshold functions over $\mathcal{X} = [2^d]$ with approximate differential privacy grow linearly or exponentially with $\log^* d$, or something in between?

7.7.3 Conjunctions (Marginals)

Unlike point functions and thresholds, the class $\mathcal{Q}^{\mathrm{conj}}$ of conjunctions is unlikely to have a polynomial-time differentially private algorithm for generating synthetic data, by Theorem 7.6.12. This suggests that we should look to other ways of summarizing the answers to conjunction queries.

Indeed, we will sketch two algorithms that beat the barrier of Theorem 7.6.12 by avoiding synthetic data. One algorithm summarizes the answers to *all* conjunction queries in subexponential ($2^{\tilde{O}(\sqrt{d})}$) time (using a subexponential-sized dataset), using low-degree approximations to Boolean functions. (Assuming the existence of digital signature schemes with exponential security and nearly linear-time verification, the proof of Theorem 7.6.12 can be extended to show that generating synthetic data requires time at least $2^{d^{1-o(1)}}$, even when $n = 2^{d^{1-o(1)}}$.) The other algorithm answers all $k = \Theta(d^2)$ 2-way conjunctions in polynomial time with error $\tilde{O}(\sqrt{d})/\varepsilon n$, in particular allowing us to answer $k = \tilde{\Omega}(n^4) \gg n^2$ such queries, using ideas from convex geometry and optimization.

Theorem 7.7.5 (Marginals via low-degree approximation [102]). *There is a constant c such that for all $\varepsilon, \alpha > 0$, $d, n, t \in \mathbb{N}$ with $d \geq t$ and $n \geq d^{c\sqrt{t}\cdot\log(1/\alpha)}/\varepsilon$, there is an ε-differentially private algorithm running in time* $\mathrm{poly}(n)$ *that takes a dataset $x \in (\{0,1\}^d)^n$ and, with high probability, outputs a "summary" (say, as a Boolean circuit) that allows for approximating the answer to all the queries in $\mathcal{Q}_t^{conj}(d)$ to within additive error α.*

A more sophisticated algorithm from [26] reduces the amount of data needed to nearly optimal ($n = O(t \cdot d^{0.51})$) at the cost of a larger (but still slightly subexponential) running time of $2^{o(d)}$.

Proof sketch: Starting with our dataset x with n rows in $\mathcal{X} = \{0,1\}^d$, the mechanism $\mathcal{M}$ will produce a "summary" S that will approximate the function f_x defined as $f_x(q) = q(x)$. S will be a polynomial of low degree.

By introducing new variables for negative literals and negating our functions, it suffices to handle *monotone* t-way *disjunctions*, which can conveniently be specified by bit strings $y \in \{0,1\}^d$:

$$q_y(w) = \bigvee_{i:y_i=1} w_i\,, \qquad w \in \mathcal{X}\,. \tag{7.6}$$

For a t-way disjunction, y has Hamming weight t, and the value of $q_y(w)$ is determined by the value of $\sum_{i=1}^t w_i y_i \in \{0,\ldots,t\}$. Specifically

$$q_y(w) = \begin{cases} 1 & \sum_{i=1}^t w_i y_i \in \{1,\ldots,t\}, \\ 0 & \sum_{i=1}^t w_i y_i = 0. \end{cases} \tag{7.7}$$

Given a dataset x, we are interested in producing a (differentially private) approximation to the function $f_x(\cdot)$ defined as

$$f_x(y) = q_y(x) = \frac{1}{n}\sum_{i=1}^n q_y(x_i) = \frac{1}{n}\sum_{i=1}^n f_{x_i}(y).$$

We will approximate f_x by a low-degree polynomial by approximating each f_{x_i} by a low-degree polynomial. We do the latter using a standard technique based on Chebychev polynomials:

Fact 7.7.6 *For all $t \in \mathbb{N}$ and $\alpha > 0$, there exists a univariate (real) polynomial g of degree at most $s = O(\sqrt{t}\log(1/\alpha))$ such that $g(0) = 0$ and for all $i \in \{1,\ldots,t\}$, $1 - \alpha \leq g(i) \leq 1 + \alpha$. Moreover, g can be constructed in time* $\mathrm{poly}(t, \log(1/\alpha))$ *and all of the coefficients of g have magnitude at most 2^s.*

Given g as in the fact and a row $w \in \mathcal{X}$, consider the following function:

$$h_w(y) = g\left(\sum_{j=1}^d w_j y_j\right), \tag{7.8}$$

where g is from Fact 7.7.6. h_w is a multivariate polynomial of degree $O(\sqrt{t} \cdot \log(1/\alpha))$. It has at most $C = d^{O(\sqrt{t}\cdot\log(1/\alpha))}$ coefficients of magnitude at most $M = d^{O(\sqrt{t}\cdot\log(1/\alpha))}$.

By construction, we have that, for all $w \in \mathcal{X}$ and all $y \in \mathcal{X}$ of Hamming weight at most t,

$$|h_w(y) - f_w(y)| \leq \alpha\,.$$

Thus, if we define

$$h_x = \frac{1}{n}\sum_{i=1}^{n} h_{x_i},$$

we have that

$$|h_x(y) - f_x(y)| \leq \alpha.$$

To obtain differential privacy, we can now add Laplace noise to each coefficient of h_x. Each coefficient is an average of the corresponding coefficients of the h_{x_i}'s, so has global sensitivity at most $2M/n$. By the Laplace mechanism and basic composition, it suffices to add noise $\mathrm{Lap}(2MC/\varepsilon n)$ to each of the C coefficients for the resulting vector of coefficients to be differentially private. With high probability, none of the coefficients will have noise more than $(\log C) \cdot 2MC/\varepsilon n$, which will add up to an error of at most $C \cdot \log C \cdot 2MC/\varepsilon n = d^{O(\sqrt{t})}/(\varepsilon n)$ when evaluating on any input y. ■

Now we turn to a different approach, which runs in polynomial time and can answer nearly n^4 low-order marginals.

Theorem 7.7.7 (Marginals via SDP projection [46]). *Let $t \in \mathbb{N}$ be an even constant. For all $n, d \in \mathbb{N}$, $\varepsilon, \delta > 0$, there is a polynomial-time (ε, δ)-differentially private algorithm that takes a dataset $x \in (\{0,1\}^d)^n$ and answers all counting queries in $\mathcal{Q}_t^{conj}(d)$ on x to within additive error*

$$\alpha = \left(\tilde{O}(d^{t/4}) \cdot \sqrt{\log(1/\delta)}/\varepsilon n\right)^{1/2}.$$

The most interesting case of this theorem is $t = 2$, when the error is $(\tilde{O}(\sqrt{d}) \cdot \sqrt{\log(1/\delta)}/\varepsilon n)^{1/2}$, matching the lower bound of Theorem 7.5.23 up to a factor of $\mathrm{poly}(\log d, \log(1/\delta))$ [21].

Proof sketch: The starting point for the algorithm is a beautiful geometric approach of Nikolov, Talwar, and Zhang [85] that was used to prove the hereditary discrepancy upper bound (Theorem 7.5.10). We will use an instantiation of their algorithm that provides near-optimal error bounds in terms of $|\mathcal{Q}|$, like the private multiplicative weights algorithm, but for ℓ_2 or ℓ_1 error rather than ℓ_∞.

We know that adding independent noise of magnitude $O(\sqrt{|\mathcal{Q}|}/\varepsilon n)$ to the answers to all the counting queries in a family $\mathcal{Q}$ provides privacy, but gives useless results (that lie outside $[0,1]$) when $|\mathcal{Q}| > n^2$. Remarkably, it turns out that simply projecting these answers back to be consistent with *some* dataset yields highly accurate results.

To formalize this, recall the convex body K used in the packing characterization of sample complexity (Theorem 7.5.15). That is, $K = \text{ConvexHull}(\{a_w : w \in \mathcal{X}\})$, where $a_w = (q(w))_{q\in\mathcal{Q}}$ is the vector in $\mathbb{R}^{\mathcal{Q}}$ giving all the query answers on row $w \in \mathcal{X}$. Recall that, for every dataset $x \in \mathcal{X}$, the tuple of answers on x is $a_x = (1/n)\sum_{i=1}^n a_{x_i} \in K$.

This leads to the following algorithm $\mathcal{M}(x, \mathcal{Q})$:

1. Calculate the exact answers
$$y = a_x = (q(x))_{q\in\mathcal{Q}} \in K.$$
2. Add *Gaussian* noise to the coordinates of y:
$$\tilde{y} = y + \frac{O(\sqrt{|\mathcal{Q}| \cdot \log(1/\delta)})}{\varepsilon n} \cdot \mathcal{N}(0,1)^{|\mathcal{Q}|}.$$
(This can be shown to achieve (ε, δ)-differential privacy, and is more convenient than Laplace noise for the geometric arguments we are about to make.)
3. Project back to K: Let
$$\hat{y} = \text{argmin}_{z\in K} \|z - \tilde{y}\|_2.$$
This step maintains (ε, δ)-differential privacy by postprocessing.

Let us analyze the error introduced by this algorithm. Consider the line ℓ through y and $\hat{y}$, and let p be the orthogonal projection of $\tilde{y}$ onto ℓ. On ℓ, p must be on the ray from $\hat{y}$ to infinity. (If p were on the segment between y and $\hat{y}$, then p would be a point in K closer to $\tilde{y}$ than $\hat{y}$. If p were on the ray from y to infinity, then y would be a point in K closer to $\tilde{y}$ than $\hat{y}$.)

$$\begin{aligned}
\|y - \hat{y}\|_2^2 &= \langle \hat{y} - y, \hat{y} - y\rangle \\
&\le \langle \hat{y} - y, p - y\rangle && \text{(because } p \text{ is on the ray from } \hat{y} \text{ to infinity)} \\
&= \langle \hat{y} - y, \tilde{y} - y\rangle && \text{(because } \tilde{y} - p \text{ is orthogonal to } \hat{y} - y) \\
&\le (|\langle \hat{y}, \tilde{y} - y\rangle| + |\langle y, \tilde{y} - y\rangle|) && \text{(triangle inequality)} \\
&\le 2 \max_{z\in K} |\langle z, \tilde{y} - y\rangle|.
\end{aligned}$$

Taking expectations, and writing $\tilde{y} - y = O(\sqrt{|\mathcal{Q}| \cdot \log(1/\delta)}/\varepsilon n) \cdot g$ for $g \sim \mathcal{N}(0,1)^{|\mathcal{Q}|}$, we have
$$\mathrm{E}\left[\|y - \hat{y}\|_2^2\right] \le \frac{O\left(\sqrt{|\mathcal{Q}| \cdot \log(1/\delta)}\right)}{\varepsilon n} \cdot \mathrm{E}_g\left[\max_{z\in K} |\langle z, g\rangle|\right].$$

The quantity
$$\ell^*(K) \stackrel{\text{def}}{=} \mathrm{E}_g \max_{z\in K} |\langle z, g\rangle|$$
is known as the *Gaussian mean width* of the polytope K, an important and well-studied quantity in convex geometry.

Let us upper bound it for K defined by an arbitrary set $\mathcal{Q}$ of counting queries. For every choice of g, the maximum of $|\langle z, g\rangle|$ over $z \in K$ will be obtained at one of the vertices of K. Recalling the definition of K, we have

$$\max_{z \in K} |\langle z, g\rangle| = \max_{w \in \mathcal{X}} |\langle a_w, g\rangle|.$$

By rotational symmetry of Gaussians, the random variable $\langle a_w, g\rangle$ is distributed as $\mathcal{N}(0, \|a_w\|_2)$. We have $\|a_w\|_2 \leq \sqrt{|\mathcal{Q}|}$ since a_w is a $\{0,1\}$ vector. Thus, with probability at least $1 - \beta$ over g, we have $|\langle a_w, g\rangle| \leq O(\sqrt{|\mathcal{Q}| \cdot \log(1/\beta)})$. Taking a union bound over $w \in \mathcal{X}$, we have

$$\max_{w \in \mathcal{X}} |\langle a_w, g\rangle| \leq O\left(\sqrt{|\mathcal{Q}| \cdot \log(|\mathcal{X}|/\beta)}\right).$$

with probability at least $1 - \beta$, for every $\beta > 0$. This implies that

$$\mathop{\mathrm{E}}_{g}\left[\max_{z \in K} |\langle z, g\rangle|\right] = \mathop{\mathrm{E}}_{g}\left[\max_{w \in \mathcal{X}} |\langle a_w, g\rangle|\right] \leq O\left(\sqrt{|\mathcal{Q}| \cdot \log |\mathcal{X}|}\right).$$

Putting it all together, we have

$$\mathrm{E}\left[\|y - \hat{y}\|_2^2\right] \leq \frac{|\mathcal{Q}| \cdot O(\sqrt{\log |\mathcal{X}| \cdot \log(1/\delta)})}{\varepsilon n}.$$

So if we look at the average error (averaged over the $|\mathcal{Q}|$ queries), we have

$$\begin{aligned}
\mathop{\mathrm{E}}_{\text{coins of } \mathcal{M},\, q \in \mathcal{Q}}\left[\left|y_q - \hat{y}_q\right|\right] &\leq \left(\mathop{\mathrm{E}}_{\text{coins of } \mathcal{M},\, q \in \mathcal{Q}}\left|y_q - \hat{y}_q\right|^2\right)^{1/2} \\
&= \left(\mathop{\mathrm{E}}_{\text{coins of } \mathcal{M}}\left[\frac{1}{|\mathcal{Q}|} \cdot \|y - \hat{y}\|_2^2\right]\right)^{1/2} \\
&= O\left(\frac{\sqrt{\log(1/\delta)}}{\sqrt{|\mathcal{Q}|} \cdot \varepsilon n} \cdot \ell^*(K)\right)^{1/2} \\
&\leq O\left(\frac{\sqrt{\log |\mathcal{X}| \cdot \log(1/\delta)}}{\varepsilon n}\right)^{1/2}.
\end{aligned}$$

This exactly matches the (optimal) bound from the private multiplicative weights algorithm, except that we only achieve small error on average for a random query from $\mathcal{Q}$. However, it can be generalized to obtain small average-case error on any given distribution of queries (just weight the coordinates in $\mathbb{R}^{\mathcal{Q}}$ according to the distribution), and then combined with a differentially private algorithm for "boosting" [42] to obtain small error on all queries with high probability (paying a factor of polylog($|\mathcal{Q}|$) in the error).

Our interest in this algorithm, however, is that it does not appear to generate synthetic data, and thus is not subject to the computational complexity lower bounds of Theorem 7.6.12. Converting the output $\hat{y}$ to synthetic data would amount to de-

composing $\hat{y}$ into a convex combination of the $|\mathcal{X}|$ vertices of K, which could take time proportional to $|\mathcal{X}|$. Unfortunately, this same reason means that the "Project back to K" step might take time proportional to $|\mathcal{X}|$, as the given description of K is in terms of its $|\mathcal{X}|$ vertices. Indeed, projection onto a convex set is known to be polynomially equivalent to optimizing linear functions on the set, and as we will see below, optimizing over K is NP-hard for the cases we are interested in.

Let us see how to make this process more efficient for the case of 2-way marginals. For t-way marginals with $t > 2$, the theorem follows by reduction to 2-way marginals. (Create $\binom{d}{t/2} \leq d^{t/2}$ variables representing the conjunctions on every subset of $t/2$ variables; and then every t-way conjunction in the original variables can be written as a 2-way conjunction in the new variables.)

Actually, releasing conjunctions of width at most 2 is equivalent to releasing parities of width at most 2, so let us focus on the latter problem. It will also be useful to work in ± 1 notation, so the parity function $q_{ij} : \{\pm 1\}^d \to \{\pm 1\}$ on variables i and j is given by $q_{ij}(v) = v_i v_j$. Thus we see that

$$K = \text{ConvexHull}(\{v \otimes v : v \in \{\pm 1\}^d\}).$$

Unfortunately, projecting onto and optimizing over K is known to be NP-hard, so we will take a cue from approximation algorithms and look at a semidefinite programming relaxation.

It is NP-hard to do this optimally. So instead, we will find a nicer L "close" to K (where $K \subseteq L$) and optimize over L. We need to ensure that the Gaussian mean width of L is comparable to that of K (or at least the bound we used on the Gaussian mean width of K).

First, we will relax to:

$$L_0 = \text{ConvexHull}(\{v \otimes v' : v, v' \in \{\pm 1\}^d\}).$$

To bound the Gaussian mean width of K, we only used the fact that K is the convex hull of $|\mathcal{X}| = 2^d$ vectors whose entries have magnitude at most 1, and the bound was linear in $\sqrt{\log |\mathcal{X}|} = \sqrt{d}$. L_0 is now the convex hull of 2^{2d} such vectors, so we only lose a constant factor in our bound.

Optimizing over L_0 is still NP-hard, but it has polynomial-time approximation algorithms. Indeed, if we relax L_0 to

$$L = \{V \in \mathbb{R}^{d^2} : \exists \{u_i\}_{i=1}^d, \{u'_j\}_{j=1}^d \text{ unit vectors with } V_{ij} = \langle u_i, u'_j \rangle\},$$

then we can optimize linear functions on L by semidefinite programming, and consequently we can project onto L. Moreover, Grothendieck's inequality (see [71]) says that the maximum of any linear objective function on L is at most a factor of $K_G < 1.783$ larger than on L_0, which implies that

$$\ell^*(L) \leq K_G \cdot \ell^*(L_0) = O(\sqrt{|\mathcal{Q}| \cdot d}).$$

To summarize, the algorithm for the set $\mathcal{Q}$ of 2-way parities operates as follows:

1. Calculate the exact answers

$$y = a_x = (q(x))_{q \in \mathcal{Q}} \in K \subseteq \mathbb{R}^{d^2}.$$

2. Add *Gaussian* noise to the coordinates of y:

$$\tilde{y} = y + \frac{O(\sqrt{|\mathcal{Q}| \cdot \log(1/\delta)})}{\varepsilon n} \cdot \mathcal{N}(0,1)^{|\mathcal{Q}|}.$$

3. Project back to L: Let

$$\hat{y} = \text{argmin}_{z \in L} \|z - \tilde{y}\|_2.$$

By the analysis we did earlier, the average error per query we obtain is at most

$$\begin{aligned} \mathop{\mathrm{E}}_{\text{coins of } \mathcal{M},\, q \in \mathcal{Q}} \left[\left|y_q - \hat{y}_q\right|\right] &\leq O\left(\frac{\sqrt{\log(1/\delta)}}{\sqrt{|\mathcal{Q}|} \cdot \varepsilon n} \cdot \ell^*(L)\right)^{1/2} \\ &\leq O\left(\frac{\sqrt{d \cdot \log(1/\delta)}}{\varepsilon n}\right)^{1/2}, \end{aligned}$$

as desired. ∎

The theorems above show that we can bypass the intractability of producing differentially private summaries by focusing on specific, structured query classes, and by avoiding synthetic data. We summarize the state of knowledge about t-way marginals in Table 7.5. (Results for all marginals, i.e., $\mathcal{Q}^{\text{conj}}(d)$, roughly correspond to the case $t = d$, but in some cases will be off by a logarithmic factor, and we do not include the result based on the hereditary partial discrepancy of $\mathcal{Q}^{\text{conj}}(d)$ being $\tilde{\Theta}((2/\sqrt{3})^d)$ [77].)

As can be seen from the table, there are still important gaps in our state of knowledge, such as:

Open Problem 7.7.8. Is there a polynomial-time differentially private algorithm for estimating all (higher-order) marginals with vanishing error $\alpha = o(1)$ on a dataset with $n = \text{poly}(d)$ rows from data universe $\mathcal{X} = \{0,1\}^d$? Or at least all t-way marginals for some $t = \omega(1)$?

Open Problem 7.7.9. Is there a polynomial-time differentially private algorithm for estimating all 3-way marginals with vanishing error $\alpha = o(1)$ on a dataset with $n = o(d)$ rows from data universe $\mathcal{X} = \{0,1\}^d$?

Open Problem 7.7.10. For what other classes of queries can one bypass the intractability of generating differentially private synthetic data and answer more than n^2 queries with polynomial- or subexponential-time algorithms?

7.8 Private PAC Learning

We now examine the possibility of machine learning in Valiant's PAC model [106], under differential privacy. (See [70] for background on the PAC model.)

Table 7.5: Error bounds for $\mathcal{Q}_t^{\text{conj}}(d)$ when $t \ll d$ with (ε, δ)-differential privacy on a dataset of size n. Computational lower bounds hold under plausible cryptographic assumptions (e.g., exponentially secure digital signatures with linear-time verification). "Synth?" indicates whether the entry refers to algorithms that generate synthetic data.

Type	Bound	Constraints	Runtime	Synth?	Ref.
Upper	$O\left(\frac{d^{t/2}\cdot\sqrt{\log(1/\delta)\cdot\log\log d}}{\varepsilon n}\right)$		$\text{poly}(n, d^t)$	no	Thm. 7.2.7
Upper	$O\left(\frac{t\log d\sqrt{d\log(1/\delta)}}{\varepsilon n}\right)^{1/2}$		$\text{poly}(n, 2^d)$	yes	Thm. 7.4.3
Upper	α	$n \geq d^{c\sqrt{t}\cdot\log(1/\alpha)}/\varepsilon$	$\text{poly}(n)$	no	Thm. 7.7.5
Upper	$\left(\tilde{O}(d^{t/4})\cdot\sqrt{\log(1/\delta)}/\varepsilon n\right)^{1/2}$	t even	$\text{poly}(n, d^t)$	no	Thm. 7.7.7
Lower	$\min\left\{\frac{\tilde{\Omega}(d^{t/2})}{n}, \tilde{\Omega}\left(\frac{1}{\sqrt{n}}\right)\right\}$	$t = O(1)$	any	no	[66]
Lower	$\Omega\left(\min\left\{\frac{t\log(d/t)}{n}, 1\right\}\right)$		any	no	[14]
Lower	$\min\left\{\tilde{\Omega}\left(\frac{t\sqrt{d}}{\varepsilon n}\right)^{1/2}, \Omega(1)\right\}$	$n \leq d^{O(1)}/\varepsilon$	any	no	Thm. 7.5.23
Lower	$\Omega(1)$	$t \geq 2$	$\leq 2^{d^{1-o(1)}}$	yes	Thm. 7.6.12

7.8.1 PAC Learning vs. Private PAC Learning

Recall that PAC learning considers, for each input length d, two sets of functions:

- A concept class $\mathcal{C} = \mathcal{C}_d = \{c : \{0,1\}^d \to \{0,1\}\}$, from which the unknown concept c we are trying to learn comes.
- A hypothesis class $\mathcal{H} = \mathcal{H}_d = \{h : \{0,1\}^d \to \{0,1\}\}$, which contains the functions we will use to try to represent our learned approximation of c.

Definition 7.8.1 (PAC learning). *A concept class $\mathcal{C}$ is* PAC-learnable *if there exist an algorithm L (called the* learner*) and a number n polynomial in d (called the* sample complexity*) such that, for every distribution D on $\{0,1\}^d$ and every $c \in \mathcal{C}$, if we sample points $x_1, \ldots, x_n, x_{n+1}$ chosen independently according to D, with high probability $L(x_1, c(x_1), \cdots, x_n, c(x_n))$ returns a function $h \in \mathcal{H}$ such that $h(x_{n+1}) = c(x_{n+1})$.*

If $\mathcal{H} = \mathcal{C}$, we call L a proper learner *and say that $\mathcal{C}$ is* properly PAC-learnable*. If L is poly-time computable as are the functions in $\mathcal{H}$ (given a* $\text{poly}(d)$*-bit description of a function $h \in \mathcal{H}$ as output by L and an input $w \in \{0,1\}^d$, we can evaluate $h(d)$ in time* $\text{poly}(d)$*), then we say that L is an* efficient learner *and say that $\mathcal{C}$ is* efficiently PAC-learnable*.*

Definition 7.8.2 (Private PAC learning). Private PAC learning *is defined in the same way as PAC learning, but with the additional requirement that L is differentially private. That is, for all sequences $(x_1, y_1), \ldots, (x_n, y_n)$ and $(x'_1, y'_1), \ldots, (x'_n, y'_n)$ that differ in one coordinate $i \in [n]$, $L((x_1, y_1), \ldots, (x_n, y_n))$ and $L((x'_1, y'_1), \ldots, (x'_n, y'_n))$*

are (ε, δ)-indistinguishable for some constant ε (e.g., $\varepsilon = 1$) and δ negligible in n and d.

Taking ε to be a constant is without loss of generality due to a generic reduction for improving ε (increase the sample size by a factor of ε/ε', and run the original learner on random subsample of the dataset). The success probability of the learner can also be amplified via "boosting", which has a differentially private analogue [42].

Note that, while the definition of PAC learning only speaks of inputs that consist of i.i.d. samples from an unknown distribution that is consistent with some concept $c \in \mathcal{C}$, we require privacy on all (worst-case) pairs of neighboring input sequences. Indeed, if our modeling assumptions about the world are wrong, we naturally expect that our learner might fail, but we do not want the privacy promises to the data subjects to be broken. Also note that we consider the output of the learner to be the entire description of the hypothesis h, not just its prediction $h(x_{n+1})$ on the challenge point.

Amazingly, there is no gap between PAC learning and Private PAC learning, if we do not care about computation time:

Theorem 7.8.3 (Generic private learner [67]). *If $\mathcal{C}$ is (nonprivately) PAC-learnable (equivalently, $\mathrm{VC}(\mathcal{C}) \leq \mathrm{poly}(d)$), then it is privately and properly PAC-learnable with sample complexity $O(\log|\mathcal{C}|) \leq O(d \cdot \mathrm{VC}(\mathcal{C})) = \mathrm{poly}(d)$.*

The relation $\log|\mathcal{C}| \leq d \cdot \mathrm{VC}(\mathcal{C})$ is the Perles–Sauer–Shelah lemma. (See [70].)

Proof: We use the exponential mechanism (Proposition 7.4.2). Let $\mathcal{H} = \mathcal{C}$. On input $(x_1, y_1) \cdots (x_n, y_n)$, we

$$\text{output } h \in \mathcal{H} \text{ with probability} \propto \mathrm{e}^{-\varepsilon \cdot |\{i : h(x_i) \neq y_i\}|}.$$

Since $\mathrm{score}(x, h) = -|\{i : h(x_i) \neq y_i\}|$ has sensitivity 1 as a function of the dataset x, Proposition 7.4.2 tells us that this mechanism is 2ε-differentially private.

To prove that the learner succeeds with high probability, consider $x_1, \cdots, x_n$ that are taken according to some unknown distribution D, and let $y_i = c(x_i)$.

If $n \geq O(\mathrm{VC}(\mathcal{C}) \cdot \log(1/\alpha)/\alpha^2)$, then Occam's razor from learning theory (cf. [70]) tells us that with high probability over $x_1 \cdots x_n$, we have

$$\forall h \in \mathcal{C} \qquad \left| \frac{\#\{i : h(x_i) = c(x_i)\}}{n} - \Pr_{w \sim D}[h(w) = c(w)] \right| \leq \alpha.$$

Combining this with Proposition 7.4.2, we know that with high probability the hypothesis h we output satisfies

$$\begin{aligned}\Pr_{w\sim D}[h(w) = c(w)] &\geq \frac{\#\{i : h(x_i) = c(x_i)\}}{n} - \alpha \\ &\geq \frac{\text{argmax}_{h^*}\#\{i : h^*(x_i) = c(x_i)\} - O(\log|\mathcal{C}|)/\varepsilon}{n} - \alpha \\ &= \frac{n - O(\log|\mathcal{C}|)/\varepsilon}{n} - \alpha \\ &\geq 1 - 2\alpha,\end{aligned}$$

provided $n \geq O(\log|\mathcal{C}|)/\varepsilon\alpha$.

We are done when taking

$$n = O\left(\max\left\{\frac{\log|\mathcal{C}|}{\varepsilon\alpha}, \frac{\text{VC}(\mathcal{C})\cdot\log(1/\alpha)}{\alpha^2}\right\}\right) \ll 1.$$

■

7.8.2 Computationally Efficient Private PAC Learning

Unfortunately, as is often the case with the exponential mechanism, Theorem 7.8.3 does not produce computationally efficient private learners. Thus, we now investigate what can be learned in polynomial time under differential privacy.

Nonprivately, most examples of computationally efficient PAC learners are learners in the *statistical query model* of Kearns [69]. This is a model where the learner does not get direct access to labeled samples $(x_i, c(x_i))$, but is allowed to obtain additive approximations to the expectation of any (efficiently computable) function $f : \{0,1\}^d \times \{0,1\} \to [0,1]$ on the labeled distribution. That is, on specifying statistical query f, the learner obtains an answer in the range $\text{E}_{w\leftarrow D}[f(w, c(w))] \pm 1/\text{poly}(n)$. Efficient statistical query learners can be simulated by efficient PAC learners because expectations $\text{E}_{w\leftarrow D}[f(w, c(w))]$ can be estimated to within $\pm 1/\text{poly}(n)$ by taking the average of $f(x_i, c(x_i))$ over $m = \text{poly}(n)$ random samples $x_i \leftarrow D$. Such estimations are also easily done with differential privacy, as an average of $f(x_i, y_i)$ over m samples (x_i, y_i) has global sensitivity at most $2/m$ as a function of the dataset, and thus can be estimated via the Laplace mechanism. Thus, we have the following:

Theorem 7.8.4 (Private SQ learning [13]). *Every concept class that is efficiently PAC learnable in the statistical query model (which includes $\mathcal{Q}^{pt}$, $\mathcal{Q}^{thr}$, and $\mathcal{Q}^{conj}$) is efficiently and privately PAC learnable.*

In fact, Kasiviswanathan et al. [67] showed that (efficient) statistical query learners are *equivalent* to (efficient) private learners in the "local model" of privacy (which will be discussed more in the next section).

However, there are some concept classes that are efficiently PAC learnable that are provably not learnable in the statistical query model, most notably the class of parity functions, that is, the class of functions $\{0,1\}^d \to \{0,1\}$ of the form $x \mapsto c \cdot x$, where $c \cdot x$ is taken modulo 2. It turns out that there is an elegant, efficient private learner for this class, showing that efficient private learning goes beyond the statistical query model:

Theorem 7.8.5 (Private learning of parities [67]). *The class $\mathcal{Q}^{par} = \mathcal{Q}^{par}(d)$ of parity functions on $\{0,1\}^d$ is efficiently and privately PAC learnable, with sample complexity $n = O(d/\varepsilon)$ for $(\varepsilon, 0)$-differential privacy.*

Since the class of parity functions on $\{0,1\}^d$ has VC dimension d, the sample complexity for private learning is within a constant factor of the sample complexity for nonprivate learning.

Proof: We have a dataset (x, y) with n rows (x_i, y_i), where $x_i \in \{0,1\}^d$ and $y_i \in \{0,1\}$. Assume that $x_1, \ldots, x_n$ are drawn independently from some distribution D, and that there is some $c \in \{0,1\}^d$ such that $y_i = c \cdot x_i$ for all $1 \le i \le n$. We wish to determine a hypothesis $h \in \{0,1\}^d$ such that, if x is drawn from D, then $h \cdot x = c \cdot x$ with probability at least 0.99.

A simple (nonprivate) algorithm is to take any h such that $y_i = h \cdot x_i$ for all i. We can do this by using Gaussian elimination to solve the system of linear equations $y = h \cdot x$. Standard calculations show that this succeeds with $n = O(d)$ samples.

Now let us consider private learning, keeping in mind that we need to ensure privacy even when the data is inconsistent with the concept class. Indeed, we need to make sure that we do not leak information by revealing whether or not the data is consistent! For instance, we need to make sure that the algorithm's output distribution only changes by ε (multiplicatively) if we add a single row (x_i, y_i) such that $y_i \neq c \cdot x_i$.

Our mechanism $\mathcal{M}$ works as follows; we use $\perp$ to denote failure. We will start by succeeding with probability about $1/2$, and amplify this probability later.

1. Take $n = O(d/\varepsilon)$ samples.
2. With probability $1/2$, output $\perp$.
3. For each $1 \le i \le n$, set $\hat{x}_i, \hat{y}_i$ independently as follows:

$$(\hat{x}_i, \hat{y}_i) = \begin{cases} (0^d, 0) & \text{with probability } 1 - \varepsilon\,, \\ (x_i, y_i) & \text{with probability } \varepsilon\,. \end{cases}$$

   Call the resulting dataset $(\hat{x}, \hat{y})$. This is effectively a random sample of the original dataset, containing an expected fraction ε of the rows. The zero entries $(\hat{x}_i, \hat{y}_i) = (0^d, 0)$ will have no effect on what follows.
4. Using Gaussian elimination, determine the affine subspace V of hypotheses h that are consistent with $(\hat{x}, \hat{y})$, i.e.,

$$V = \{h \mid \forall i : \hat{y}_i = h \cdot \hat{x}_i\}\,.$$

   Output an h chosen uniformly from V. If $V = \varnothing$ (i.e., if no consistent h exists), then output $\perp$.

Since the nonprivate algorithm described above succeeds with probability 0.99, if the data is consistent then $\mathcal{M}$ succeeds with probability at least 0.49. We can amplify by repeating this t times, in which case the sample complexity is $n = O(td/\varepsilon)$.

Now we analyze $\mathcal{M}$'s privacy. We willfully identify $1 \pm \varepsilon$ with $e^{\pm\varepsilon}$, neglecting $O(\varepsilon^2)$ terms.

Claim 7.8.6. *$\mathcal{M}$ is $(2\varepsilon, 0)$-differentially private.*

Proof of claim: Let $x \sim x'$ be two neighboring datasets that differ at one row i. Assume that $(x'_i, y'_i) = (0^d, 0)$. Since we can get from any x to any x'' by going through such an x', if we can show that $\mathcal{M}(x)$ and $\mathcal{M}(x')$ are $(\varepsilon, 0)$-indistinguishable, then $\mathcal{M}$ will be $(2\varepsilon, 0)$-differentially private.

With probability $1 - \varepsilon$, we replace (x_i, y_i) with $(0^d, 0)$ in step 3 (assuming we make it past step 2). In that case, $(\hat{x}, \hat{y}) = (\hat{x}', \hat{y}')$, and the output probabilities are the same. Thus for all possible outputs z,

$$\Pr[\mathcal{M}(x) = z] \geq (1 - \varepsilon) \Pr[\mathcal{M}(x') = z]\,. \tag{7.9}$$

But we are not done. The problem is that x' is special (by our assumption) so the reverse inequality does not automatically hold. We also need to prove

$$\Pr[\mathcal{M}(x) = z] \leq (1 + \varepsilon) \Pr[\mathcal{M}(x') = z]\,. \tag{7.10}$$

To prove (7.10), start by fixing $(\hat{x}_j, \hat{y}_j) = (\hat{x}'_j, \hat{y}'_j)$ for all $j \neq i$. (Thus, we are coupling the algorithm's random choices on the two datasets.) Let V_{-i} be the affine subspace consistent with these rows:

$$V_{-i} = \{h \mid \forall j \neq i : \hat{y}_j = h \cdot \hat{x}_j\}\,.$$

As before, if we fail or if we set $(\hat{x}_i, \hat{y}_i) = (0^d, 0) = (\hat{x}'_i, \hat{y}'_i)$, the output probabilities are the same. On the other hand, with probability $\varepsilon/2$ we pass step 2 and set $(\hat{x}_i, \hat{y}_i) = (x_i, y_i)$ in step 3. In that case, $\mathcal{M}(x')$ is uniform in V_{-i} (or $\mathcal{M}(x') = \perp$ if $V_{-i} = \varnothing$), while $\mathcal{M}(x)$ is uniform in

$$V = V_{-i} \cap \{h \mid y_i = h \cdot x_i\}$$

(or $\mathcal{M}(x) = \perp$ if $V = \varnothing$).

Let us compare the probabilities that $\mathcal{M}(x)$ and $\mathcal{M}(x')$ fail. If $V_{-i} = \varnothing$, then $\mathcal{M}(x) = \mathcal{M}(x') = \perp$. But if $V_{-i} \neq \varnothing$ but $V = \varnothing$, the probability that $\mathcal{M}(x)$ fails is at most $1/2 + \varepsilon/2$; and since $\mathcal{M}(x')$ fails with probability at least $1/2$, we have

$$\Pr[\mathcal{M}(x) = \perp] \leq \frac{1 + \varepsilon}{2} \leq (1 + \varepsilon) \cdot \Pr[\mathcal{M}(x') = \perp]\,.$$

Finally, we come to the most interesting case: comparing the probabilities that $\mathcal{M}(x)$ and $\mathcal{M}(x')$ output some hypothesis h, where both V_{-i} and V are nonempty and contain h. Since V is obtained by adding one linear constraint to V_{-i}, we have

$$|V| \geq \frac{1}{2}|V_{-i}|\,.$$

Since $\mathcal{M}(x)$ and $\mathcal{M}(x')$ are uniform in V and V_{-i}, respectively, for every $h \in V_{-i}$ we have

$$\Pr[\mathcal{M}(x) = h] \leq \frac{1}{2}\left(\frac{1-\varepsilon}{|V_{-i}|} + \frac{\varepsilon}{|V|}\right) \leq \frac{1}{2} \cdot \frac{1+\varepsilon}{|V_{-i}|} = (1+\varepsilon)\Pr[\mathcal{M}(x') = h],$$

which completes the proof. ■ ■

Since linear algebra was essentially the only known technique for efficient private learning outside the statistical query model, this result suggested that perhaps every concept that is efficiently PAC learnable is also efficiently and privately PAC learnable. Bun and Zhandry [20] recently gave evidence that this is not the case.

Theorem 7.8.7 (Hardness of private learning [20]). *If "indistinguishability obfuscation" and "perfectly sound noninteractive zero-knowledge proofs for NP" exist, then there is a concept class that is efficiently PAC learnable but not efficiently PAC learnable with differential privacy.*

7.8.3 The Sample Complexity of Private PAC Learning

Another gap between PAC learning and private PAC learning is in sample complexity. The sample complexity of nonprivate learning is characterized by $\Theta(\mathrm{VC}(\mathcal{C}))$, whereas for private learning we have the upper bound $O(\log|\mathcal{C}|)$ from Theorem 7.8.5, which can be as large as $d \cdot \mathrm{VC}(\mathcal{C})$ on a domain of size 2^d. Two classes that illustrate this gap are the classes of point functions and threshold functions ($\mathcal{Q}^{\mathrm{pt}}$ and $\mathcal{Q}^{\mathrm{thr}}$). In both cases, we have $\mathrm{VC}(\mathcal{C}) = 1$ but $\log|\mathcal{C}| = d$.

For the class $\mathcal{C} = \mathcal{Q}^{\mathrm{pt}}(d)$ of point functions on $\{0,1\}^d$ and $(\varepsilon, 0)$-differentially private *proper* learners, Beimel, Brenner, Kasiviswanathan, and Nissim [10] showed that the best possible sample complexity is $\Theta(d)$, similarly to the situation with releasing approximate answers to all point functions (Proposition 7.2.8 and Theorem 7.5.14). If we relax the requirement to *either* improper learning or approximate differential privacy, then, similarly to Theorem 7.3.5, the sample complexity becomes independent of d, namely $O(1)$ or $O(\log(1/\delta))$, respectively [10, 9].

For the class $\mathcal{C} = \mathcal{Q}^{\mathrm{thr}}([2^d])$ of threshold functions on $\{1, \ldots, 2^d\}$, again it is known that $\Theta(d)$ sample complexity is the best possible sample complexity for $(\varepsilon, 0)$-differentially private proper learners [10], similarly to Theorem 7.7.2. In contrast to point functions, however, it is known that relaxing to either (ε, δ)-differential privacy or to improper learning is *not* enough to achieve sample complexity $O(1)$. For (ε, δ)-differentially private proper learners, the sample complexity is somewhere between $2^{(1+o(1))\log^* d} \cdot \log(1/\delta)$ and $\Omega(\log^* d)$, similarly to Theorem 7.7.3. For $(\varepsilon, 0)$-differentially private learners, the sample complexity was recently shown to be $\Omega(d)$ by Feldman and Xiao [50]. We present the proof of this result, because it uses beautiful connections between VC dimension, private learning, and communication complexity.

Every concept class $\mathcal{C}$ defines a one-way communication problem as follows: Alice has a function $c \in \mathcal{C}$, Bob has a string $w \in \{0,1\}^d$, and together they want to compute $c(w)$. The *one-way communication complexity* of this problem is the length of the shortest message m that Alice needs to send to Bob that lets him compute $c(w)$. We will consider randomized, distributional communication complexity,

where the inputs are chosen according to some distribution μ on $\mathcal{C} \times \{0,1\}^d$, and Bob should compute $c(w)$ correctly with high probability over the choice of the inputs and the (shared) randomness between Alice and Bob. We write $\mathrm{CC}^{\rightarrow,\mathrm{pub}}_{\mu,\alpha}(\mathcal{C})$ to denote the minimum message length over all protocols where Bob computes $c(w)$ with probability at least $1-\alpha$.

It was known that maximizing this communication complexity over all *product distributions* characterizes the sample complexity of nonprivate learning (i.e., VC dimension):

Theorem 7.8.8 (CC characterization of nonprivate learning [73]). *For every constant $\alpha \in (0, 1/8)$,*

$$\mathrm{VC}(\mathcal{C}) = \Theta\left(\max_{\mu_A,\mu_B} \mathrm{CC}^{\rightarrow,\mathrm{pub}}_{\mu_A\otimes\mu_B,\alpha}(\mathcal{C})\right),$$

where μ_A and μ_B are distributions on $\mathcal{C}$ and $\{0,1\}^d$, respectively.

Building on Beimel et al. [8], Feldman and Xiao [50] showed that the sample complexity of learning $\mathcal{C}$ with pure differential privacy is related to the one-way communication complexity maximized over all *joint* distributions on $\mathcal{C} \times \{0,1\}^d$.

Theorem 7.8.9 (CC characterization of learning with pure differential privacy [50]). *For all constants $\varepsilon > 0$, $\alpha \in (0, 1/2)$, the smallest sample complexity for learning $\mathcal{C}$ under $(\varepsilon, 0)$-differential privacy is $\Theta(\max_\mu \mathrm{CC}^{\rightarrow,\mathrm{pub}}_{\mu,\alpha}(\mathcal{C}))$.*

We note that, by Yao's minimax principle, $\max_\mu \mathrm{CC}^{\rightarrow,\mathrm{pub}}_{\mu,\alpha}(\mathcal{C})$ is simply equal to the worst-case randomized communication complexity of $\mathcal{C}$, where we want a protocol such that, on every input, Bob computes the correct answer with probability at least $1-\alpha$ over the public coins of the protocol. Returning to threshold functions, computing $c_y(w)$ is equivalent to computing the "greater than" function. Miltersen et al. [80] showed that for this problem the randomized communication complexity is $\Omega(d)$, proving that learning thresholds with pure differential privacy requires sample complexity $\Omega(d)$.

Proof sketch of Theorem 7.8.9: We begin by showing that the communication complexity is upper-bounded by the sample complexity of private learning. Let L be an $(\varepsilon, 0)$-differentially private learner for $\mathcal{C}$ with a given sample complexity n; we will use L to construct a communication protocol. Using their shared randomness, Alice and Bob both run L on the all-zeroes dataset $x^{(0)}$. They do this M times for M to be determined in a moment, giving a list of shared functions $h_1, \ldots, h_M \in \mathcal{H}$.

Since L is $(\varepsilon, 0)$-differentially private, by group privacy, the distribution of functions returned by L "covers" the distribution on every other dataset $x \in \mathcal{X}^n$, in the sense that, for each $h \in \mathcal{H}$,

$$\Pr[L(x^{(0)}) = h] \geq \mathrm{e}^{-\varepsilon n} \Pr[L(x) = h].$$

Thus with $M = \mathrm{e}^{O(\varepsilon n)}$ samples, Alice and Bob can ensure that, with high probability, at least one h_i in their shared list is a good hypothesis for any particular dataset.

In particular, let μ be a distribution on pairs (c, w), and let $c_0 \in \mathcal{C}$ be Alice's function. Then there is some $1 \leq i \leq M$ such that h_i is a good hypothesis for the dataset x we would get by sampling the rows of x from the conditional distribution $\mu(w \mid c = c_0)$: that is, $h_i(w) = c_0(w)$ with high probability in w. Alice can send Bob this index i with communication complexity $\log M = O(\varepsilon n)$.

Conversely, suppose that we have a randomized, public-coin protocol for $\mathcal{C}$ with communication complexity at most n. Every setting r of the public randomness and message m from Alice defines a hypothesis $h_{r,m}$ which Bob uses to compute the output of the protocol (by applying it to his input w). Given a dataset $(x_1, y_1), \ldots, (x_n, y_n)$, our differentially private learner will choose r uniformly at random, and then use the exponential mechanism to select an m approximately maximizing $|\{i : h_{r,m}(x_i) = y_i\}|$, similarly to the use of the exponential mechanism in the proof of Theorem 7.8.3. The sample complexity n required by the exponential mechanism is logarithmic in the size of the hypothesis class $\mathcal{H}_r = \{h_{r,m}\}$, so we have $n = O(|m|)$. ■

While this provides a tight characterization of the sample complexity of learning with pure differential privacy, the case of approximate differential privacy is still very much open.

Open Problem 7.8.10. Does every concept class $\mathcal{C}$ over $\{0,1\}^d$ have an (ε, δ)-differentially private learner with sample complexity $n = O(\mathrm{VC}(\mathcal{C}) \cdot \mathrm{polylog}(1/\delta))$ (for δ negligible in n and d)? Or are there concept classes where the sample complexity must be $n = \Omega(d \cdot \mathrm{VC}(\mathcal{C}))$?

These questions are open for both proper and improper learning. In the case of proper learning, there are concept classes known where the sample complexity is at least $\Omega(\log^* d \cdot \mathrm{VC}(\mathcal{C}) \cdot \log(1/\delta))$, such as threshold functions [22], but this does not rule out an upper bound of $n = O(\mathrm{VC}(\mathcal{C}) \cdot \mathrm{polylog}(1/\delta))$ when δ is negligible in n and d.

7.9 Multiparty Differential Privacy

7.9.1 The Definition

We now consider an extension of differential privacy to a multiparty setting, where the data is divided among some m parties $P_1, \ldots, P_m$. For simplicity, we will assume that m divides n and each party P_k has exactly n/m rows of the dataset, which we will denote by $x_k = (x_{k,1}, x_{k,2}, \ldots, x_{k,n/m})$. (Note the change in notation; now x_k is a subdataset, not an individual row.) We consider the case that P_k wants to ensure the privacy of the rows in x_k against an adversary who may control the other parties.

As in the studies of secure multiparty computation (cf. [52]), there are many variants of the adversary model that we can consider:

- Passive versus active: for simplicity, we will restrict to passive adversaries — ones that follow the specified protocol — but try to extract information from the communication seen (also known as "honest-but-curious" adversaries). Since

our emphasis is on lower bounds, this only strengthens the results. However, all of the upper bounds we mention are also known to hold for active adversaries.
- Threshold adversaries: we can restrict the adversary to control at most t parties for some $t \leq m-1$. For simplicity, we will only consider the case $t = m-1$. Consequently we may assume without loss of generality that all communication occurs on a broadcast channel, as the adversary would anyhow see all communication on point-to-point channels.
- Computationally bounded versus unbounded: as in the basic definition of differential privacy, we will (implicitly) consider computationally unbounded adversaries. In the next section, we will discuss computationally bounded adversaries.

A *protocol* proceeds in a sequence of rounds until all honest parties terminate. Informally, in each round, each party P_k selects a message to be broadcast based on its input $x^{(k)}$, internal coin tosses, and all messages received in previous rounds. The *output* of the protocol is specified by a deterministic function of the transcript of messages exchanged. (As in secure multiparty computation, one can also consider individual outputs computed by the parties P_k, which may depend on their private input and coin tosses, but we do not do that for simplicity.) Given a particular adversary strategy A, we write $\mathrm{View}_A((A \leftrightarrow (P_1, \ldots, P_m))(x))$ for the random variable that includes everything that A sees when participating in the protocol $(P_1, \ldots, P_m)$ on input x. In the case we consider, where A is a passive adversary controlling $P_{-k} = (P_1, P_2, \ldots, P_{k-1}, P_{k+1}, \ldots, P_m)$, $\mathrm{View}_A(A \leftrightarrow (P_1, \ldots, P_m)(x))$ is determined by the inputs and coin tosses of all parties other than P_k as well as the messages sent by P_k.

Definition 7.9.1 (Multiparty differential privacy [7]). *For a protocol $P = (P_1, \ldots, P_m)$ taking as input datasets $(x_1, \ldots, x_m) \in (\mathcal{X}^{n/m})^m$, we say that P is (ε, δ)-differentially private (for passive adversaries) if, for every $k \in [m]$ and every two datasets $x, x' \in (\mathcal{X}^{n/m})^m$ that differ on one row of P_k's input (and are equal otherwise), the following holds for every set T:*

$$\Pr[\mathrm{View}_{P_{-k}}(P_{-k} \leftrightarrow (P_1, \ldots, P_m)(x)) \in T] \leq e^{\varepsilon} \cdot \Pr[\mathrm{View}_{P_{-k}}(P_{-k} \leftrightarrow (P_1, \ldots, P_m)(x')) \in T] + \delta.$$

7.9.2 The Local Model

Constructing useful differentially private multiparty protocols for $m \geq 2$ parties is harder than constructing them in the standard centralized model (corresponding to $m = 1$), as a trusted curator could just simulate the entire protocol and provide only the output. An extreme case is when $m = n$, in which case the individual data subjects need not trust anyone else, because they can just play the role of a party in the protocol. This is the *local model* that we've alluded to several times in earlier sections. While this is the hardest model of distributed differential privacy, there are nontrivial protocols in it, namely *randomized response* (as in Section 7.1.5):

Theorem 7.9.2 (Randomized response). *For every counting query $q : \mathcal{X} \to \{0,1\}$, $n \in \mathbb{N}$, and $\varepsilon > 0$, there is an $(\varepsilon, 0)$-differentially private n-party protocol in the local model for computing q to within error $\alpha = O(1/(\varepsilon\sqrt{n}))$ with high probability.*

This can be extended to estimating statistical queries $q : \mathcal{X} \to [0,1]$ over the dataset—first randomly round $q(x_k)$ to a bit $b_k \in \{0,1\}$ with expectation $q(x_k)$ (i.e., set $b_k = 1$ with probability $q(x_k)$), and then apply randomized response to b_k. This gives some intuition for why everything that is PAC learnable in the statistical query model is PAC learnable in the local model, as mentioned in Section 7.8.

Note that the error in Theorem 7.9.2 is significantly worse than the error $O(1/\varepsilon n)$ we get with a centralized curator. Building on [7, 78], Chan et al. [25] proved that the $1/\sqrt{n}$ decay is in fact optimal:

Theorem 7.9.3 (Randomized response is optimal in the local model [25]). *For every* nonconstant *counting query $q : \mathcal{X} \to \{0,1\}$, $n \in \mathbb{N}$, and $(1,0)$-differentially private n-party protocol P for approximating q, there is an input dataset $x \in \mathcal{X}^n$ on which P has error $\alpha = \Omega(1/\sqrt{n})$ with high probability.*

Proof sketch: We first prove it for $\mathcal{X} = \{0,1\}$, and q being the identity function (i.e., we are computing the average of the input bits). Consider a uniformly random input dataset $X = (X_1, \ldots, X_n) \leftarrow \{0,1\}^n$, let $R = (R_1, \ldots, R_n)$ denote the randomness of the n parties, and let $T = T(X,R)$ be the random variable denoting the transcript of the protocol. Let $t \in \mathrm{Supp}(T)$ be any value of T. We claim that, conditioned on $T = t$:

1. The n random variables $(X_1, R_1), \ldots, (X_n, R_n)$ are independent, and in particular $X_1, \ldots, X_n$ are independent.
2. Each $\Pr[X_i = 1] \in (1/4, 3/4)$.

Item 1 is a general fact about interactive protocols—if the parties' inputs start independent, they remain independent conditioned on the transcript—and can be proven by induction on the number of rounds of the protocol. Item 2 uses $(\varepsilon = 1, 0)$-differential privacy and Bayes' rule:

$$\begin{aligned}\frac{\Pr[X_i = 1 | T = t]}{\Pr[X_i = 0 | T = t]} &= \frac{\Pr[T = t | X_i = 1] \cdot \Pr[X_i = 1] / \Pr[T = t]}{\Pr[T = t | X_i = 0] \cdot \Pr[X_i = 0] / \Pr[T = t]} \\ &= \frac{\Pr[T = t | X_i = 1]}{\Pr[T = t | X_i = 0]} \\ &\in \left[e^{-\varepsilon}, e^{\varepsilon}\right].\end{aligned}$$

This implies that

$$\Pr[X_i = 1 | T = t] \in \left[\frac{1}{e^{\varepsilon} + 1}, \frac{e^{\varepsilon}}{e^{\varepsilon} + 1}\right] \subset (1/4, 3/4)$$

for $\varepsilon = 1$.

Consequently, conditioned on $T = t$, $(1/n) \cdot (\sum_i X_i)$ is the average of n independent $\{0,1\}$ random variables with bounded bias. In particular, the standard deviation of

$\sum_i X_i$ is $\Omega(1/\sqrt{n})$, and by anticoncentration bounds, with high probability we will have

$$\left|(1/n)\sum_i X_i - \text{output}(t)\right| = \Omega(1/\sqrt{n}),$$

where output$(\cdot)$ is the output function of the protocol. Since the protocol has error $\Omega(1/\sqrt{n})$ on a random dataset with high probability, there is some fixed dataset on which it has error $\Omega(1/\sqrt{n})$ with high probability.

To obtain the result for general nonconstant counting queries $q : \mathcal{X} \to \{0,1\}$, fix two inputs $w_0, w_1 \in \mathcal{X}$ such that $q(w_b) = b$, and restrict to datasets of the form $(w_{b_1}, \ldots, w_{b_n})$ for $b_1, \ldots, b_n \in \{0,1\}$. Estimating the counting query q on such datasets with differential privacy is equivalent to estimating the average function on datasets of the form $(b_1, \ldots, b_n)$ with differential privacy. ■

Effectively, what the above proof is using is a "randomness extraction" property of the SUM function. Specifically, for every source Y consisting of n independent bits $Y = (Y_1, \ldots, Y_n)$ that are not too biased, $\sum_i Y_i$ has a lot of "randomness"—it is not concentrated in any interval of width $O(\sqrt{n})$. (In the proof, $Y_i = X_i|_{T=t}$.) In fact, a stronger statement is true: $\sum_i Y_i \mod k$ can be shown to be almost uniformly distributed in $\mathbb{Z}_k$ for some $k = \Omega(\sqrt{n})$. In the language of randomness extractors (see [94, 105]), we would say that "the sum modulo k function is a (deterministic) randomness extractor for the class of sources consisting of n independent bits with bounded bias."

7.9.3 Two-Party Differential Privacy

Now let us look at the case of $m = 2$ parties each holding $n/2$ rows of the dataset, which seems closer to the trusted curator case than to the local model. Indeed, in this model, any counting query can be computed with error $O(1/\varepsilon n)$: each party just adds $\text{Lap}(1/(\varepsilon \cdot (n/2)))$ noise to the counting query on her own dataset and announces the result; we average the two results to estimate the overall counting query. However, there are other simple queries where again there is a quadratic gap between the single curator ($m = 1$) and two-party case, namely the (normalized) inner product function $\text{IP} : \{0,1\}^{n/2} \times \{0,1\}^{n/2} \to [0,1]$ given by $\text{IP}(x,y) = \langle x, y\rangle/(n/2)$. IP has global sensitivity $2/n$, and hence can be computed by a single trusted curator with error $O(1/n)$). But for two parties (one given x and one given y), the best possible error is again $\tilde{\Theta}(1/\sqrt{n})$:

Theorem 7.9.4 (Two-party DP protocols for inner product [81, 78]).

1. *There is a two-party differentially private protocol that estimates* IP *to within error* $O(1/\varepsilon \cdot \sqrt{n})$ *with high probability, and*
2. *Every two party* $(1,0)$*-differentially private protocol for* IP *incurs error* $\tilde{\Omega}(1/\sqrt{n})$ *with high probability on some dataset.*

Proof sketch: For the upper bound, we again use randomized response:

1. On input $x \in \{0,1\}^{n/2}$, Alice uses randomized response to send a noisy version $\hat{x}$ of x to Bob.

2. Upon receiving $\hat{x}$ and his input $y \in \{0,1\}^{n/2}$, Bob computes

$$z = \frac{2}{n} \sum_{i=1}^{n/2} \frac{y_i}{\varepsilon} \cdot \left(\hat{x}_i - \frac{(1-\varepsilon)}{2} \right),$$

which will approximate $\mathrm{IP}(x,y)$ to within $O(1/\varepsilon\sqrt{n})$.
3. Bob sends the output $z + \mathrm{Lap}(O(1/\varepsilon^2 n))$ to Alice, where this Laplace noise is to protect the privacy of y, since z has global sensitivity $O(1/\varepsilon n)$ as a function of y.

For the lower bound, we follow the same outline as Theorem 7.9.3. Let $X = (X_1, \ldots, X_{n/2})$ and $Y = (Y_1, \ldots, Y_{n/2})$ each be uniformly distributed over $\{0,1\}^{n/2}$ and independent of each other. Then, conditioned on a transcript t of an $(\varepsilon, 0)$-differentially private protocol, we have:

1. X and Y are independent, and
2. For every $i \in [n/2]$, $x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_n$,

$$\Pr[X_i = 1 | X_1 = x_1, \ldots, X_{i-1} = x_{i-1}, X_{i+1} = x_{i+1}, \ldots, X_n = x_n] \in (1/4, 3/4),$$

and similarly for Y.

Item 2 again follows from differential privacy and Bayes' rule. (Consider the two neighboring datasets $(x_1, \ldots, x_{i-1}, 0, x_{i+1}, \ldots, x_n)$ and $(x_1, \ldots, x_{i-1}, 1, x_{i+1}, \ldots, x_n)$.) In the literature on randomness extractors, sources satisfying item 2 are known as "Santha–Vazirani sources" or "unpredictable-bit sources", because no bit can be predicted with high probability given the others. (Actually, the usual definition only requires that item 2 hold when conditioning on past bits $X_1 = x_1, \ldots, X_{i-1} = x_{i-1}$, so the sources we have are a special case.)

One of the early results in randomness extractors is that the (nonnormalized) inner product modulo 2 function is an extractor for Santha–Vazirani sources [107]. This result can be generalized to the inner product modulo $k = \tilde{\Omega}(\sqrt{n})$, so we know that $\langle X, Y \rangle \mod k$ is almost uniformly distributed in $\mathbb{Z}_k$ (even conditioned on the transcript t). In particular, it cannot be concentrated in an interval of width $o(k)$ around output(t). Thus the protocol must have error $\Omega(k)$ with high probability. ■

The above theorems show there can be a $\tilde{\Theta}(\sqrt{n})$ factor gap between the worst-case error achievable with a centralized curator (which is captured by global sensitivity) and multiparty (even two-party) differential privacy. Both lower bounds extend to (ε, δ)-differential privacy when $\delta = o(1/n)$. When $\delta = 0$, the largest possible gap, namely $\Omega(n)$, can be proven using a connection to *information complexity*. Before defining information-complexity, let us look at an information-theoretic consequence of differential privacy.

Theorem 7.9.5 (Differential privacy implies low mutual information [78]). *Let $\mathcal{M} : \mathcal{X}^n \to \mathcal{Y}$ be an $(\varepsilon, 0)$-differentially private mechanism. Then for every random variable X distributed on $\mathcal{X}^n$, we have*

$$I(X; \mathcal{M}(X)) \leq 1.5\varepsilon n,$$

where $I(\cdot;\cdot)$ denotes mutual information.

Note that, without the DP constraint, the largest the mutual information could be is when X is the uniform distribution and $\mathcal{M}$ is the identity function, in which case $I(X;\mathcal{M}(X)) = n \cdot \log_2 |\mathcal{X}|$, so the above bound can be much smaller. We remark that, for approximate differential privacy, one can bound the mutual information $I(X;\mathcal{M}(X))$ in case the rows of X are independent [78, 92], but these bounds do not hold for general correlated distributions [29].

Proof: The mutual information between X and $\mathcal{M}(X)$ is the expectation over $(x,y) \leftarrow (X,\mathcal{M}(X))$ of the following quantity:

$$\log_2\left(\frac{\Pr[\mathcal{M}(X)=y|X=x]}{\Pr[\mathcal{M}(X)=y]}\right).$$

By group privacy (Lemma 7.2.2), the quantity inside the logarithm is always at most $e^{\varepsilon n}$, so the mutual information is at most $(\log_2 e)\cdot\varepsilon n < 1.5\varepsilon n$. ■

To apply this to two-party protocols, we can consider the mechanism $\mathcal{M}$ that takes both parties' inputs and outputs the transcript of the protocol, in which case the mutual information is known as *external information cost*. Or we can fix one party's input x, and consider the mechanism $\mathcal{M}_x(y)$ that takes the other party's input y and outputs the former party's view of the protocol, yielding a bound on *internal information cost*. The information cost of two-party protocols has been very widely studied in recent years (with initial motivations from communication complexity), and there are a number of known, explicit Boolean functions f and input distributions (X,Y) such that any protocol computing f on (X,Y) has information cost $\Omega(n)$. These can be leveraged to construct a low-sensitivity function g such that any two-party differentially private protocol for g incurs error $\Omega(n\cdot \mathrm{GS}_g)$ [78]. This is within a constant factor of the largest possible gap, since the range of g has size at most $n\cdot \mathrm{GS}_g$. It is open to obtain a similar gap for approximate differential privacy:

Open Problem 7.9.6. Is there a function $f : \mathcal{X}^n \to \mathbb{R}$ such that any multiparty (ε,δ)-differentially private protocol (with constant ε and $\delta = \mathrm{neg}(n)$) for f incurs error $\omega(\sqrt{n}\cdot \mathrm{GS}_f)$ with high probability on some dataset? What about $\Omega(n\cdot \mathrm{GS}_f)$? These are open in both the two-party and local models.

More generally, it would be good to develop our understanding of multiparty differential privacy computation of specific functions such as IP and towards a more general classification.

Open Problem 7.9.7. Characterize the optimal privacy–accuracy tradeoffs for estimating a wide class of functions (more generally, solving a wide set of data analysis tasks) in two-party or multiparty differential privacy.

As the results of Section 7.9.2 suggest, we have a better understanding of the local model than for a smaller number of parties, such as $m = 2$. (See also [4] and the references therein.) However, it still lags quite far behind our understanding of the single-curator model, for example, when we want to answer a set $\mathcal{Q}$ of queries (as opposed to a single query).

7.10 Computational Differential Privacy

7.10.1 The Definition

The basic definition of differential privacy provides protection even against adversaries with unlimited computational power. It is natural to ask whether one can gain from restricting to computationally bounded adversaries, given the amazing effects of such a restriction in modern cryptography.

To obtain a computational analogue of differential privacy, we can simply take the inequalities defining differential privacy, namely

$$\forall T \subseteq \mathcal{Y},\ \Pr[\mathcal{M}(x) \in T] \leq \mathrm{e}^{\varepsilon} \cdot \Pr[\mathcal{M}(x') \in T] + \delta$$

and restrict our attention to tests T defined by feasible algorithms.

Definition 7.10.1 (Computational differential privacy [7]). *Let $\mathcal{M} = \{\mathcal{M}_n : \mathcal{X}_n^n \to \mathcal{Y}_n\}_{n\in\mathbb{N}}$ be a sequence of randomized algorithms, where elements in $\mathcal{X}_n$ and $\mathcal{Y}_n$ can be represented by* poly(n)*-bit strings. We say that $\mathcal{M}$ is* computationally ε-differentially private *if there is a superpolynomial function $s(n) = n^{\omega(1)}$ and a negligible function $\delta(n) = n^{-\omega(1)}$ such that, for all n, all pairs of datasets $x, x' \in \mathcal{X}^n$ differing on one row, and all Boolean circuits $T : \mathcal{X}^n \to \{0,1\}$ of size at most $s(n)$, we have*

$$\Pr[T(\mathcal{M}(x)) = 1] \leq \mathrm{e}^{\varepsilon} \cdot \Pr[T(\mathcal{M}(x')) = 1] + \delta(n).$$

We make a few remarks on the definition:

- We always allow for a nonzero $\delta = \delta(n)$ term in the definition of computational differential privacy. If we did not do so, then the definition would collapse to that of ordinary (information-theoretic) $(\varepsilon, 0)$-differential privacy, because the latter is equivalent to requiring $(\varepsilon, 0)$-differential privacy for sets T of size 1, which are computable by Boolean circuits of size poly(n).
- We generally are only interested in computationally differentially private mechanisms $\mathcal{M}$ that are themselves computable by randomized polynomial-time algorithms, as we should allow the adversary T to invest more computation time than the privacy mechanism.
- For simplicity, we have used the number n of rows as a security parameter, but it is often preferable to decouple these two parameters. We will often drop the index of n from the notation, and make the asymptotics implicit, for sake of readability.

7.10.2 Constructions via Secure Multiparty Computation

The most significant gains we know how to get from computational differential privacy are in the multiparty case. Indeed, by using powerful results on secure multiparty computation, everything that is achievable by a differentially private centralized curator can also be emulated by a multiparty protocol with computational differential privacy.

Theorem 7.10.2 (Computational differential privacy via cryptography [38, 7]). *Assume that oblivious transfer protocols exist. Let $\mathcal{M} : \mathcal{X}^n \to \mathcal{Y}$ be computationally ε-differentially private for $\varepsilon \leq 1$ and computable in time* poly(n). *Then for every $m|n$, there is an m-party protocol $P = (P_1, \ldots, P_m) : (\mathcal{X}^{n/m})^m \to \mathcal{Y}$ such that:*

1. *P is computationally ε-differentially private,*
2. *For every input $x \in \mathcal{X}^n$, the output distribution of $P(x)$ is the same as that of $\mathcal{M} : (\mathcal{X}^{n/m})^m \to \mathcal{Y}$,*
3. *P is computable in time* poly(n).

Proof sketch: By classic results on secure multiparty computation [109, 53], there exists an m-party protocol P for evaluating $\mathcal{M}$ that is secure against passive adversaries, assuming the existence of oblivious transfer protocols. (See [**?** 52] for full definitions and constructions of secure multiparty computation.) Items 2 and 3 are immediately guaranteed by the properties of secure multiparty computation protocols. For item 1, we recall that each party learns nothing from a secure multiparty computation protocol other than what is implied by their own input and the output of the function being evaluated (in this case $\mathcal{M}$). More precisely, for every poly(n)-size adversary A, controlling all parties other than P_k, there is a poly(n)-size simulator S such that $\text{View}_A(A \leftrightarrow (P_1, \ldots, P_m(x))$ is computationally indistinguishable from $S(\mathcal{M}(x), x_1, \ldots, x_{k-1}, x_{k+1}, \ldots, x_m)$. Thus, for every x and x' that differ only by changing one row of the input to party j, and every poly(n)-size T, we have

$$\begin{aligned}
&\Pr[T(\text{View}_A(A \leftrightarrow (P_1, \ldots, P_m)(x))) = 1] \\
&\leq \Pr[T(S(\mathcal{M}(x), x_1, \ldots, x_{k-1}, x_{k+1}, \ldots, x_m)) = 1] + \text{neg}(n) \\
&= \left(e^{\varepsilon} \cdot \Pr[T(S(\mathcal{M}(x'), x'_1, \ldots, x'_{k-1}, x'_{k+1}, \ldots, x'_m)) = 1] + \text{neg}(n)\right) + \text{neg}(n) \\
&\leq e^{\varepsilon} \cdot (\Pr[T(\text{View}_A(A \leftrightarrow (P_1, \ldots, P_m)(x'))) = 1] + \text{neg}(n)) + \text{neg}(n) + \text{neg}(n) \\
&= e^{\varepsilon} \cdot \Pr[T(\text{View}_A(A \leftrightarrow (P_1, \ldots, P_m)(x'))) = 1] + \text{neg}(n).
\end{aligned}$$

■

In particular, with computational differential privacy, we have n-party protocols for computing any counting query or the normalized inner product function with error $O(1/\varepsilon n)$, significantly better than the $\tilde{\Theta}(1/\sqrt{n})$ error achievable with information-theoretic differential privacy. It is interesting to understand to what extent general secure multiparty computation (whose existence is equivalent to oblivious transfer) is necessary for such separations between information-theoretic and computational differential privacy. Haitner et al. [57] showed that black-box use of one-way functions does *not* suffice to construct two-party protocols for the inner product function with error smaller than $\tilde{\Theta}(1/\sqrt{n})$, but a tight characterization remains open.

Open Problem 7.10.3. What is the minimal complexity assumption needed to construct a computational task that can be solved by a computationally differentially private protocol but is impossible to solve by an information-theoretically differentially private protocol?

Recent works have made progress on understanding this question for computing *Boolean* functions with differential privacy, for example showing that achieving

near-optimal accuracy requires oblivious transfer in some cases [54], but it remains open whether there can be a separation based on a weaker assumption, and whether oblivious transfer is needed to have an asymptotic separation in accuracy for a more natural statistical task (e.g., estimating a function with bounded global sensitivity, such as normalized inner product).

7.10.3 Usefulness with a Trusted Curator?

For the single-curator case ($m = 1$), computational and information-theoretic differential privacy seem closer in power. Indeed, Groce et al. [56] showed that, in the case of real-valued outputs, we can often convert computational differentially private mechanisms into information-theoretically differentially private mechanisms.

Theorem 7.10.4 (From computational to information-theoretic differential privacy [56]). *Let $\mathcal{M} : \mathcal{X}^n \to \mathbb{R}$ be an ε-computationally differentially private mechanism with the property that, for every dataset $x \in \mathcal{X}^n$, there is an interval I_x of width at most $w(n)$ such that $\Pr[\mathcal{M}(x) \notin I_x] \leq \mathrm{neg}(n)$, and the endpoints of I_x are rational numbers with $\mathrm{poly}(n)$ bits of precision. Define $\mathcal{M}'(x)$ to be the mechanism that runs $\mathcal{M}(x)$ and rounds the result to the nearest multiple of $\alpha(n) = w(n)/n^c$, for any desired constant c. Then $\mathcal{M}'$ is $(\varepsilon, \mathrm{neg}(n))$-differentially private.*

Thus, the error incurred is an arbitrary polynomial small fraction of the "spread" of $\mathcal{M}$'s outputs.

Proof: Let I'_x denote the rounding of all points in I_x to the nearest multiple of $\alpha(n)$; notice that $|I'_x| \leq w(n)/\alpha(n) + 1 \leq n^c + 1$. $\mathcal{M}'$ is computationally differentially private because $\mathcal{M}$ is, and we will use this to show that it is actually information-theoretically differential private: For every $x, x' \in \mathcal{X}^n$ that differ on one row and every $T \subseteq \mathbb{R}$, we have

$$\begin{aligned}
\Pr[\mathcal{M}'(x) \in T] &\leq \left(\sum_{y \in I'_x \cap T} \Pr[\mathcal{M}'(x) = y] \right) + \Pr[\mathcal{M}'(x) \notin I'_x] \\
&\leq \left(\sum_{y \in I'_x \cap T} \left(\mathrm{e}^{\varepsilon} \cdot \Pr[\mathcal{M}'(x') = y] + \mathrm{neg}(n) \right) \right) + \mathrm{neg}(n) \\
&\leq \mathrm{e}^{\varepsilon} \cdot \Pr[\mathcal{M}'(x') \in T] + (n^c + 1) \cdot \mathrm{neg}(n) + \mathrm{neg}(n) \\
&= \mathrm{e}^{\varepsilon} \cdot \Pr[\mathcal{M}'(x') \in T] + \mathrm{neg}(n),
\end{aligned}$$

where the second inequality uses the fact that testing equality with a fixed value y or testing membership in an interval can be done by polynomial-sized circuits, provided the numbers have only $\mathrm{poly}(n)$ bits of precision. ∎

This proof technique extends to low-dimensional outputs (e.g., answering a logarithmic number of real-valued queries) as well as outputs in polynomial-sized discrete sets [56, 23]. So to get a separation between computational and information-theoretic differential privacy with a single curator, we need to use large or high-

dimensional output spaces, or measure utility in a different way (not by a low-dimensional metric). Such a separation was recently obtained by Bun et al. [23]:

Theorem 7.10.5 (Separating computational and information-theoretic differentially private curators [23]). *Assuming the existence of subexponentially secure one-way functions and "exponentially extractable noninteractive witness indistinguishable (NIWI) proofs for NP", there exists an efficiently computable utility function* $u : \mathcal{X}^n \times \mathcal{Y} \to \{0, 1\}$ *such that*

1. *There exists a polynomial-time CDP mechanism* $\mathcal{M}^{\mathrm{CDP}}$ *such that, for every dataset* $x \in \mathcal{X}^n$*, we have* $\Pr[u(x, \mathcal{M}^{\mathrm{CDP}}(x)) = 1] \geq 2/3$.
2. *There exists a computationally unbounded differentially private mechanism* $\mathcal{M}^{\mathrm{unb}}$ *such that, for every dataset* $x \in \mathcal{X}^n$*, we have* $\Pr[u(x, \mathcal{M}^{\mathrm{unb}}(x)) = 1] \geq 2/3$.
3. *For every polynomial-time differentially private* $\mathcal{M}$*, there exists a dataset* $x \in \mathcal{X}^n$ *such that* $\Pr[u(x, \mathcal{M}(x)) = 1] \leq 1/3$.

Note that this theorem provides a task where achieving information-theoretic differential privacy is infeasible—not impossible. Moreover, it is for a rather unnatural, cryptographic utility function u. It would be interesting to overcome either of these limitations:

Open Problem 7.10.6. Is there a computational task that is solvable by a single curator with computational differential privacy but is *impossible* to solve with information-theoretic differential privacy?

Open Problem 7.10.7. Can an analogue of Theorem 7.10.5 be proven for a more "natural" utility function u, such as one that measures the error in answering or summarizing the answers to a set of counting queries?

7.10.4 Relation to Pseudodensity

The definition of computational differential privacy is related to concepts studied in the literature on pseudorandomness. For random variables Y, Z taking values in $\mathcal{Y}$ and $\rho \in [0, 1]$, we say that Y has *density at least* ρ in Z if, for every event $T \subseteq \mathcal{Y}$, we have

$$\rho \cdot \Pr[Y \in T] \leq \Pr[Z \in T].$$

For intuition, suppose that Y and Z are uniform on their supports. Then this definition says that $\mathrm{Supp}(Y) \subseteq \mathrm{Supp}(Z)$ and $|\mathrm{Supp}(Y)| \geq \rho \cdot |\mathrm{Supp}(Z)|$. Additionally, if Z is the uniform distribution on $\mathcal{Y}$, then Y having density at least ρ in Z is equivalent to Y having "min-entropy" at least $\log(\rho|\mathcal{Y}|)$. Notice that a mechanism $\mathcal{M}$ is $(\varepsilon, 0)$-differentially private iff, for every two neighboring datasets $x \sim x'$, $\mathcal{M}(x)$ has density at least $e^{-\varepsilon}$ in $\mathcal{M}(x')$.

Just like computational analogues of statistical distance (namely, computational indistinguishability and pseudorandomness) have proven to be powerful concepts in computational complexity and cryptography, computational analogues of density

and min-entropy have also turned out to be quite useful, with applications including additive number theory [55], leakage-resilient cryptography [49], and constructions of cryptographic primitives from one-way functions [62].

One of the computational analogues of density that has been studied, called *pseudodensity* (or sometimes *metric entropy* when Z is uniform on $\mathcal{Y}$) [3, 90], is precisely the one used in the definition of computational differential privacy, namely that, for every polynomial-sized Boolean circuit T, we have

$$\rho \cdot \Pr[T(Y) = 1] \leq \Pr[T(Z) = 1] + \text{neg}(n).$$

When considering a *single pair* of random variables (Y, Z), the dense model theorem of [55, 100, 90] says that pseudodensity is *equivalent* to Y being computationally indistinguishable from a random variable $\tilde{Y}$ that truly has density at least ρ in Z. Mironov et al. [81] asked whether something similar can be said about (computationally) differentially private mechanisms, which require (pseudo)density *simultaneously* for all pairs $\mathcal{M}(x), \mathcal{M}(x')$ where $x \sim x'$:

Open Problem 7.10.8. For every ε-computationally differentially private and polynomial-time computable mechanism $\mathcal{M} : \mathcal{X}^n \to \mathcal{Y}$, is there an $(O(\varepsilon), \text{neg}(n))$-differentially private mechanism $\tilde{\mathcal{M}} : \mathcal{X}^n \to \mathcal{Y}$ such that, for all datasets $x \in \mathcal{X}^n$, $\mathcal{M}(x)$ is computationally indistinguishable from $\tilde{\mathcal{M}}(x)$?

A positive answer to this question would imply a negative answer to Open Problem 7.10.6.

7.11 Conclusions

We have illustrated rich connections between the theory of differential privacy and numerous topics in theoretical computer science and mathematics, such as learning theory, convex geometry and optimization, cryptographic tools for preventing piracy, probabilistically checkable proofs and approximability, randomness extractors, information complexity, secure multiparty computation, and notions of pseudoentropy. There have also been very fruitful interactions with other areas. In particular, in both game theory and in statistics, differential privacy has proved to be a powerful tool for some applications where privacy is not the goal—such as designing approximately truthful mechanisms [79, 87] and preventing false discovery in adaptive data analysis [44]. Remarkably, both positive and negative results for differential privacy (including both information-theoretic and computational lower bounds as we have seen in this tutorial) have found analogues for the false discovery problem [44, 60, 98, 6], suggesting that it will also be a very fertile area for complexity-theoretic investigation.

We now mention some more directions for future work in differential privacy, beyond the many open problems stated in earlier sections. As illustrated in previous sections, there has been a thorough investigation of the complexity of answering *counting queries* under differential privacy, with many algorithms and lower bounds that provide nearly matching results. While there remain numerous important open

questions, it would also be good to develop a similar kind of understanding for other types of computations. There is now a wide literature on differentially private algorithms for many types of data analysis tasks, but what is missing are negative results to delineate the border between possible and impossible.

Open Problem 7.11.1. Classify large classes of problems (other than counting queries) in differential privacy according to their privacy–utility tradeoffs and their computational tractability.

Two areas of particular interest, both in theory and in practice, are:

Statistical inference and machine learning. In this tutorial, we have mostly been measuring accuracy relative to the particular (worst-case) dataset that is given as input to our differentially private algorithm. However, in statistical inference and machine learning, the goal is usually to infer properties of the *population* from which the dataset is (randomly) drawn. The PAC model studied in Section 7.8 is a theoretically appealing framework in which to study how such tasks can be done with differential privacy, but there are many inference and learning problems outside the PAC model that are also of great interest. These problems include tasks like hypothesis testing, parameter estimation, regression, and distribution learning, and a variety of utility measures such as convergence rates, p values, risk minimization, and sizes of confidence intervals. Moreover, the data distributions are often assumed to have a significant amount of structure (or enough samples are taken for central limit theorems to provide such structure), in contrast to the worst-case distributions considered in the PAC model. Some broad positive results are provided in Smith [95] and Bassily et al. [5] and some negative results in [32, 21, 5], but our understanding of these types of problems is still quite incomplete.

Graph privacy. As mentioned in Section 7.3, there has been some very interesting work on differentially private graph analysis, where our dataset is a graph and we are interested in protecting either relationships (edge-level privacy) or everything about an individual/vertex (node-level privacy). We refer to Raskhodnikova and Smith [88] for a broader survey of the area. Again, most of the work to date has been algorithmic, and we still lack a systematic understanding of impossibility and intractability.

If the existing study of differential privacy is any indication, these studies are likely to uncover a rich theoretical landscape, with even more connections to the rest of theoretical computer science.

Acknowledgements This tutorial was written starting from notes taken during a minicourse given at the 26th McGill Invitational Workshop on Computational Complexity in February 2014, at the Bellairs Institute in Holetown, Barbados [1]. Special thanks go to Kunal Talwar for giving several of the lectures (leading to material in Sections 7.5.1 and 7.7.3 here), to the workshop attendees who wrote up the lecture notes (Eric Allender, Borja Balle, Anne-Sophie Charest, Lila Fontes, Antonina Kolokolova, Swastik Kopparty, Michal Koucký, Cristopher Moore, Shubhangi Saraf, and Luc Segoufin), to Alexander Russell for collecting and editing the notes, to Denis Thérien for

organizing the workshop, to all of the participants for illuminating comments during the workshop, and to Sammy Innis for surfing lessons.

I am grateful to Cynthia Dwork, Ilya Mironov, and Guy Rothblum, who got me started on differential privacy during a month-long visit to the wonderful (sadly, now defunct) Microsoft Research Silicon Valley lab in 2008, and numerous collaborators and students since then, whose insights are reflected throughout this tutorial.

I thank Mark Bun, Iftach Haitner, Jack Murtagh, Sasho Nikolov, Adam D. Smith, and Uri Stemmer for extensive comments and corrections that have improved the tutorial. I also thank Yehuda Lindell for his leadership in producing this volume of tutorials in Oded's honor, and for his patience with all of my delays.

References

[1] Lecture notes for the 26th McGill Invitational Workshop on Computational Complexity, February 2014. Lectures given by Salil Vadhan and Kunal Talwar. Notes edited by Alexander Russell.

[2] Victor Balcer and Salil Vadhan. Efficient algorithms for differentially private histograms with worst-case accuracy over large domains. Manuscript, February 2017.

[3] Boaz Barak, Ronen Shaltiel, and Avi Wigderson. Computational analogues of entropy. In *Approximation, randomization, and combinatorial optimization*, volume 2764 of *Lecture Notes in Comput. Sci.*, pages 200–215. Springer, Berlin, 2003. doi: 10.1007/978-3-540-45198-3_18. URL http://dx.doi.org/10.1007/978-3-540-45198-3_18.

[4] Raef Bassily and Adam Smith. Local, private, efficient protocols for succinct histograms. In *STOC'15—Proceedings of the 2015 ACM Symposium on Theory of Computing*, pages 127–135. ACM, New York, 2015.

[5] Raef Bassily, Adam Smith, and Abhradeep Thakurta. Private empirical risk minimization: efficient algorithms and tight error bounds. In *55th Annual IEEE Symposium on Foundations of Computer Science—FOCS 2014*, pages 464–473. IEEE Computer Soc., Los Alamitos, CA, 2014. doi: 10.1109/FOCS.2014.56. URL http://dx.doi.org/10.1109/FOCS.2014.56.

[6] Raef Bassily, Kobbi Nissim, Adam Smith, Thomas Steinke, Uri Stemmer, and Jonathan Ullman. Algorithmic stability for adaptive data analysis. In *48th Annual Symposium on the Theory of Computing (STOC'16)*, June 2016. Preliminary version available at http://arxiv.org/abs/1511.02513.

[7] Amos Beimel, Kobbi Nissim, and Eran Omri. Distributed private data analysis: simultaneously solving how and what. In *Advances in cryptology—CRYPTO 2008*, volume 5157 of *Lecture Notes in Comput. Sci.*, pages 451–468. Springer, Berlin, 2008. doi: 10.1007/978-3-540-85174-5_25. URL http://dx.doi.org/10.1007/978-3-540-85174-5_25.

[8] Amos Beimel, Kobbi Nissim, and Uri Stemmer. Characterizing the sample complexity of private learners. In *ITCS'13—Proceedings of the 2013 ACM Conference on Innovations in Theoretical Computer Science*, pages 97–109. ACM, New York, 2013.

[9] Amos Beimel, Kobbi Nissim, and Uri Stemmer. Private learning and sanitization: Pure vs. approximate differential privacy. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques*, pages 363–378. Springer, 2013.

[10] Amos Beimel, Hai Brenner, Shiva Prasad Kasiviswanathan, and Kobbi Nissim. Bounds on the sample complexity for private learning and private data release. *Machine Learning*, 94(3):401–437, 2014. ISSN 0885-6125. doi: 10.1007/s10994-013-5404-1. URL http://dx.doi.org/10.1007/s10994-013-5404-1.

[11] Aditya Bhaskara, Daniel Dadush, Ravishankar Krishnaswamy, and Kunal Talwar. Unconditional differentially private mechanisms for linear queries.

In *STOC'12—Proceedings of the 2012 ACM Symposium on Theory of Computing*, pages 1269–1283. ACM, New York, 2012. doi: 10.1145/2213977.2214089. URL http://dx.doi.org/10.1145/2213977.2214089.

[12] Jeremiah Blocki, Avrim Blum, Anupam Datta, and Or Sheffet. Differentially private data analysis of social networks via restricted sensitivity. In *ITCS'13—Proceedings of the 2013 ACM Conference on Innovations in Theoretical Computer Science*, pages 87–96. ACM, New York, 2013.

[13] Avrim Blum, Cynthia Dwork, Frank McSherry, and Kobbi Nissim. Practical privacy: the SuLQ framework. In *Proceedings of the 24th ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems*, pages 128–138. ACM, 2005.

[14] Avrim Blum, Katrina Ligett, and Aaron Roth. A learning theory approach to noninteractive database privacy. *Journal of the ACM*, 60(2):Art. 12, 25, 2013. ISSN 0004-5411. doi: 10.1145/2450142.2450148. URL http://dx.doi.org/10.1145/2450142.2450148.

[15] Dan Boneh and James Shaw. Collusion-secure fingerprinting for digital data. *IEEE Transactions on Information Theory*, 44(5):1897–1905, Sep 1998.

[16] Dan Boneh and Mark Zhandry. Multiparty key exchange, efficient traitor tracing, and more from indistinguishability obfuscation. In *Advances in cryptology—CRYPTO 2014. Part I*, volume 8616 of *Lecture Notes in Comput. Sci.*, pages 480–499. Springer, Heidelberg, 2014. doi: 10.1007/978-3-662-44371-2_27. URL http://dx.doi.org/10.1007/978-3-662-44371-2_27.

[17] Dan Boneh, Amit Sahai, and Brent Waters. Fully collusion resistant traitor tracing with short ciphertexts and private keys. In *Advances in cryptology—EUROCRYPT 2006*, volume 4004 of *Lecture Notes in Comput. Sci.*, pages 573–592. Springer, Berlin, 2006. doi: 10.1007/11761679_34. URL http://dx.doi.org/10.1007/11761679_34.

[18] Mark Bun. *New Separations in the Complexity of Differential Privacy*. PhD thesis, Harvard University, August 2016.

[19] Mark Bun and Thomas Steinke. Concentrated differential privacy: Simplifications, extensions, and lower bounds. *CoRR*, abs/1605.02065, 2016. URL http://arxiv.org/abs/1605.02065.

[20] Mark Bun and Mark Zhandry. Order-revealing encryption and the hardness of private learning. In *Theory of Cryptography Conference (TCC '16A)*, pages 176–206. Springer, 10–13 January 2016. Full version available at https://eprint.iacr.org/2015/417.

[21] Mark Bun, Jonathan Ullman, and Salil Vadhan. Fingerprinting codes and the price of approximate differential privacy. In *Proceedings of the 46th Annual ACM Symposium on Theory of Computing*, STOC '14, pages 1–10, New York, NY, USA, 2014. ACM.

[22] Mark Bun, Kobbi Nissim, Uri Stemmer, and Salil Vadhan. Differentially private release and learning of threshold functions. In *Proceedings of the 56th Annual IEEE Symposium on Foundations of Computer Science (FOCS*

2015), pages 634–649. IEEE, 18–20 October 2015. Full version posted as arXiv:1504.07553.

[23] Mark Bun, Yi-Hsiu Chen, and Salil Vadhan. Separating computational and statistical differential privacy in the client-server model. In Martin Hirt and Adam D. Smith, editors, *Proceedings of the 14th IACR Theory of Cryptography Conference (TCC '16-B)*, Lecture Notes in Computer Science. Springer-Verlag, 31 October–3 November 2016. Full version posted on *Cryptology ePrint Archive*, Report 2016/820.

[24] Mark Bun, Kobbi Nissim, and Uri Stemmer. Simultaneous private learning of multiple concepts. In *Innovations in Theoretical Computer Science (ITCS '16)*, pages 369–380. ACM, 14–16 January 2016. Full version available at http://arxiv.org/abs/1511.08552.

[25] T.-H. Hubert Chan, Elaine Shi, and Dawn Song. Optimal lower bound for differentially private multi-party aggregation. In Leah Epstein and Paolo Ferragina, editors, *Algorithms - ESA 2012 - 20th Annual European Symposium, Ljubljana, Slovenia, September 10-12, 2012. Proceedings*, volume 7501 of *Lecture Notes in Computer Science*, pages 277–288. Springer, 2012. ISBN 978-3-642-33089-6. doi: 10.1007/978-3-642-33090-2_25. URL http://dx.doi.org/10.1007/978-3-642-33090-2_25.

[26] Karthekeyan Chandrasekaran, Justin Thaler, Jonathan Ullman, and Andrew Wan. Faster private release of marginals on small databases. In *ITCS*, pages 387–402, 2014. doi: 10.1145/2554797.2554833.

[27] Shixi Chen and Shuigeng Zhou. Recursive mechanism: Towards node differential privacy and unrestricted joins. In *Proceedings of the 2013 ACM SIGMOD International Conference on Management of Data*, SIGMOD '13, pages 653–664, New York, NY, USA, 2013. ACM. ISBN 978-1-4503-2037-5. doi: 10.1145/2463676.2465304. URL http://doi.acm.org/10.1145/2463676.2465304.

[28] Benny Chor, A Fiat, M. Naor, and B. Pinkas. Tracing traitors. *IEEE Transactions on Information Theory*, 46(3):893–910, May 2000.

[29] Anindya De. Lower bounds in differential privacy. In *Theory of cryptography*, volume 7194 of *Lecture Notes in Comput. Sci.*, pages 321–338. Springer, Heidelberg, 2012. doi: 10.1007/978-3-642-28914-9_18. URL http://dx.doi.org/10.1007/978-3-642-28914-9_18.

[30] Irit Dinur, editor. *IEEE 57th Annual Symposium on Foundations of Computer Science, FOCS 2016, 9-11 October 2016, New Brunswick, New Jersey, USA*, 2016. IEEE Computer Society. ISBN 978-1-5090-3933-3. URL http://ieeexplore.ieee.org/xpl/mostRecentIssue.jsp?punumber=7781469.

[31] Irit Dinur and Kobbi Nissim. Revealing information while preserving privacy. In *Proceedings of the Twenty-second ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems*, PODS '03, pages 202–210, New York, NY, USA, 2003. ACM. doi: 10.1145/773153.773173.

[32] John C. Duchi, Michael I. Jordan, and Martin J. Wainwright. Privacy aware learning. *Journal of the ACM*, 61(6):Art. 38, 57, 2014. ISSN 0004-5411. doi: 10.1145/2666468. URL http://dx.doi.org/10.1145/2666468.

[33] Cynthia Dwork. Differential privacy. In *Automata, languages and programming. Part II*, volume 4052 of *Lecture Notes in Comput. Sci.*, pages 1–12. Springer, Berlin, 2006. doi: 10.1007/11787006_1. URL http://dx.doi.org/10.1007/11787006_1.

[34] Cynthia Dwork and Jing Lei. Differential privacy and robust statistics. In *STOC'09—Proceedings of the 2009 ACM International Symposium on Theory of Computing*, pages 371–380. ACM, New York, 2009.

[35] Cynthia Dwork and Kobbi Nissim. Privacy-preserving datamining on vertically partitioned databases. In *Advances in Cryptology–CRYPTO 2004*, pages 528–544. Springer, 2004.

[36] Cynthia Dwork and Aaron Roth. The algorithmic foundations of differential privacy. *Foundations and Trends in Theoretical Computer Science*, 9(34): 211–407, 2013. ISSN 1551-305X. doi: 10.1561/0400000042. URL http://dx.doi.org/10.1561/0400000042.

[37] Cynthia Dwork and Sergey Yekhanin. New efficient attacks on statistical disclosure control mechanisms. In *Advances in cryptology—CRYPTO 2008*, volume 5157 of *Lecture Notes in Comput. Sci.*, pages 469–480. Springer, Berlin, 2008. doi: 10.1007/978-3-540-85174-5_26. URL http://dx.doi.org/10.1007/978-3-540-85174-5_26.

[38] Cynthia Dwork, Krishnaram Kenthapadi, Frank McSherry, Ilya Mironov, and Moni Naor. Our data, ourselves: privacy via distributed noise generation. In *Advances in cryptology—EUROCRYPT 2006*, volume 4004 of *Lecture Notes in Comput. Sci.*, pages 486–503. Springer, Berlin, 2006. doi: 10.1007/11761679_29. URL http://dx.doi.org/10.1007/11761679_29.

[39] Cynthia Dwork, Frank McSherry, and Kunal Talwar. The price of privacy and the limits of LP decoding. In *Proceedings of the 39th Annual ACM Symposium on Theory of Computing (STOC)*, pages 85–94, San Diego, California, USA, June 2007. Association for Computing Machinery, Inc. URL http://research.microsoft.com/apps/pubs/default.aspx?id=64343.

[40] Cynthia Dwork, Moni Naor, Omer Reingold, Guy N. Rothblum, and Salil Vadhan. On the complexity of differentially private data release: Efficient algorithms and hardness results. In *Proceedings of the Forty-first Annual ACM Symposium on Theory of Computing*, STOC '09, pages 381–390, New York, NY, USA, 2009. ACM.

[41] Cynthia Dwork, Moni Naor, Toniann Pitassi, and Guy N. Rothblum. Differential privacy under continual observation. In *STOC'10—Proceedings of the 2010 ACM International Symposium on Theory of Computing*, pages 715–724. ACM, New York, 2010.

[42] Cynthia Dwork, Guy Rothblum, and Salil Vadhan. Boosting and differential privacy. In *Proceedings of the 51st Annual IEEE Symposium on Foundations of Computer Science (FOCS 2010)*, pages 51–60. IEEE, 23–26 October 2010.

[43] Cynthia Dwork, Moni Naor, and Salil Vadhan. The privacy of the analyst and the power of the state. In *Proceedings of the 53rd Annual IEEE Symposium on Foundations of Computer Science (FOCS 2012)*, pages 400–409. IEEE, 20–23 October 2012.

[44] Cynthia Dwork, Vitaly Feldman, Moritz Hardt, Toniann Pitassi, Omer Reingold, and Aaron Roth. Preserving statistical validity in adaptive data analysis [extended abstract]. In *STOC'15—Proceedings of the 2015 ACM Symposium on Theory of Computing*, pages 117–126. ACM, New York, 2015.

[45] Cynthia Dwork, Moni Naor, Omer Reingold, and Guy N. Rothblum. Pure differential privacy for rectangle queries via private partitions. In Tetsu Iwata and Jung Hee Cheon, editors, *Advances in Cryptology - ASIACRYPT 2015 - 21st International Conference on the Theory and Application of Cryptology and Information Security, Auckland, New Zealand, November 29 - December 3, 2015, Proceedings, Part II*, volume 9453 of *Lecture Notes in Computer Science*, pages 735–751. Springer, 2015. ISBN 978-3-662-48799-0. doi: 10.1007/978-3-662-48800-3_30. URL http://dx.doi.org/10.1007/978-3-662-48800-3_30.

[46] Cynthia Dwork, Aleksandar Nikolov, and Kunal Talwar. Efficient algorithms for privately releasing marginals via convex relaxations. *Discrete Comput. Geom.*, 53(3):650–673, 2015. ISSN 0179-5376. doi: 10.1007/s00454-015-9678-x. URL http://dx.doi.org/10.1007/s00454-015-9678-x.

[47] Cynthia Dwork, Adam Smith, Thomas Steinke, Jonathan Ullman, and Salil Vadhan. Robust traceability from trace amounts. In *Proceedings of the 56th Annual IEEE Symposium on Foundations of Computer Science (FOCS '15)*, pages 650–669. IEEE, 18–20 October 2015.

[48] Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. *Journal of Privacy and Confidentiality*, 7(3), 2016. To appear. Preliminary version in *Proc. TCC '06*.

[49] Stefan Dziembowski and Krzysztof Pietrzak. Leakage-resilient cryptography. In *FOCS*, pages 293–302. IEEE Computer Society, 2008. ISBN 978-0-7695-3436-7. URL http://ieeexplore.ieee.org/xpl/mostRecentIssue.jsp?punumber=4690923.

[50] Vitaly Feldman and David Xiao. Sample complexity bounds on differentially private learning via communication complexity. In *Proceedings of COLT 2014*, pages 1000–1019, 2014.

[51] Sanjam Garg, Craig Gentry, Shai Halevi, Mariana Raykova, Amit Sahai, and Brent Waters. Candidate indistinguishability obfuscation and functional encryption for all circuits. In *Foundations of Computer Science (FOCS), 2013 IEEE 54th Annual Symposium on*, pages 40–49. IEEE, 2013.

[52] Oded Goldreich. *Foundations of cryptography. II*. Cambridge University Press, Cambridge, 2004. ISBN 0-521-83084-2. doi: 10.1017/CBO9780511721656.002. URL http://dx.doi.org/10.1017/CBO9780511721656.002. Basic Applications.

[53] Oded Goldreich, Silvio Micali, and Avi Wigderson. How to play any mental game or a completeness theorem for protocols with honest majority. In Alfred V. Aho, editor, *Proceedings of the 19th Annual ACM Symposium on Theory of Computing, 1987, New York, New York, USA*, pages 218–229. ACM, 1987. ISBN 0-89791-221-7. doi: 10.1145/28395.28420. URL http://doi.acm.org/10.1145/28395.28420.

[54] Vipul Goyal, Dakshita Khurana, Ilya Mironov, Omkant Pandey, and Amit Sahai. Do distributed differentially-private protocols require oblivious transfer? *IACR Cryptology ePrint Archive*, 2015:1090, 2015. URL http://eprint.iacr.org/2015/1090.

[55] Ben Green and Terence Tao. The primes contain arbitrarily long arithmetic progressions. *Annals of Mathematics. Second Series*, 167(2):481–547, 2008. ISSN 0003-486X. doi: 10.4007/annals.2008.167.481. URL http://dx.doi.org/10.4007/annals.2008.167.481.

[56] Adam Groce, Jonathan Katz, and Arkady Yerukhimovich. Limits of computational differential privacy in the client/server setting. In *Theory of cryptography*, volume 6597 of *Lecture Notes in Comput. Sci.*, pages 417–431. Springer, Heidelberg, 2011. doi: 10.1007/978-3-642-19571-6_25. URL http://dx.doi.org/10.1007/978-3-642-19571-6_25.

[57] Iftach Haitner, Eran Omri, and Hila Zarosim. Limits on the usefulness of random oracles. *Journal of Cryptology*, 29(2):283–335, 2016. ISSN 0933-2790. doi: 10.1007/s00145-014-9194-9. URL http://dx.doi.org/10.1007/s00145-014-9194-9.

[58] Moritz Hardt and Guy N. Rothblum. A multiplicative weights mechanism for privacy-preserving data analysis. In *Proceedings of the 51st Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 61–70, Oct 2010. doi: 10.1109/FOCS.2010.85.

[59] Moritz Hardt and Kunal Talwar. On the geometry of differential privacy. In *STOC'10—Proceedings of the 2010 ACM International Symposium on Theory of Computing*, pages 705–714. ACM, New York, 2010.

[60] Moritz Hardt and Jonathan Ullman. Preventing false discovery in interactive data analysis is hard. In *Symposium on Foundations of Computer Science (FOCS '14)*, pages 454–463. IEEE, Oct 18–21 2014.

[61] Moritz Hardt, Katrina Ligett, and Frank Mcsherry. A simple and practical algorithm for differentially private data release. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 2339–2347. Curran Associates, Inc., 2012.

[62] Johan Håstad, Russell Impagliazzo, Leonid A. Levin, and Michael Luby. A pseudorandom generator from any one-way function. *SIAM Journal on Computing*, 28(4):1364–1396 (electronic), 1999. ISSN 1095-7111.

[63] Nils Homer, Szabolcs Szelinger, Margot Redman, David Duggan, Waibhav Tembe, Jill Muehling, John V. Pearson, Dietrich A. Stephan, Stanley F. Nelson, and David W. Craig. Resolving individuals contributing trace amounts of DNA to highly complex mixtures using high-density SNP genotyping microarrays. *PLoS genetics*, 4(8):e1000167, 2008.

[64] Peter Kairouz, Sewoong Oh, and Pramod Viswanath. The composition theorem for differential privacy. In Francis R. Bach and David M. Blei, editors, *Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, Lille, France, 6-11 July 2015*, volume 37 of *JMLR Proceedings*, pages 1376–1385. JMLR.org, 2015. URL http://jmlr.org/proceedings/papers/v37/kairouz15.html.

[65] Shiva P. Kasiviswanathan and Adam Smith. On the 'semantics' of differential privacy: A bayesian formulation. *Journal of Privacy and Confidentiality*, 6 (1), 2014.

[66] Shiva Prasad Kasiviswanathan, Mark Rudelson, Adam Smith, and Jonathan Ullman. The price of privately releasing contingency tables and the spectra of random matrices with correlated rows. In *STOC'10—Proceedings of the 2010 ACM International Symposium on Theory of Computing*, pages 775–784. ACM, New York, 2010.

[67] Shiva Prasad Kasiviswanathan, Homin K. Lee, Kobbi Nissim, Sofya Raskhodnikova, and Adam Smith. What can we learn privately? *SIAM J. Comput.*, 40(3):793–826, 2011. doi: 10.1137/090756090.

[68] Shiva Prasad Kasiviswanathan, Kobbi Nissim, Sofya Raskhodnikova, and Adam Smith. Analyzing graphs with node differential privacy. In *TCC*, pages 457–476, 2013. doi: 10.1007/978-3-642-36594-2_26. URL http://dx.doi.org/10.1007/978-3-642-36594-2_26.

[69] Michael J. Kearns. Efficient noise-tolerant learning from statistical queries. *Journal of the Association for Computing Machinery*, 45(6):983–1006, 1998. doi: 10.1145/293347.293351. URL http://doi.acm.org/10.1145/293347.293351.

[70] Michael J. Kearns and Umesh V. Vazirani. *An introduction to computational learning theory*. MIT Press, Cambridge, MA, 1994. ISBN 0-262-11193-4.

[71] Subhash Khot and Assaf Naor. Grothendieck-type inequalities in combinatorial optimization. *Communications on Pure and Applied Mathematics*, 65(7):992–1035, 2012. ISSN 0010-3640. doi: 10.1002/cpa.21398. URL http://dx.doi.org/10.1002/cpa.21398.

[72] Lucas Kowalczyk, Tal Malkin, Jonathan Ullman, and Mark Zhandry. Strong hardness of privacy from weak traitor tracing. In Martin Hirt and Adam D. Smith, editors, *Theory of Cryptography - 14th International Conference, TCC 2016-B, Beijing, China, October 31 - November 3, 2016, Proceedings, Part I*, volume 9985 of *Lecture Notes in Computer Science*, pages 659–689, 2016. ISBN 978-3-662-53640-7. doi: 10.1007/978-3-662-53641-4_25. URL http://dx.doi.org/10.1007/978-3-662-53641-4_25.

[73] Ilan Kremer, Noam Nisan, and Dana Ron. On randomized one-round communication complexity. In *Proceedings of STOC 1995*, pages 596–605, 1995.

[74] Yehuda Lindell and Benny Pinkas. Secure multiparty computation for privacy-preserving data mining. *Journal of Privacy and Confidentiality*, 1 (1), 2009. URL http://repository.cmu.edu/jpc/vol1/iss1/5.

[75] Jiří Matoušek. *Geometric discrepancy*, volume 18 of *Algorithms and Combinatorics*. Springer-Verlag, Berlin, 2010. ISBN 978-3-642-03941-6.

doi: 10.1007/978-3-642-03942-3. URL http://dx.doi.org/10.1007/978-3-642-03942-3. An illustrated guide, Revised paperback reprint of the 1999 original.

[76] Jiří Matoušek and Jan Vondrák. The probabilistic method (lecture notes). http://kam.mff.cuni.cz/~matousek/lectnotes.html, March 2008.

[77] Jiří Matoušek, Aleksandar Nikolov, and Kunal Talwar. Factorization norms and hereditary discrepancy. arXiv:1408.1376 [math.CO], August 2014.

[78] Andrew McGregor, Ilya Mironov, Toniann Pitassi, Omer Reingold, Kunal Talwar, and Salil Vadhan. The limits of two-party differential privacy. In *Proceedings of the 51st Annual IEEE Symposium on Foundations of Computer Science (FOCS '10)*, pages 81–90. IEEE, 23–26 October 2010.

[79] Frank McSherry and Kunal Talwar. Mechanism design via differential privacy. In *Proceedings of the 48th Annual IEEE Symposium on Foundations of Computer Science*, FOCS '07, pages 94–103, Washington, DC, USA, 2007. IEEE Computer Society. doi: 10.1109/FOCS.2007.41.

[80] P. B. Miltersen, N. Nisan, S. Safra, and A. Wigderson. On data structures and asymmetric communication complexity. *J. Computer & System Sciences*, 57 (1):37–49, 1998.

[81] Ilya Mironov, Omkant Pandey, Omer Reingold, and Salil Vadhan. Computational differential privacy. In S. Halevi, editor, *Advances in Cryptology—CRYPTO '09*, volume 5677 of *Lecture Notes in Computer Science*, pages 126–142. Springer-Verlag, 16–20 August 2009.

[82] Jack Murtagh and Salil Vadhan. The complexity of computing the optimal composition of differential privacy. In Eyal Kushilevitz and Tal Malkin, editors, *Proceedings of the 13th IACR Theory of Cryptography Conference (TCC '16-A)*, volume 9562 of *Lecture Notes in Computer Science*, pages 157–175. Springer-Verlag, 10–13 January 2016. ISBN 978-3-662-49095-2. doi: 10.1007/978-3-662-49096-9. URL http://dx.doi.org/10.1007/978-3-662-49096-9. Full version posted on *CoRR*, abs/1507.03113, July 2015.

[83] S. Muthukrishnan and Aleksandar Nikolov. Optimal private halfspace counting via discrepancy. In *Proceedings of the Forty-fourth Annual ACM Symposium on Theory of Computing*, STOC '12, pages 1285–1292, New York, NY, USA, 2012. ACM. doi: 10.1145/2213977.2214090.

[84] Arvind Narayanan, Joanna Huey, and Edward W. Felten. A precautionary approach to big data privacy. In *Computers, Privacy & Data Protection*, 2015.

[85] Aleksandar Nikolov, Kunal Talwar, and Li Zhang. The geometry of differential privacy: the small database and approximate cases. *SIAM Journal on Computing*, 45(2):575–616, 2016. ISSN 0097-5397. doi: 10.1137/130938943. URL http://dx.doi.org/10.1137/130938943. Preliminary version in *Proc. STOC 2013*.

[86] Kobbi Nissim, Sofya Raskhodnikova, and Adam Smith. Smooth sensitivity and sampling in private data analysis. In *STOC'07—Proceedings of the 39th Annual ACM Symposium on Theory of Computing*, pages 75–84. ACM, New

York, 2007. doi: 10.1145/1250790.1250803. URL http://dx.doi.org/10.1145/1250790.1250803.

[87] Mallesh M. Pai and Aaron Roth. Privacy and mechanism design. *SIGecom Exch.*, 12(1):8–29, June 2013. ISSN 1551-9031. doi: 10.1145/2509013.2509016. URL http://doi.acm.org/10.1145/2509013.2509016.

[88] Sofya Raskhodnikova and Adam Smith. In Ming-Yang Kao, editor, *Encyclopedia of Algorithms*, chapter Private Analysis of Graph Data, pages 1–6. Springer Berlin Heidelberg, Berlin, Heidelberg, 2014. ISBN 978-3-642-27848-8. doi: 10.1007/978-3-642-27848-8_549-1. URL http://dx.doi.org/10.1007/978-3-642-27848-8_549-1.

[89] Sofya Raskhodnikova and Adam D. Smith. Lipschitz extensions for node-private graph statistics and the generalized exponential mechanism. In I. Dinur [30], pages 495–504. ISBN 978-1-5090-3933-3. doi: 10.1109/FOCS.2016.60. URL http://dx.doi.org/10.1109/FOCS.2016.60.

[90] Omer Reingold, Luca Trevisan, Madhur Tulsiani, and Salil Vadhan. Dense subsets of pseudorandom sets. In *Proceedings of the 49th Annual IEEE Symposium on Foundations of Computer Science (FOCS '08)*, pages 76–85. IEEE, 26–28 October 2008.

[91] Renato Renner and Stefan Wolf. Simple and tight bounds for information reconciliation and privacy amplification. In *Advances in cryptology—ASIACRYPT 2005*, volume 3788 of *Lecture Notes in Comput. Sci.*, pages 199–216. Springer, Berlin, 2005. doi: 10.1007/11593447_11. URL http://dx.doi.org/10.1007/11593447_11.

[92] Ryan M. Rogers, Aaron Roth, Adam D. Smith, and Om Thakkar. Max-information, differential privacy, and post-selection hypothesis testing. In I. Dinur [30], pages 487–494. ISBN 978-1-5090-3933-3. doi: 10.1109/FOCS.2016.59. URL http://dx.doi.org/10.1109/FOCS.2016.59.

[93] Aaron Roth. The algorithmic foundations of data privacy (course lecture notes). http://www.cis.upenn.edu/~aaroth/courses/privacyF11.html, Fall 2011.

[94] Ronen Shaltiel. An introduction to randomness extractors. In *Automata, languages and programming. Part II*, volume 6756 of *Lecture Notes in Comput. Sci.*, pages 21–41. Springer, Heidelberg, 2011. doi: 10.1007/978-3-642-22012-8_2. URL http://dx.doi.org/10.1007/978-3-642-22012-8_2.

[95] Adam Smith. Privacy-preserving statistical estimation with optimal convergence rates. In Lance Fortnow and Salil P. Vadhan, editors, *Proceedings of the 43rd ACM Symposium on Theory of Computing, STOC 2011, San Jose, CA, USA, 6-8 June 2011*, pages 813–822. ACM, 2011. ISBN 978-1-4503-0691-1. doi: 10.1145/1993636.1993743. URL http://doi.acm.org/10.1145/1993636.1993743.

[96] Adam D. Smith and Abhradeep Guha Thakurta. Differentially private feature selection via stability arguments, and the robustness of the lasso. *Journal of Machine Laerning Research: Workshop and Conference Proceedings*, 30: 1–32, 2013.

[97] Joel Spencer. Six standard deviations suffice. *Transactions of the American Mathematical Society*, 289(2):679–706, 1985. ISSN 0002-9947. doi: 10.2307/2000258. URL http://dx.doi.org/10.2307/2000258.

[98] Thomas Steinke and Jonathan Ullman. Interactive fingerprinting codes and the hardness of preventing false discovery. In *Proceedings of The 28th Conference on Learning Theory (COLT 2015), Paris, France, July 3-6*, pages 1588–1628, 2015. URL http://jmlr.org/proceedings/papers/v40/Steinke15.html. Preliminary version posted as arXiv:1410.1228 [cs.CR].

[99] Thomas Steinke and Jonathan Ullman. Between pure and approximate differential privacy. *Journal of Privacy and Confidentiality*, 7(2):3–22, 2016. Special Issue on TPDP '15. Preliminary version posted as arXiv:1501.06095.

[100] Terence Tao and Tamar Ziegler. The primes contain arbitrarily long polynomial progressions. *Acta Mathematica*, 201(2):213–305, 2008. ISSN 0001-5962. doi: 10.1007/s11511-008-0032-5. URL http://dx.doi.org/10.1007/s11511-008-0032-5.

[101] Gábor Tardos. Optimal probabilistic fingerprint codes. In *Proceedings of the Thirty-fifth Annual ACM Symposium on Theory of Computing*, STOC '03, pages 116–125, New York, NY, USA, 2003. ACM.

[102] Justin Thaler, Jonathan Ullman, and Salil P. Vadhan. Faster algorithms for privately releasing marginals. In *ICALP (1)*, pages 810–821, 2012. doi: 10.1007/978-3-642-31594-7_68.

[103] Jonathan Ullman. Answering $n^{2+o(1)}$ counting queries with differential privacy is hard. In *Proceedings of the 45th annual ACM Symposium on Theory of Computing*, pages 361–370. ACM, 2013.

[104] Jonathan Ullman and Salil Vadhan. PCPs and the hardness of generating private synthetic data. In *Theory of Cryptography*, pages 400–416. Springer, 2011.

[105] Salil P. Vadhan. *Pseudorandomness*, volume 7 (1–3) of *Foundations and Trends in Theoretical Computer Science*. now publishers, December 2012. 336 pages.

[106] Leslie G. Valiant. A theory of the learnable. *Communications of the ACM*, 27(11):1134–1142, 1984.

[107] Umesh V. Vazirani. Strong communication complexity or generating quasirandom sequences from two communicating semirandom sources. *Combinatorica*, 7(4):375–392, 1987.

[108] Stanley L. Warner. Randomized response: A survey technique for eliminating evasive answer bias. *Journal of the American Statistical Association*, 60 (309):63–69, 1965.

[109] Andrew Chi-Chih Yao. Protocols for secure computations (extended abstract). In *23rd Annual Symposium on Foundations of Computer Science, Chicago, Illinois, USA, 3-5 November 1982*, pages 160–164. IEEE Computer Society, 1982. doi: 10.1109/SFCS.1982.38. URL http://dx.doi.org/10.1109/SFCS.1982.38.

[110] Elias A. Zerhouni and Elizabeth G. Nabel. Protecting aggregate genomic data. *Science*, 322(5898):44–44, 2008. ISSN 0036-8075. doi: 10.1126/

science.1165490. URL http://science.sciencemag.org/content/322/5898/44.1.

[111] Jun Zhang, Graham Cormode, Cecilia M. Procopiuc, Divesh Srivastava, and Xiaokui Xiao. Private release of graph statistics using ladder functions. In Timos K. Sellis, Susan B. Davidson, and Zachary G. Ives, editors, *Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data, Melbourne, Victoria, Australia, May 31 - June 4, 2015*, pages 731–745. ACM, 2015. ISBN 978-1-4503-2758-9. doi: 10.1145/2723372.2737785. URL http://doi.acm.org/10.1145/2723372.2737785.

Nomenclature

$\mathrm{Avg}_{j\in T} f(j)$ The average of $f(j)$ over j in the set T, page 396

$\mathcal{M}$ A (randomized) mechanism $\mathcal{M} : \mathcal{X}^n \times \mathcal{Q} \to \mathcal{Y}$ or $\mathcal{M} : \mathcal{X}^n \to \mathcal{Y}$, page 349

$\mathcal{Q}$ A $k \times |\mathcal{X}|$ matrix with $\{0, 1\}$ entries whose rows correspond to a set of counting queries over $\mathcal{X}$. Abusing notation, we also denote this set of counting queries by $\mathcal{Q}$, page 385

$\mathcal{Q}$ A set of queries $q : \mathcal{X}^n \to \mathcal{Y}$, page 349

$\mathcal{Q}_S$ The restriction of counting query family (i.e. $\{0, 1\}$ matrix) $\mathcal{Q}$ to the data universe elements (i.e. columns) in S, page 385

$\mathcal{X}$ A data universe for dataset rows, page 349

$\mathcal{Y}$ A (discrete) output space for a mechanism, page 349

δ The additive privacy parameter of differential privacy, page 351

$\mathsf{Disc}(\mathcal{Q})$ The discrepancy of matrix $\mathcal{Q}$, i.e. $\min_{z\in\{\pm 1\}^n} \|\mathcal{Q}z\|_\infty$, page 383

$\ell^*(K)$ The Gaussian mean width of K: $\mathsf{Exp}_g \max_{z\in K} |\langle z, g\rangle|$, page 414

ε The multiplicative privacy parameter of differential privacy, page 351

GS_q The global sensitivity of q, i.e. $\max_{x\sim x'} |q(x) - q(x')|$, page 353

$\mathsf{HerDisc}(\mathcal{Q})$ The hereditary discrepancy of matrix $\mathcal{Q}$, i.e. $\max_{S\subseteq\mathcal{X}} \mathsf{Disc}(\mathcal{Q}_S)$, page 387

$\mathsf{HerPDisc}(\mathcal{Q})$ The hereditary partial discrepancy of matrix $\mathcal{Q}$, i.e. $\max_{S\subseteq\mathcal{X}} \mathsf{PDisc}(\mathcal{Q}_S)$, page 386

$\mathrm{Lap}(\sigma)$ The Laplace distribution with scale σ, page 353

ln The natural logarithm function, page 353

log Base 2 logarithm function, page 353

$\mathrm{LS}_q(x)$ The local sensitivity of query q on dataset x, i.e. $\max_{x'\sim x} |q(x) - q(x')|$, page 367

$\|v\|_p$ The ℓ_p norm of vector v, i.e. $(\sum_i |v_i|^p)^{1/p}$, page 383

$\mathsf{PDisc}(\mathcal{Q})$ The partial discrepancy of matrix $\mathcal{Q}$, i.e. $\min_{\substack{z\in\{0,+1,-1\}^n,\\ \|z\|_1 > n/10}} \|\mathcal{Q}z\|_\infty$, page 383

$\mathcal{Q}^{\mathrm{conj}} = \mathcal{Q}^{\mathrm{conj}}(d)$ $\bigcup_{t=0}^{d} \mathcal{Q}_t^{\mathrm{conj}}(d)$, page 351

$\mathcal{Q}_t^{\mathrm{conj}} = \mathcal{Q}_t^{\mathrm{conj}}(d)$ Set of t-way marginal queries, i.e. counting queries corresponding to t-way conjunctions on $\mathcal{X} = \{0, 1\}^d$, page 351

$\mathcal{Q}^{\mathrm{means}} = \mathcal{Q}^{\mathrm{means}}(d)$ Set of d attribute means on dataset with d boolean attributes, i.e. counting queries corresponding to coordinate functions on $\mathcal{X} = \{0, 1\}^d$, page 351

$\mathcal{Q}^{\mathrm{pt}} = \mathcal{Q}^{\mathrm{pt}}(\mathcal{X})$ Set of counting queries corresponding to point functions on $\mathcal{X}$, page 351

$\mathcal{Q}^{\mathrm{thr}} = \mathcal{Q}^{\mathrm{thr}}(\mathcal{X})$ Set of counting queries corresponding to threshold functions on totally ordered $\mathcal{X}$, page 351

$\mathrm{SD}(Y, Y')$ The statistical distance between random variables Y and Y', page 354

$\sigma_{\min}(M)$ The smallest singular value of matrix M, i.e. $\inf_{z\neq 0} \|Mz\|_2 / \|z\|_2$, page 387

$\mathrm{Supp}(Z)$ The support of random variable Z, i.e. $\{z : \Pr[Z = z] > 0\}$, page 356

$\mathrm{VC}(\mathcal{Q})$ The Vapnik–Chervonenkis dimension of $\mathcal{Q}$, i.e. the largest number k such that there exist $x_1, \ldots, x_k \in \mathcal{X}$ for which $\{(q(x_1), \ldots, q(x_k)) : q \in \mathcal{Q}\} = \{0, 1\}^k$, page 374

$D(p\|q)$ The Kullback–Leibler divergence (a.k.a. relative entropy) between discrete probability measures p and q, i.e. $\sum_y p(y) \cdot \log(p(y)/q(y))$, page 363

$d(x, x')$ The Hamming distance between datasets $x, x' \in \mathcal{X}^n$, page 354

K The convex hull of the answer vectors $a_w = (q(w))_{q \in \mathcal{Q}} \in \mathbb{R}^{\mathcal{Q}}$ over $w \in \mathcal{X}$, page 392

n The number of rows in a dataset, page 349

$P_\alpha(K)$ The largest number of points that we can pack in K with all pairwise ℓ_∞ distances larger than α, page 393

$q : \mathcal{X} \to \{0, 1\}$ A predicate inducing a counting query $q : \mathcal{X}^n \to [0, 1]$, page 351

$q : \mathcal{X}^n \to \mathcal{Y}$ A query, page 351

$x = (x_1, \ldots, x_n) \in \mathcal{X}^n$ A dataset of n rows, page 349

$x \sim x'$ Datasets $x, x' \in \mathcal{X}^n$ differ in one row, page 351