LECTURE NOTES FOR THE 26TH MCGILL INVITATIONAL WORKSHOP ON COMPUTATIONAL COMPLEXITY

Bellairs Institute
Holetown, Barbados

Primary Lecturer:
Salil Vadhan

Guest Lecturer:
Kunal Talwar

Contents

1 Introduction **5**
- 1.1 Differential privacy . . . 6
- 1.2 Remarks on the definition . . . 8
  - 1.2.1 Interpretations of DP . . . 9
  - 1.2.2 Variants of definitions and notation . . . 9
  - 1.2.3 Plan for this week . . . 9

2 Composition **10**
- 2.1 Composition theorems for differential privacy . . . 10
  - 2.1.1 Group privacy . . . 10
  - 2.1.2 Answering many queries . . . 10
- 2.2 Alternatives to Global Sensitivity . . . 13
  - 2.2.1 Smooth Sensitivity . . . 13
  - 2.2.2 Propose-Test-Release (PTR) . . . 14
  - 2.2.3 A generalization of PTR . . . 14

3 Counting Queries **16**
- 3.1 Counting queries . . . 16
- 3.2 Private multiplicative weights . . . 18

4 Lower Bounds for Counting Queries **20**
- 4.1 Counting queries: Basic lower bounds . . . 20
- 4.2 General sets of counting queries . . . 22
- 4.3 Lower bounds for $(\epsilon, 0)$-differential privacy . . . 23

5 Pirates of the Caribbean I **25**
- 5.1 Two lower bounds via cryptography . . . 25
- 5.2 Fingerprinting codes . . . 25
- 5.3 Traitor-tracing schemes . . . 28

6 Pirates of the Caribbean II **30**
- 6.1 Cryptographic lower bounds . . . 30
- 6.2 Known results for interval queries . . . 31
- 6.3 Computational lower bounds for t-way marginals . . . 31

7 2-way and t-way Marginals Using Geometric Algorithms **33**

8 Large Conjunctions **38**
8.1 Computational complexity of t-way conjunctions 38
8.2 Private PAC learning 39

9 Private Learning **41**
9.1 A computationally efficient private mechanism for learning parity 41
9.2 Private PAC learning and communication complexity 43
References 44

Foreword

These notes reflect a series of lectures given by Salil Vadhan and Kunal Talwar at the 26th McGill Invitational Workshop on Computational Complexity. The workshop was held at the Bellairs Research Institute in Holetown, Barbados in February, 2014.

LECTURE 1

Introduction

Lecturer: Salil Vadhan *Scribe: Antonina Kolokolova*

The basic setting: A trusted curator holds a dataset x about n individuals. How can the curator permit others to analyse the dataset without compromising the privacy of subjects?

The traditional approach to this problem is to anonymize the dataset by removing identifiers such as names and month/day of birth. The problem with this approach is that the rest of information may uniquely specify individuals and be used to "re-identify" them. There have been a number of such instances with real-life datasets, such as identifying Netflix subscribers in the anonymized "Netflix Challenge" dataset by matching reviews posted on the Internet Movie Database with times when movies were watched [32]. Alternatively, if somebody already knows a lot about an individual and the person is unique based on the known data, they can learn everything else about that individual. So people can often be re-identified; and it is a difficult game to play to figure out how much information to remove.

Another approach is to only release "aggregate" statistics, which is in fact the approach we will study (but as we will see it is more subtle than it seems). We abstract this task by defining a particular "interface" to the dataset, called *mechanism*. We issue a query of the dataset and the mechanism—if the query is safe to answer—will provide a result.

x_1	Salil	5/9/73	M	0 1 0 1
x_2	Denis Therien			
x_3				
$\vdots$				
x_n				

$\in \mathcal{X}^n \leftrightarrows \boxed{\text{M}} \overset{q}{\underset{q(x)}{\leftrightarrows}}$ Data Analyst/ Adversary

As the US census does, a mechanism might effectively determine what kind of statistics to release, and only permit release of a bounded number of aggregate statistics on a dataset. An example of an aggregate statistics is a *counting query* $q : \mathcal{X} \to \{0, 1\}$: given a predicate on the rows identifying which people have a certain property, return

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

the fraction of people satisfying q.

However, it is not hard to design queries focused on a specific person: "how many males of age 40 have a childhood fear of sharks"—if there is only one person in the database satisfying this property, you know who that is. Thus, even counting queries can target a specific individual. A common remedy to this is to only release statistics where the answer is large enough. But then there is a problem that you can do several queries where each has a large answer. Take a query which focuses on an individual "or" with a query that has many rows satisfying it, then take the difference.

Even given approximate answers to many random queries, it may be possible compromise privacy ("reconstruction attacks"). In fact, an adversary may be able to reconstruct almost an entire database given many different random queries. Again, a question is "how much is too much?" NIH took down genomics dataset posted online after an attack of this flavour. Tomorrow, we will discuss a paper by Dinur and Nissim [11]. (Reconstruction from pairwise correlations will come up in a different way.)

1.1 Differential privacy

Differential privacy is a quantitative theory of privacy protection that enables us to reason about how much statistical information is safe to reveal in general, without knowing specifics of the data.

It emerged through a series of papers following the aforementioned reconstruction attacks of Dinur and Nissim [11]: Dwork and Nissim [13], Blum et al. [3], and Dwork et al. [15]; the last paper crystallized the definition of differential privacy.

The basic idea is to require that no individual's data has much effect on what an adversary sees. We give the precise definition below.

Definition 1.1.1. We say that a randomized mechanism $\mathcal{M} : \mathcal{X}^n \to \mathcal{Y}$ is *differentially private* if for any two neighboring datasets $\forall x \sim x' \in \mathcal{X}^n$ (x and x' differ in one row), $\forall q \in Q$, the distribution of $\mathcal{M}(x, q) \approx_\epsilon$ distribution of $\mathcal{M}(x', q)$: $\forall T \subseteq \mathcal{Y}$,

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

Here we typically take ϵ is small, but non-negligible (not cryptographically small); for example, a small constant, such as $\epsilon = 0.1$. Smaller ϵ provides better privacy, but the definition is no longer useful when $\epsilon < 1/n$.

We will think of the query as fixed, and remove q from notation. In this lecture, we consider answering only one query; the major focus later will be many queries.

In our treatment, it will be convenient to think of n as known and public information, and we will study asymptotic behavior as $n \to \infty$.[1]

It is technically convenient to have e^ϵ rather than $1 + \epsilon$ (makes squaring easier) so we will actually work with the definition

$$\Pr[\mathcal{M}(x) \in T] \leq e^\epsilon \Pr[\mathcal{M}(x') \in T] .$$

Equivalently, $\forall y \in \mathcal{Y}$,

$$\Pr[\mathcal{M}(x) = y] \leq e^\epsilon \Pr[\mathcal{M}(x') = y] .$$

Example 1.1.2 (Randomized response [40])**.** Let $X = \{0, 1\}$ (one bit per person). For each x_i, define

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

and

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

[1] Cynthia's comment: considering a definition where n is not fixed, and we consider datasets x and x' that differ in addition or removal of one row, is a good idea. This allows us to interpret the definition as hiding my presence in the database (as you can remove me, and the adversary's view remains similar).

Here, x_i only affects x'_i:

$$\frac{\Pr[x'_i = x_i]}{\Pr[x'_i = \neg x_i]} = \frac{1+\epsilon}{1-\epsilon} = e^{O(\epsilon)}.$$

This is an example of a mechanism in the "local model": no trusted, centralized curator is needed for the data.

Note that $\mathbb{E}[x'_i] = \epsilon x_i + (1-\epsilon)/2$. By the Chernoff bound,

$$\frac{1}{n}\sum_i \frac{1}{\epsilon}\cdot\left(x'_i - \frac{(1-\epsilon)}{2}\right) - \left(\frac{1}{n}\sum_i x_i\right) \le O\left(\frac{1}{\sqrt{n}\cdot\epsilon}\right).$$

As $n \to \infty$, difference $\to 0$.

This idea was introduced by Warner in the 60's for getting people to answer sensitive survey questions truthfully: make people flip coin. If heads, answer truthfully. If tails, flip again, and answer truthfully if heads, falsely if tails.

Example 1.1.3 (Laplace mechanism [15])**.** Let q be a counting query; we define $M(x) = q(x) + \text{noise}$. Note that if $x \sim x'$, $|q(x) - q(x')| \le 1/n$. This suggests "noise" of the magnitude $1/(\epsilon n)$. Which distribution will satisfy this? Density should be the same up to e^{ϵ}. Density of $\mathcal{M}(x)$ at $q(x') + z$ is the density of noise at $z + 1/n$; density of $\mathcal{M}(x')$ at $q(x)$ is density of noise at z. Thus we wish

$$(\text{density at } z + 1/n) \in [e^{\pm\epsilon}] \cdot (\text{density at } z).$$

The Laplace distribution $\mathrm{Lap}(\sigma)$ has density $\alpha e^{-|z|/\sigma}$. (The Gaussian distribution does not quite work: multiplicative factor does not work in the tail.)

Discrete version: Place probability mass at each integer mutiple of $1/n$ proportional to density.

For $z > 0$, $\sigma = 1/\epsilon n$; then,

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

and

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

Proposition 1.1.4. *For every $q : \mathcal{X}^n \to \mathbb{R}$, $\mathcal{M}(x) = q(x) + \mathrm{Lap}(\mathrm{GS}_q/\epsilon)$ is ϵ-DP.*

Here, GS_q is a global sensitivity $= \max_{x\sim x'} |q(x) - q(x')|$. For example,

1. Counting queries, $\mathrm{GS}_q \le 1/n$.
2. $q(x) = \max q_1(x), q_2(x), \ldots, q_t(x)$, where q_i are counting queries. $\mathrm{GS}_q \le \max_i \mathrm{GS}_{q_i}$.
3. $q(x) = d(x, H)$, the Hamming distance, for $H \subseteq \mathcal{X}^n$. ("is my data set close to one that satisfies my hypothesis?"). There, $\mathrm{GS}_q \le 1$.
4. Linear queries: given a bounded function $q : \mathcal{X} \to [0,1]$ on rows, we define:

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

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

1.2 Remarks on the definition

1. Why not require

$$\mathrm{SD}(\mathcal{M}(x), \mathcal{M}(x')) \triangleq \max_{T \subseteq \mathcal{Y}} \left| \Pr[(x) \in T] - \Pr[\mathcal{M}(x') \in T] \right| \leq \delta \, ?$$

What can δ be?

   (a) $\delta \leq 1/2n$. Then $\forall x, x' \in \mathcal{X}^n$ (even x, x' that are not neighbours), we have $\mathrm{SD}(\mathcal{M}(x), \mathcal{M}(x')) \leq n\delta \leq 1/2$. Then with probability $1/2$ get an answer independent of data. So in this case there is effectively no utility: this mechanism is useless.

   (b) $\delta \geq 1/2n$. In this case, "$\mathcal{M}(x)$= with probability $1/2$, output a random row of the database" satisfies the definition.

   However, we will work with the following relaxation of differential privacy that incorporates a δ statistical distance term in addition to the the multiplicative ϵ:

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

   Here, we will insist that δ is cryptographically negligible; in particular, $\delta \leq n^{-\omega(1)}$. This setting is often called (ϵ, δ)-differential privacy.

2. Why a multiplicative definition? This has an immediate Bayesian interpretation. Suppose the adversary has prior X_i on data of a person. Fix the rest of database $x_{-i} \in \mathcal{X}^{n-1}$, which models that either adversary knows the rest of database, or the rest of database is independent from x_i. If the adversary sees $y \leftarrow \mathcal{M}(X, x_{-i})$, then its posterior belief about i^{th} row is the conditional distribution $X_i|_{\mathcal{M}(X_i, x_{-i})} = y$.

   Comparing the prior X_i and posterior we have:

$$\begin{aligned} \Pr[\text{posterior} = x_i] &= \Pr[X_i = x_i \mid \mathcal{M}(X_i, x_{-i}) = y] \\ &=_{\text{Bayes}} \frac{\Pr[\mathcal{M}(X_i, x_{-i}) = y \mid X_i = x_i]}{\Pr[\mathcal{M}(X_i, x_{-i}) = y]} \cdot \Pr[X_i = x_i] \\ &\in [e^{\pm\epsilon}] \cdot \Pr[\text{prior} = x_i] \, . \end{aligned}$$

   In particular, $\mathrm{SD}(\text{prior}, \text{posterior}) \leq O(\epsilon)$.

   Conversely, if for every prior X_i, output y, we have $\mathrm{SD}(\text{prior}, \text{posterior}) \leq \epsilon$, then $\mathcal{M}$ is $O(\epsilon)$-DP.

   Proof. Let $x \sim x'$, so $x = (x_i, x_{-i})$, $x'_i = (x'_i, x_{-i})$. A natural prior would be

$$X_i = \begin{cases} x_i & \text{with prob. } 1/2 \\ x'_i \, , & \text{with prob. } 1/2 \, . \end{cases}$$

   Then

$$\frac{\Pr[\mathcal{M}(x_i, x_{-i}) = y]}{\Pr[\mathcal{M}(X_i, x_{-i})]} = \frac{\Pr[\text{posterior} = x_i]}{\Pr[\text{prior} = x_i]} = 1 \pm O(\epsilon) \, .$$

   □

1.2.1 Interpretations of DP

- Whatever the adversary learns about me he could have learned from the rest of the database (if an adversary knows smoking correlates with lung cancer, and knows that you smoke, can derive high chance of lung cancer).
- Mechanism would not leak significant amount of information specific to an individual or a small group (not large groups). Thus it is not useful to take action against a specific individual (e.g., NSA looking for a terrorist). (Cynthia: DP can be used to figure out "what is normal," to detect "abnormal" people).
- What data sets may adversary use? DP does not need to model that: arbitrary prior is captured.

1.2.2 Variants of definitions and notation

So far, data set is ordered. $x \in \mathcal{X}^n$: n is known/public. Alternative: x is a multiset of elements of $\mathcal{X}$. Can represent x as a histogram of $x \in \mathbb{N}^{\mathcal{X}}$. Here, mostly interested in symmetric queries. In multiset definition, distance is symmetric difference $|x \Delta x'|$, which is the same as ℓ_1 distance in histogram notation. The two notions may differ by a factor of 2 in distance (of two datasets of the same size n), and hence in the privacy parameter ϵ.

1.2.3 Plan for this week

Given that this is a complexity workshop, we will focus on complexity aspects of differential privacy, and connections to other topics in theoretical computer science. More on the algorithmic aspects of differential privacy can be found in the new monograph of Dwork and Roth [14].

Tentatively, we will cover the following topics, with some of the connections to other TCS/Math topics mentioned in parentheses.

- Composition theorems, answering many queries independently.
- Alternatives to worst-case global sensitivity GS.
- Answering many queries in a correlated fashion to protect privacy (learning theory).
- Information-theoretic limitations of DP (geometric, discrepancy ideas).
- Computational lower bounds (crypto, PCPs).
- Geometric algorithms (convex geometry, SDP).
- Multiparty DP (communication complexity, extractors).
- Differentially private PAC learning

LECTURE 2

Composition

Lecturer: Salil Vadhan *Scribe: Eric Allender and Shubhangi Saraf*

2.1 Composition theorems for differential privacy

2.1.1 Group privacy

$\forall x, x' \in \mathcal{X}^n$, let $d(x, x')$ denotes 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'. We say $x \sim x'$ if $d(x, x') \leq 1$.

Theorem 2.1.1. *If $\mathcal{M}$ is an (ϵ, δ)-differentially private mechanism, then $\forall x, x' \in \mathcal{X}^n$, if $k = d(x, x')$, $\mathcal{M}(x)$ and $\mathcal{M}(x')$ are $(k\epsilon, e^{k\epsilon} \cdot \delta)$-indistinguishable.*

Proof. 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 at most one row. Then, for all $T \subseteq \mathcal{Y}$, since $\mathcal{M}$ is (ϵ, δ)-differentially private,

$$\begin{aligned}
\Pr[\mathcal{M}(x_0) \in T] &\leq e^{\epsilon} \Pr[\mathcal{M}(x_1) \in T] + \delta \\
&\leq e^{\epsilon} \left(e^{\epsilon} \Pr[\mathcal{M}(x_2) \in T] + \delta \right) + \delta \\
&\ldots \\
&\leq e^{k\epsilon} \cdot \Pr[\mathcal{M}(x_k) \in T] + e^{k\epsilon} \cdot \delta.
\end{aligned}$$

□

2.1.2 Answering many queries

Let $\mathcal{M}_1, \mathcal{M}_2, \ldots, \mathcal{M}_k$ be (ϵ, δ)-differentially private mechanisms. Let $\mathcal{M}(x) = (\mathcal{M}_1(x), \mathcal{M}_2(x), \ldots, \mathcal{M}_k(x))$. This is the mechanism answering a k-tuple of queries using the mechanisms $\mathcal{M}_1, \mathcal{M}_2, \ldots, \mathcal{M}_k$, where each M_i is run with independent coin tosses.

Theorem 2.1.2. *$\mathcal{M}$ is $(k\epsilon, k\delta)$-differentially private.*

Theorem 2.1.3. *$\forall \delta' > 0$, $\mathcal{M}$ is $\left(O(\sqrt{k \log(1/\delta')}) \cdot \epsilon, k\delta + \delta'\right)$-differentially private, as long as $k < 1/\epsilon^2$.*

Theorem 2.1.3 is from Dwork et al. [18].
We now prove the above theorems starting with Theorem 2.1.2.

Proof of Theorem 2.1.2. Let us focus on the case $\delta = 0$. Fix databases x, x' such that $x \sim x'$. For an output $y \in \mathcal{Y}$, define the *privacy loss* to be

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

Notice that ϵ-differential privacy of $\mathcal{M}$ is equivalent to the statement that $\forall x \sim x'$ and $\forall y$,

$$L_{\mathcal{M}}^{x \to x'}(y) \leq \epsilon,$$

and

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

Also when $\delta \neq 0$, (ϵ, δ)-differential privacy of $\mathcal{M}$ is essentially equivalent to the statement that $\forall x \sim x'$, with probability $\geq (1 - \delta)$ over the choice of y sampled from $\mathcal{M}(x)$,

$$L_{\mathcal{M}}^{x \to x'}(y) \leq \epsilon,$$

and

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

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

$$L_{\mathcal{M}}^{x \to x'}(y) = \ln \left(\frac{\Pr[\mathcal{M}_1(x) = y_1 \wedge \mathcal{M}_2(x) = y_2 \wedge \cdots \wedge \mathcal{M}_k(x) = y_k]}{\Pr[\mathcal{M}_1(x') = y_1 \wedge \mathcal{M}_2(x') = y_2 \wedge \cdots \wedge \mathcal{M}_k(x') = y_k]} \right).$$

By the independence of the coin tosses of the various $\mathcal{M}_i$, the probabilities of the $\wedge$ of the various events break into a product of probabilities of individual events, and thus we get that

$$L_{\mathcal{M}}^{x \to x'}(y) = \sum_{i=1}^{k} L_{\mathcal{M}_i}^{x \to x'}(y_i).$$

By the union bound, with probability $1 - k\delta$, $y_1, y_2, \ldots, y_k$ all are such for all $i \in [k]$, the privacy loss

$$L_{\mathcal{M}_i}^{x \to x'}(y_i) \leq \epsilon.$$

Thus the proof of the theorem immediately follows. □

We now prove Theorem 2.1.3.

Proof of Theorem 2.1.3. Consider

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

By definition, this is the KL divergence

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

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 privacy loss we showed earlier.

Claim 2.1.4. *If $\mathcal{M}_i$ is ϵ-differentially private, where $\epsilon \leq 1$, then*

$$\mathop{\mathbb{E}}_{y_i \leftarrow \mathcal{M}_i(x)} [L_{\mathcal{M}_i}^{x \to x'}(y_i)] \leq O(\epsilon^2).$$

Proof. We will show that

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

and then the result will follow by the non-negativity of divergence. Now,

$$D(\mathcal{M}_i(x)\|\mathcal{M}_i(x')) + D(\mathcal{M}_i(x')\|\mathcal{M}_i(x)) = \mathop{\mathbb{E}}_{y_i \leftarrow \mathcal{M}_i(x)}[L_{\mathcal{M}_i}^{x \to x'}(y_i)] - \mathop{\mathbb{E}}_{y_i \leftarrow \mathcal{M}_i(x')}[L_{\mathcal{M}_i}^{x \to x'}(y_i)]$$

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

$$\mathop{\mathbb{E}}_{y_i \leftarrow \mathcal{M}_i(x)}[L_{\mathcal{M}_i}^{x \to x'}(y_i)] - \mathop{\mathbb{E}}_{y_i \leftarrow \mathcal{M}_i(x')}[L_{\mathcal{M}_i}^{x \to x'}(y_i)] \leq 2\epsilon \cdot \text{Statistical distance}(\mathcal{M}_i(x), \mathcal{M}_i(x')).$$

Since ϵ-differential privacy implies that the statistical distance is at most $O(\epsilon)$, the proof of the claim follows. □

Thus by linearity of expectation,

$$\mathop{\mathbb{E}}_{y \leftarrow \mathcal{M}(x)}[L_{\mathcal{M}}^{x \to x'}(y)] = k \cdot O(\epsilon^2).$$

Applying the Hoeffding bounds we get that the probability that $L_{\mathcal{M}}^{x \to x'}(y)$ deviates from its expected value by a factor of more than

$$O\left(\sqrt{k \log(1/\delta')}\right) \cdot \epsilon$$

is at most $1/\delta'$. This combined with the union bound gives the final bound on the parameters of differential privacy. □

This completes the proofs of both composition theorems.

It should be noted that, although Theorem 2.1.3 is stated in terms of queries being asked *simultaneously* (that is: *nonadaptively*), a nearly-identical proof (appealing to Azuma's Inequality, instead of Hoeffding) shows that the same conclusion holds even when the queries are asked *adaptively*.

Observe that if we have k 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 and then use the composition theorems. If we get an initial ϵ_0 differential privacy for each query, then to use the composition theorem, we would have to set

$$\epsilon_0 = \frac{\epsilon}{c \cdot \sqrt{k \log(1/\delta')}}.$$

Thus the noise to be added per query is

$$O\left(\frac{1}{\epsilon_0 n}\right) = O\left(\frac{\sqrt{k \log(1/\delta')}}{\epsilon n}\right).$$

Actually, if we want a bound on the *maximum* noise added to any of the queries, we should do a union bound over the k queries. Since the Laplace distribution has exponentially vanishing tails, this costs an additional factor of $\log k$, so with high probability, our maximum error over the k queries is:

$$O\left(\frac{\sqrt{k \log(1/\delta')} \cdot \log k}{\epsilon n}\right).$$

Thus if we want noise $o\left(1/\sqrt{n}\right)$, we can take k to be close to n (which we will see is essentially optimal for any notion of privacy). If we want noise $o(1)$, we can achieve this by taking k close to n^2 (which is known to be optimal if the answers are not coordinated or if the queries are completely unrelated.)

2.2 Alternatives to Global Sensitivity

In this section, we consider the question of whether we can do better than adding noise $\text{Lap}(\text{GS}_q/\epsilon)$, where GS_q denotes the Global Sensitivity of query q (as discussed in an earlier lecture).

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 problem with trying to use local sensitivity instead of global sensitivity is that we don't want the amount of allowable 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 $x_1 \sim x$ where x_1 has a neighbor x_2 such that $q(x_2) = 10^9$. $\text{LS}_q(x) = 0$, but $\text{LS}_q(x_1)$ is large, and answering queries noisily based on LS_q could violate privacy because it distinguishes between $x \sim x_1$. To avoid this problem, we would have to add noise roughly to the average of $\text{LS}_q(x)$ and $\text{LS}_q(x')$ for *every* pair $x \sim x'$.

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. (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 to the instance $x' \sim x$ 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 LS_q is small, even in a large neighborhood of 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 three related approaches that have been taken, in formulating alternatives to Global Sensitivity.

1. Smooth Sensitivity [34].
2. Propose-Test-Release (PTR) [12].
3. A generalization of PTR [27].

Below, we discuss each of these approaches in turn.

2.2.1 Smooth Sensitivity

Define Smooth Sensitivity (at x) as follows:

$$\text{SS}(x) = \max\{\text{LS}_q(x')e^{-\epsilon d(x,x')} : x' \in \mathcal{X}^n\}.$$

Here, ϵ is a parameter.

The main facts that were presented about Smooth Sensitivity are:

- Adding noise $O(\text{SS}_q(x)/\epsilon)$ (according to a Cauchy distribution) is sufficient for ϵ-differential privacy.
- SS_q can be computed efficiently when q is the Median query (and thus this leads to an efficient mechanism).

2.2.2 Propose-Test-Release (PTR)

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 database might be queried, an alternative is to initially *propose* an amount of noise that seems tolerable (or, more precisely, a bound on the local sensitivity of the current database), and then *test* whether answering a query with this amount of noise would violate privacy. If the test passes, then everything's fine. But perhaps you detect that adding this (small) amount of noise *would* violate privacy. In that case, you simply refuse to answer.

More precisely: PTR consists of the following three steps (parameterized by ϵ and δ), yielding a mechanism $\mathcal{M}$:

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

Claim 2.2.1. *This scheme is $(2\epsilon, \delta)$-differentially private.*

Proof. Consider some fixed x and x' with $x \sim x'$.

Because of the Laplacian noise in the definition of $\hat{d}$, it follows that

$$\Pr[\mathcal{M}(x) = \perp] \in [e^{\pm\epsilon}] \Pr[\mathcal{M}(x') = \perp].$$

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

Case 1: $\mathrm{LS}_q(x) > \beta$. In this case, $\Pr[\mathcal{M}(x) \neq \perp] \leq \delta$ and $\Pr[\mathcal{M}(x') \neq \perp] \leq \delta$.

Case 2: $\mathrm{LS}_q(x) \leq \beta$. In this case, $|q(x) - q(x')| \leq \beta$, which in turn implies the ϵ-indistinguishability of $q(x) + \mathrm{Lap}(\beta/\epsilon)$ and $q(x') + \mathrm{Lap}(\beta/\epsilon)$. □

2.2.3 A generalization of PTR

Rather than *proposing* (arbitrarily) a threshold β, might it not be possible to (occasionally) *compute* a suitable β? This is the approach which we now consider. That is, we will investigate whether it is possible to compute a differential privacy estimate $\hat{\beta} = \hat{\beta}(x)$, such that, with probability $1 - \delta$, $\mathrm{LS}_q(x) \leq \hat{\beta}$, in which case we will output $q(x) + \mathrm{Lap}(\hat{\beta}/\epsilon)$.

The setting in which we will explore this possibility is where we are querying a graph, to find out the number of triangles in the graph.

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

- *Edge-level Privacy*. In this setting, we say that $G \sim G'$ if the graphs G and G' differ on one edge.
- *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.

Here, we will consider only edge-level privacy. Let $q_\Delta(G)$ be the number of triangles in G. One can easily verify 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, $\mathrm{GS}_{q_\Delta} = n - 2$. Observe that $\mathrm{GS}_{\mathrm{LS}_{q_\Delta}} = 1$.

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

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

This mechanism is (ϵ, δ)-differentially private, and the total noise is

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

Also, it turns out that—for this particular query—all of the computations can in fact be done efficiently.

LECTURE 3

Counting Queries

Lecturer: Salil Vadhan *Scribe: Anne-Sophie Charest*

3.1 Counting queries

We have seen the following results:

- Answering one query: For $q : \mathcal{X} \to \{0,1\}, \mathcal{M}(x) = q(x) + \text{Lap}(1/\epsilon n)$ is ϵ-differentially private.
- Answering k counting queries: with high probability error

$$\alpha \leq O\left(\frac{\sqrt{k\log(1/\delta)}\log(k)}{\epsilon n}\right)$$

  for all k queries.

We now consider a way to answer $\gg n^2$ queries with differential privacy due to Blum et al. [4].

Theorem 3.1.1 ([4])**.** *For every set Q of counting queries on a data universe $\mathcal{X}$, $\forall \epsilon > 0$, there exists an ϵ-differentially private mechanism M such that $\forall x \in \mathcal{X}^n$ with high probability $M(x)$ answers all queries in Q to with error*

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

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

Example 3.1.2. Let $\mathcal{X} = \{0,1\}^d$ and Q be the set of conjunctions, e.g., $q(x) = X_1 \wedge \overline{X_2} \ldots \wedge X_7 \wedge X_{10}$. Then $|X| = 2^d$ and $|Q| = 3^d$; if $n = \omega(d^2/\epsilon)$,

$$\alpha = O\left(\left(\frac{d^2}{\epsilon n}\right)^{1/3}\right) \to 0\,.$$

Remark 3.1.3. There does not exists any polynomial time algorithm to do this yet. In general, constructing this dataset will be easier to do for structured sets of queries.

Proof of Theorem 3.1.1. We begin by establishing the existence of an accurate m-row synthetic dataset y^*: Let y^* be a random sample of m rows from $\mathcal{X}$, say with replacement for simplicity. By the Chernoff bound,

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

Note that this argument is inspired by "Occam's Razor"-type argument from Learning Theory.

In order to guarantee privacy, we use the *exponential mechanism* of McSherry and Talwar [29], i.e.,

$$\text{output } y \in \mathcal{X}^m \text{ with probability } \propto e^{-\epsilon n \max\limits_{q\in Q} |q(y)-q(x)|},$$

where ϵ is the usual privacy parameter. We refer to the right-hand size as expression as $\text{weight}_x(y)$. More generally, the exponential mechanism can be used to design differentially private mechanisms for sampling "good" outputs from any output space $\mathcal{Y}$, replacing 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 database x, and replacing the factor of n in the exponent with

$$1/\max_{y} \text{GS}_{\text{score}(\cdot,y)}\ .$$

So to get good performance from the exponential mechanism, score should have low sensitivity as a function of its first argument — as that will mean we put higher relative weight on good outputs than bad outputs.

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

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

Thus, we have 2ϵ-differential privacy.

Accuracy: We will show 2α accuracy.

$$\begin{aligned}\Pr[M \text{ is not } 2\alpha \text{ accurate}] &= \Pr[\exists\ q \text{ such that } |q(M(x)) - q(x)| > 2\alpha] \\ &\leq \frac{1}{\text{weight}_x(y^*)} \cdot \sum_{\substack{y\in\mathcal{X}^m,\\ y \text{ not } 2\alpha\text{-accurate}}} \text{weight}_x(y)\end{aligned}$$

(We are lower-bounding the sum of all weights by the weight of the single y^* we showed to exist by the Occam argument above.)

$$\begin{aligned}&\leq \frac{|X|^m e^{-\epsilon n(2\alpha)}}{e^{-\epsilon n}} \\ &\ll 1 \qquad \text{if } \alpha\epsilon n > 2m \log |X|\,.\end{aligned}$$

Recall that $m = O(\log |Q|/\alpha^2)$. Solve for α to obtain the desired result. □

Remark 3.1.4. The computational time is roughly

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

so it is really expensive. For example, we get $\exp(d^2/\alpha^2)$ for the conjunctions.

3.2 Private multiplicative weights

We now present the state of the art for general queries due to Hardt and Rothblum [23].
Comparison with the small databases method (Blum et al. [4], just discussed):

- error
$$\alpha = O\left(\frac{\sqrt{\log|\mathcal{X}|\log(1/\delta)}(\log|Q|)}{\epsilon n}\right)^{1/2},$$
i.e., 1/2 instead of 1/3 and $\log|\mathcal{X}|$ is in a square root;
- queries can arrive on-line, and the algorithm is adaptive;
- running time is polynomial $(n, |\mathcal{X}|)$ per query, which is better than we had before, but still sometimes too much.

The algorithm: We view the database x as a distribution on $\mathcal{X}$:
$$x(i) = \frac{\#\text{ rows of type } i \text{ in } x}{n}.$$
Then, $q(x) \triangleq \mathbb{E}_{i\sim x}[q(i)]$. The algorithm will maintain a distribution h on $\mathcal{X}$, some hypothesis for what the data is. It will try to answer queries with h, and update h when it leads to too much error. Here are the details:

1. Initially, set h to the uniform distribution on $\mathcal{X}$.

Outer Loop REPEAT at most $O(\log|\mathcal{X}|/\alpha^2)$ times

(a) Randomize the accuracy threshold: $\hat{\alpha} = \alpha + \text{Lap}(1/\epsilon_0 n)$.

Inner loop REPEAT

i. Receive next query q
ii. Set $a = q(x) + \text{Lap}(1/\epsilon_0 n)$.
iii. If $|a - q(h)| < \hat{\alpha}$, then output $b = q(h)$ and CONTINUE inner loop. Otherwise, output $b = q(x) + \text{Lap}(1/\epsilon_0 n)$ (with fresh noise) and EXIT inner loop.

(b) Reweight using query q: $\forall i\ g(i) = \begin{cases} h(i)e^{(\alpha/4)\cdot q(i)} & \text{if } \alpha > q(h), \\ h(i)e^{-(\alpha/4)\cdot q(i)} & \text{if } \alpha < q(h). \end{cases}$

(c) Renormalize: $\forall i\ h(i) = \dfrac{g(i)}{\sum_j g(j)}$.

(d) CONTINUE outer loop.

Utility analysis: With high probability, the maximum noise magnitude is no more than
$$O\left(\frac{\log|Q|}{\epsilon_0 n}\right)$$
because of the property of the tail of the Laplace distribution and the union bound. We constrain α so that
$$O(\log(|Q|/\epsilon_0 n) \le \alpha/4.$$
This implies that all answers that we provide are within $\pm 3\alpha/2$ of $q(x)$ by the triangle inequality.

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

Claim 3.2.1. *Assuming the maximum noise magnitude is at most $\alpha/4$, at most $O(\log|\mathcal{X}|/\alpha^2)$ updates (i.e. iterations of the outer loop) will be required.*

Proof. Consider the potential function $D(x \mid\mid h)$. Initially,

$$D(x||\text{uniform}) = \log|\mathcal{X}| - H(x) \leq \log|\mathcal{X}|,$$

where H is the Shannon entropy function. When we do an update, we know that $|q(x) - q(h)| \geq \alpha/2$. It can be shown that this implies that:

$$D(x \mid\mid h') \leq D(x \mid\mid h) - \Omega(\alpha^2),$$

which follows from a tedious but not very hard calculation. □

Privacy analysis: Note that the entire output of the algorithm is determined by three items: the indicators for when the updates occur, the queries, and the noisy answers b when an update occurs. Define an epoch to be one execution of the outer loop, up to the next update.

Claim 3.2.2. *One epoch is 4ϵ-differentially private.*

Proof. Fix a transcript of an epoch in which the update occurs in the t^{th} query. Let $q_1, \ldots, q_t$ be the t queries in the epoch and $a_1, \ldots, a_t$ the corresponding noisy answers in the inner loop. Fix the noise in $a_1, \ldots, a_{t-1}$. Then,

$$\Pr[\text{no update for queries } q_1, \ldots, q_{t-1}] = \Pr\left[\hat{\alpha} > \max_i |a_i - q_i(h)|\right]. \tag{3.1}$$

Note that this condition is determined by the quantity $\beta = \hat{\alpha} - \max_i |a_i - q_i(h)| = \alpha - \max_i |a_i - q_i(h)| + \text{Lap}(1/\epsilon_0)$. Since $\alpha - \max_i |a_i - q_i(h)|$ has global sensitivity 1 as a function of the database x, β is differentially private.

Now conditioning on the value of β (such that $\beta > 0$ so that no updates happen in $q_1, \ldots, q_{t-1}$, we consider the probability of having an update in query q_t. Since $a_t = q_t(x) + \text{Lap}(1/\epsilon_0 n)$, this is the probability that $|q_t(x) + \text{Lap}(1/\epsilon_0 n) - q_t(h)| \geq \hat{\alpha} = \beta + \max_i |a_i - q_i(h)|$. That is,

$$\begin{aligned}\Pr[\text{update at } t^{th} \text{ query} \mid \beta] = &\Pr[\text{Lap}(1/\epsilon_0 n) \geq \beta + \max_i |a_i - q_i(h)| + q_t(h) - q_t(x)] + \\ &\Pr[\text{Lap}(1/\epsilon_0 n) \leq -\beta - \max_i |a_i - q_i(h)| + q_t(h) - q_t(x)].\end{aligned}$$

Since $\beta + \max_i |a_i - q_i(h)| + q_t(h) - q_t(x)$ and $-\beta - \max_i |a_i - q_i(h)| + q_t(h) - q_t(x)$ both have global sensitivity 2, by the differential privacy of the Laplace mechanism, the probabilities above can vary by a factor of at most $e^{2\epsilon_0}$ on neighboring databases, and thus the decision to do an update on q_t is $2\epsilon_0$-differentially private. The noisy answer $b_t = q_t(x) + \text{Lap}(1/\epsilon_0)$ is ϵ_0-differentially private, giving a total of $4\epsilon_0$-differential privacy.

By composition, we get

$$O\left(\sqrt{\frac{\log|\mathcal{X}|\log(1/\delta)}{\alpha^2}}\epsilon_0, \delta\right)$$

differential privacy. Then, we use $O(\log|Q|/\epsilon_0 n) \leq \alpha/4$. Solve for α to get the final answer. □

Remark 3.2.3.

- There is a way to get a synthetic dataset at the end of the algorithm, but you need to do a little bit of work to make sure that all queries are correctly answered simultaneously.
- If all queries are given simultaneously, we can go faster by first picking queries which should generate an update (e.g., with the exponential mechanism) [24].

LECTURE 4

Lower Bounds for Counting Queries

Lecturer: Kunal Talwar *Scribe: Swastik Kopparty*

In this lecture we will prove lower bounds on the problem of privately answering k counting queries from a database of with n records, where each record comes from the space $\mathcal{X}$. We will consider even weaker notions of privacy.

4.1 Counting queries: Basic lower bounds

We begin with the definition of a very weak standard for privacy.

Definition 4.1.1. A mechanism $\mathcal{M} : \mathcal{X}^n \to \mathcal{Y}$ is called *blatantly non-private* 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 $n/100$ coordinates (with high probability over the randomness of $\mathcal{M}$).

Any mechanism which is $(1, 0.1)$ differentially private is blatantly not blatantly non-private.

We now give some basic lower bounds, due to Dinur and Nissim [11], on the tradeoff between the error and the number of counting queries that can be answered while maintaining privacy under the above definition.

Let $\mathcal{X} = \{0, 1\}$. Then a database of n people is simply a vector $x \in \{0, 1\}^n$. We will consider queries $q \in \{0, 1\}^n$: the intended answer to the query q is $\langle q, x\rangle$. These are not exactly counting queries, but they can be transformed into counting queries over a larger data universe. (Set $\mathcal{X}' = [n] \times \{0, 1\}$, $x' = ((1, x_1), (2, x_2), \ldots, (n, x_n))$, $q'((i, b)) = q_i \cdot b$.)

Theorem 4.1.2. *Given $x \in \{0, 1\}^n$, suppose $\mathcal{M}$ outputs, for each $q \in \{0, 1\}^n$, a value $y_q \in \mathbb{R}$ such that*

$$\left| y_q - \langle q, x\rangle \right| < E.$$

Then one can use the y_q's to compute $x' \in \{0, 1\}^n$ such that $|x - x'|_1 < 4E$ (and thus x and x' differ in at most $4E$ coordinates).

Corollary 4.1.3. *If $\mathcal{M}$ is a mechanism as above with $E < n/400$, then $\mathcal{M}$ is blatantly non-private.*

Thus at least $\Omega(1)$ additive error is needed for privately answering all queries.

The procedure for computing x' that we will see is not efficient; this is sufficient since we are dealing with an information-theoretic notion of privacy.

Why would we ever consider a mechanism that answers 2^n queries? This is very natural: if the mechanism privately outputs a sanitized database or a synopsis (like in the previous lecture), we might hope that it can preserve answers to a huge number of counting queries. Indeed, the mechanisms from the previous lecture can answer $\exp(\Omega(\epsilon \cdot \alpha^3 \cdot n/(\log |X|)))$ and $\exp(\Omega(\epsilon \cdot \alpha^2 \cdot n/\sqrt{\log |X| \cdot \log(1/\delta)}))$ counting queries, respectively. These get close to 2^n; the lower bound we are about to prove shows that we cannot hope to reach 2^n.

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

$$\left|y_q - \langle q, x' \rangle\right| < E.$$

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

Claim 4.1.4. $|x - x'|_1 < 4E$.

Proof. Set

$$I_0 = \{j \in [n] \mid x_j = 0\},$$
$$I_1 = \{j \in [n] \mid x_j = 1\}.$$

Define $q_0 \in \{0,1\}^n$ to be the vector which is nonzero in coordinates I_0. Define $q_1 \in \{0,1\}^n$ to be the vector which is nonzero in coordinates I_1.

Note that $\langle q_0, x \rangle = 0$. Thus $y_{q_0} < E$, and so $0 \leq \langle q_0, x' \rangle < 2E$. Thus x' has at most $2E$ 1's in the coordinates of I_0. Similarly $\langle q_1, x \rangle = |I_1|$. Thus $y_{q_1} \in |I_1| \pm E$, and so $\langle x', q_1 \rangle \in |I_1| \pm 2E$. Thus x' has at most $2E$ 0's in the coordinates of I_1.

Thus x' and x differ on at most $4E$ coordinates. □

This completes the proof. □

Our next theorem says that for privately answering a fixed set of even $O(n)$ counting queries, an additive error of $\Omega\left(1/\sqrt{n}\right)$ is sometimes needed.

We will in fact study the more general question of what additive error is needed for privately answering any set of counting queries.

Let $q_1, \ldots, q_k$ be a collection of queries. Suppose we are given $y_1, \ldots, y_k$ with

$$\left|y_i - \langle q_i, x \rangle\right| < E.$$

Our privacy-breaking strategy is the same: take any $x' \in \{0,1\}^n$ with

$$\left|y_i - \langle q_i, x' \rangle\right| < E$$

for each i.

Then, as in the previous argument, we deduce that $|\langle q_i, x - x' \rangle| < 2E$. We want to use this to deduce that $|x - x'|_1 < n/100$.

Suppose not. Let $z = x - x'$. Let Q denote $k \times n$ matrix whose rows are q_i. Thus we have:

1. z is a $\{0, +1, -1\}$ vector with $|z|_1 > n/100$,
2. $|Qz|_\infty < 2\alpha\sqrt{n}$.

Definition 4.1.5. We define the partial discrepancy of a matrix Q, denoted $\mathsf{PDisc}(Q)$, by:

$$\mathsf{PDisc}(Q) = \min_{\substack{z \in \{0,+1,-1\}^n, \\ |z|_1 > n/100}} |Qz|_\infty.$$

Summarizing the above discussion: any mechanism for answering $q_1, \ldots, q_k$ within error $\mathsf{PDisc}(Q)/2$ is blatantly non-private.

Let us see a quick application of this, the second theorem of Dinur and Nissim [11].

Theorem 4.1.6. *There exists $\alpha > 0$ and a set of queries $q_1, \ldots, q_k \in \{0,1\}^n$ (where $k = O(n)$) such that: Given $x \in \{0,1\}^n$, suppose $\mathcal{M}$ outputs, for each $i \in [k]$, a value $y_i \in \mathbb{R}$ such that*

$$\left|y_i - \langle q_i, x\rangle\right| < \alpha\sqrt{n}.$$

Then one can use the y_i's to compute $x' \in \{0,1\}^n$ such that $|x - x'|_1 < n/100$.

Corollary 4.1.7. *If $\mathcal{M}$ is a mechanism answering queries $q_1, \ldots, q_k$ with the accuracy given as above, then $\mathcal{M}$ is blatantly non-private.*

The collection of queries above is uniformly random. In fact, this can be derandomized with an explicit collection of $O(n)$ queries. The privacy-breaking procedure can also be implemented to run efficiently, and can be implemented in real world databases using hashing.

The proof of Theorem 4.1.6 follows immediately by combining earlier observations connecting PDisc and blatant nonprivacy, with the following basic claim about the PDisc of a random matrix.

Claim 4.1.8. *There exists a constant $\alpha > 0$ such that for $k = \Omega(n)$, a random $0,1$ $k \times n$ matrix Q has $\mathsf{PDisc}(Q) > \alpha\sqrt{n}$.*

Proof. Pick the rows $q_1, \ldots, q_k \in \{0,1\}^n$ uniformly. Fix $z \in \{0,+1,-1\}^n$ with $|z|_1 > n/100$. Then we have:

$$\Pr_{q_i}[|\langle q_i, z\rangle| < \beta\sqrt{n}] < O(\beta),$$

(by the anticoncentration of the Binomial distribution, which is the distribution of $\langle q_i, z\rangle$).

Thus, for each z we have

$$\Pr[\forall i \in [k], |\langle q_i, z\rangle| < \beta\sqrt{n}] < O(\beta)^k.$$

Taking a union bound over all such z, the probability that for some $z \in \{0,+1,-1\}^n$ with $|z|_1 > n/100$, we have that $|\langle q_i, z\rangle| < \beta\sqrt{n}$ for each $i \in [k]$, is at most $3^n \cdot O(\beta)^k$, which for $\beta = 1/2$ and $k = O(n)$ is exponentially small in n.

Thus a random matrix has the desired discrepancy property. □

This completes the proof of Theorem 4.1.6.

One can consider an even more relaxed version of error, where the mechanism is allowed to give answers with $O\left(\sqrt{n}\right)$ additive error for 51% of the queries, and for the remaining 49% it is free to make arbitrary error. Even such a mechanism is necessarily non-private. If one wants this theorem with an efficient privacy-breaking algorithm, then this can also be done with the 51% replaced by about 77% (this is a theorem of Dwork et al. [16], and is based on connections to compressed sensing).

4.2 General sets of counting queries

We now work towards a complete understanding the error required for differential privacy for answering a given set of counting queries.

Let $q_1, \ldots, q_k \in \{0,1\}^N$ be a given set of counting queries, where $N = |\mathcal{X}|$. Let Q be the $k \times N$ matrix whose rows are the q_i. We let $\mathsf{Error}(Q)$ denote the smallest quantity η, such that there is a $(1, 0.1)$-differentially private mechanism for answering the queries Q, where the error in the answer is at most η with probability at least $2/3$.

Remark 4.2.1. Kunal's normalization is different from Salil's (relative error vs. absolute error).

For a set $S \subseteq [N]$, we let Q_S denote the restriction of Q to the columns of S (this corresponds to restricting the set of users to S). Now a trivial but important observation: an (ϵ, δ)-private mechanism for Q is also an (ϵ, δ) private mechanism for Q_S.

Thus:

$$\mathsf{Error}(Q) \geq \mathsf{Error}(Q|_S) \geq \mathsf{PDisc}(Q|_S).$$

Remark 4.2.2. This does not hold for blatant nonprivacy! (Because S can be very small.)

Define the hereditary partial discrepancy $\mathsf{HerPDisc}(Q)$ to be $\min_{S \subseteq [N]} \mathsf{PDisc}(Q_S)$. In this language, we just proved the theorem of Muthukrishnan and Nikolov [31]:

Theorem 4.2.3.

$$\mathsf{Error}(Q) \geq \mathsf{HerPDisc}(Q).$$

Amazingly, this bound is nearly tight. The following theorem is due to Nikolov, Talwar and Zhang.

Theorem 4.2.4 ([33])**.** *For every Q,*

$$\mathsf{Error}(Q) \leq \mathsf{HerPDisc}(Q) \cdot \mathrm{poly}(\log(k/\delta)).$$

4.3 Lower bounds for $(\epsilon, 0)$-differential privacy

We will now see some lower bounds for pure $(\epsilon, 0)$-differential privacy that distinguish it from (ϵ, δ)-differential privacy. In particular, we will prove that for answering k counting queries with $(\epsilon, 0)$-differential privacy, one must have $\mathsf{Error}(Q) \geq \Omega(k)$ (this contrasts with the result we saw in Lecture 2 which shows that one can have $O(\sqrt{k})$ error if one only requires $(1, 0.1)$ differential privacy, by applying advanced composition to the Laplace mechanism).

Let $x \in \mathbb{R}^N$ represent our database in the histogram representation. (Recall what this means: N is the number of possible "user types," and the coordinates of x represent the frequency of each user type.) We will also represent our queries q_i as vectors in $\mathbb{R}^N$. Q will be the matrix with q_is as rows. Suppose all the entries of Q are in $[-1, 1]$.

Lemma 4.3.1. *Let $s, R > 0$ be such that there exist vectors vectors $x_1, \ldots, x_\ell \in \mathbb{R}^N$ with:*

- $|x_i|_1 \leq s$ *for each i,*
- $|Qx_i - Qx_j|_\infty \geq R$ *for each $i \neq j$.*

Suppose $\ell > 2e^{2\epsilon s}$. Then for any $(\epsilon, 0)$ differentially private mechanism $\mathcal{M}$ answering queries Q, there exists some $i \in [\ell]$ with:

$$\Pr[|\mathcal{M}(x_i) - Qx_i|_\infty > R/2] > 1/2\,.$$

(In particular $\mathsf{Error}(Q) \geq R/2$).

Proof. Suppose not. Then for every i, there is a good probability that $\mathcal{M}(x_i)$ is pretty close to Qx_i.

Then by "group privacy" (using the fact that $|x_1 - x_i|_1 < 2s$), we have for each $i \in [\ell]$:

$$\Pr[|\mathcal{M}(x_1) - Qx_i| < R/2] > e^{-2\epsilon s} \cdot \frac{1}{2}\,.$$

Thus $\mathcal{M}(x_1)$ has a decent chance of lying close to each Qx_i. But since all the Qx_i are far away from each other, and there are many of them, this cannot be.

Formally, we have

$$\begin{aligned}\Pr\left[\mathcal{M}(x_1) \in \bigcup_{i=1}^{\ell} B_{\ell_\infty}(Q(x_i), R/2)\right] &= \sum_{i=1}^{\ell} \Pr\left[\mathcal{M}(x_1) \in B_{\ell_\infty}(Q(x_i), R/2)\right] \\ &\geq \ell \cdot e^{-2\epsilon s} \cdot \frac{1}{2} \\ &> 1.\end{aligned}$$

Contradiction. □

Instantiating the above framework, take $N = 2^k$. Then for $i \in [k]$, take $q_i \in \{0,1\}^N$ to be the vector with 1 in those coordinates $j \in [N]$ for which the base-2 representation of j has a 1 in coordinate i. Now we can take $\epsilon = 1/4$ and $\ell = 2^k > 2 \cdot e^{2\epsilon s}$ in the above framework (to get $s = R = k$) by choosing $x_i = k \cdot e_i$ for each $i \in [N]$. (Here e_i is the elementary unit vector with 1 in coordinate i). Translating back to our old language, here our data universe is $\mathcal{X} = \{0,1\}^k$, our queries are the k column sums (i.e. one-way marginals), and we are considering the 2^k datasets that have $n = k$ identical rows.

Observe that the required properties hold. Thus we see that $\mathsf{Error}(Q) > R/2 = k/2$.

In general, by choosing a suitable Q and collection of vectors x_i, we can prove a lower bound of

$$\sqrt{k \log \frac{|X|}{k}}$$

on the error required for achieving $(\epsilon, 0)$ differential privacy while answering k counting queries.

It is a good exercise to see what breaks down in the above argument when we consider (ϵ, δ) differential privacy. Do it!

LECTURE 5

Pirates of the Caribbean I

Lecturer: Salil Vadhan *Scribe: Michal Koucký*

5.1 Two lower bounds via cryptography

In this lecture we cover two lower bounds on the abilities of differentially private mechanisms. The lower bounds are obtained from two different cryptographic primitives. The first lower bound shows that in order to answer d attribute means (the same as the column sums we considered in the last lecture, normalized by a factor of n) with error $\alpha < 1/2$ in differential private manner we need $n \in \tilde{\Omega}(\sqrt{d})$, which is tight up to the hidden polylogarithmic factors, by the Laplace mechanism and composition (cf. Lecture 2). This lower bound is obtained using the *fingerprinting codes* of Boneh and Shaw [5] and holds unconditionally. The second lower bound shows that to answer $d = n^{2+o(1)}$ certain efficiently computable counting queries with error $\alpha < 1/2$ one needs time exponential in d (assuming the existence of exponentially strong one-way functions). This latter lower bound is obtained using *traitor-tracing schemes* of Chor et al. [10].

5.2 Fingerprinting codes

Fingerprinting codes were designed to solve the following scenario. Imagine a digital movie distribution company which 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 costumers 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 the two angles from which the movie was shot.) The colluding costumers may want to merge 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 with high probability from the code and a pirated copy one of the colluding pirates.

We will use the framework of Boneh and Shaw [5]. We will have a randomized *generating algorithm* $\mathsf{Gen}(1^n)$ which will produce $n \times d$ binary fingerprinting matrix C where $C_{i,j}$ determines the watermark of customer i in scene j. The generating algorithm will take as its only parameter n, d will be determined as a function of n. Associated with the generating algorithm $\mathsf{Gen}(1^n) \to \{0,1\}^{n \times d}$ is the (randomized) *tracing algorithm* $\mathsf{Trace} : \{0,1\}^{n \times d} \times \{0,1\}^d \to \{1,\ldots,n\}$ which on input C and a pirated copy w (w is the sequence of watermarks of the pirated copy) returns an index of a customer, hopefully one who contributed to the creation of the pirated copy with watermarks w.

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

rows in S.

We consider a (randomized) pirating algorithm $P : \{0,1\}^{|S|\times d} \to \{0,1\}^d$ which takes as its input C_S for a generating matrix C and produces a watermark sequence w for the pirated copy. The pirates can read off the matrix C_S from their copies of the movie.

The fingerprinting scheme is *secure* against any coalition $S \subseteq \{1, \ldots, n\}$ and any randomized pirating algorithm P if:

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

The probability is *negligible* ($\leq \text{negl}(n)$) when it is asymptotically smaller than any polynomial $1/n^k$, for any fixed k.

Boneh and Shaw [5] construct a fingerprinting code for $d \in \tilde{O}(n^3)$ as follows. $\mathsf{Gen}(1^n)$ outputs a matrix obtained by randomly permuting columns of the matrix

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

Each block spans $\tilde{O}(n^2)$ identical columns. For such a randomly generated matrix, a coalition which does not include say the second player cannot distinguish columns that come from the first and the second block of the matrix, these columns look identical to the coalition. The tracing algorithm takes advantage of this observation. The tracing algorithm $\mathsf{Trace}(C, w)$ outputs the first i such that

$$\mathop{\mathbb{E}}_{j \text{ in } i\text{th block}}[w_j] - \mathop{\mathbb{E}}_{j \text{ in } (i-1)\text{th block}}[w_j] \geq \frac{1}{n}$$

where the expectation is taken over a randomly chosen index j among the ones coming from the ith block or $(i-1)$th block. Notice, for a feasible w such i is guaranteed to exist since $\mathbb{E}_{j \text{ in } 0 \text{ block}}[w_j] = 0$, $\mathbb{E}_{j \text{ in } n\text{th block}}[w_j] = 1$, and the sum of the differences over all i forms a telescoping sum. The correctness of the tracing algorithm follows from the following claim:

Claim 5.2.1. *For a given coalition S, a randomly generated $C \leftarrow \mathsf{Gen}(1^n)$ and a random pirated feasible $w \leftarrow P(C_S)$ with probability greater than $1 - \text{negl}(n)$, for all $i \notin S$:*

$$\mathop{\mathbb{E}}_{j \text{ in } i\text{th block}}[w_j] - \mathop{\mathbb{E}}_{j \text{ in } (i-1)\text{th block}}[w_j] < \frac{1}{n}.$$

Proof. Fix $i \notin S$. Condition the randomized processes on a specific C_S and w. Columns from the ith block and $(i-1)$th block look identical in C_S. Each such column comes with the same probability from either of the two blocks. There are $\tilde{O}(n^2)$ such columns and $\mathbb{E}_{j \text{ in } i\text{th block}}[w_j]$ sums over a randomly chosen half of them. The same goes for $\mathbb{E}_{j \text{ in } (i-1)\text{th block}}[w_j]$. These sums have the same expectation and their standard deviation is $1/\tilde{O}(n)$. The probability that these sums would deviate from their mean by at least $1/2n$ is bounded by Chernoff bound, and by suitable choice of polylog(n) factors in $\tilde{O}(n^2)$ can be made negligibly small. Thus the probability that there will be a large deviation for any of the $i \notin S$ is negligible. □

The dependence of d on n is not optimal in the fingerprinting codes of Boneh and Shaw [5]. One would like to minimize d as a function of n. Tardos [36] provides a different construction and shows optimality of his parameters. In his construction $d \in \tilde{O}(n^2)$. His generating matrix is chosen at random by the following process: From a certain (artificial) probability distribution, for each column j, pick $p_j \in (0, 1)$. Then chose each entry in jth column of the generating matrix independently according to Bernoulli distribution with probability p_j. It is still open whether choosing the p_j's uniformly from $(0, 1)$ for all columns would give good fingerprinting code.[1]

Now, we will use fingerprinting codes to derive lower bounds on differential privacy using the technique of Bun et al. [7].

Theorem 5.2.2. *If there is a fingerprinting scheme with codewords of length d for $n + 1$ users then there is no $(1, 1/10n)$-differentially private mechanism for answering d attribute means with error $\alpha < 1/2$.*

Proof. The proof proceeds by contradiction. Fix a fingerprinting scheme and let M be the $(1, 1/10n)$-differentially private mechanism for answering attribute means. Generate a (random) fingerprinting code C. Chose $i \in \{1, \ldots, n + 1\}$ at random and set $S = \{1, \ldots, n + 1\} \setminus \{i\}$. Let $(a_1, \ldots, a_d)$ be attribute means obtained from 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, M makes error less than $1/2$, w is a feasible pirated copy for C_S. By the properties of the fingerprinting code

$$\Pr[\mathsf{Trace}(C, w) \in \{1, \ldots, n + 1\} \setminus \{i\}] \geq 1 - \mathrm{negl}(n).$$

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

$$\Pr[\mathsf{Trace}(C, w) = i^*] \geq \frac{1}{2n}.$$

This is a probability over randomly chosen i and hence randomly chosen set S. So there must be a specific i_0 so that if we condition on $i = i_0$

$$\Pr[\mathsf{Trace}(C, w) = i^* \mid i = i_0] \geq \frac{1}{2n}.$$

M is $(1, 1/10n)$-differentially private so for any C its output on $C_{\{1,\ldots,n+1\}\setminus\{i_0\}}$ and $C_{\{1,\ldots,n+1\}\setminus\{i^*\}}$ should be essentially indistinguishable. Hence,

$$\Pr[\mathsf{Trace}(C, w) = i^* \mid i = i_0] \leq \mathrm{e}^1 \cdot \Pr[\mathsf{Trace}(C, w) = i^* \mid i = i^*] + \frac{1}{10n}.$$

This implies

$$\Pr[\mathsf{Trace}(C, w) = i^* \mid i = i^*] \geq \frac{1}{2\mathrm{e}n} - \frac{1}{10\mathrm{e}n} \geq \Omega(1/n),$$

which contradicts the correctness of the tracing algorithm as with non-negligible probability we are accusing someone not among the pirates. □

Notice that the proof is fairly strong and convincing in the sense that for example for Tardos' fingerprinting scheme the data on which M is run are chosen from a distribution of essentially random samples. Hence it is not just some contrived data set on which M fails but it fails on a plausible data set.

[1] This has been proven in work subsequent to our workshop [20].

5.3 Traitor-tracing schemes

A traitor-tracing is intended to solve a scenario related but different from fingerprinting codes. Imagine a satellite broadcast company which for a fee distributes set-top boxes capable of decoding their satellite signal. Each customer gets his own set-top box with a unique decryption key. A party of S customers wants to collude to create (and sell) unauthorized set-top boxes. They will build their set-top box using the decryption keys found in their set-top boxes. The goal of the satellite company is to be able to identify at least one of the colluding customers who contributed by his decryption key.

We can formalize this problem as follows. For a security parameter d, we will have a randomized key generating function $\mathsf{Gen}(1^d, 1^n) \to (k_1, k_2, \ldots, k_n, bk, tk)$ where $k_1, \ldots, k_n \in \{0,1\}^d$ will be customer decryption keys, bk will be the broadcast key and tk will be a key used to trace pirating users. We will have a randomized encryption procedure parametrized by bk which for each plain-text message $b \in \{0,1\}$ will give a ciphertext c, i.e., $\mathsf{Enc}_{bk}(b) \to c$. Decoding is provided by a decoding procedure $\mathsf{Dec}_{k_i}(c) \to b$ which is parametrized by private keys of customers.

We may have two alternative requirements on traitor-tracing procedure:

1. In this case we assume that a pirate procedure $P((k_i)_{i\in S})$ on the input of private keys of users in S outputs a pirate decoder $\tilde{P}$ that distinguishes between $\mathsf{Enc}_{bk}(0)$ and $\mathsf{Enc}_{bk}(1)$. We require that the tracing algorithm $\mathsf{Trace}(\tilde{P}, th)$ outputs a user in S with high probability. The tracing algorithm has an oracle (black-box) access to the pirate decoder $\tilde{P}$ so it cannot peek inside the box to see what kind of decryption keys are hidden inside.

2. Here we assume an existence of a pirate procedure $P((k_i)_{i\in S}, c_1, \ldots, c_k)$ which gets decryption keys of users in S and ciphertext messages $c_1, \ldots, c_k$, and outputs $a_1, \ldots, a_d$ such that if $\mathsf{Dec}_{k_i}(c_i)$ is the same b_j for all $i \in S$ then $a_j = b_j$. The tracing algorithm gets as its input the tracing key tk and it gets oracle access to $P((k_i)_{i\in S}, \cdot)$. It can probe the oracle with arbitrary ciphertexts $c_1, \ldots, c_k$, even malformed ones. The tracing algorithm should identify a member of S with high probability.

A construction for the second alternative was given in Chor et al. [10], see also Ullman [38]. Fix a secure private-key encryption system $(\mathsf{Enc}^0, \mathsf{Dec}^0)$. $\mathsf{Gen}(1^d, 1^n)$ 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:

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

and decoding for user i by:

$$\mathsf{Dec}_{k_i}(c) = \mathsf{Dec}^0_{k_i}(c_i)$$

where c_i is the ith component of the ciphertext.

The tracing algorithm with oracle access to $P((k_i)_{i\in S}, \cdot)$ for some set S works as follows. It generates a fingerprinting code $n \times k$ matrix $C \leftarrow \mathsf{Gen}_{\text{f.p.}}(1^n)$, and it creates a 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 on $c^{(1)}, c^{(2)}, \ldots, c^{(k)}$ to get answers $w = (b_1, \ldots, b_k)$, 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.

For 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. Since w is also feasible for C and S, the fingerprinting tracing algorithm correctly identifies someone in S with high probability.

The traitor-tracing scheme can be used to provide the following computational bound on differentially private mechanisms as shown by Dwork et al. [17] and Ullman [38].

Theorem 5.3.1. *If a traitor-tracing scheme for alternative 2 exists, then any* $(1, 1/10n)$*-differentially private mechanism for answering* $k = k(n, d)$ *efficiently computable counting queries with error* $\alpha < 1/2$ *on data sets with n rows must run in time at least* $s(d)$*, where* $s(d)$ *is the minimum running time of successful pirates in the traitor-tracing scheme.*

Proof. We provide a brief proof sketch. We consider a differentially private mechanism M whose data set is a set of decryption keys (of colluding users). Queries to this mechanism are indexed by ciphertexts, the output of a query is the fraction of keys that decode the ciphertext to one. Assuming that the decoding procedure is efficient this is an efficiently computable query. A pirated set-top box can be build by fixing the data set of M to the decryption keys of colluding users, and on ciphertexts $c^{(1)}, c^{(2)}, \ldots, c^{(k)}$, the box runs M with queries $c^{(1)}, c^{(2)}, \ldots, c^{(k)}$, to get answers $(a_1, \ldots, a_k)$ which are rounded to the nearest integer to get decoding $(b_1, \ldots, b_k)$. By an argument similar to the proof of Theorem 5.2.2 we conclude that if M is differentially private then the traitor-tracing scheme must fail. If the tracing scheme does not fail then either M is not differentially private or M uses at least as much resources as a successful pirates for the traitor-tracing scheme. □

LECTURE 6

Pirates of the Caribbean II

Lecturer: Salil Vadhan *Scribe: Borja Balle*

This lecture continues the discussion on computational lower bounds for differentially private mechanisms. We will be using *finger printing codes* (FPCS) and *traitor tracing schemes* (TTS) as discussed in the previous lecture.

6.1 Cryptographic lower bounds

Theorem 6.1.1 ([17, 38])**.** *If a TTS of type 2 with the above notation exists, then any* $(1, 0.1/n)$*-differentially private mechanism for answering* $k = k(n, d)$ *efficiently computable counting queries with error* $\alpha < 1/2$ *on datasets with n individuals from* $\mathcal{X} = \{0,1\}^d$ *must run in time* $\gtrsim s(d)$*.*

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 TTS 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 a TTS of type 2 with $n+1$ users and take a database x containing the keys of a coalition of n users obtained by removing one user at random. We consider counting queries on this dataset given by ciphertext decryption: for any 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 ith user with its key k_i. Therefore, when query q_c is answered by $\mathcal{M}$ on the dataset x we obtain an α-approximation $a = \mathcal{M}(q_c, x)$ to the number of users in x whose key decrypts c to 1; we denote by b the $\{0,1\}$ rounding of a. With this notation, we define a pirate P for the type 2 TTS that given ciphertexts $c_1, \ldots, c_k$ returns $P(x, c_1, \ldots, c_k) = (b_1, \ldots, b_k)$, where b_i is the rounding of $a_i = \mathcal{M}(q_{c_i}, x)$.

Note that assumption $\alpha < 1/2$ implies that P is feasible. In addition, using the same argument as in the lower bounds involving FPCS one can see that the tracing property of TTS implies that $\mathcal{M}$ cannot be differentially private unless the running time of P is $\gtrsim s(d)$. Now the bound on the running time of $\mathcal{M}$ follows from the construction of P. □

A similar construction using type 1 TTS can be used to prove lower bounds for the time complexity of mechanisms that produce differentially private digests from datasets. In particular, using the recent type 1 TTS candidate of Garg et al. [22] with ciphertext length $\mathrm{poly}(d)$ one can show that it is computationally hard to produce a digest for a dataset with $n = \mathrm{poly}(d)$ individuals from which $k = 2^{\mathrm{poly}(d)}$ queries can be answered with error $\alpha < 1/2$. However, it is important to note that this hardness result relies on non-standard cryptographic assumptions.

The previous lower bounds involve query families based on cryptographic primitives. An important question is whether we can obtain similar bounds for families of queries that arise in practical situations. Although some results of this sort are known, many question remain open. We discuss some of these in the next two sections.

6.2 Known results for interval queries

A simple but important problem is that of answering *interval queries*. Given a universe $\mathcal{X} = \{1, \ldots, D = 2^d\}$, an interval query q_θ for $\theta \in \mathcal{X}$ is defined by $q_\theta(x) = \mathbb{I}[x \geq \theta]$. The following table summarizes known bounds on the order of the error needed to answer all interval queries with a differentially private mechanism. See Beimel et al. [2] and references therein for details. An important open question is whether it is possible to prove a lower bound on the noise required to answer interval queries in the (ϵ, δ)-differential privacy setting which exhibits an explicit dependence on the dimension d.[1]

	Upper bound	Lower bound
$(\epsilon, 0)$-dp	$\frac{\text{poly}(d)}{\epsilon n}$	$\frac{d}{\epsilon n}$
(ϵ, δ)-dp	$\left(\frac{2^{O(\log^* d)} \log(1/\delta)}{\epsilon n}\right)^{2/5}$	$\frac{\log(1/\delta)}{\epsilon n}$

6.3 Computational lower bounds for t-way marginals

The set of all t-way conjunctions is a natural set of queries for which we know how to prove some computational lower bounds. Let us consider the universe $\mathcal{X} = \{0,1\}^d$ and denote by Q the set of all t-way conjunctions over d literals for some fixed t. Note that we have $k = |Q| = 2^t \binom{d}{t}$.

For small t, we can use the Laplace mechanism with independent noise for each query to give differentially private answers to all queries in Q within error $o(1)$ in a data set with $n \gtrsim \sqrt{k} \gtrsim d^{t/2}$ individuals. This has running time $\text{poly}(n)$ per query. An interesting question is whether we can achieve similar privacy results for databases with $n < d^{t/2}$ individuals.

On the other hand, instead of giving explicit answers to all possible queries in Q or providing interactive access to the dataset x, a common approach is to produce a *synthetic dataset*, like Theorem 3.1.1. This is basically a differentially private digest of the original dataset which can be used to answer any query in Q with small error. The following result shows that under some standard cryptographic assumptions it is hard to produce such digests in polynomial time.

Theorem 6.3.1 ([39])**.** *Assuming exponentially secure digital signatures exists, there exists a constant $\alpha > 0$ such that there is no $(1, 0.1/n)$-differentially private mechanism that given a dataset with n individuals over $\mathcal{X} = \{0,1\}^d$ outputs a sanitized dataset approximating all 2-way marginals to within error α in time $2^{d^{1-o(1)}}$ for any $n \leq 2^{d^{1-o(1)}}$.*

Before we can proof this results we need to review digital signatures and the PCP theorem.
A *digital signature scheme* (DSS) is given by a triple of algorithms as follows:

1. A *key generation* randomized 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 *signing* randomized algorithm that given a message $m \in \{0,1\}^d$ and a secret key produces a signature $\sigma = \mathsf{Sign}_{sk}(m)$.

3. A *verification* algorithm that accepts $\mathsf{Ver}_{pk}(m, \sigma)$ if and only if σ is a signature for m generated using the secret key sk corresponding to pk.

[1] This has been done in work subsequent to our workshop [8], but there still is a gap between the lower bound of $\log^* d$ and the upper bound of $2^{\log^* d}$.

We say that a DSS is *exponentially 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 $2^{o(d)}$ cannot generate a new pair (m', σ') such that $\mathsf{Ver}_{pk}(m', \sigma') = 1$.

Secure DSS can be used to show the existence of hard to sanitize datasets for cryptographically defined queries. This will be the starting point for the construction of a hard to sanitize dataset with respect to t-way marginals.

Lemma 6.3.2 ([17])**.** *Assuming secure DSS exist, there exist datasets which are hard to sanitize.*

Proof. Let us consider a dataset x with n individuals, where each row contains a pair (m_i, σ_i) with m_i randomly generated and $\sigma_i = \mathsf{Sign}_{sk}(m_i)$ for some fixed secret key. Take the counting query q defined by applying $\mathsf{Ver}_{pk}(\cdot)$ to each row of x, where pk is the public key associated with sk. This query is efficiently computable and evaluates to 1 on the whole dataset. Now suppose there exists a differently private mechanism $\mathcal{M}$ with running time $2^{o(d)}$ that given x produces a digest y which is accurate with respect to q. By the accuracy, y must contain at least one row $y_j = (m'_j, \sigma'_j)$ such that $q(y_j) = \mathsf{Ver}_{pk}(m'_j, \sigma'_j) = 1$. We note that this is a contradiction: if $y_j \notin x$, then $\mathcal{M}$ succeeded in creating a forgery for the DSS in time $2^{o(d)}$, and if $y_j \in x$ then $\mathcal{M}$ is not differentially private. □

Recall that *circuit SAT* is an NP-hard problem. Then, by 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 output a 3-CNF $\mathsf{Red}(C) = \phi = \phi_1 \wedge \ldots \wedge \phi_m$ 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.

Proof of Theorem 6.3.1. We prove the result for 3-way marginals; the proof for 2-way marginals is more technical and requires another version of the PCP theorem.

Let x be the dataset from the proof of Lemma 6.3.2 based on DSS. Let C be the circuit corresponding to the verification algorithm Ver_{pk} from the DSS. 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: $z_i = \mathsf{Enc}(C, x_i)$. We denote by $\phi = \phi_1 \wedge \cdots \wedge \phi_m$ the 3-CNF obtained by $\mathsf{Red}(\mathsf{Ver}_{pk})$. Note that for every row z_i in z we have $\phi(z_i) = 1$, and for every clause ϕ_j in ϕ we have $\phi_j(z) = n^{-1} \sum_{i \in [n]} \phi_j(z_i) = 1$. Suppose $\mathcal{M}$ is a differentially private mechanism that produces sanitized datasets which are α-accurate with respect to 3-way conjunctions and let $z' = \mathcal{M}(z)$. Then for every $j \in [m]$ we have $\phi_j(z') \geq 1 - \alpha$, which implies that there exists some row z'_i of z' that satisfies at least $(1 - \alpha)m$ clauses from ϕ. Therefore, using this row from the sanitized dataset we can obtain $(m', \sigma') = \mathsf{Dec}(\mathsf{Ver}_{pk}, z'_i)$ such that $\mathsf{Ver}_{pk}(m', \sigma') = 1$. Now the same argument used in Lemma 6.3.2 shows that either (m', σ') is a forgery or a violation of privacy. Thus, we conclude that $\mathcal{M}$ must have running time $\gtrsim 2^{d^{1-o(1)}}$. □

LECTURE 7

2-way and t-way Marginals Using Geometric Algorithms

Lecturer: Salil Vadhan *Scribe: Lila Fontes*

The setup:

- people: $\vec{v} \in \{-1,-1\}^d$.
- queries: 2-way marginals (conjunctions).
- n known (or at least a factor of 2 upper bound on n is known).
- $k \approx d^2$ queries.

With independent noise, we can get error $\tilde{O}(\sqrt{k}) = \tilde{O}(d)$ efficiently.

With private multiplicative weights, we can get error $\tilde{O}(\sqrt{n} \cdot d^{1/4})$ and time polynomial in n and 2^d. (For some parameters, this error cannot be improved—see previous lecture notes for lower bounds.)

We'll represent the database as a histogram $x \in \mathbb{R}^{\{-1,1\}^d}$, where x_v is the number of people of type $v \in \{-1,1\}^d$. (For convenience, we use the relaxation of a database over $\mathbb{R}$, not $\mathbb{N}$.)

We'll define neighbors:

$$x \sim x' \iff |x - x'|_1 \leq 1\,.$$

One-way marginals are simple counts, e.g.:

$$q_1^1 = \sum_{v \in \{-1,1\}^d} \frac{1+v_1}{2} \cdot x_v$$

is the count of people in the database with first coordinate = 1.

Two-way marginals are counts of conjunctions, e.g., the count of people with i^{th} coordinate = 1 and j^{th} coordinate = 1 is:

$$\begin{aligned} q_{ij}^{11} &= \sum_{v \in \{-1,1\}^d} \left(\frac{1+v_i}{2}\right)\left(\frac{1+v_j}{2}\right) x_v \\ &= \sum_{v \in \{-1,1\}^d} \left(\frac{1}{4} \cdot x_v + \frac{v_i}{4} \cdot x_v + \frac{v_j}{4} \cdot x_v + \frac{v_i \cdot v_j}{4} \cdot x_v\right)\,. \end{aligned}$$

So if we can release all parities up to 2 with error ϵ, then we can release all marginals up to 2 with error ϵ (because the sum of the magnitude of coefficients is less than 1, so clever Fourier business lets us compute the function from these coefficients).

We can devise a matrix Q with d^2 rows (one for each pair (i, j)) and 2^d columns (one for each vector $v \in \{-1, 1\}^d$) such that

$$Qx = y$$

is a vector in $\mathbb{R}^{d^2}$ which gives answers to all the 2-way parities on database x:

$$y_{ij} = \sum_{v \in \{-1,1\}^d} (v_i v_j) x_v \,.$$

If $|x| = 1$ (the database has only one member!), what answers are possible? Just columns of Q. But since x is a $\mathbb{R}$ vector, we might actually get a convex hull of $\pm$ columns of Q. Call this convex hull K (see Figure 7.1).

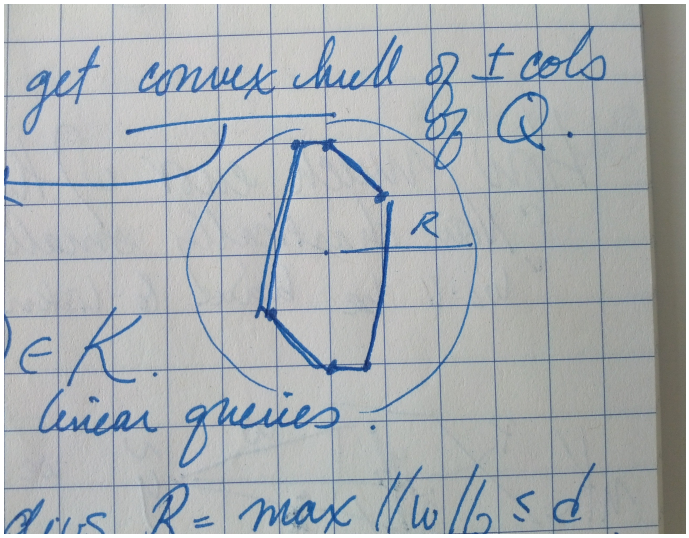

Figure 7.1: The convex hull K.

For databases $x \sim x'$ of arbitrary size, $y - y' = Q(x - x') \in K$. (Think of Q as an arbitrary set of linear queries.)

There is a ball containing this K of radius $R = \max_{w \in \text{cols}(Q)} \|w\|_2 \leq d$. (This radius depends only on d, not n—so it's bad for small n and large $|Q|$, which is what intuition about privacy would suggest: it will be hard to preserve privacy when asked many queries about a small population.)

What we'll try first for privacy:

1. Calculate the answer!
$$y = Qx$$
2. Make it differentially private.
$$\tilde{y} = y + R\frac{\sqrt{\log(1/\delta)}}{\epsilon} \cdot N(0,1)^{d^2}$$
where for $N(0, 1)$ we sample each coordinate independently from a Gaussian distribution.

   Cynthia: This is "noise statisticians are more comfortable with."

   Kunal: This is "noise complex geometers are more comfortable with, which is convenient for us here."

It's likely that the noise we've added is much too large, so we'll need to project this back into the convex hull K. For example, if $n = o(d)$ then the noise will be $Rd \approx ||y - \tilde{y}||_2$, but $||y|| \leq nd$.

We're shooting for $n \approx \sqrt{d}$, with constant relative error. We know n, but $\tilde{y}$ is likely tou end up outside the nK polytope. (Kunal: "Anyone can look at this $\tilde{y}$ and say, this is bullshit.")

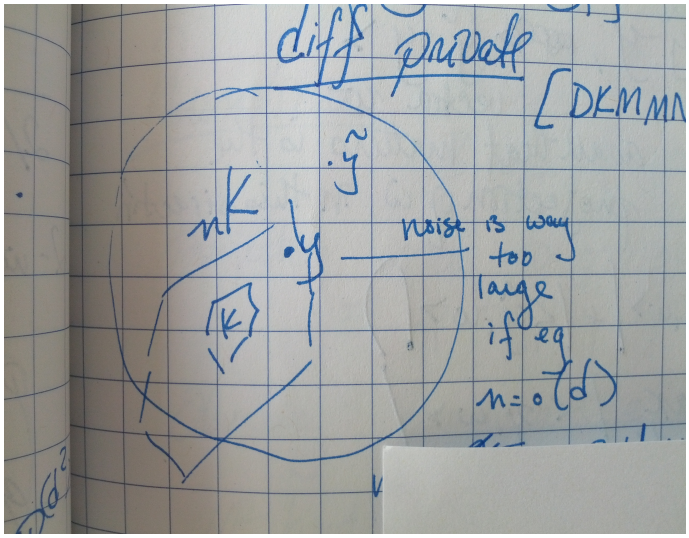

Figure 7.2: The noise is way too large.

So we need to add a step 3:

3 Adjust this by projecting.

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

Note that this step can be done in public (there's no privacy to preserve here, we have already added the privacy-protecting noise).

How much error does this mechanism (steps 1-2-3) have? Let p be the point where the perpendicular from $\tilde{y}$ intersects the line defined by y and $\hat{y}$.

We know $||p - \tilde{y}||_2 \leq ||y - p||_2$. Then

$$\begin{aligned}
||y - \hat{y}||_2^2 &= \langle y - \hat{y}, y - \hat{y} \rangle \\
&\leq 2\langle \hat{y} - y, p - y \rangle && \text{because } y - \hat{y} \leq p - y \\
&= 2\langle \hat{y} - y, \tilde{y} - y \rangle && \text{all that matters is the projection of } w = \tilde{y} - y \text{ in this direction} \\
&\leq 2\left(|\langle y, w \rangle| + |\langle \hat{y}, w \rangle|\right) && \text{triangle inequality} \\
&\leq 4 \max_{z \in nK} |\langle z, w \rangle|
\end{aligned}$$

This is the squared error mechanism. Then it follows that

$$\mathop{\mathbb{E}}_{w \text{ Gaussian}} ||y - \hat{y}||_2^2 \leq 4 \mathop{\mathbb{E}}_{w} \max_{z \in nK} |\langle z, w \rangle|$$

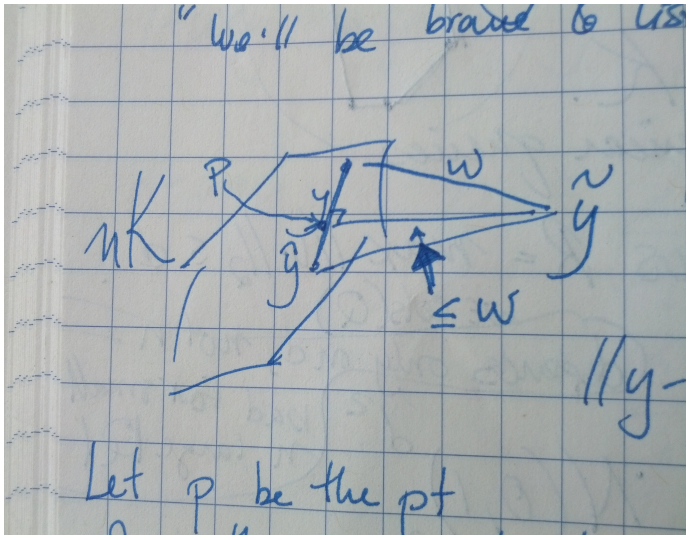

Figure 7.3: This is how we project back into the nK polytope.

and

$$\begin{aligned} 4Rn(\text{width of polytope}) &\le 4Rn \mathop{\mathbb{E}}_{w\sim N(0,1)^{d^2}} \max_{z\in K} |\langle z, w\rangle| \\ &\le 4dn \cdot d\sqrt{d}\,. \end{aligned}$$

As

$\mathop{\mathbb{E}}_{w\in N(0,1)} \max_{z\in K} \lvert w - z\rvert$	this max obtained at one of the 2^d vertices of polytope K
$\le \mathbb{E} \max 2^d \cdot N(0, d^2)$	total L_2 weight of z is $\approx d$ and random variables $-d \cdot \mathbb{E}\max(g_1, g_2, \dots, g_{d^2})$, each $N(0,1)$
$\le d\sqrt{\log 2^d} = d\sqrt{d}\,.$	by union bound.

And generally, this is $\sqrt{kd}$ for k queries.

$$\mathop{\mathbb{E}}_{ij\in[d]\times[d]} |\hat{y}_{ij} - y_{ij}|^2 \le n\sqrt{d}$$

So average root mean square error of this mechanism is $\le \tilde{O}_{\epsilon\delta}(\sqrt{n} \cdot d^{1/4})$. This matches the PMW error.

What is the runtime of this mechanism? Steps 1 ($nk = nd^2$ time) and 2 (linear in K) are no problem. Step 3 is ok if you use a heuristic, but definitely dominates the runtime. Projection is equivalent (up to a polynomial) to optimization (of a linear function over K).

The obvious way to implement step 3 is to enumerate over all vertices; this is poly$(n, 2^d, k)$ and we prefer something which is simply poly(d).

It is NP-hard to do this optimally. So instead, we'll find an L close to K (where $K \subseteq L$) and optimize over L. We need to ensure that the width of L is comparable to the width of K.

New task: find $K \subseteq L$ such that:

- we can efficiently optimize over L, and
- $\ell^*(L) \lesssim \ell^*(K)$.

Recall that for $v \in \{-1,1\}^d$, we have $K = $ convex hull$(\pm v \otimes v)$, where the ij^{th} entry is $v_i v_j$, containing 2^d vertices.

$$K \subseteq L_0 = \text{ convex hull}(\pm v \otimes v' : v, v' \in \{-1,1\}^d)$$

The optimizing problem is:

$$\max_{v,v' \in \{-1,1\}^d} \sum_{ij} g_{ij} v_i v'_j .$$

This is still NP-hard, but Grothendieck's inequality applies:

$$\begin{aligned} &\leq \max_{\{u_i\}_{i=1}^d, \{u'_j\}_{j=1}^d \text{ unit vectors}} \sum_{ij} g_{ij} \langle u_i, u'_j \rangle \\ &\leq 2 \max_{v,v' \in \{-1,1\}^d} \sum_{ij} g_{ij} v_i v'_j . \end{aligned}$$

So if

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

then efficient optimization over L is easy: you solve a small SDP.

We can update step 3 with argmin to get an efficient algorithm with polynomial runtime and similar error.

(Note: for all distributions D there exists a polytime algorithm such that

$$\mathop{\mathbb{E}}_{ij \in D} \mathop{\mathbb{E}}_{\text{mechanism}} |\hat{y_{ij}} - y_{ij}| \leq \sqrt{n} d^{1/4} .$$

That is, small error over any distribution. But the minimax argument doesn't work on this claim: it's no longer differential privacy.)

Theorem 7.0.1 (Boosting for queries)**.** *If, for any distribution D, we can privately obtain average error λ with a summary of s bits, then we can privately obtain worst-case error $\lambda + \tilde{O}(\sqrt{s})$ (ignoring constants and log terms).*

Kunal: "This is an old result—from 2010."

Therefore, two-way marginals can be privately obtained with worst-case error guarantees in polynomial time.

t-way marginals are reducible to two-way marginals over ¿d^2? coordinates.

Open question: Let K be the convex hull of $v \otimes v \otimes v$ where $v \in \{-1,1\}^d$. Does there exist a set L such that $K \subseteq L$ and $\ell^*(L) \leq c \cdot \ell^*(K)$?

We can't do this with linear programs if we ask the stronger requirement $K \subseteq L \subseteq c \cdot K$.

LECTURE 8

Large Conjunctions

Lecturer: Salil Vadhan *Scribe: Luc Segoufin*

8.1 Computational complexity of t-way conjunctions

In this lecture $\mathcal{X} = \{0,1\}^d$.

Recap:

- It is hard to generate synthetic data even for $t = 2$ (NP-hard).
- We have mechanisms with error in $\tilde{O}(d^{t/4}/(\epsilon n))$ and running time Poly(n, d^t) [33, 19].

This lecture: A mechanism with error $o(1)$ and running time Poly$(n, d^{O(\sqrt{t})})$, assuming $n \geq d^{O(\sqrt{t})}$ [25, 37].

There is another result that we will not cover: A mechanism with error $o(1)$ and running time Poly$(n, 2^{o(d)})$, assuming $n \geq O(t \cdot d^{.51})$ [9].

Starting with our database x with n rows in $\mathcal{X}$ the mechanism $\mathcal{M}$ will produce a "summary" S which will add error to the function f_x defined as $f_x(q) = q(x)$. S will be a polynomial of low degree. For notational convenience the queries will be monotone disjunctions specified by $y \in \{0,1\}^d$:

$$q_y(w) = \bigvee_{i\,:\,y_i=1} w_i\,, \qquad w \in \mathcal{X}\,. \tag{8.1}$$

("Monotone" is without loss of generality as we can duplicate attributes otherwise. Disjunction is the same as conjunction modulo negation.)

Notice that:

$$q_y(w) = \begin{cases} 1 & \exists i\ w_i = y_i = 1, \\ 0 & \text{otherwise.} \end{cases} \tag{8.2}$$

Fact 8.1.1 (Chebishev Polynomials)**.** $\forall t \in \mathbb{N}, \forall \alpha > 0$ there exists a univariate polynomial g of degree $O\big(\sqrt{t}\log(1/\alpha)\big)$ such that $g(0) = 0$ and for $i \in \{1 \cdots t\}$, $1 - \alpha \leq g(i) \leq 1 + \alpha$.

Given $w \in \mathcal{X}$, consider the following function defined for y with at most t entries set to 1:

$$f_w(y) = g(\Sigma_j w_j y_j), \tag{8.3}$$

where g is the Chebishev polynomial of the Fact. f_w can be viewed as a multivariate polynomial $g_w(y_1, \cdots, y_d)$ of degree $\sqrt{t}$. Its coefficients have magnitude $d^{O(\sqrt{t})}$.

By construction we have that for all $x, y \in \mathcal{X}$,

$$|g_x(y) - q_y(x)| \leq \alpha\,. \tag{8.4}$$

The desired mechanism adds $\mathrm{Lap}(d^{O(\sqrt{t})}/(\epsilon n))$ to each coefficient. The total error is $\alpha + d^{O(\sqrt{t})}/(\epsilon n)$. S is the resulting polynomial, $g_{\mathcal{X}}$ + error.

Open Question 8.1.2. Can we summarize all conjunctions in time poly(d)?

8.2 Private PAC learning

These results are based on Kasiviswanathan et al. [26].

PAC learning:

- a concept class $C = \{c : \{0,1\}^d \to \{0,1\}\}$,
- an hypothesis class $H = \{h : \{0,1\}^d \to \{0,1\}\}$,

both poly(d)-time computable.

Definition 8.2.1. A concept class C is *PAC-learnable* if there exists an algorithm L and a number n polynomial in d, called the *sample complexity*, such that $\forall D$ distribution on $\{0,1\}^d$, $\forall c \in C$, and $\forall x_i$ chosen independently according to D, $L(x_1, c(x_1), \cdots, x_n, c(x_n), d)$ returns a function $h \in H$ such that with high probability $h(x_{n+1}) = c(x_{n+1})$.

We speak of *proper learning* when $H = C$. We speak of *efficient learning* when L is poly-time computable. (We will write $L(x, c(x))$ for $L(x_1, c(x_1), \cdots, x_n, c(x_n), d)$).

We obtain the notion of *Private PAC-learning* by further requiring that L be differentially private: $\forall x, x', y, y'$ such that $(x, y) \sim (x', y')$, $L(x, y)$ and $L(x', y')$ should be (ϵ, δ)-indistinguishable.

Theorem 8.2.2 ([26])**.** *If C is (non-privately) PAC-learnable then it is (ϵ, δ)-privately and properly PAC-learnable with sample complexity $O(d \cdot \mathrm{VC}(C))$.*

Proof. Let $H = C$. On input $(x_1, y_1) \cdots (x_n, y_n)$ we output $h \in H$ with probability

$$\propto e^{-\epsilon \#\{i : h(x_i) \neq y_i\}}\,.$$

By what we have seen in the proof of Theorem 3.1.1 with the exponential mechanism, this is 2ϵ-differentially private. Assume now that $x_1, \cdots, x_n$ are taken according to some distribution D. Let $y_i = c(x_i)$.

We record a few facts about this state of affairs:

1. If $n \geq O(\mathrm{VC}(C)/\alpha^2)$ then with high probability over $x_1 \cdots x_n$

$$\forall h \in C, \frac{\#\{i : h(x_i) = c(x_i)\}}{n} = \Pr_{w \sim D}[h(w) = c(w)] \mp \alpha.$$

2. $|C| \leq (2^d)^{\mathrm{VC}(C)}$.

From this we derive:

$$\begin{aligned}\Pr_{\text{coins of } L}[L \text{ outputs a } h \text{ with error } > 2\alpha] &\leq \frac{\Sigma_{h \text{ with error } >\alpha} e^{-\epsilon\alpha n}}{e^{-\epsilon\cdot 0}} \\ &\leq |C| e^{-\epsilon\alpha n} \\ &\leq 2^{d\cdot \mathrm{VC}(C)} e^{-\epsilon\alpha n}.\end{aligned}$$

We are done when taking

$$n = O\left(\max\left\{\frac{d\cdot \mathrm{VC}(C)}{\epsilon\alpha}, \frac{\mathrm{VC}(C)}{\alpha^2}\right\}\right) \ll 1. \qquad \square$$

Questions.

Open Question 8.2.3. What about efficient private PAC-learning? We know that everything that is efficiently PAC-learnable in the "statistical query model" is efficiently and privately PAC-learnable; we also know that parities are efficiently and privately PAC-learnable.

It is open whether there is a separation between private and non private PAC-learning.[1]

Open Question 8.2.4. Is the gap between sample complexities inherent? ($\mathrm{VC}(C)$ vs. $d\cdot \mathrm{VC}(C)$)? Yes for $(\epsilon, 0)$-differentially private (next lecture). Open for (ϵ, δ)-differentially private.

Open Question 8.2.5. What about learning problems outside the PAC framework, such as learning parameters of structured distributions as studied in traditional statistical inference and applied machine learning? Some very general results along these lines are given in [35], but we still are far from having a complete understanding of what is possible.

[1] Subsequent to the workshop, evidence of a separation was provided in [6].

LECTURE 9

Private Learning

Lecturer: Salil Vadhan *Scribe: Cristopher Moore*

9.1 A computationally efficient private mechanism for learning parity

In this lecture we continue our discussion of private PAC learning. To recap, Kasiviswanathan et al. [26] showed that if a concept class $\mathcal{C} \in \{0,1\}^d$ is PAC-learnable, then it is privately PAC-learnable with sample complexity $O(d \cdot \mathrm{VC}(\mathcal{C}))$ using the exponential mechanism. This mechanism is not computationally efficient, but for specific problems we can do better. In particular, they gave a computationally efficient private learner for parity functions, whose sample complexity is within a constant of that of the best non-private learner.

Let $\mathcal{C}$ be the class of parities: that is, the class of functions $\{0,1\}^d \to \{0,1\}$ of the form $x \mapsto c \cdot x$. We have a dataset (x, y) with n rows (x_i, y_i), where $x_i \in \{0,1\}^d$ and $y_i \in \{0,1\}$. We are promised that the data is consistent with the class: that is, there is some $c \in \{0,1\}^d$ such that $y_i = c \cdot x_i$ for all $1 \le i \le n$. As always in PAC learning, our goal is the following. Assume that $x_1, \ldots, x_n$ are drawn independently from some distribution D. 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 (non-private) algorithm is to take any h such that $y_i = h \cdot x_i$ for all i. We can do this by Gaussian elimination, i.e., by solving the system of linear equations $y = h \cdot x$. Standard results show that this succeeds with $n = O(d)$ samples. Since the class $\mathcal{C}$ of parities has $\mathrm{VC}(\mathcal{C}) = d$, we have $n = O(\mathrm{VC}(\mathcal{C}))$.

Now let's consider private learning. Keep 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 don't 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 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 $V = \emptyset$, i.e., if no consistent h exists, then output $\perp$.

Since the non-private algorithm decribed above succeeds with probability 0.99, if the data is consistent then $\mathcal{M}$ succeeds with probablity 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.

Theorem 9.1.1. *$\mathcal{M}$ is $(2\varepsilon, 0)$-differentially private.*

Proof. 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 $\mathcal{M}(x)$ and $\mathcal{M}(x')$ are ε-close, 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,
$$\Pr[\mathcal{M}(x) = z] \geq (1 - \varepsilon) \Pr[\mathcal{M}(x') = z]. \tag{9.1}$$
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
$$\Pr[\mathcal{M}(x) = z] \leq (1 + \varepsilon) \Pr[\mathcal{M}(x') = z]. \tag{9.2}$$

To prove (9.2), start by fixing $(\hat{x}_j, \hat{y}_j) = (\hat{x}'_j, \hat{y}'_j)$ for all $j \neq i$. (If you like, we are coupling the algorithm's choices on the two inputs, so that $\mathcal{M}(x)$ and $\mathcal{M}(x')$ agree on these rows after step 3. We also couple the events that they fail in step 2.) 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 $V_{-i} = \emptyset$), 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 $V = \emptyset$.

Let's compare the probabilities that $\mathcal{M}(x)$ and $\mathcal{M}(x')$ fail. If $V_{-i} = \emptyset$, then $\mathcal{M}(x) = \mathcal{M}(x') = \perp$. But if $V_{-i} \neq \emptyset$ but $V = \emptyset$, 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
$$\Pr[\mathcal{M}(x) = \perp] \leq \frac{1 + \varepsilon}{2} \leq (1 + \varepsilon) \Pr[\mathcal{M}(x') = \perp].$$

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_i are nonempty and contain h. Linear algebra tells us that
$$|V| \geq \frac{1}{2}|V_{-i}|.$$
Since $\mathcal{M}(x)$ and $\mathcal{M}(x')$ are uniform in V and V_{-i} respectively, for any $h \in V_{-i}$ we have
$$\Pr[\mathcal{M}(x) = h] \leq \frac{1}{2}\left(\frac{1 - \varepsilon}{|V_{-i}|} + \frac{\varepsilon}{|V|}\right) \leq \frac{1}{2}\frac{1 + \varepsilon}{|V_{-i}|} = (1 + \varepsilon) \Pr[\mathcal{M}(x') = h],$$
which completes the proof. □

9.2 Private PAC learning and communication complexity

The results of [26] show that $O(d \cdot \mathrm{VC}(\mathcal{C}))$ samples suffice for private PAC learning. On the other hand, even without privacy, PAC learning requires at least $\mathrm{VC}(\mathcal{C})$ samples. Is this factor of d necessary?

To get some sense of why it might be natural, note that

$$\mathrm{VC}(\mathcal{C}) \leq \log |\mathcal{C}| \leq d \cdot \mathrm{VC}(\mathcal{C}) .$$

The first inequality follows trivially from the shattering definition of the VC dimension; the second inequality comes from Sauer's lemma (more properly known as the Perles-Sauer-Shelah lemma).

In fact there are classes $\mathcal{C}$ that can be learned non-privately with $O(\mathrm{VC}(\mathcal{C}))$ samples, but for which learning with $(\varepsilon, 0)$-differential privacy requires $\Omega(d \cdot \mathrm{VC}(\mathcal{C}))$ samples. One such class is the class of threshold functions $\{c_\theta \mid 0 \leq \theta \leq 2^d\}$, where

$$c_\theta(x) = \begin{cases} 1 & x \geq \theta , \\ 0 & x < \theta . \end{cases}$$

It is easy to see that $\mathrm{VC}(\mathcal{C}) = 1$. However, Beimel et al. [1] showed that the sample complexity of proper private learning is $\Omega(d)$, and this was extended to improver private learning by Feldman and Xiao [21].

This gap is proven using beautiful connections between VC dimension, private learning, and communication complexity. Suppose that Alice has a function $c \in \mathcal{C}$, Bob has a string $x \in \{0,1\}^d$, and together they want to compute $c(x)$. 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 $c(x)$.

We can also consider randomized communication complexity, where Alice and Bob want to succeed with some probability $1 - \delta$. (We will assume that Alice and Bob have access to shared randomness, i.e., a public coin.) The complexity then depends on the distribution of c and x. Given a distribution μ on $\mathcal{C} \times \{0,1\}^d$, we denote one-way, public-coin communication complexity as $\mathrm{CC}^{\rightarrow,\mathrm{pub}}_{\mu}(\mathcal{C})$.

Kremer et al. [28] showed that the VC dimension is closely related to the one-way communication complexity maximized over all product distributions,

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

where μ_A and μ_B are distributions on $\mathcal{C}$ and $\{0,1\}^d$ respectively. (The constant hidden in Θ depends on the success probability $1 - \delta$.) In contrast, Feldman and Xiao [21] showed that the sample complexity of privately learning $\mathcal{C}$ is related to the one-way communication complexity maximized over all *joint* distributions on $\mathcal{C} \times \{0,1\}^d$. This can be larger than the maximum over product distributions, and thus larger than $\mathrm{VC}(\mathcal{C})$, due to perverse correlations between c and x. In particular, Alice and Bob now need to succeed with probability $1 - \delta$ on all pairs (c, x), so we can simply write

$$\mathrm{CC}^{\rightarrow,\mathrm{pub}}(\mathcal{C}) = \max_{\mu} \mathrm{CC}^{\rightarrow,\mathrm{pub}}_{\mu}(\mathcal{C})$$

and

$$\text{private sample complexity} = \Theta\big(\mathrm{CC}^{\rightarrow,\mathrm{pub}}(\mathcal{C})\big) .$$

Returning to the threshold function, computing $c_\theta(x)$ is equivalent to computing the "greater than" function. Miltersen et al. [30] showed that for this problem we have $\mathrm{CC}^{\rightarrow,\mathrm{pub}} = \Omega(d)$, giving a more general proof [28] that private learning requires $\Omega(d)$ samples.

We conclude by sketching the proof from [21] that $\mathrm{CC}^{\rightarrow,\mathrm{pub}}(\mathcal{C}) = O(\text{sample complexity})$. Let L be a $(\varepsilon, 0)$-differentially private learner with a given sample complexity n. Using their shared randomness, Alice and Bob both run L on the all-zeros database $(0^d, 0)^n$. 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 $(0^d, 0)^n$ is at most n rows different from any database, and since L is $(\varepsilon, 0)$-differentially private, the distribution of functions returned by L "covers" the distribution on any other database D, in the sense that for each $h \in \mathcal{H}$,

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

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 database.

In particular, let μ be a distribution on pairs (c, x), 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 database we would get by sampling x from the conditional distribution $\mu(x \mid c = c_0)$: that is, $h_i(x) = c_0(x)$ with high probability in x. Alice can send Bob this index i with communication complexity $\log M = O(\varepsilon n)$.

Given the fact that $\mathrm{CC}^{\rightarrow,\mathrm{pub}}(\mathcal{C}) = \Omega(d \cdot \mathrm{VC}(\mathcal{C}))$ for some classes, this shows that the gap of d between public and private learning is sometimes unavoidable.

We give an even quicker sketch of the other direction, that the private sample complexity is $O(\mathrm{CC}^{\rightarrow,\mathrm{pub}}(\mathcal{C}))$. Namely, given a low-communication protocol between Alice and Bob, the messages from Alice define a small set of hypotheses, from which we can privately sample using the exponential mechanism.

Bibliography

[1] Amos Beimel, Shiva Prasad Kasiviswanathan, and Kobbi Nissim. Bounds on the sample complexity for private learning and private data release. In *Proceedings of TCC 2010*, 2010. 43

[2] 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. 31

[3] Avrim Blum, Cynthia Dwork, Frank McSherry, and Kobbi Nissim. Practical privacy: the sulq framework. In *Proceedings of the twenty-fourth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, pages 128–138. ACM, 2005. 6

[4] Avrim Blum, Katrina Ligett, and Aaron Roth. A learning theory approach to non-interactive database privacy. In *Proceedings of the Fortieth Annual ACM Symposium on Theory of Computing*, STOC '08, pages 609–618, New York, NY, USA, 2008. ACM. doi:10.1145/1374376.1374464. 16, 18

[5] D. Boneh and J. Shaw. Collusion-secure fingerprinting for digital data. *IEEE Transactions on Information Theory*, 44(5):1897–1905, Sep 1998. 25, 26, 27

[6] 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. URL `http://eprint.iacr.org/2015/417`. 40

[7] 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. 27

[8] 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 '15)*, pages 634–649. IEEE, 18–20 October 2015. Full version posted as arXiv:1504.07553. 31

[9] 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. 38

[10] Benny Chor, A Fiat, M. Naor, and B. Pinkas. Tracing traitors. *IEEE Transactions on Information Theory*, 46(3):893–910, May 2000. 25, 28

[11] 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. 6, 20, 21

[12] Cynthia Dwork and Jing Lei. Differential privacy and robust statistics. In *STOC*, pages 371–380, 2009. 13

[13] Cynthia Dwork and Kobbi Nissim. Privacy-preserving datamining on vertically partitioned databases. In *Advances in Cryptology–CRYPTO 2004*, pages 528–544. Springer, 2004. 6

[14] Cynthia Dwork and Aaron Roth. The algorithmic foundations of differential privacy. *Foundations and Trends in Theoretical Computer Science*, 9(3-4):211–407, 2013. ISSN 1551-305X. doi:10.1561/0400000042. 9

[15] Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. In *Theory of cryptography*, pages 265–284. Springer, 2006. 6, 7

[16] 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. 22

[17] 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. 29, 30, 32

[18] Cynthia Dwork, Guy N. Rothblum, and Salil P. Vadhan. Boosting and differential privacy. In *FOCS*, pages 51–60, 2010. 10

[19] Cynthia Dwork, Aleksandar Nikolov, and Kunal Talwar. Efficient algorithms for privately releasing marginals via convex relaxations. *CoRR*, abs/1308.1385, 2013. 38

[20] 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. 27

[21] Vitaly Feldman and David Xiao. Sample complexity bounds on differentially private learning via communication complexity. In *Proceedings of COLT 2014*, pages 1000–1019, 2014. 43, 44

[22] Sanjam Garg, Craig Gentry, Shai Halevi, Mariana Raykova, Amit Sahai, and Brent Waters. Candidate indistinguishability obfuscation and functional encryption for all circuits. In *2013 IEEE 54th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 40–49. IEEE, 2013. 30

[23] M. Hardt and G. 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. 18

[24] 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. URL http://papers.nips.cc/paper/4548-a-simple-and-practical-algorithm-for-differentially-private-data-release.pdf. 19

[25] Moritz Hardt, Guy N. Rothblum, and Rocco A. Servedio. Private data release via learning thresholds. In *SODA*, pages 168–187, 2012. 38

[26] 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. 39, 41, 43

[27] 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. 13

[28] Ilan Kremer, Noam Nisan, and Dana Ron. On randomized one-round communication complexity. In *Proceedings of STOC 1995*, pages 596–605, 1995. 43

[29] 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. 17

[30] P. B. Miltersen, N. Nisan, S. Safra, and A. Wigderson. On data structures and asymmetric communication complexity. *J. Comput. Syst. Sci.*, 57(1):37–49, 1998. 43

[31] 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. 23

[32] Arvind Narayanan and Vitaly Shmatikov. Robust de-anonymization of large sparse datasets. In *2008 IEEE Symposium on Security and Privacy (S&P 2008), 18-21 May 2008, Oakland, California, USA*, pages 111–125. IEEE Computer Society, 2008. ISBN 978-0-7695-3168-7. doi:10.1109/SP.2008.33. URL `http://ieeexplore.ieee.org/xpl/mostRecentIssue.jsp?punumber=4531131`. 5

[33] Aleksandar Nikolov, Kunal Talwar, and Li Zhang. The geometry of differential privacy: the sparse and approximate cases. In *STOC*, pages 351–360, 2013. doi:10.1145/2488608.2488652. 23, 38

[34] Kobbi Nissim, Sofya Raskhodnikova, and Adam Smith. Smooth sensitivity and sampling in private data analysis. In *STOC*, pages 75–84, 2007. 13

[35] 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. 40

[36] 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. 27

[37] 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. 38

[38] Jonathan Ullman. Answering $n^{2+o(1)}$ counting queries with differential privacy is hard. In *Proceedings of the 45th annual ACM symposium on Symposium on theory of computing*, pages 361–370. ACM, 2013. 28, 29, 30

[39] Jonathan Ullman and Salil Vadhan. PCPs and the hardness of generating private synthetic data. In *Theory of Cryptography*, pages 400–416. Springer, 2011. 31

[40] Stanley L. Warner. Randomized response: A survey technique for eliminating evasive answer bias. *Journal of the American Statistical Association*, 60(309):63–69, 1965. 6