Chapter 2 — What is Aggregate?

Draft version: September 1, 2021

Contents

Acknowledgement. This chapter is mostly written by Jonathan Ullman. Please feel free to provide feedback.

We've talked about the fact that a lot of the public discussion around data privacy centers on concepts like *identifiability*, specifically whether or not an attacker can identify information belonging to a specific individual. So we might come into this subject with the intuition that *aggregate statistics*—those which combine the information of many individuals—are automatically private because there is no direct way to identify any individual in the output. The goal of this chapter is to disabuse the reader of this notion, and argue that releasing seemingly harmless statistics can reveal a lot about individuals. We'll begin with simple examples of *difference attacks* showing that even questions that seem to aggregate information, when constructed carefully, can actually single out individuals precisely. However, the bulk of this chapter is devoted to *reconstruction attacks*, which allow an attacker to recover all or part of a dataset using noisy statistics. Not only will we see that reconstruction attacks are possible, we will see that they are, in fact, inevitable if we don't impose limits on the kind and amount of information we release about the dataset.

1 Difference Attacks

Let's start with a very simple example. You've just started a new job and signed up for the health insurance provided by your employer. Your employer has access to aggregate statistics from the insurance provider, and can, in particular ask questions like *How many employees joined the company on December 10th, 2020 and have lupus?* Suppose the answer to this query is 1. Since the employer knows your exact start date, and perhaps even knows that you are the only person with this start date, your employer just learned that you suffer from lupus.

Maybe this example was a little too obvious. If the answer is 1, are we really aggregating anything? What if we just suppress these small answers and return something like "less than 5" instead of the exact answer? It turns out that small answers are not the only issue. We can come up with two queries whose answers are both large, but whose *difference* reveals information about a specific person. Consider the following pair of queries: (1) *How many employees joined the company before December 10th, 2020 and suffer from lupus?* (2) *How many employees joined the company before December 11th, 2020 and suffer from lupus?* Suppose the answers to these queries are 417 and 418, respectively. Then, your employer has just learned that you suffer from lupus. Notice that we don't need to allow a user to construct carefully related queries for this type of attack to occur. These are exactly the queries that would arise from a system that reports a running count of the number of employees with lupus over time, and these sorts of frequently updated statistics often directly result in difference attacks.

Attacks like these that exploit the difference between two aggregate queries to single out a specific person are called, unsurprisingly, *difference attacks*.

> *Difference attacks might seem trivial and easily avoided, but they are a useful test case for any proposed method for protecting privacy.*

2 Reconstruction Attacks

We will turn our attention to a much more general class of attacks called *reconstruction attacks*, where seemingly benign information can be used to recover a large amount of information about the dataset.

2.1 Example: Reconstruction Attacks and the Census

We'll introduce reconstruction attacks by way of an example inspired by the U.S. Census Bureau's disclosure control methods for the 2010 Decennial Census [GAM19]. The Census collects the age, sex, race, location, and other demographic information of those living in the United States.

Microdata				
ID	Block	Age	Sex	Race
001	1402	26	M	Black
002	1402	34	F	White
003	1402	46	M	White
004	1402	30	M	Black
005	1402	9	F	White
006	1402	5	F	White
007	1403	58	F	Asian
008	1403	18	F	Latina

Tabulations for Block 1402				
			Age	
Statistic	Group	Count	Median	Mean
1	Total	6	26	25
2A	Female	3	9	16
2B	Male	3	30	34
3A	Black Male	$*$	$*$	$*$
3B	Black Female	$*$	$*$	$*$
3C	White Female	3	9	16
3D	White Male	$*$	$*$	$*$

Figure 1: An oversimplified example of Census microdata (left) and summary statistics (right).

This data is made available for various purposes, but federal law prohibits releasing the individual responses—called *microdata*—so the Census instead releases tabulations of various summary statistics, such as those in Figure 1.[1] As we discussed, releasing small counts is problematic, so certain cells in the table are *suppressed*, which we indicate with the $*$ symbol.

What can we learn about the microdata from the tabulations given in Figure 1? Let's look at a concrete example and focus on just Statistic 2B in the table, which tells us about the set of individuals in the dataset who reported their sex as male. The first column tells us that there are three such people in the dataset, so let's denote their ages as $A \leq B \leq C$. We have the background information that $0 \leq A, B, C \leq 122$, where the upper bound is the oldest reported age of any human.[2] Using only our background information, there are $\binom{125}{3} = 317{,}750$ possible choices for A, B, and C. However, we also know that the median age of these individuals is $B = 30$, which leaves us with $31 \times 93 = 2{,}883$ possible choices—more than 100-fold fewer choices! Finally, we know that the mean age of these individuals is $\frac{1}{3}(A + B + C) = 34$. Since we know $B = 30$ and $A + B + C = 108$, we also have $C = 78 - A$, so C is actually determined once we pick a value of A. That leaves us with only 31 possible tuples (A, B, C) that are consistent with the given statistics—over 10,000-fold fewer choices than we started with, from just two statistics! It's not hard to see how adding more constaints will quickly reveal even more information about the underlying microdata.

Exercise 2.1. Suppose we unsuppress Statistic 3A, and it shows that the dataset contains two Black males, whose mean age is 28. Is this enough information to uniquely determine the values A, B, and C, or are there multiple choices of A, B, C that are consistent with the tabulated statistics?

[1] Note that the example statistics in Figure 1 are intentionally oversimplified for pedagogical purposes. In particular, the census asks two separate questions for race and ethnicity and the set of possible responses is much richerd.

[2] Although it's the subject of some controvery, the French woman Jeanne Calment reportedly lived to 122 [Wik21].

2.2 The Reconstruction Paradigm

The previous reconstruction attack relied on ad hoc reasoning to perform the reconstruction. However, the approach we took is actually very general, and we can introduce a model to study this paradigm. We start with some *dataset* $\mathbf{x}$ that represents the underlying microdata. Each *statistic* (or *query*) on the dataset is some function $f(\mathbf{x})$, and for each statistic we obtain some answer $f(\mathbf{x}) = a$. For example, the first column of Statistic 2B tells us that $f(\mathbf{x}) = 3$ where f counts the number of individuals who entered male as their sex. Each statistic gives us a set of *constraints* on the dataset, such as the fact that $\mathbf{x}$ must be a dataset with three males.

Tables like the one in Figure 1 give us a large set of constraints like

$$f_1(\mathbf{x}) = a_1 \quad f_2(\mathbf{x}) = a_2 \quad \cdots \quad f_k(\mathbf{x}) = a_k$$

Given all these constraints, the *dataset reconstruction problem* is to find a dataset $\hat{\mathbf{x}}$ that is *consistent* with these constraints.

> *Aggregate statistics place constraints on the dataset that can be used to reconstruct all or part of the underlying microdata.*

If we are given enough constraints, then there may only be a single consistent dataset, in which case we have reconstructed the microdata! More generally, the statistics might severely limit what datasets are consistent, revealing a lot of information about the microdata. In our previous example we were not able to reconstruct the entire dataset $\mathbf{x}$ exactly, but we were able to determine that any consistent dataset must have three males, one of whom was exactly 30, one younger than 30, and one of whom was no older than 78.

In general, solving dataset reconstruction is an instance of what are called *constraint-satisfaction problems.* Although most interesting constraint-satisfaction problems are NP-hard in the worst case, we can often use powerful tools like SAT-solvers to do dataset reconstruction in practice, so NP-hardness offers little-to-no practical defense against reconstruction. Moreover, in the next section we'll see examples where the reconstruction problem can be solved in polynomial time.

2.3 Reconstruction from Noisy Statistics

Our example of the Census relied only on *exact* statistics, such as the fact that there were exactly three males in the dataset. Reconstructing using exact statistics is already enough to cause concern, and to suggest that some explicit steps must be taken to limit access to sensitive data, such attacks can be rather brittle and easily avoided. In realistic settings we are not only given a limited set of statistics to work with, but are also given statistical information that is *noisy.* For example, the tabulations in Figure 1 already anticipate certain simple attacks by suppressing certain cells with small counts, which can be thought of as a form of noise if we think of each suppressed cell as representing some default answer. Noisy statistics can also arise by explicitly perturbing the data or the statistics with random noise. Noisy statistical information can be viewed as giving us soft constraints on the dataset of the form

$$f_1(\mathbf{x}) \approx a_1 \quad f_2(\mathbf{x}) \approx a_2 \quad \cdots \quad f_k(\mathbf{x}) \approx a_k$$

and reconstruction attacks aim to find a dataset $\mathbf{x}$ that is consistent with these soft constraints.

3 Provable Reconstruction Attacks: Linear Reconstruction

In this section we'll explore settings where we can rigorously analyze reconstruction attacks, which will lead to insights about what types of statistical information will leave the microdata vulnerable to reconstruction, how robust these reconstruction attacks are to sources of noise in the statistics, and how to carry out these attacks in a computationally efficient way. This theory was introduced in a seminal paper by Dinur and Nissim [DN03] that was ultimately the beginning of the development of differential privacy, however we will see the theory presented in a slightly different way that emphasizes how these attacks might arise in practice.

3.1 Linear Queries and Linear Reconstruction Attacks

Let's start by introducing a model for reconstruction attacks that will allow us to study the problem formally. Suppose we start with a dataset containing identifying attributes like first name, postal code, age, and sex, as well as a sensitive attribute that, for simplicity, we represent as a single bit (Figure 2 left). We will design attacks in which an attacker who already knows the identifying attributes, and receives approximate answers to certain statistics on the dataset will be able to approximately *reconstruct* the column of sensitive bits.

Microdata				
ID	Block	Age	Sex	Race
001	1402	26	M	Black
002	1402	34	F	White
003	1402	46	M	White
004	1402	30	M	Black
005	1402	9	F	White
006	1402	5	F	White

Stylized Microdata	
Identifiers (ID, Block, Age, Sex)	Secret (White / Black)
z_1	s_1
z_2	s_2
z_3	s_3
z_4	s_4
z_5	s_5
z_6	s_6

Figure 2: Going from real microdata data (left) to a stylized dataset (right) in our model

To start, we will simplify the dataset by writing each individual's data as a pair (z_j, s_j) where $z_j \in \mathcal{Z}$ contains user j's identifying infomation and $s_j \in \{0, 1\}$ is some binary piece of secret information belonging to user j, depicted in Figure 2. For our toy dataset

$$\mathcal{Z} = \{\text{IDs}\} \times \{\text{blocks}\} \times \{\text{ages}\} \times \{\text{sexes}\}$$

and, for example, $z_1 = (001, 1402, 26, \text{M})$ and $s_1 = 1$ indicating that this individual is black. Given a data $\mathbf{x} = ((z_1, s_1), \ldots, (z_n, s_n))$ we will use $\mathbf{z} = (z_1, \ldots, z_n)$ to denote the identifying part of the dataset and $\mathbf{s} = (s_1, \ldots, s_n)$ to denote the vector of secrets.

Note that the distinction between the identifiers and the secret bits is arbitrary, and is merely convenient notation for representing the goal of the reconstruction attacker. If the attacker were interested in reconstructing, say, the sex of each individual, we could treat sex as the secret bit and the remaining attributes would be folded into the identifiers.

Next, we will define a natural type of *count statistics* that captures many summary statistics that one would release about a dataset like this one. Intuitively, count statistics ask for the number of individuals in the dataset that satisfy some specific property. For example, how many individuals are older than 28 and have the secret bit 1? Since we're interested in reconstructing the secret bits s_j, we will only consider statistics of the form

$$\begin{aligned} f(\mathbf{x}) &= \sum_{j=1}^{n} \varphi(z_j)s_j \text{ for } \varphi : \mathcal{Z} \to \{0,1\} \\ &= \#\{j : \varphi(z_j) = 1 \text{ and } s_j = 1\} \end{aligned} \tag{1}$$

Here, φ in the example above would be $\varphi(z_j) = 1$ if and only if user j is older than 28. Note that if the microdata contains n individuals, then these statistics return an integer between 0 and n.

The nice thing about these statistics—and the reason they are often called *linear statistics*—is because the can be expressed nicely in the language of linear algebra. For a given statistic f and dataset $\mathbf{x}$ with identifiers $\mathbf{z}$ and secret vector $\mathbf{s}$, the value of the statistic has the form

$$f(\mathbf{x}) = \mathbf{f}^{\mathbf{z}} \cdot \mathbf{s} \text{ where } \mathbf{f}^{\mathbf{z}} = (\varphi(z_1), \ldots, \varphi(z_n)) \tag{2}$$

where $u \cdot v$ is the *dot product* between the two vectors. We used the notation $\mathbf{f}^{\mathbf{z}}$ to indicate that the vector depends on both the specification of the query and on the identifying information in the dataset, however, from now on we will simply write $\mathbf{f}$ without the superscript because the identifiers will be fixed and unchanging throughout our analysis.

Given a set of queries $f_1, \ldots, f_k$ we can write the evaluation of all the queries as a matrix-vector product $\mathbf{F} \cdot \mathbf{s}$

$$\begin{bmatrix} f_1(\mathbf{x}) \\ \vdots \\ f_k(\mathbf{x}) \end{bmatrix} = \begin{bmatrix} - & \mathbf{f}_1 & - \\ & \vdots & \\ - & \mathbf{f}_k & - \end{bmatrix} \begin{bmatrix} s_1 \\ \vdots \\ s_n \end{bmatrix} \tag{3}$$

An important thing to note is that, in this notation, the answer to f_i is the dot product of the i-th row of the matrix, denoted F_i, with the secret vector s. In other words

$$f_i(\mathbf{x}) = (\mathbf{F} \cdot s)_i = \mathbf{f}_i \cdot s \tag{4}$$

Internalizing this notation will help with what comes next. Linear algebra is tricky, and it might seem daunting to go from something relatively intuitive like a count statistic to matrices and vectors. However, this linear algebraic representation is going to be crucial both for analyzing how reconstruction is possible and for making the algorithm computationally efficient.

Exercise 3.1. Consider the set of queries f_1, f_2, f_3 specified by:

- $\varphi_1(z_j) = 1$ if and only if user j is older than 40
- $\varphi_2(z_j) = 1$ if and only if user j is older than 40 and male
- $\varphi_3(z_j) = 1$ if and only if user j is older than 20 and male

For the dataset in Figure 2, write the matrix $\mathbf{F}$, the secret vector s, and the product $\mathbf{F} \cdot \mathbf{s}$.

The reconstruction problem we're going to try to understand is how to take a vector of approximate answers $a \approx \mathbf{F} \cdot \mathbf{s}$ and recover a vector $\tilde{\mathbf{s}} \approx \mathbf{s}$. We'll mostly focus on when the constraints give enough information to reconstruct something close to $\mathbf{s}$, and only touch upon how to actually find $\tilde{\mathbf{s}}$ in a computationally efficient way.

In this chapter, we will only see one actual reconstruction attack, although we'll prove different things about it depending on which queries we're given. All the attack does is try to find a vector of secrets $\tilde{\mathbf{s}} \in \{0,1\}^n$ that is *consistent* with the information we're given, in the sense that we would have obtained similar answers if the true secrets were $\tilde{\mathbf{s}}$. See Algorithm 1 for a description of the attack.

Algorithm 1 The Reconstruction Attack

```
Reconstruct($f_1, \dots, f_k; a_1, \dots, a_k; \mathbf{z}$)
    input: queries $f_1, \dots, f_k$, answers $a_1, \dots, a_k \in \mathbb{R}$, and identifiers $\mathbf{z} \in \mathcal{Z}^n$
    output: a vector of secrets $\tilde{\mathbf{s}} \in \{0,1\}^n$
    form the vectors $\mathbf{f}_1, \dots, \mathbf{f}_k$ using the queries and identifiers
    let $\tilde{\mathbf{s}} \in \{0,1\}^n$ be the vector that minimizes the quantity $\max_{i \in [k]} |\mathbf{f}_i \cdot \tilde{\mathbf{s}} - a_i|$
    return $\tilde{\mathbf{s}}$
```

To be explicit, we've defined the reconstruction attack in terms of the queries $f_1, \dots, f_k$ and the identifiers, but ultimately what the attack needs is only the vector representation $\mathbf{f}_1, \dots, \mathbf{f}_k$, so as a shorthand we will sometimes say that the attack is "given" the vector representation of the queries.

The next claim captures a simple, but important statement about what happens in this reconstruction attack when the answers are all accurate to within some error bound.

Claim 3.2. *If every query is answered to within error $\leq \alpha n$, i.e.*

$$\max_{i \in [k]} |\mathbf{f}_i \cdot \mathbf{s} - a_i| \leq \alpha n,$$

then the reconstruction attack returns $\tilde{\mathbf{s}}$ such that $\max_{i \in [k]} |\mathbf{f}_i \cdot \tilde{\mathbf{s}} - a_i| \leq \alpha n$.

To see why this claim is true, observe that the true vector of secrets $\mathbf{s}$ satisfies $\max_{i \in [k]} |\mathbf{f}_i \cdot \mathbf{s} - a_i| \leq \alpha n$. Thus, the vector $\tilde{\mathbf{s}}$ that *minimizes* this quantity must make it no greater than αn. We may actually find a vector $\tilde{\mathbf{s}}$ that minimizes the quantity even further, but for our analysis we will only use the fact that there is some $\tilde{\mathbf{s}}$ that has error $\leq \alpha n$ for these queries.

The main idea for how we're going to analyze this attack is to show that for every $\tilde{\mathbf{s}}$ that disagrees with the real $\mathbf{s}$ in many coordinates, there is some query vector $\mathbf{f}_i$ that prevents $\tilde{\mathbf{s}}$ from being the minimum in the sense that $|\mathbf{f}_i \cdot \tilde{\mathbf{s}} - \mathbf{f}_i \cdot \mathbf{s}|$ is too large, and therefore $|\mathbf{f}_i \cdot \tilde{\mathbf{s}} - a_i|$ is also large.

3.2 Privacy is an Exhaustible Resource: Reconstruction from Many Queries

In case you're wondering, so far we have not shown that there is *any* set of statistics that can be used to reconstruct the secret vector $\mathbf{s}$, even when we are given *exact* answers to each of these statistics. But such sets do exist! We leave it as an exercise to write down an explicit example.

Exercise 3.3. Suppose the identifiers $z_1, \dots, z_n$ are unique and known to you. Construct a set of statistics $f_1, \dots, f_n$ such that, for every secret vector $\mathbf{s} \in \{0,1\}^n$, the reconstruction attack will recover $\tilde{\mathbf{s}} = \mathbf{s}$ provided it is given exact answers to each of these queries so that $a_i = f_i(\mathbf{x})$.

But can reconstruction succeed when the answers are noisy? In this section we'll prove a simple, but absolutely crucial result, showing that if we are given answers to *all possible statistics*, then reconstruction is possible *even if the noise in the queries is so large as to render the answers almost useless.*

Let's start by making it precise what we mean by all possible statistics. For a given set of identifiers $\mathbf{z} \in \mathcal{Z}^n$ and a predicate $\varphi : \mathcal{Z} \to \{0,1\}$, the vector representation is some vector $\mathbf{f}^{\mathbf{z}} \in \{0,1\}^n$. Thus, while there are $|\mathcal{Z}|^n$ distinct predicates φ, there are at most $k = 2^n$ distinct possibilities for the resulting vector $\mathbf{f}^{\mathbf{z}} \in \{0,1\}^n$. We will say that a set of queries $f_1, \dots, f_k$ *contains all statistics* if each vector $\mathbf{f} \in \{0,1\}^n$ appears as the vector representation of some f_i. For purposes of our analysis we will give a concrete construction of these queries. First, we will assume public identifiers $z_1, \dots, z_n$ are *unique*, and it may help to fix them to be $z_j = j$. In our example table, this corresponds to using the User ID attribute for the public identifier, but even if such an attribute doesn't exist, notice that combinations of attributes like (Block, Age, Sex) are also unique in our table. For every vector $v \in \{0,1\}^n$, we can define a query

$$\varphi_v(z) = \begin{cases} 1 & \text{if } z = z_j \text{ and } v_j = 1 \\ 0 & \text{otherwise} \end{cases} \tag{5}$$

For notational simplicity, we will index the queries by the vector v,[3] and write $(f_v)_{v \in \{0,1\}^n}$.

Exercise 3.4. For $n = 3$, and $z_j = j$ for $j \in \{1,2,3\}$, write down the matrix $\mathbf{F} \in \{0,1\}^{k \times n}$

We will prove the following theorem about reconstruction attacks in the case where we are given noisy answers to all possible statistics.

Theorem 3.5 ([DN03]). *Let* $\mathbf{x}$ *be any dataset with distinct identifiers* $\mathbf{z}$ *and secret vector* $\mathbf{s}$. *Then if* $f_1, \dots, f_k$ *contain all statistics (as described above), and* $a_1, \dots, a_k$ *are answers satisfying*

$$|a_i - f_i(\mathbf{x})| \leq \alpha n \text{ for every } i = 1, \dots, k,$$

then Reconstruct$(f_1, \dots, f_k; a_1, \dots, a_k; \mathbf{z})$ *(Algorithm 1) returns* $\tilde{\mathbf{s}}$ *that disgrees with* $\mathbf{s}$ *on at most* $4\alpha n$ *coordinates.*

An important note: we're placing *no assumptions* on the secret vector $\mathbf{s}$, so without the query answers the best we could hope to do is guess at random, which would be right for only $\frac{n}{2}$ of the users on average.

Before we prove the theorem, let's take a moment to appreciate what it says. Suppose you are tasked with coming up with a private method for answering count statistics. While you are willing to reveal some information about the secrets, you cannot allow an attacker to reconstruct the secret vector with better than, say 60% accuracy (errors on at most $0.4n$ coordinates). Then no matter how you construct the answers, there will have to be some query that has error at least $\pm 0.1n$. Since the answer ranges from 0 to n, error $0.1n$ is enough error to render the answer unusable for most applications. Thus, this theorem teaches us an important principle to keep in mind when evaluating claims about privacy.

> *Any method that satisfies any meaningful privacy and accuracy guarantee cannot answer an arbitrary number of statistics to high accuracy.*

Now we can return to the theorem's proof.

[3] This notation where we index an array with a vector, rather than a number, might look a bit unusual. Formally, you can think of v as an integer written in binary.

Proof. Our goal is to show that, under our assumptions, the output $\tilde{\mathbf{s}}$ of Algorithm 1 must agree with the true private bits $\mathbf{s}$ on all but $4\alpha n$ users. The reason is that if $\tilde{\mathbf{s}}$ disagreed with more than $4\alpha n$ of the secret bits, then the answer to some query would have *eliminated* $\tilde{\mathbf{s}}$ from contention. To see this, fix some $\tilde{\mathbf{s}} \in \{0,1\}^n$, and let

$$S_{01} = \{j : \tilde{s}_j = 0, s_j = 1\} \text{ and } S_{10} = \{j : \tilde{s}_j = 1, s_j = 0\} \tag{6}$$

If $\tilde{s}$ and s disagree on more than $4\alpha n$ bits, then at least one of these two sets has size larger than $2\alpha n$. Let us assume that this set is S_{01}, and we'll deal with the other case by symmetry. Let v be the *indicator vector* for the set S_{01}, which is defined by $v_j = 1 \iff j \in S_{01}$. This vector defines a statistic f_v that counts how many users have $j \in S_{01}$ and secret bit $s_j = 1$. Then we have

$$|\mathbf{F}_v \cdot \mathbf{s} - \mathbf{F}_v \cdot \tilde{\mathbf{s}}| = |S_{01}| > 2\alpha n. \tag{7}$$

At the same time, if $\tilde{\mathbf{s}}$ were output by the attacker, then $|a_v - \mathbf{F}_v \cdot \tilde{\mathbf{s}}| \leq \alpha n$. We would then have

$$|\mathbf{F}_v \cdot \mathbf{s} - \mathbf{F}_v \cdot \tilde{\mathbf{s}}| \leq |a_v - \mathbf{F}_v \cdot \tilde{\mathbf{s}}| + |\mathbf{F}_v \cdot \mathbf{s} - a_v| \leq 2\alpha n, \tag{8}$$

which is a contradiction. Similarly, S_{10} must also have size at most $2\alpha n$. Thus, any vector $\tilde{\mathbf{s}}$ output by the attack disagrees with $\mathbf{s}$ on at most $4\alpha n$ coordinates. □

An important point about the proof is that the attacker does not need to know the set S_{10}, or the corresponding statistic f_v. Since the attacker asks all possible queries, we can be sure f_v is one of these statistics, and an accurate answer to it rules out this particular bad choice of $\tilde{\mathbf{s}}$ as the output of the reconstruction attack.

3.3 Reconstruction Using Fewer Queries

The reconstruction attack we just discussed is quite powerful, but it is arguably unrealistic because it requires answers to an enormous set of 2^n statistics. What if we simply bound the number of statistics we release to something much less than $\ll 2^n$? Does reconstruction from noisy answers suddenly become impossible? The answer turns out to be no! What we will now show is that $O(n)$ queries are enough to reconstruct the dataset to high accuracy, provided that the answers have error $\ll \sqrt{n}$. In other words, our attack now uses just a small number of queries, but requires answers that are much more accurate. There are good reasons why this $\sqrt{n}$ bound appears, and why it's interesting, which we'll return to in Section 3.5 after we describe and analyze the attack.

Describing the queries. The attack is the same algorithm we used before, Algorithm 1, but instead of asking all 2^n queries, we will ask just a small subset of $O(n)$ queries. In particular, we will use k *independently and randomly chosen* functions $\varphi_i : \mathcal{Z} \to \{0,1\}$. Observe crucially that if φ_i is a uniformly random function mapping $\mathcal{Z}$ to $\{0,1\}$, and we continue to assume that the identifiers $z_1, \ldots, z_n$ are unique, then each corresponding vector $\mathbf{f}_i$ is a uniformly random element of $\{0,1\}^n$. Moreover, if the functions $\varphi_1, \ldots, \varphi_k$ are independent then so are the corresponding vectors $\mathbf{f}_1, \ldots, \mathbf{f}_k$.

Now we can prove the following theorem about the reconstruction attack when the queries are chosen randomly as we described.

Theorem 3.6 ([DN03]). *Let* $\mathbf{x}$ *be any dataset with distinct identifiers* $\mathbf{z}$ *and secret vector* $\mathbf{s}$*. Then if* $f_1, \ldots, f_k$ *are* $k = 20n$ *independent, uniformly random statistics (as described above) then with high probability (over the choice of the queries), if* $a_1, \ldots, a_k$ *are answers satisfying*

$$|a_i - f_i(\mathbf{x})| \leq \alpha n \text{ for every } i = 1, \ldots, k,$$

then Reconstruct$(f_1, \ldots, f_k; a_1, \ldots, a_k; \mathbf{z})$ *(Algorithm 1) returns* $\tilde{\mathbf{s}}$ *that disagrees with* $\mathbf{s}$ *on at most* $256\alpha^2 n^2$ *coordinates.*[4]

Observe that when $\alpha \ll 1/\sqrt{n}$, the reconstruction error is $\ll n$, meaning that we recover nearly all of the secret bits. As we discuss in Section 3.5, introducing error that is proportional to $\sqrt{n}$ is significant, because this is the scale of the error that would arise naturally if the data were randomly sampled from some population. Thus, this theorem teaches us another important lesson.

> *Any method that offers a meaningful privacy guarantee and has "insignificant" error is severely limited in how many queries it can answer.*

The proof that this attack has low reconstruction error is much trickier, but ultimately uses the same idea we used for the exponential reconstruction attack—if $\mathbf{s}$ and $\tilde{\mathbf{s}}$ disagree on many bits, then there will be some query that proves $\tilde{\mathbf{s}}$ cannot be close to $\mathbf{s}$, and thus cannot be the output of the reconstruction attack.

3.3.1 ** Proving Theorem 3.6

Before giving the proof, we'll need the following technical fact.

Claim 3.7. *Let* $\mathbf{t} \in \{-1, 0, +1\}^n$ *be a vector with at least* m *non-zero entries and let* $\mathbf{u} \in \{0, 1\}^n$ *be a uniformly random vector. Then*

$$\mathbb{P}\left(|\mathbf{u} \cdot \mathbf{t}| > \sqrt{m}/4\right) \leq \frac{1}{10} \tag{9}$$

Proof sketch. Intuitively, what we want to show is that $u \cdot t$ behaves somewhat like a Gaussian random variable with standard deviation at least $\sqrt{m}/2$. If it were truly Gaussian with this standard deviation, then the probability that it is contained in an any interval of width $\sqrt{m}/2$ would be at most $\frac{7}{10}$. The reason the right-hand side above is $\frac{9}{10}$ is because the Gaussian approximation isn't exactly correct, and we need to account for the difference, which can be done in several ways. We do not include a complete proof. □

Now let's return to the proof of Theorem 3.6

Proof of Theorem 3.6. Our goal will be to show that any vector $\tilde{\mathbf{s}} \in \{0, 1\}^n$ that disagrees with $\mathbf{s}$ on more than $256\alpha^2 n^2$ bits cannot satisfy

$$\forall i \in [k] \;\; |\mathbf{f}_i \cdot \tilde{\mathbf{s}} - a_i| \leq \alpha n. \tag{10}$$

and thus cannot be the output of the reconstruction attack. To this end, fix any true secret vector $\mathbf{s} \in \{0, 1\}^n$ and let

$$\mathcal{B} = \left\{\tilde{\mathbf{s}} : \tilde{\mathbf{s}} \text{ and } \mathbf{s} \text{ disagree on at least } 256\alpha^2 n^2 \text{ coordinates}\right\} \tag{11}$$

[4] The constants 20 and 256 in this theorem are somewhat arbitrary and can definitely be improved substantially with a more careful analysis.

Our goal is to show that the reconstruction attack does not output any vector in $\mathcal{B}$. Here $\mathcal{B}$ is a mnemonic for the set of "bad" outputs that we want to show are not the ones returned by the reconstruction attack. To this end, we will say that statistic i *eliminates* vector $\tilde{\mathbf{s}}$ if

$$|\mathbf{f}_i \cdot (\mathbf{s} - \tilde{\mathbf{s}})| \geq 4\alpha n. \tag{12}$$

If $\tilde{\mathbf{s}}$ is eliminated by some statistic i then $\tilde{\mathbf{s}}$ cannot be the output of the reconstruction attack because

$$|\mathbf{f}_i \cdot \tilde{\mathbf{s}} - a_i| \geq |\mathbf{f}_i \cdot (\mathbf{s} - \tilde{\mathbf{s}}) - a_i| - |\mathbf{f}_i \cdot \mathbf{s} - a_i| \geq 4\alpha n - \alpha n = 3\alpha n. \tag{13}$$

Thus, our goal is to show that every vector in $\mathcal{B}$ is eliminated by some query. In other words

$$\forall \tilde{\mathbf{s}} \in \mathcal{B} \;\; \exists i \in [k] \;\;\; |F_i \cdot (\mathbf{s} - \tilde{\mathbf{s}})| \geq 4\alpha n \tag{14}$$

To do so, let's fix *some* particular vector $\tilde{\mathbf{s}} \in \mathcal{B}$ and show that it is eliminated with extremely high probability. Specifically, suppose $\tilde{\mathbf{s}} \in \{0,1\}^n$ differs from $\mathbf{s}$ on at least $m = 256\alpha^2 n^2$ coordinates. We will argue

$$\exists i \in [k] \;\;\; |\mathbf{f}_i \cdot (\mathbf{s} - \tilde{\mathbf{s}})| \geq 4\alpha n \tag{15}$$

We will show how (15) can be deduced from Claim 3.7. Fix vectors $\mathbf{s}$ and $\tilde{\mathbf{s}}$ that differ on at least m coordinates, and define $\mathbf{t} = \mathbf{s} - \tilde{\mathbf{s}}$. Then $\mathbf{t} \in \{-1, 0, +1\}^n$ and $\mathbf{t}$ has at least m non-zero entries. Moreover, since the queries are chosen uniformly at random, $\mathbf{u} = \mathbf{f}_i$ is a uniformly random vector in $\{0,1\}^n$. Thus $\mathbf{u}$ and $\mathbf{t}$ satisfy the assumptions of Claim 3.7, so we conclude that

$$\mathbb{P}\left(|\mathbf{f}_i \cdot (\mathbf{s} - \tilde{\mathbf{s}})| \leq 4\alpha n\right) \leq \frac{9}{10} \tag{16}$$

Thus, each query f_i has a reasonable chance of eliminating $\tilde{\mathbf{s}}$. Now, since the $k = 20n$ queries are independent, we have that

$$\mathbb{P}\left(\forall i \in [k] \;:\; |\mathbf{f}_i \cdot (\mathbf{s} - \tilde{\mathbf{s}})| \leq 4\alpha n\right) \leq \left(\frac{9}{10}\right)^{20n} \leq 2^{-2n}. \tag{17}$$

The last step is to argue that *every* $\tilde{\mathbf{s}} \in \mathcal{B}$ will be eliminated by some statistic i. Since there are only 2^n possible choices for $\tilde{\mathbf{s}}$, we know that $|\mathcal{B}| \leq 2^n$ Therefore, we have

$$\mathbb{P}\left(\exists \tilde{\mathbf{s}} \in \mathcal{B}\,,\; \forall i \in [k] \;:\; |\mathbf{f}_i \cdot (\mathbf{s} - \tilde{\mathbf{s}})| \leq 4\alpha n\right) \leq 2^n \cdot 2^{-2n} = 2^{-n}. \tag{18}$$

We now know that (except with probability $\leq 2^{-n}$), every vector $\tilde{\mathbf{s}}$ that disagrees with $\mathbf{s}$ on more than m coordinates will be eliminated from contention, so the attacker must return a vector $\tilde{\mathbf{s}}$ that disagrees on at most m coordinates. Note that the only way reconstruction can fail is if we get unlucky with the choice of queries, which happens with probability at most 2^{-n}. □

3.3.2 ∗∗ Do the queries have to be random?

Although we modeled the queries, and thus the matrix $\mathbf{F}$, as uniformly random, it's important to note that we really only relied on the fact that for every pair of vectors $\mathbf{s}, \tilde{\mathbf{s}}$,

$$\max_{i \in [k]} |\mathbf{F}_i \cdot (\mathbf{s} - \tilde{\mathbf{s}})| \gtrsim \sqrt{\mathrm{err}(\mathbf{s}, \tilde{\mathbf{s}})}, \tag{19}$$

where we define $\mathrm{err}(\mathbf{s}, \tilde{\mathbf{s}})$ is the number of coordinates on which they disagree. We can perform noisy reconstruction with error $\approx \sqrt{n}$ for any family of queries that gives rise to a matrix with this property. Moreover, quantitatively weaker versions of this property lead to reconstruction attacks as well, albeit with less tolerance to noise. Not every matrix satisfies a property like this one, and later on we will see examples of special types of queries that are much easier to make private than random queries. However, any family of *random enough* queries will satisfy such a property. More specifically, this property, or similar properties, are satisfied by any matrix with no small singular values [DY08] or high discrepancy [MN12, NTZ13], and these conditions are known to hold for some natural families such as *marginal statistics* and *contingency tables* [KRSU10], and various geometric families [MN12].

3.4 Computationally Efficient Reconstruction: Linear Programming

Algorithm 1 is not computationally efficient, even if the number of queries k is small, since the attacker might have to enumerate all 2^n vectors $\tilde{\mathbf{s}} \in \{0,1\}^n$ to find one that minimizes

$$\max_{i \in [k]} |\mathbf{f}_i \cdot \tilde{\mathbf{s}} - a_i| \tag{20}$$

However, we can modify the attack slightly to run in time polynomial in n using *linear programming* (see the cutout). To do so, we have to start by finding some *real-valued* vector $\hat{\mathbf{s}} \in [0,1]^n$ that solves the following optimization problem

$$\arg\min_{\hat{s} \in [0,1]^n} \max_{i \in [k]} |\mathbf{f}_i \cdot \hat{\mathbf{s}} - a_i| \tag{21}$$

Then, to obtain the reconstruction we will round each entry to 0 or 1 to obtain a vector $\tilde{\mathbf{s}} \in \{0,1\}^n$. Pseudocode for the new attack is in Algorithm 2

Algorithm 2 The LP-Based Reconstruction Attack

```
1: LP-RECONSTRUCT($f_1, \dots, f_k; a_1, \dots, a_k; \mathbf{z}$)
2:     input: queries $f_1, \dots, f_k$, answers $a_1, \dots, a_k \in \mathbb{R}$, and identifiers $\mathbf{z} \in \mathcal{Z}^n$
3:     output: a vector of secrets $\tilde{\mathbf{s}} \in \{0,1\}^n$
4:     form the vectors $\mathbf{f}_1, \dots, \mathbf{f}_k$ using the queries and identifiers
5:     using an LP, find $\hat{\mathbf{s}} \in [0,1]^n$ that minimizes the quantity $\max_{i \in [k]} |\mathbf{f}_i \cdot \hat{\mathbf{s}} - a_i|$
6:     let $\tilde{\mathbf{s}} \in \{0,1\}^n$ be the vector obtained by rounding each value $\hat{\mathbf{s}}_i$ to $\{0,1\}$
7:     return $\tilde{\mathbf{s}}$
```

Exercise 3.8. The optimization problem in LP-Reconstruct (Algorithm 2, Line 5) does not look like LPs as they are defined in the cutout but can indeed be written as one. Show how to write an LP that solves this optimization problem.

Using a very slightly more careful analysis, one can prove that when we are given answers to $O(n)$ random queries, each with error at most αn, the solution to the linear program will satisfy

$$\sum_{j=1}^{n} |s_j - \hat{s}_j| \lesssim \alpha^2 n^2 \tag{22}$$

Linear Programming. A *linear program* with d variables and m constraints asks us to maximize a linear objective function over $\mathbb{R}^d$ subject to m linear inequality constraints. Specifically, given an objective, represented by a vector $\mathbf{c} \in \mathbb{R}^d$, and m constraints, each represented by a vector $\mathbf{a}_i \in \mathbb{R}^d$ and a scalar $b_i \in \mathbb{R}$, a linear program can be written as

$$\max_{\mathbf{x} \in \mathbb{R}^d} \ \mathbf{c} \cdot \mathbf{x}$$

$$\text{s.t. } \forall i \in [m] \ \mathbf{a}_i \cdot \mathbf{x} \leq b_i$$

Algorithms for solving linear programs are a very interesting and deep subject, but beyond the scope of this course. All you need to know is that linear programs can be solved in polynomial time and can be solved very efficiently in practice.

and therefore the rounded solution satisfies

$$\#\{j : s_j \neq \tilde{s}_j\} \lesssim \alpha^2 n^2 \tag{23}$$

The linear program will have n variables and $O(k)$ constraints, so it can be solved in polynomial time and the rounding from $\hat{\mathbf{s}}$ to $\tilde{\mathbf{s}}$ is linear in n. Thus we have succeeding in getting the reconstruction attack to run in polynomial time. We can summarize with the following theorem

Theorem 3.9 (Strengthening of [DN03]). *Let* $\mathbf{x}$ *be any dataset with distinct identifiers* $\mathbf{z}$ *and secret vector* $\mathbf{s}$*. Then if* $f_1, \ldots, f_k$ *are* $k = 20n$ *independent, uniformly random statistics (as described above) then with high probability (over the choice of the queries), if* $a_1, \ldots, a_k$ *are answers satisfying*

$$|a_i - f_i(\mathbf{x})| \leq \alpha n \textit{ for every } i = 1, \ldots, k,$$

then LP-Reconstruct$(f_1, \ldots, f_k; a_1, \ldots, a_k; \mathbf{z})$ *(Algorithm 2) runs in time polynomial in n and k and returns* $\tilde{\mathbf{s}}$ *that disagrees with* $\mathbf{s}$ *on at most* $O\alpha^2 \log(1/\alpha) n^2)$ *coordinates.*[5]

3.5 Can we Improve The Attacks?

Why does the efficient reconstruction attack work when the error is $\ll \sqrt{n}$ but not when it is $\gg \sqrt{n}$? Why is reconstruction possible with error $\frac{n}{100}$ if we have 2^n queries, but not with n^2 queries? It turns out that there is good reason why the reconstruction attacks we have seen cannot tolerate more noise. Specifically, we will see an algorithm that provably thwarts reconstruction attacks, but still allows us to answer counts with reasonable bounds on the error.

To defeat reconstruction attacks, we will consider taking a *random subsample of the dataset*. Recall the dataset is $\mathbf{x} = (x_1, \ldots, x_n)$. Now fix $m = \frac{n}{5}$ and we will define the *subsampled dataset* $Y = (y_1, \ldots, y_m)$ as follows. For each $j \in [m]$, independently choose a random element $r \in [n]$ and set $y_j = x_r$. Note that the sampling is *independent* and *with replacement*. Suppose we now use Y to

[5] The constants 20 and 256 in this theorem are somewhat arbitrary and can definitely be improved substantially with a more careful analysis.

compute the statistics in place of $\mathbf{x}$. That is, we return the answer

$$5f(Y) = \sum_{j=1}^{m} 5\varphi(y_j) \tag{24}$$

in place of the true answer

$$f(X) = \sum_{j=1}^{n} \varphi(x_j) \tag{25}$$

Note that we multiply by 5 to account for the fact that $m = \frac{n}{5}$.

Releasing the subsampled dataset Y doesn't seem to provide much, if any, "privacy" to the users. If a user would be unhappy if you released the full dataset $\mathbf{x}$, then someone who is in the subsample Y would be just as unhappy or more if you released Y, and since each user in $\mathbf{x}$ has a reasonably large change of being in the subsample, they would probably not be happy with your decision to release Y before knowing whether or not they land in Y.

However, any reconstruction attack, when given the statistics computed on Y, *must* have reconstruction error at least $\frac{4n}{10}$, because the answers we are revealing do not depend on the users whose data wasn't subsampled. Thus the best the attacker can do is learn the secret bit of the users in the subsample exactly and then guess the secret bit of the other users at random.

However, the random subsample will simultaneously give a good estimate of the answers to many statistics. Specifically, one can prove the following result:

Exercise 3.10. Prove that for any set of statistics $f_1, \ldots, f_k$, with probability at least $\frac{99}{100}$,

$$\forall i \in [k] \quad \left| \sum_{j=1}^{m} 5\varphi_i(y_j) - \sum_{i=1}^{n} \varphi_i(x_j) \right| \leq O\left(\sqrt{n \log k}\right) \tag{26}$$

Therefore, we see that for $k = 20n$ as in the case of the efficient reconstruction attack, a random subsample will prevent reconstruction and give answers with error $\sqrt{n \log n}$. In contrast, reconstruction provably succeeds any time the error is $\ll \sqrt{n}$, so our reconstruction attack cannot be improved significantly. Also, when $k \ll 2^{o(n)}$, a random subsample will prevent reconstruction and answer all queries with error $o(n)$, meaning that no reconstruction attack that makes $k = 2^{o(n)}$ queries can tolerate noise $\frac{n}{100}$!

So this is a bit unsatisfying. The reconstruction attacks we have are the best possible, and can be defeated by a method that doesn't give a meaningful privacy guarantee. As the course goes on we will see how to give accurate answers to these statistics with rigorous privacy guarantees via *differential privacy* [DMNS06]. In some cases, the accuracy will match the limits imposed by reconstruction attacks and in some cases it won't. We will also see a more subtle type of privacy attack called *membership inference* that can help explain these gaps.

Exercise 3.11. (More general subsampling) Consider a dataset $\mathbf{x} = (x_1, \ldots, x_n)$. For some parameter m, we will define the *subsampled dataset* $Y = (y_1, \ldots, y_m)$ as follows. For each $j \in [m]$, independently choose a random element $r \in [n]$ and set $y_j = x_r$. Note that the sampling is *independent* and *with replacement*. Suppose we now use Y to compute the statistics in place of $\mathbf{x}$.

1. Given Y, how can we obtain an *unbiased* estimate of a count statistic $f(\mathbf{x}) = \sum_{j=1}^{n} \varphi(x_j)$? That is, output an estimate $f(Y)$ such that for every $\mathbf{x}$, $\mathbb{E}\left(f(Y)\right) = f(\mathbf{x})$.

2. What is the *variance* of your estimate? That is, $\mathbb{E}\left((f(Y) - f(\mathbf{x}))^2\right)$?

3. Suppose we are given statistics $f_1, \dots, f_k$. Prove the tightest bound you can on the maximum error of your estimates of $f_i(Y)$ over all $i = 1, \dots, k$. That is, prove that, for every dataset $\mathbf{x}$,

$$\mathbb{P}\left(\max_{i=1}^{k} |f_i(Y) - f_i(\mathbf{x})| \leq \blacksquare\right) \geq 1 - \beta \tag{27}$$

where you should fill in $\blacksquare$ with the best expression you can come up with in terms of m, k, and β.

4 Summary

Key Points

- Aggregate statistics place constraints on the dataset that can be used to reconstruct all or part of the underlying data about individuals.

- Any method that satisfies any meaningful privacy and accuracy guarantee cannot answer an arbitrary number of statistics.

- Any method that answers too many queries with an “insignificant” amount of error cannot satisfy any meaningful privacy guarantee.

Additional Reading

- A survey on privacy attacks against aggregate statistics [DSSU17]
- More discussion of reconstruction attacks at
  - https://differentialprivacy.org/reconstruction-theory/
  - https://differentialprivacy.org/diffix-attack/

References

[DMNS06] Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. In *Conference on Theory of Cryptography*, TCC ’06, 2006.

[DN03] Irit Dinur and Kobbi Nissim. Revealing information while preserving privacy. In *Proceedings of the 22nd ACM Symposium on Principles of Database Systems*, PODS ’03. ACM, 2003.

[DSSU17] Cynthia Dwork, Adam Smith, Thomas Steinke, and Jonathan Ullman. Exposed! A Survey of Attacks on Private Data. *Annual Review of Statistics and Its Application*, 4:61–84, 2017.

[DY08] Cynthia Dwork and Sergey Yekhanin. New efficient attacks on statistical disclosure control mechanisms. In *Annual International Cryptology Conference*. Springer, 2008.

[GAM19] Simson Garfinkel, John M Abowd, and Christian Martindale. Understanding database reconstruction attacks on public data. *Communications of the ACM*, 62(3):46–53, 2019.

[KRSU10] Shiva Prasad Kasiviswanathan, Mark Rudelson, Adam Smith, and Jonathan Ullman. The price of privately releasing contingency tables and the spectra of random matrices with correlated rows. In *Proceedings of the 42nd ACM Symposium on Theory of Computing*, STOC '10. ACM, 2010.

[MN12] S Muthukrishnan and Aleksandar Nikolov. Optimal private halfspace counting via discrepancy. In *Proceedings of the forty-fourth annual ACM symposium on Theory of computing*. ACM, 2012.

[NTZ13] Aleksandar Nikolov, Kunal Talwar, and Li Zhang. The geometry of differential privacy: The small database and approximate cases. In *ACM Symposium on Theory of Computing*, STOC '13, 2013.

[Wik21] Wikipedia contributors. Jeanne calment — Wikipedia, the free encyclopedia. https://en.wikipedia.org/w/index.php?title=Jeanne_Calment&oldid=1030629685, 2021. [Online; accessed 28-June-2021].