# Algorithms for Private Data Analysis — Spring 2021
# Lecture 2 & 3: Reconstruction Attacks

**Steven Wu**

How can releasing statistical information reveal information about individuals? At first glance, statistics like the average age of males in the dataset might seem harmless, but, as we will see, releasing many such statistics, even noisy versions of these statistics, can reveal a tremendous amount of information about individuals.

Although we will start with some very simpler ways that statistical information can violate privacy, this lecture is mostly about *reconstruction attacks.* At a high level, a reconstruction attack allows an attacker to recover all or part of a dataset using statistical information about that dataset. What we'll see is that reconstruction is not only possible, it is in fact it's inevitable if we don't limit what we want to reveal about the dataset.

## 1 What is Aggregate?

Before we try to convince you that releasing aggregate statistical information about a dataset *inevitably* reveals a lot about individuals, let's start with the claim that aggregate statistics *might* reveal something about individuals by working through some examples.

### 1.1 Warmup: 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 were born on [your birthday], and live in [your zipcode] and suffer from [stigmatizing medical condition]?

Suppose the answer to this query is 1. Seeing as how the employer knows your birthday and address, and you are likely the only person at your company with this birthday and zipcode, it is now very likely that your employer just learned about your stigmatizing medical condition.

Maybe this example is a little too obvious. I 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? It turns out that small answers are not the real issue, because we can come up with two queries whose answers are both large, but whose *difference* reveals information about a specific person. Unsurprisingly, this route for breaching privacy is called a *difference attack.* Consider the following pair of queries

> How many employees joined the company before [your start date] and suffer from [stigmatizing medical condition]?
>
> How many employees joined the company before [your start date + 1] and suffer from [stigmatizing medical condition]?

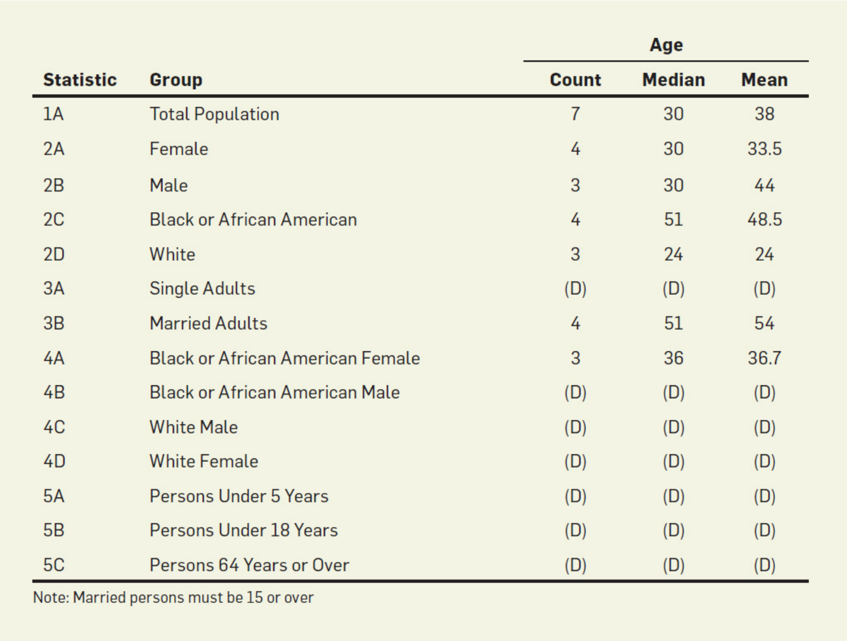
Suppose the answers to these queries are 417 and 418, respectively. Then, assuming you are the only employee who joined on your start date, you've just lost privacy.

Difference attacks might seem like a trivial example, but they are actually useful to keep in mind as a simple test case for many different issues in privacy.

## 1.2 Reconstruction Example: The Census

While difference attacks are simple, they do require the ability to ask semi-specific questions about the data. We'll now see an example of how far more benign-looking statistical information can reveal a lot about individuals in the dataset.

To do so, we'll look at an example of a recent reconstruction attack on the statistical disclosure control methods used by the U.S. Census Bureau up through the 2010 Decennial Census [GAM19]. The Census collects the age, sex, race, location, and other demographic information of those living in the United States. 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 statistics, such as those in Figure 1. As we discussed, releasing small counts is problematic, so certain cells in the table are *suppressed*, marked with a (D).



| | | Age | | |
|---|---|---|---|---|
| **Statistic** | **Group** | **Count** | **Median** | **Mean** |
| 1A | Total Population | 7 | 30 | 38 |
| 2A | Female | 4 | 30 | 33.5 |
| 2B | Male | 3 | 30 | 44 |
| 2C | Black or African American | 4 | 51 | 48.5 |
| 2D | White | 3 | 24 | 24 |
| 3A | Single Adults | (D) | (D) | (D) |
| 3B | Married Adults | 4 | 51 | 54 |
| 4A | Black or African American Female | 3 | 36 | 36.7 |
| 4B | Black or African American Male | (D) | (D) | (D) |
| 4C | White Male | (D) | (D) | (D) |
| 4D | White Female | (D) | (D) | (D) |
| 5A | Persons Under 5 Years | (D) | (D) | (D) |
| 5B | Persons Under 18 Years | (D) | (D) | (D) |
| 5C | Persons 64 Years or Over | (D) | (D) | (D) |

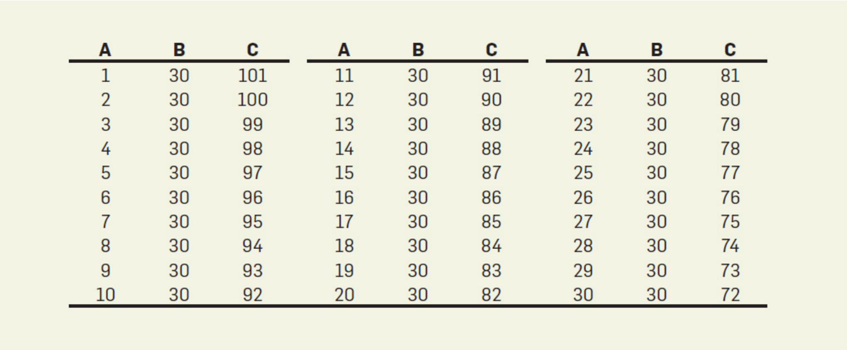
Note: Married persons must be 15 or over

Figure 1: An example of census tabulations from [GAM19].

What can we learn from this table? 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 $A \leq B \leq C$ (note that sorting the ages is just for notational purposes). We have the background information that $1 \leq A, B, C \leq 125$, where the upper bound is based on the oldest verified age of any human[1]. (The fact that ages can't be 0 is mysterious to me, but that's what they do in [GAM19] so we'll be consistent with that.) Since we are told that the median age of these three individuals is 30, we know that $B = 30$, and thus $1 \leq A \leq 30$ and $30 \leq C \leq 125$. Moreover, since the mean age of these three

[1] The French woman Jeanne Calment reportedly lived to 122 https://en.wikipedia.org/wiki/Jeanne_Calment.

individuals is 44, we know that $(A + B + C)/3 = 44$. This actually leaves relatively few possible choices for $A, B, C$, depicted in Figure 2.



| A | B | C | A | B | C | A | B | C |
|---|---|---|---|---|---|---|---|---|
| 1 | 30 | 101 | 11 | 30 | 91 | 21 | 30 | 81 |
| 2 | 30 | 100 | 12 | 30 | 90 | 22 | 30 | 80 |
| 3 | 30 | 99 | 13 | 30 | 89 | 23 | 30 | 79 |
| 4 | 30 | 98 | 14 | 30 | 88 | 24 | 30 | 78 |
| 5 | 30 | 97 | 15 | 30 | 87 | 25 | 30 | 77 |
| 6 | 30 | 96 | 16 | 30 | 86 | 26 | 30 | 76 |
| 7 | 30 | 95 | 17 | 30 | 85 | 27 | 30 | 75 |
| 8 | 30 | 94 | 18 | 30 | 84 | 28 | 30 | 74 |
| 9 | 30 | 93 | 19 | 30 | 83 | 29 | 30 | 73 |
| 10 | 30 | 92 | 20 | 30 | 82 | 30 | 30 | 72 |

Figure 2: Values for $A, B, C$ consistent with the constraints of Statistic 2B [GAM19].

Note that *a priori* there were $\binom{125+3+1}{3} = 341,376$ possible choices for $A, B, C$ and now we are down to just 30! Just knowing this information already tells us a lot—for example there is one 30 year old male and no male over 101—but it's not hard to see that adding more constraints will reveal even more information about the microdata.

**Exercise 1.1.** Suppose we unsuppress Statistic 4B, and it shows that the dataset contains two African-American males, whose mean age is 28. Is this enough information to uniquely determine the values $A$, $B$, and $C$? Either determine the unique choice of $A, B, C$ consistent with the aggregate statistics, or give two distinct choices for $A, B, C$.

### 1.3 The Reconstruction Paradigm

How can we think about what just happened? We start with some *dataset* $X$ that represents the underlying microdata. Each *statistic* (or *query*) on the dataset is some function $f(X)$, and for each statistic we obtain some answer $f(X) = a$. For example, the first column of Statistic 2B tells us that $f(X) = 3$ 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 $X$ must be a dataset with three males. More generally, some noise might have been introduced into the answers, and we might only know something like $f_i(X) \approx a_i$ up to some error bound $\alpha$ on the approximation. For example, the suppressed cells give a constrait of the form $f_i(X) \leq 5$. Tables like the one in Figure 1 give us a large set of constraints like

$$\begin{gathered} f_1(X) \approx a_1 \\ f_2(X) \approx a_2 \\ \vdots \\ f_k(X) \approx a_k \end{gathered}$$

Given all these constraints, we can define the *dataset reconstruction problem* as follows:

> Given a set of statistics $\{f_i(X) \approx a_i\}$ and an error bound $\alpha$, find a dataset $\tilde{X}$ that is consistent with all of the constraints.

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. Notice that in our above example we were not able to reconstruct *all* of $X$ exactly, but we were able to determine that any consistent dataset had three males, one of whom was age 30, and none of which were older than 101.

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 powertools like modern SAT-solvers to do dataset reconstruction in practice, so one shouldn't take much comfort in the fact that the problem is NP-hard. Moreover, in the next section we'll see examples where the reconstruction problem can be solved in polynomial time.

## 2 Linear Reconstruction Attacks

We'll now explore *linear reconstruction attacks*. Aside from capturing a lot of natural settings, these are an important case to study because we can analyze the attacks mathematically to understand the conditions under which reconstruction is possible. In particular, we'll see that these attacks are also fairly robust to noise introduced in the answers to the statistics, which means we can analyze tradeoffs between how much we want to release and how much noise we need to add.

### 2.1 The Model

This theory was introduced in a seminal paper by Dinur and Nissim [DN03] that led to the development of differential privacy, however we will see it presented in a somewhat different model that highlights how these attacks arise in practice.

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 "Has Disease?" that we represent as a single bit (Figure 3 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.

| Name | Postal Code | Age | Sex | Has Disease? |
|---|---|---|---|---|
| Alice | 02445 | 36 | F | 1 |
| Bob | 02446 | 18 | M | 0 |
| Charlie | 02118 | 66 | M | 1 |
| ⋮ | ⋮ | ⋮ | ⋮ | ⋮ |
| Zora | 02120 | 40 | F | 1 |

| Identifiers | Secret |
|---|---|
| $z_1$ | $s_1$ |
| $z_2$ | $s_2$ |
| $z_3$ | $s_3$ |
| ⋮ | ⋮ |
| $z_n$ | $s_n$ |

Figure 3: A toy dataset (left) and its representation in our model (right)

To start, we will simply 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 user $j$'s secret infomation (Figure 3 right). For our toy dataset

$$\mathcal{Z} = \{\text{names}\} \times \{\text{postal codes}\} \times \{\text{ages}\} \times \{\text{sexes}\}$$

and, for example, $z_1 = (\text{Alice, 02445, 36, F})$ and $s_1 = 1$.

Next, we will define a natural type of *count statistics* that captures many of the types of statistics one would release about this sort of dataset. 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 40 and have secret bit 1? Since we're interested in reconstructing the secret bits $s_j$, we will only consider statistics that have the form

$$f(X) = \sum_{j=1}^{n} \varphi(z_j)s_j \text{ for some } \varphi : \mathcal{Z} \to \{0,1\} \tag{1}$$

More generally we could consider queries of the form $f(X) = \sum_{j=1}^{n} \varphi(z_j, s_j)$, but for reconstruction attacks it helps to consider the special case in (1). Note that these statistics return an integer between 0 and $n$. In the example above $\varphi(z_j) = 1$ if and only if user $j$ is over 40.

The nice thing about these statistics, and the reason they are often called *linear statistics*, is because they can be expressed in the language of linear algebra. For a given statistic $f$ and dataset $X$ the value of the statistic has the form

$$f(X) = (\varphi(z_1), \dots, \varphi(z_n)) \cdot (s_1, \dots, s_n) \tag{2}$$

where $u \cdot v$ is the *dot product* between the two vectors. More generally, given a set of queries $f_1, \dots, f_k$ we can represent the answers as a matrix-vector product $F \cdot s$ defined as follows

$$\begin{bmatrix} f_1(X) \\ \vdots \\ f_k(X) \end{bmatrix} = \begin{bmatrix} \varphi_1(z_1) & \dots & \varphi_1(z_n) \\ \vdots & \ddots & \vdots \\ \varphi_k(z_1) & \dots & \varphi_k(z_n) \end{bmatrix} \begin{bmatrix} s_1 \\ \vdots \\ s_n \end{bmatrix} \tag{3}$$

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

$$f_i(X) = F_i \cdot s = (F \cdot s)_i \tag{4}$$

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 2.1.** 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 3 (ignoring the …parts of the table), write the matrix $F$, the secret vector $s$, and the product $F \cdot s$.

The reconstruction problem we're going to try to understand is how to take a vector of approximate answers $a \approx F \cdot s$ and recover a vector $\tilde{s} \approx s$. We'll mostly focus on when such reconstruction is possible, and only touch upon how to actually find $\tilde{s}$ efficiently.

## 2.2 A General Reconstruction Attack

For this lecture we're basically only going to 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

**Input:** a set of query vectors $F_1, \dots, F_k \in \{0,1\}^n$ and a set of answers $a_1, \dots, a_k \in \mathbb{R}$
**Output:** a vector of secrets $\tilde{s} \in \{0,1\}^n$

Return $\tilde{s} \in \{0,1\}^n$ that *minimizes* the quantity $\max_{i \in [k]} |F_i \cdot \tilde{s} - a_i|$

Figure 4: The reconstruction attack.

vector of secrets $\tilde{s} \in \{0,1\}^n$ that is *consistent* with the information we're given, in the sense that we would have obtained similar answers if the true secrets were $\tilde{s}$.

Note that the reconstruction attack is defined in terms of the queries $F_1, \dots, F_k$ in this vector representation. We could have defined the attack in terms of the queries $\varphi_1, \dots, \varphi_k$ and the public information $z_1, \dots, z_n$, which is a more realistic way of thinking of what the attacker has access to, but at this point we want to strip away some of the details of the model and focus on the underlying math. In our model we can always go from the queries and public identifiers to the vectors $F_1, \dots, F_k$, although it can take some work to do so in any given application.

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

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

*then the reconstruction attack returns $\tilde{s}$ such that* $\max_{i \in [k]} |F_i \cdot \tilde{s} - a_v| \leq \alpha n$.

To see why this claim is true, not that the true vector of secrets $s$ satisfies $\max_{i \in [k]} |F_i \cdot s - a_i| \leq \alpha n$. Thus, the vector $\tilde{s}$ that *minimizes* this quantity must make it no greater than $\alpha n$. We may actually find a vector $\tilde{s}$ that minimizes the quantity even further, but for our analysis we will only ue the fact that there is some $\tilde{s}$ that has error $\leq \alpha n$ with respect to the queries.

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

## 2.3 Reconstruction Using Many Queries

We'll start with a simple baseline, in which we try to release *all possible statistics* on the dataset $X$. We will show that any set of answers to this enormous set of queries allows reconstruction, even when the error is large enough to render the aswers almost useless. For simplicity, let's assume that the public identifiers $z_1, \dots, z_n$ are all *unique*, and it may help to fix them to be $z_j = j$. For every vector $v \in \{0,1\}^n$, we can define a query

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

Note that there are $k = 2^n$ such queries, so exponential in the size of the dataset. For notational simplicity, we wil index the queries by the vector $v$,[2] and denote the vector of queries $(f_v)_{v \in \{0,1\}^n}$.

**Exercise 2.3.** For $n = 3$, and $z_j = j$ for $j \in \{1,2,3\}$, write down the matrix $F \in \{0,1\}^{k \times n}$

[2] If indexing an array with a vector bothers you, think of $v$ as an integer written in binary.

We'll show that given answers to all these queries that are accurate to within $\pm\frac{n}{100}$, we can recover $\tilde{s}$ that is correct for at least $\frac{95n}{100}$ users. Thus, even very noisy answers can lead to almost perfect reconstruction! An important note: we're placing *no assumptions* on the secret vector $s$, 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.

We will prove the following theorem about this reconstruction attack.

**Theorem 2.4** ([DN03]). *If all queries have error at most $\alpha n$, then the reconstruction error (the number of entries on which $\tilde{s}$ and $s$ disagree) is at most $4\alpha n$.*

Before we present the proof. Here are two observations: First,without any prior information and statistics, the attacker can "guess" roughly 50% of the secret bits. Second, the attacker can't do this without *your data* in the dataset.

*Proof.* Our claim is that any such guess $\tilde{s}$ in fact agrees with the true private bits $s$ for all but $4\alpha n$ of the users. The reason is that if $\tilde{s}$ disagreed with more than $4\alpha n$ of the secret bits, then the answer to some query would have eliminated $\tilde{s}$ from contention. To see this, fix some $\tilde{s} \in \{0,1\}^n$, 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

$$|F_v \cdot s - F_v \cdot \tilde{s}| = |S_{01}| > 2\alpha n, \tag{7}$$

but, at the same time, if $\tilde{s}$ were output by the attacker, we would have

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

which is a contradiction. □

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 $\tilde{s}$.

## 2.4 Reconstruction Using Fewer Queries

The reconstruction attack we just discussed is quite powerful, as it recovers 96% of the secret bits correctly even from answers with 1% error. Even that is a non-obvious an important message. But arguably the attack is unrealistic, since it requires answers to an enormous set of $2^n$ counts, which is infeasible even for relatively small datasets.

What if we simply bound the number of counts 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 $O(n)$ *random* queries are enough to reconstruct the dataset to high-accuracy, provided that the answers have error $\ll \sqrt{n}$. There are good reasons why this $\sqrt{n}$ bound appears, and why it's interesting, which we'll return to after we describe and analyze the attack.

However, Dinur and Nissim showed that if we obtain *highly accurate* answers—still noisy, but with error smaller than the sampling error—then we can reconstruct the dataset to high accuracy. We can also make the reconstruction process computationally efficient by using linear programming to replace the exhaustive search over all $2^n$ possible vectors of secrets.

**Describing the attack.** Specifically, in the attack, the attacker will now choose $k = 20n$ *randomly chosen* functions[3] $\varphi_i : \mathcal{Z} \to \{0,1\}$. Note that these functions correspond to random vectors $(F_i)_{i \in [k]}$.[4] The reconstruction procedure will be the same as before. The attacker will receive a vector of answers $a = (a_v)$ with the guarantee

$$\forall i \in [k] \;\; |F_i \cdot s - a_i| \leq \alpha n \tag{9}$$

Note that we want the error bound to be $\ll \sqrt{n}$ so think of $\alpha$ as $\ll 1/\sqrt{n}$. For now, reconstruction will proceed the same way, by finding any vector $\tilde{s} \in \{0,1\}^n$ such that

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

As before, a solution always exists because the actual private bits $s$ will satisfy the constraints. Note that even though there are only $20n$ queries, finding $\tilde{s}$ might still require time proportional to $2^n$ because we have to search over $2^n$ possible vectors $\tilde{s} \in \{0,1\}^n$. After analyzing this attack, we wil see a method for reconstructing in polynomial time.

We can prove the following theorem about this reconstruction attack.

**Theorem 2.5** ([DN03]). *If all queries have error at most $\alpha n$, then with extremely high probability, the reconstruction error (the number of entries on which $\tilde{s}$ and $s$ disagree) is at most $256\alpha^2 n^2$. Note that the constant 256 is somewhat arbitrary and can definitely be improved with more careful analysis.*

Observe that when $\alpha \ll 1/\sqrt{n}$, the reconstruction error is $\ll n$, meaning that we recover nearly all of the secret bits. 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 $s$ and $\tilde{s}$ disagree on many bits, then there will be some query that proves $\tilde{s}$ cannot be close to $s$.

In order to complete the proof, we'll need the following technical fact.

**Claim 2.6.** *Let $t \in \{-1, 0, +1\}^n$ be a vector with at least $m$ non-zero entries and let $u \in \{0,1\}^n$ be a uniformly random vector. Then*

$$\mathbb{P}\big(|u \cdot t| \leq \sqrt{m}/4\big) \leq \frac{9}{10} \tag{11}$$

*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. □

Now let's return to the proof of Theorem 2.5

*Proof of Theorem 2.5.* Our goal will be to show that any vector $\tilde{s} \in \{0,1\}^n$ that disagrees with $s$ on more than $256\alpha^2 n^2$ bits cannot satisfy

$$\forall i \in [k] \;\; |F_i \cdot \tilde{s} - a_i| \leq \alpha n. \tag{12}$$

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

$$\mathcal{B} = \left\{\tilde{s} : \tilde{s} \text{ and } s \text{ disagree on at least } 256\alpha^2 n^2 \text{ coordinates}\right\} \tag{13}$$

[3] The choice of the constant 20 is somewhat arbitrary

[4] We'll use the handy notation $[k] = \{1, \ldots, k\}$ a lot in this course.

Our goal is to show that the reconstruction attack does not output any vector in $\mathcal{B}$. To this end, we will say that statistic $i$ *eliminates* vector $\tilde{s}$ if

$$\left|F_i \cdot (s - \tilde{S})\right| \geq 4\alpha n \tag{14}$$

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

$$|F_i \cdot \tilde{s} - a_i| \geq |F_i \cdot (s - \tilde{s}) - a_i| - |F_i \cdot s - a_i| \geq 4\alpha n - \alpha n = 3\alpha n. \tag{15}$$

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

$$\forall \tilde{s} i \in \mathcal{B} \;\; \exists i \in [k] \;\; |F_i \cdot (s - \tilde{s})| \geq 4\alpha n \tag{16}$$

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

$$\exists i \in [k] \;\; |F_i \cdot (s - \tilde{s})| \geq 4\alpha n \tag{17}$$

We will show how (17) can be deduced from Claim 2.6. Fix some vectors $s$ and $\tilde{s}$ that differ on at least $m$ coordinates, and define $t = s - \tilde{s}$. Then $t \in \{-1, 0, +1\}^n$ and $t$ has at least $m$ non-zero entries. Moreover, since the queries are chosen uniformly at random, $u = F_i$ is a uniformly random vector in $\{0,1\}^n$. Thus $u$ and $t$ satisfy the assumptions of Claim 2.6, so we conclude that

$$\mathbb{P}(|F_i \cdot (s - \tilde{s})| \leq 4\alpha n) \leq \frac{9}{10} \tag{18}$$

Thus, each query $f_i$ has a small chance of catching the difference between $s$ and $\tilde{s}$. Now, since the $k = 20n$ queries are independent, we have that

$$\mathbb{P}(\forall i \in [k] \;\; |F_i \cdot (s - \tilde{s})| \leq 4\alpha n) \leq \left(\frac{9}{10}\right)^{20n} \leq 2^{-2n} \tag{19}$$

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

$$\mathbb{P}(\exists \tilde{s} \in \mathcal{B} \;\; \forall i \in [k] \;\; |F_i \cdot (s - \tilde{s})| \leq 4\alpha n) \leq 2^n \cdot 2^{-2n} = 2^{-n} \tag{20}$$

We now know that (except with probability $\leq 2^{-n}$), every vector $\tilde{s}$ that disagrees with $s$ on more than $m$ coordiantes will be eliminated from contention, so the attacker must return a vector $\tilde{s}$ 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 $2^{-n}$. □

**Do the queries have to be random?** Although we modeled the queries, and thus the matrix $F$ as uniformly random, it's important to note that we really only relied on the fact that

$$\max_{i \in [k]} |F_i \cdot (s - \tilde{s})| \gtrsim \sqrt{\mathrm{err}(s, \tilde{s})}, \tag{21}$$

where we define $\mathrm{err}(s, \tilde{s})$ is the number of coordiantes on which they disagree. We can reconstruct when the error is $\ll \sqrt{n}$ for any family of queries that gives rise to a matrix with this property. Not every matrix satisfies this property, and later in the course 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 have this property. More specifically, the property is satisfied by any matrix with no small singular values [DY08] or high discrepancy [MN12]. There is a large body of work showing that many specific families of queries lead to reconstruction, such as the *conjunction* queries we began this lecture with [KRSU10].

**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 $c \in \mathbb{R}^d$, and $m$ constraints, each represented by a vector $a_i \in \mathbb{R}^d$ and a scalar $b_i \in \mathbb{R}$, a linear program can be written as

$$\max_{x \in \mathbb{R}^d} \; c \cdot x$$
$$\text{s.t. } \forall i \in [m] \; a_i \cdot x \leq b_i$$

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.

### 2.4.1 Making the attack computationally efficient

The attack we analyzed above is still not computationally efficient, since the attacker might have to enumerate all $2^n$ vectors $\tilde{s} \in \{0,1\}^n$ to find one that minimizes

$$\max_{i \in [k]} |F_i \cdot \tilde{s} - a_i| \tag{22}$$

Note that to analyze reconstruction it was sufficient to find any vector where this maximum is at most $\leq \alpha n$ but there is no reason not to find the *optimal* vector $\tilde{s}$.

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 $\hat{s} \in [0,1]^n$ that solves the following

$$\min_{\hat{s} \in [0,1]^n} \max_{i \in [k]} |F_i \cdot \hat{s} - a_i| \tag{23}$$

Then, to obtain the reconstruction we will round each entry to 0 or 1 to obtain a vector $\tilde{s} \in \{0,1\}^n$.

**Exercise 2.7.** The optimization problem in (23) does not look like LPs as they are defined in the cutout but can indeed be written as one. Show how to write a linear program whose solution solves (23).

Using a very slightly more complex analysis, one can show that if all the answers have error at most $\alpha n$, then the solution to the linear program will satisfy

$$\sum_{j=1}^{n} |s_j - \hat{s}_j| \leq O(\alpha^2 n^2) \tag{24}$$

and therefore the rounded solution satisfies

$$|\{j : s_j \neq \tilde{s}_j\}| \leq O(\alpha^2 n^2) \tag{25}$$

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

### 2.4.2 Discussion

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 $\frac{n}{100}$ but only if we have $2^n$ 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 "defeats reconstruction" 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 $X = (x_1, \dots, x_n)$. Now fix $m = \frac{n}{5}$ and we will define the *subsampled dataset* $Y = (y_1, \dots, y_m)$ as follows. For each $j \in [m]$, independently choose a random element $j' \in [n]$ and set $y_j = x_{j'}$. Note that the sampling is *independent* and *with replacement.* Suppose we now use $Y$ to compute the statistics in place of $X$. That is, $f$, we return the answer

$$5 \cdot f(Y) = 5 \cdot \sum_{j=1}^{m} \varphi(y_j) \tag{26}$$

in place of the true answer

$$f(X) = \sum_{j=1}^{n} \varphi(x_j) \tag{27}$$

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

The subsampled dataset $Y$ doesn't seem inherently "private," since each user has their data in the subsample with probability $\frac{1}{5}$ so whatever I was worried about happening with my data in $X$ still has a reasonable chance of happening in $Y$. However, any reconstruction attack like those we studied above *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 2.8.** Prove that for any set of statistics $f_1, \dots, f_k$, with probability at least $\frac{99}{100}$,

$$\forall i \in [k] \quad \left| 5 \cdot \sum_{j=1}^{m} \varphi_i(y_j) - \sum_{i=1}^{n} \varphi_i(x_j) \right| \leq O\Big(\sqrt{n \log k}\Big) \tag{28}$$

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 wont. We will also see a more subtle type of privacy attack called *membership-inference* that can help explain these gaps.

## 3 Linear Reconstruction in Practice

Often it takes some work to see how a real system would give rise to the structure and conditions we need to make reconstruction attacks work. Before we wrap up, let's show how reconstruction attacks can be applied in practice to a commercial system called *Diffix*. Diffix is a system designed by the startup Aircloak for computing statistics on a private database.

The goal of Diffix is to answer SQL queries, such as:

```
SELECT count(*) FROM loans
WHERE loanStatus = 'C'
AND clientId BETWEEN 2000 and 3000
```

on a database while preventing the disclosures about individual records in the dataset. To map these queries to the setting we considered, denote the records in the dataset $x_1, \dots, x_n$ and let each record be of the form $x_j = (\texttt{clientId}, \texttt{loanStatus})$. Then these queries fit our model and return

$$\sum_{j=1}^{n} \varphi(x_j) \tag{29}$$

where $\varphi$ returns 1 if and only $2000 \leq \texttt{clientId} \leq 3000$ and `loanStatus = 'C'`. They are also in the special form required for reconstructing the secret bits $s_j$ that indicate whether `loanStatus = 'C'`.

We we've seen, such a system must not provide exact answers or else the attacker could run a differencing attack, thus Diffix uses a variety of heuristics for adding noise to the answers, but the noise they add does not satisfy formal privacy guarantees like the methods we'll see later in the course.

Recall that to mount a reconstruction attack, we'd like to consider *random* queries. We can construct a random query by randomly choosing whether to add each `clientId` to the query, giving us a query like this:

```
SELECT count(*) FROM loans
WHERE loanStatus = 'C'
AND (clientId = 2007
OR clientId = 2018
...
OR clientId = 2991)
```

The makers of Diffix were aware that random queries of this sort lead to reconstruction attacks, so they wanted to ensure that such queries would be answered with noise $\Omega(\sqrt{n})$. Thus, they have a system for adding noise that scales with the square root of the *number of terms* in the query. If we are trying to reconstruct a dataset of size $n$, and we add each `clientId` at random, then these queries will have $\Omega(n)$ terms and thus noise $\Omega(\sqrt{n})$. So far, so good!

However, it turns out that truly random queries are not necessary for reconstruction. In particular, Cohen and Nissim [CN18] showed how to construct queries that are *syntactically simple* but are *random enough* for reconstruction attacks. Specifically, these queries have the form

```
SELECT COUNT(clientId) FROM loans
WHERE FLOOR(100 * ((clientId * 2)^0.7))
    = FLOOR(100 * ((clientId * 2)^0.7) + 0.5)
AND clientId BETWEEN 2000 and 3000
AND loanStatus = 'C'
```

Why queries liket hese are useful work is a combination of theoretical principles beyond the scope of this course and some experimental optimizations. The upshot is that because these queries do not require many terms to specify, Diffix answered the queries with very small noise $O(1)$, and thus full victim to reconstruction.

In response Diffix was modified so as not to allow mathematical operations on attributes like `clientId` that are *unique* for each record (or most records). However, this system also fell victim to reconstruction attacks by specifying conjunctions of many attributes that are individually not unique but that can be combined to product unique identifiers. Think of queries like

```
handedness = 'right' AND age ≤ 50 AND state = MA AND eyeColor = 'blue'
```

To their credit, AirCloak runs a bounty program that encourages people to catch these attacks and thereby harden the system.[5] Nonetheless these attacks show that it's challenging to avoid reconstruction through *ad hoc* methods.

## Summary

### Key Points

- Merely releasing "aggregate" statistics can reveal information about individuals because of small counts and *difference attacks*.
- Releasing aggregate statistics imposes constraints on the underlying data, and releasing many such statistics, even with noise, likely allows an attacker to *reconstruct* all or part of the dataset.
- Reconstruction attacks show that it is impossible to release an *arbitrarily large* set of statistics with any non-trivial accuracy guarantee.
- Releasing even a linear number of sufficiently rich statistics with noise $o(\sqrt{n})$ will allow reconstruction of the dataset.

### 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

[CN18] Aloni Cohen and Kobbi Nissim. Linear program reconstruction in practice. *arXiv preprint arXiv:1810.05692*, 2018.

[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.

[5] If you'd like a great course project, try to claim the bounty!

[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.