DS 102 Data, Inference, and Decisions **Fall 2019**

Randomized Response

Lecturer: Moritz Hardt

Privacy is an issue you'll almost certainly face in practice. Many valuable applications of data science touch on personal data, from health data, and location data and mobile phone activity to smart meter data.

This lecture and the next focus on how we can perform useful data analysis on sensitive data sets in a way that protects the privacy of individuals in your data set. Most of this lecture will examine data privacy attacks, while the next lecture will focus on methods to promote privacy while still allowing for useful data analysis. Understanding common attacks is one of the best ways to mitigate them; that said, we expect you to be responsible with the material in this lecture.

1 De-anonymization techniques

Many data sets must be scrubbed of personally identifiable information (PII) before release, so that given a dataset, one cannot determine the identity of any individual in the data set. A common first step is to remove "sensitive attributes" from data to anonymize data. For example, the HIPAA safe harbor provision specifies the following as sensitive attributes for medical data: name, location, phone number, email, IP, SSN, medical record numbers, health plan numbers, and more. All of these must be removed from a medical data set to insure HIPAA compliance. However, just removing all sensitive attributes from your data set is not sufficient to guarantee that individuals cannot be identified.

In 1997, researcher Latanya Sweeney showed that she could de-anonymize a HIPAA-compliant medical data set by matching it with entries in a public ally available data set of voter records. There were three attributes in common between these data sets: zip code, date of birth, and sex. On their own, any of these three attributes would not be enough to identify any individual. But taken together, they were enough to uniquely identify the medical records of the governor of Massachusetts.

This is an example of a **linkage attack** in which one links multiple data sources, some of which are seemingly anonymized sensitive data (e.g. medical records) with a seemingly innocuous data sets (e.g. voter list). Having your name on a voter is not harmful, but when other attributes overlap (as in the case of the governor of Massachusetts), it can be enough to link you to sensitive records you did not want you name attached to. Even today, many data privacy attacks come from unforeseen data linkage attacks.

Understanding the dangers of linkage attacks gave a rise to a definition k-anonymity. This notion distinguishes between "quasi-identifiers" (e.g. zip code, date of birth) and "sensitive attributes" (e.g. name). A data set preserves k-anonymity if for any setting of the quasi-identifiers and sensitive

attributes, there are at least k individuals who match this setting. To achieve k-anonynimity, one needs in general to modify their data set (by removing attributes or by combining them into coarser bins). However, ensuring k-anonymity is not enough to ensure privacy, as we'll see in the next example.

In 2006, movie rental company Netflix released a dataset of 18,000 movies and 480,000 users, with roughly 100 million ratings of the movies by those users. Ratings were in $\{0, 1, .., 5\}$, and the official challenge goal was to predict the missing entries (the ratings that users would give to movies had they watched them). The movie ratings data was potentially sensitive, as the ratings (or that fact than one had watched) certain movies could be embarassing. To make the data set publicly available, Netfilx replaced user names with random numbers. The challenge ran from 2006 to 2009.

However, Netflix is not the only curator of movie ratings. The public database IMDB also collects individuals' ratings of movies. By posting on IMDB users opt-in to having these ratings be publicly available. In another linkage attack, researchers Arvind Narayanan and Vitaly Shmatikov showed that using the publicly available IMDB ratings, they could identify some of the Netlix users, and thus obtain their ratings of movies that they didn't post publicly to IMDB.

2 The fundamental law of information recovery

Arvind Narayanan articulated the term "33-bits of entropy" to describe data susceptibility to linkage attacks. Why 33? $2^{33} \approx 8.5$ billion, more than the number of people in the world. This means, as a rule of thumb, given any data set with > 33 bits of entropy, de-anonymization is in principle possible. [1]

As another example on linkage attacks, we'll talk about Genome Wide Association Studies (GWAS). In these studies, the National Institute of Health (NIH) collects data from test candidates with a common disease, and releases minor allele frequencies (MAF) of the test population at positions in the DNA sequence. The goal is to find the commonalities in the individuals' DNA sequences that are associated with certain diseases.

There was an interesting attack on this dataset that took a different flavor from the attacks we've discussed so far. In particular, in 2018, Homer et al.[2] showed that they could *infer whether an individual was in the published test group* from a DNA sequence of the individual and the published aggregated data in a GWAS study.

First, they compared the individual's minor allele locations with that the the average of the published NIH test population (first two rows of Table 1.1). They also compared the individual's data to a reference population data set with MAFs sampled from a larger, more representative population, which was also publicly available (third column of Table 1.1).

To determine if the individual was in the study or not, they looked at each DNA location: for each

[1] see `https://33bits.wordpress.com/` for more.

[2] you can find the original paper here: `https://journals.plos.org/plosgenetics/article?id=10.1371/journal.pgen.1000167`

DNA loc.	1	2	3	...	100,000
Test Pop. MAF	0.02	0.03	0.05	...	0.02
Individual MA	No	No	Yes	...	Yes
Reference Pop. MAF	0.01	0.04	0.04	...	0.01
Closer to?	Ref	Test	Test	...	Test

Table 1.1: Caption

they computed whether the individual's MA was closer to the MAF of the test population, or the reference population. They then took a majority vote over all the DNA locations to determine whether the individual was in the GWAS data set (more MA's were closer to Test than Ref. population) or not. For this data set, with $n = 1,000$ and with $100,000$ DNA locations, this attack could identify whether individuals where in the NIH data set or not, with high accuracy.

Analysis of each DNA location separately gives a very weak signal, but when combined, these weak signals produce a very strong signal. This particular example is evidence of a technical principle common in several attacks: many weak signals combine into one strong signal. This is encoded in the **fundamental law of information recovery**. It's an informal law[3], that states:

> "Overly accurate information about too many queries to a data source allows for partial or full reconstruction of data (i.e. *blatant non-privacy*)."

Note that for any reasonable definition of privacy, full reconstruction of data will violate that definition. This type of attack is called a *reconstruction attack*.

The following lemma is one way to make this statement more formal, but also more precise. Many reconstruction attacks to reduce to some variant of the following **signal boost lemma**:

Lemma 1.1 (Signal boost lemma)**.** *Let $b \in \{-1, 1\}$ be an unknown bit. Given access to draws from the Bernoulli distribution, $\mathcal{B} = Bernoulli(1/2 + \epsilon \cdot b)$ (where draws are either 1 or -1, not 1 or 0) $\Theta(1/\epsilon^2)$ samples are both necessary and sufficient to determine the bit b with high confidence.*

Proof. (Sufficiency). To prove sufficiency, we will show that $n = \Theta(1/\epsilon^2)$ draws of our choice will suffice to determing b with high confidence. Consider the following procedure: sample bits $b_1, ..., b_n$ i.i.d. from the distribution $\mathcal{B}$ and compute their average, $\bar{b} = \frac{1}{n}\sum_i b_i$. If $\bar{b} > 0$, guess $\hat{b} = 1$, otherwise guess that $\hat{b} = -1$. The expectation of $\bar{b}$ is given by:

$$\begin{aligned}\mathbb{E}[\bar{b}] &= \mathbb{E}[\frac{1}{n}\sum_i b_i] \\ &= (\frac{1}{2} + \epsilon b)1 + (\frac{1}{2} - \epsilon b)(-1) \\ &= 2\epsilon b\end{aligned}$$

[3]The law is commonly used within the privacy community. See `https://www.cis.upenn.edu/~aaroth/Papers/privacybook.pdf` for more).

so if we knew the expected value, we would be in good shape to predict b. A sample average will suffice if the variance of the estimator is small. Computing the variance of $\bar{b}$, we get:

$$\begin{aligned}
\text{Var}[\bar{b}] &= \mathbb{E}[\bar{b}^2] - \mathbb{E}[\bar{b}]^2 \\
&= \frac{1}{n^2}\left(\sum_{i=1}^{n}\sum_{j=1}^{n}\mathbb{E}[b_i \cdot b_j]\right) - 4\epsilon^2 b^2 \\
&= \frac{1}{n^2}\left(\sum_{i=1}^{n}\mathbb{E}[b_i^2] + \sum_{i=1}^{n}\sum_{j\neq i}\mathbb{E}[b_i \cdot b_j]\right) - 4\epsilon^2 \\
&= \frac{1}{n^2}\left(\sum_{i=1}^{n} 1 + \sum_{i=1}^{n}\sum_{j\neq i} 4\epsilon^2\right) - 4\epsilon^2 \\
&= \frac{1}{n^2}\left(n + 4(n^2-n)\epsilon^2 - 4n^2\epsilon^2\right) \\
&= \frac{1}{n}(1 - 4\epsilon^2)
\end{aligned}$$

which for small ϵ is approximately $\frac{1}{n}$, showing that for large enough samples, the sample mean will suffice to recover b. In particular, your guess $\hat{b}$ will be good with high probability if $\epsilon > \frac{c}{\sqrt{n}}$ for some constant c.

(Necessary). A future version of the notes will cover the necessary direction. This direction of the proof material outside of the scope of DS102 Fall 2019 so stay tuned if you're interested. □

3 Privacy Attacks

The next attack we will study is an **approximate inversion attack** (also known as a **linear reconstruction attack**). In this setting you have a binary vector $a \in \{-1, 1\}^n$ where each bit corresponds to sensitive information of one of the n individuals. The attacker may query these bits by specifying some vector $w \in \{-1, 1\}^n$, the data curator will return $\langle a, w\rangle + \epsilon$, where ϵ is an unknown noise term. The goal is to reconstruct the vector a using a sequence of queries w.

A natural question is then, assuming some bound on ϵ, how many queries do we need to approximately reconstruct a?

If $\epsilon = 0$, then there is no noise. In this case, picking n queries $w_1, ..., w_n$ that create matrix W with full rank then inverting the linear system will suffice. So we need at most n queries when there is no noise.

The same type of strategy will work even when there is nonzero noise. Denoting our queries $w_1, ..., w_n$ as rows of an $n \times n$ matrix, the responses to those queries are given as u, where:

$$u = Wa + \epsilon = \begin{bmatrix} - & w_1 & - \\ - & w_2 & - \\ & \vdots & \\ - & w_n & - \end{bmatrix} \begin{bmatrix} | \\ a \\ | \end{bmatrix} + \begin{bmatrix} | \\ \epsilon \\ | \end{bmatrix}$$

To recover a from the observed u, we can compute $v = W^{-1}u$ (since we have choice over W we can always ensure that it is in invertible). However, due to the noise ϵ, $v \neq u$. Rather,

$$v = W^{-1}u = W^{-1}(Wa + \epsilon) = a + W^{-1}\epsilon$$

We can reconstruct a up to the error $W^{-1}\epsilon$. Therefore, we want to ensure that $W^{-1}\epsilon$ is as small as possible. We'll consider the ℓ_2 norm[4], and recall that

$$\|W^{-1}\epsilon\|_2 \leq \|W^{-1}\|_2 \|\epsilon\|_2 !.$$

$\|W^{-1}\|_2$ is the operator norm of W^{-1}, and is equal to $1/\sigma_n(W)$, where $\sigma_n(W)$ is the smallest singular value of W. That is, we want to pick a W with large singular values.

It turns out that a matrix $W \in \{-1, 1\}^{n \times n}$ with each entry randomly -1 or 1, independently of each other, will in have $\sigma_n(W) \gtrsim \sqrt{n}$.

Altogether, we have

$$\|W^{-1}\epsilon\| \leq \|W^{-1}\|\|\epsilon\| = \|\epsilon\|/\sigma_n(W) \lesssim n^{-1/2}\|\epsilon\|$$

Assuming $\|\epsilon\|^2 = o(n^2)$, then

$$\|v - a\|^2 = \|W^{-1}\epsilon\| \lesssim o(n^2)$$

The following corollary summarizes our work in a specific setting of the errors:

Corollary 1.2. *Assuming each coordinate of the perturbation ϵ has magnitude $o(n^{-1/2})$, the linear reconstruction attack (defined above) reconstructs a up to an average coordinate error of $o(1)$. That is, the reconstruction is close to the true $-1/1$ values up to rounding error, for a large fraction of the coordinates.*

Stepping back, a large class of privacy attacks centers on finding noisy, but invertible linear measurements of a system.

4 Randomized Response

Now that we've discussed many ways to break privacy, we will look at some methods to promote privacy. We'll cover more on this in the next lecture as well.

[4]Specifically, we use the Euclidean norm on vectors, and the operator norm on matrices

Randomized response methods came from a realization that sensitive questions elicit *evasive answers bias*, as the responder was reluctant to reveal the truth.

Suppose $b \in \{-1, 1\}$ is the answer to the sensitive question. The basic idea of randomized response methods is to not ask for b directly, but rather to sample b' from Bernoulli$(1/2 + \epsilon b)$. The surveyor never observes b directly, and the responder has plausible deniability that their answer was due to the randomness in the sample of b'.

Still, each bit b' on it's own is not very useful. However, if n individuals report noisy bits $b'_i \sim$ Bernoulli$(1/2 + \epsilon b)$. We're interested in the average sensitive value, $\frac{1}{n}\sum_i b_i$.

Similar to our analysis of the signal boost lemma, one can show that $\mathbb{E}[\frac{1}{n}\sum_i b'_i] = 1/2 + \epsilon\frac{1}{n}\sum_i b_i$ and var$[\frac{1}{n}\sum_i b'_i] = O(1/n)$. Even though every bit alone is useless, by averaging them we can construct a useful statistic.

5 Summary

In summary, sufficiently rich information about data can always be used to re-identify individuals in the data set. The signal boost lemma characterized the idea behind a general attack strategy: identify many sources of mild correlation, and boost those into a large correlation. We also saw two common attack schemes: linkage attacks and linear reconstruction attacks, as well as an early attempt to maintain privacy but still allow for statistical analysis of sensitive data sets.

Setting the stage for the next lecture, the signal boost lemme showed that we can't invoke randomized response too many times, or else the private bit will be compromised. In fact, it's not clear yet how to generalize the randomization scheme to multiple analysis such that the privacy guarantee *composes well*. Next time, we'll see how to do this when we talk about *differential privacy*.