Section 2: Attacks Review

CS 208 Applied Privacy for Data Science, Spring 2022

February 15, 2022

1 Agenda

- Discuss any lingering questions related to PSET 1.
- Briefly discuss reconstruction attacks.
- Briefly discuss membership attacks.
- Create and run a membership attack together.

2 Overview of Reconstruction Attacks

Recall that in reconstruction attacks, in the simplified setting, there is a n-bit vector we want to reconstruct that is sensitive: $x \in \{0,1\}^n$. For any $S \subseteq [n]$, the adversary can ask for (approximate) answers to $q_S(x) = \sum_{i \in S} x_i$. If answers to $q_S(x)$ are too exact (in particular, within $o(\sqrt{n})$ of the truth), the adversary can reconstruct x almost exactly (i.e., a $1 - o(1)$ fraction of x) with high probability. See lecture notes for more details.

3 Overview of Membership Attacks

In the membership attack, we have the following components:

- Population probabilities, $p = (p_1, \ldots, p_d) \in [0,1]^d$, where d is the number of attributes and p_j is the proportion of the jth attribute in the population. We assume that the p_j's are iid uniform in $[0,1]$, and that in any draw from the population, the jth attribute is iid Bernoulli(p_j) and independent from all other attributes.
- Dataset containing n random samples from the population. Each sample is a vector $\in \{0,1\}^d$, where the jth bit indicates whether or not the individual has the jth attribute.
- Sample means, $\bar{x} \in [0,1]^d$.
- Noisy means released by the mechanism, $a = M(\bar{x}) \approx \bar{x} \in \{0,1\}^d$. We do not assume the noise added is independent or unbiased; only that $|a_j - \bar{x}_j| < \alpha$ with high probability.
- An independent draw from the population, $y \in \{0,1\}^d$, representing Alice's data.

The adversary gets y, a, and p and tries to determine whether Alice is in or out of the dataset.

Theorem 3.1 (Dwork et al., 2015)**.** *There is an attacker A such that when $d < O(n)$ and $\alpha < \min\{\sqrt{d/O(n^2 \log(1/\delta))}, 1/2\}$:*

- *If Alice is IN,* $\Pr[A(y,a,p) = IN] \geq \frac{1}{O(\alpha^2 n)}$.
- *If Alice is OUT,* $\Pr[A(y,a,p) = IN] \leq \delta$

The attack is very simple. Let T be a threshold set to $O\big(\sqrt{d \log(1/\delta)}\big)$. The attacker outputs:

$$A(y,a,p) = \begin{cases} \text{IN} & \text{if} \quad \langle y-p, a-p \rangle \; > \; T \\ \text{OUT} & \text{if} \quad \langle y-p, a-p \rangle \; \leq \; T \end{cases}$$

Note that in practice T can be set via simulating the null distribution for the test statistic. If Alice is OUT, and $A(y,a,p) = IN$, this is called a "false positive" and if Alice is IN and $A(y,a,p) = IN$, this is called a "true positive."

4 Running a membership attack

Now, we will run an experiment to evaluate the effectiveness of the membership attack (similar to the attack covered in class). We will use `membership_attack.ipynb` from the 02/03 lecture as a starting point. However, we will not think of `pub` as known to us as attacker, but rather that we only know of `alice` for the individual Alice for whom we are trying to determine membership in the dataset.

Recall that the membership attack we saw in lecture requires the means of many boolean attributes. Since the PUMS dataset is very low-dimensional (not all boolean), we will use random predicates q_j to create derived boolean attributes for each individual. That is, we'll treat $q_j(\texttt{pub}_i)$ as the jth boolean attribute of user i in the membership attack. When we issue $q = q_j$ as a query, we will get back its mean over the dataset (or an approximation of the mean):

$$\frac{1}{n}\sum_{i=1}^{n} q(\texttt{pub}_i), \tag{1}$$

where n is the number of rows in the dataset.

1. Create a function `execute_means_exact(predicates)`, which takes as input a list `predicates` q on the `pub` variables and returns the list of means on `data`, computed as in Equation (1).
2. Write a function `membership_attack(predicates,answers,alice,pop_params)` that takes as input a list `predicates` of some d predicates on the public attributes, a list of d (possibly approximate) `answers` to the queries, the data `alice` of a target individual Alice, and list `pop_params` of d population parameters (each in $[0,1]$) and tests whether or not Alice is in the dataset or is a random independent member of the population.
   (a) You should be able to write this function by modifying the code from the class on 2/3 in `membership_attack.ipynb`. We suggest using the Dwork et al. test statistic.
   (b) Set the false positive probability to be $\delta = 1/(20n)$. To determine the corresponding threshold $T = T_{p,a}$, you can approximate the null distribution of your test statistic using the resampling method shown in class on 2/3.

3. Implement defenses to obtain functions `execute_means_round`, `execute_means_noise`, and `execute_means_sample`. To keep the parameters comparable between counts and means, the first defense should round to the nearest multiple of R/n and the second defense should add noise of variance $(\sigma/n)^2$.

4. For increasing values of d starting at $d = 2n$, carry out the membership inference attack using d predicates and estimate the true positive and false positive probabilities by averaging 1000 trials (each time picking Alice to either be a random member of the dataset or a random member of the population). To calculate the vector p of population probabilities, you can either use the Fulton Georgia PUMS dataset that we have provided (`FultonPUMS5full.csv` consisting of 25,766 individuals or the `FultonPUMS5reconstruction.csv` homework dataset) or do an analytic calculation based on the random predicates you use.

5. Make plots of d versus the true positive and false positive probabilities. Confirm that the false positive probabilities remain below δ. Keep increasing d until either the true positive probabilities start to converge or it becomes computationally infeasible.

In your problem set, you will be asked to interpret and briefly discuss the results of both the reconstruction and membership attack experiments, how they relate to each other, and how they relate to the theoretical results we saw in lecture.

A solution to this problem has been posted: `https://github.com/opendp/cs208/blob/main/spring2022/homework/hw2_membership.ipynb`.

5 Theory exercises

Note: the following exercises will not be covered in section. We provide them in these notes as a reference.

Claim 5.1 (A Chernoff-Hoeffding Bound)**.** *For $i = 1, \ldots, k$, let X_i be an independent random variable within $[a, b]$ with mean μ. Then,*

$$\Pr\left[\frac{1}{k}\sum_{i=1}^{k} X_i - \mu \geq t\right] \leq \exp\left(-\frac{2t^2 k}{(b-a)^2}\right).$$

Exercise 5.2. Show that subsampling k out of n rows allows us to estimate m averages each to within $\pm O\left(\frac{1}{\sqrt{k}}\sqrt{\log(m)}\right)$.

Solution. First, we recall the Chernoff bound (above) where we assume the bounds of the random variables (i.e., a, b) are constant.

$$\Pr[\frac{1}{k}\sum_{i=1}^{k}(X_i - \mu) \geq t] \leq e^{-\Omega(2kt^2)}$$

Next, we set $t = \sqrt{\frac{\log(m)}{k}}$. Then, we can re-write the right side of the inequality as

$$\begin{aligned} e^{-2kt^2} &= e^{-2k\left(\sqrt{\frac{\log(m)}{k}}\right)^2} \\ &= e^{-2\log(m)} \quad (\leq e^{-2\ln(m)} \text{ for } m > 1) \\ &\leq \frac{1}{m^2} \end{aligned}$$

Taking the union bound over m queries, we have that that the probability of any sample mean deviating from the true mean by greater than $\sqrt{\frac{\log(m)}{k}}$ is upper bounded by $\frac{1}{m}$. Thus, subsampling k of n rows allows us to estimate m averages each to within $\pm O\left(\frac{1}{\sqrt{k}}\sqrt{\log(m)}\right)$ with high probability.

Exercise 5.3. Show that on average, we can successfully trace $\Omega(\frac{1}{\alpha^2})$ individuals in a dataset at best.

Solution. If we subsample $\Omega(\frac{1}{\alpha^2})$ rows, then the sample mean has a standard deviation of

$$O\left(\frac{1}{\sqrt{1/\alpha^2}}\right) = O\left(\frac{1}{1/\alpha}\right) = O(\alpha)$$

Recall our assumption that the error in answers is bounded by α, $|a_j - \bar{x}_j| < \alpha$, with high probability. Since subsampling $\Omega(\frac{1}{\alpha^2})$ rows maxes out the allowed error to $O(\alpha)$, we are on average only able to trace the individuals included in the subsample.

Exercise 5.4. Show that $\langle y - p, a - p\rangle$ approaches a normal distribution.

Solution. We can express this inner product as follows:

$$\langle y - p, a - p\rangle = \sum_{j=1}^{d}(y_j - p_j) \cdot (a_j - p_j)$$

First, note that p and a are given to the adversary, and our only randomness comes from the sampling of y. From our assumption that the d attributes are independent, we know that each y_j is an independent draw from a Bernoulli distribution with expectation p_j.

Since the d attributes of y are independent samples, we can apply the central limit theorem: if n is sufficiently large, then the distribution of $\langle y - p, a - p\rangle$ is approximately normal, $N(0, \sigma^2)$. (This is because a is a noisy mean of the form $\bar{a}_n = \frac{1}{n}\sum_{i=1}^{n} a_i$.)

The variance σ^2 is equal to $\sum_{j=1}^{d}(a_j - p_j)^2 \cdot p_j \cdot (1 - p_j)$. Note that σ^2 is at most $\frac{d}{4}$, since $p_j = 0.5$ yields the maximum variance of $(p_j)(1 - p_j) = 0.25$ for every j. However, we can get a better hypothesis test by using the actual variance instead of this upper bound $\frac{d}{4}$. This is why using a normal approximation in the homework gives us a $T_{a,p}$ that is better than the T given by the Hoeffding bound.

Exercise 5.5. To make the false positive probability at most δ, show that we can choose $T = O\left(\sqrt{d\log(1/\delta)}\right)$ using Chernoff-Hoeffding bounds.

Solution. We translate the Hoeffding bound for this problem. Note that since $(y_j - p_j) \cdot (a_j - p_j)$ is in $[-1, 1]$, we can replace $(b-a)^2$ with 4.

$$\Pr[\sum_{j=1}^{d} (y_j - p_j) \cdot (a_j - p_j) \geq T] \leq e^{-\Omega\left(\frac{2T^2}{4d}\right)}$$

Then, we simply solve for T in terms of d and δ.

$$\begin{aligned}
e^{-\frac{2T^2}{4d}} &= \delta \\
-\frac{T^2}{2d} &= \ln(\delta) \\
\frac{T^2}{2d} &= -\ln(\delta) = \ln(1/\delta) \\
T^2 &= 2d \ln(1/\delta) \\
T &= \sqrt{2d \ln(1/\delta)} = O\big(\sqrt{d \log(1/\delta)}\big)
\end{aligned}$$