Section 3: Membership Inference Attacks

CS 208 Applied Privacy for Data Science, Spring 2025

February 10, 2025

1 Agenda

- Review membership inference attacks
- Membership inference attack theory exercise
- Membership inference attack coding exercise

2 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 2.1 (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."

3 Exercise: Analyzing MIAs

Let $\mathcal{P}$ be a finite set of N individuals' data, all unique, and let $k \leq n < N$. Consider the subsampling mechanism M that takes a dataset $x = \{x_1, \ldots, x_n\}$ of n individuals $x_i \in \mathcal{P}$ and outputs a random subset $S \subseteq x$ of size k. Let $y \in \mathcal{P}$ be Alice's data, and consider the following Null and Alternative hypotheses.

H_0**:** x is a uniformly random subset of $\mathcal{P} - \{y\}$ of size n.

H_1**:** $x = \{y\} \cup z$, where z is a uniformly random subset of $\mathcal{P} - \{y\}$ of size $n - 1$.

(Note that here we're using sampling from $\mathcal{P}$ without replacement rather than iid samples.)

Consider the following randomized test statistic that takes the mechanism's output $S = M(x)$ and Alice's data y:

$$T(S, y) = \begin{cases} 1 & y \in S \\ U & \text{otherwise} \end{cases}$$

where U is uniformly distributed in $[0, 1]$.

Determine the ROC curve and AUC for the Hypothesis test that rejects H_0 if $T \geq t$, as we vary the threshold $t \in [0, 1]$. Note that if $k \ll n$, the AUC is very close to $1/2$, even though this mechanism is very non-private (publishing k individual's data), as shown by taking $t = 1$.

The Neyman–Pearson Lemma actually implies that the this is the optimal ROC curve for this mechanism. (In class, we elided the fact that the optimality of the Likelihood Ratio test at all FPR requires allowing a randomized rejection rule at the threshold, which we've mimicked above by randomizing the test statistic T.)

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 02/10 lecture as a starting point - you can download the notebook here. However, we will not think of `pub` as known to us as an 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 2/10 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 2/10.

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.

References

[1] Dwork, Cynthia, and Aaron Roth. "The algorithmic foundations of differential privacy." Foundations and Trends in Theoretical Computer Science 9.34 (2014): 211-407.

[2] http://www-bcf.usc.edu/ korolova/teaching/CSCI599Privacy/