Privacy In Machine Learning Homework 1

Kinan Dak Albab

17 October 2018

Collaborater: Rawane Issa

The code and higher resolution plots plus some raw data is available at:
https://github.com/KinanBab/PrivacyInMachineLearning

The repository contains README files and instructions that will guide you through the code, the file structure, and how to run things.

1 Problem 1

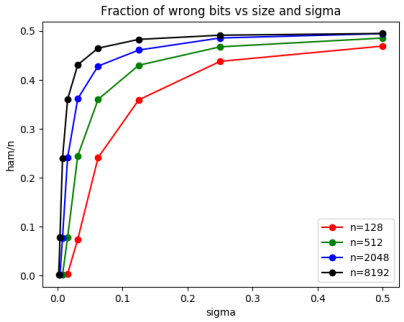
1.1 Discussion

Disclamer For Random Query attacks, the benchmarks I ran stop short of a couple of combinations of very large n and m. The least squares optimization becomes extremely slow for those cases. My Computer has a lot of memory and several cores, if I could parallelize or otherwise optimize that step, I would be able to run the experiments much quicker. In all cases, the experiments did not fail for these combinations, they just took a long time to finish a few runs, so I stopped them as to not waste time.

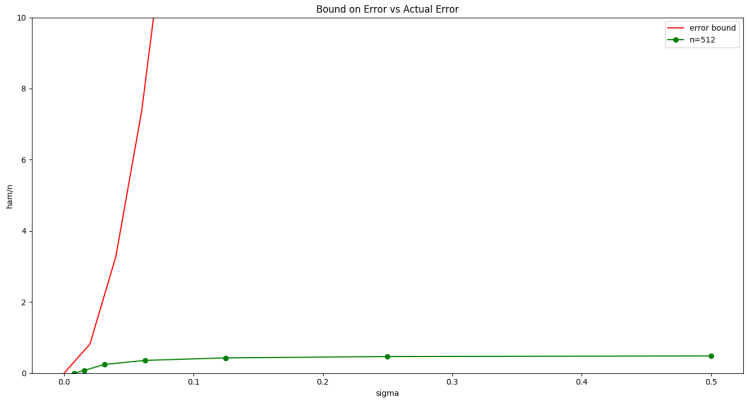
Plots vs Theory The plots seem to confirm the theory. If we treat $\alpha = \sigma$, then we can see that the accuracy of the attack is severely low for α close to half, the noise is very big (as big as rounding) to the extent that we are left guessing randomly, hence the accuracy is close to 0.5.

As α becomes smaller, the accuracy increases near the theoretical limit of $4\sigma^2 n$. I plotted the accuracy vs the theoretical limit for each n attempted in figure 1b. I only show the figure for $N = 512$ for brevity, but similar figures for all n are available (and in higher quality) at:
https://github.com/KinanBab/PrivacyInMachineLearning/tree/master/Homework1/1-Hadmard-Random-Attacks/plots

Comparing the two attacks: The nice thing about the Hadamard load (or the bad thing, depending on how you look at it), is that you can reconstruct with



(a) Overall Accuracy for All n



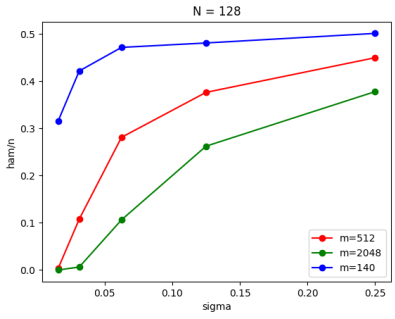
(b) Accuracy for $n = 2048$ Compared To Theoretical Bound

Figure 1: Plots for Hadamard Attack

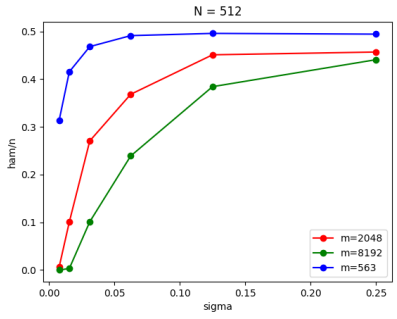
high accuracy with n queries. The runtime for it is also fast, since the attack does not involve optimizations or matrix inversions. The random query attack needs m to be much larger than n to become effective, as shown in the plots, which means the number of queries is large, and the run-time needed to carry out the attack is larger (larger dimensions and more expensive operations).

Choice of m: Random Query attacks perform poorly when m is small compared to n. However, if sigma is very small (close to $\frac{1}{\sqrt{(32n)}}$), the accuracy is improved even for small m.

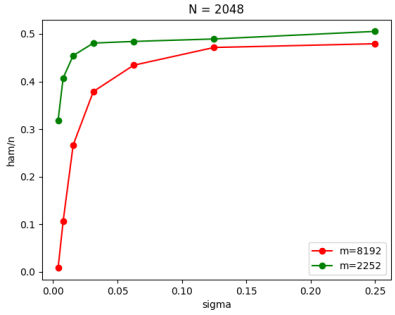
Average Deviation: In All the experiments (both hadamard and random querys), the average deviation is small (¡ 0.02). It decreases very quickly in n and m, but seems independent from σ.



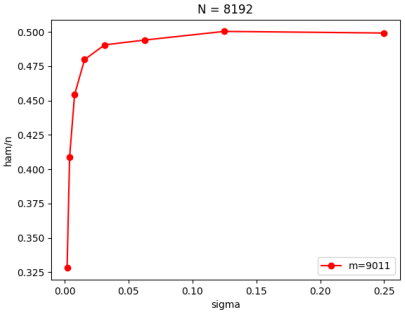
(a) Accuracy for $n = 128$ and several m



(b) Accuracy for $n = 512$ and several m



(c) Accuracy for $n = 2048$ and several m



(d) Accuracy for $n = 8192$ and $m = 1.1n$

Figure 2: Plots for Random Query Attack

2 Problem 2

2.1 Description Of The Attack

The attacker is given $a = (a_1, ..., a_n)$, where $a_i = \sum_{j=1}^{i} a_j + Z_i$, and Z_i are chosen uniformly at random from $\{0, 1\}$.

The attack is very simple:

1. Compute $z = \textit{leftPrefixSum}(a)$.
   Now $z = (a_1, a_1 + a_2, a_1 + a_2 + a_3, ...)$.

2. Compute z' such that $z'_i = z_i - i/2.0$
   Because z is a prefix sum, every entry i contains the sum of all the random Z_j for $j \leq i$. This is very helpful, although any single Z_i can be either $0, 1$ with the same probability (and thus has relatively large variance), the sum of k many such Z_i has low variance, and expected value $k/2$.
   Intuition: With high probability, we have removed the noise from many entries (at least enough for rounding to be able to fix it)

3. Compute b such that $b_i = z_i - z_{i-1}$ and $b_1 = z_1$.
   In effect, this undos step 1, making b equal to the original prefix sums, with much of the noise removed.

4. Let $A = (A_{i,j})$ such that $A_{i,j} = 1$ if $i \leq j$ and 0 otherwise.
   This is a lower triangular matrix that contains only 1s in the lower part, and it match the prefix sum transformation.

5. Efficiently Solve $A\hat{x} = b$ and perform rounding on elements of $\hat{x}$ to get the desired reconstruction.
   This step is very efficient because: (1) A is already in lower triangular form. (2) A has a special form where it contains 1 in all the lower part. A Forward pass can be executed in $O(n)$, by using a running accumulator/counter.

I did not perform detailed analysis on the bounds of the error here, beyond the simple intuition about the variance of sums of many independent uniform binary variables, since I was able to test the accuracy empirically in the experiments. Given more time, I would be interested in carrying out such an analysis.

Disclaimer: I did not create another algorithm/attack for the case where additional information is known about x. I ran out of time, and the attack I have achieved the desired accuracy. I will try to come back to this soon to practice.

2.2 Discussion

In a sense, this is a form of composition, where several queries are being released about a data set as time progresses. The data set itself is "logically" changed,

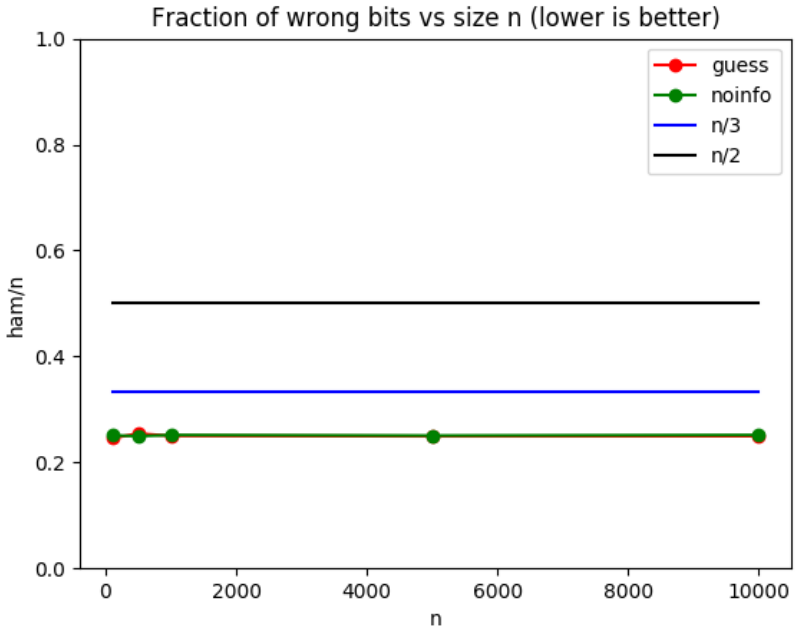


Figure 3: Overall Accuracy for Problem 2 with and without additional info

since new elements $x(i)$ are inserted into the sum with future queries (logically, this is as if such $x(i)$ was 0 in previous query).

It is very tricky to perform something like while preserving privacy without rigorous analysis. Especially to determine (1) the amount/distribution of noise, (2) the dependence/independence of noise between the several steps.

For example, even if the Z_i's were not binary, but instead had some distribution over $\{1, ..., l\}$, the attack can still be carried through, by changing step (2) to remove the new expectation as opposed to $i/2$. This will probably result in some lower accuracy, but the attacker will still be able to reconstruct a decent portion of the data-set.

Instead, it seems like introducing some correlation or dependence between the different Z_i can be a more interesting approach, as that will potentially increase the variance of the sum (e.g. cannot use independent to get the variance reduced to the sum of the individual variances). Utilizing this in addition to (moderatly) increasing the range of the Z_i's could lead to significant improvement in privacy (the cost at utility is unclear to me right now).

3 Problem 3

We have $a_1, ..., a_n$ randomly chosen in $[-1, 1]$,
and $\bar{a} = \frac{1}{n}(a_1 + ... + a_n)$.

Let $s_1, ..., s_k$ be each chosen independently uniformly at random from $\{a_1, ..., a_n\}$,
and $\hat{a} = \frac{1}{k}(s_1 + ... + s_k)$.

we define error as $|\hat{a} - \bar{a}|$.

3.1 Lower bound on k to guarantee small error

We want to find a bound on k, such that for all larger k the following is true:

$$Pr[error \leq \alpha] \geq 1 - \delta$$

Note that this is equivalent to:

$$Pr[|\hat{a} - \bar{a}| > \alpha] \leq \delta \qquad (1)$$

Consider $\mu_{\hat{a}} = E[\hat{a}] = \frac{1}{k}(E[s_1] + ... + E[s_k])$
Since all of $s_1, ..., s_k$ have identitcal distribution, they all have the same expectation.
Furthermore, since $\forall i \in \{1, ..., n\} : s_1 = a_i$ with probability $\frac{1}{n}$, then

$$E[s_1] = \frac{1}{n}(a_1 + ... + a_n) = \bar{a}$$

$$\mu_{\hat{a}} = \bar{a}$$

This means our desired inequality (1) became equivalent to:

$$Pr[|\hat{a} - \mu_{\hat{a}}| > \alpha] \leq \delta \qquad (2)$$

To apply the Chernoff bound, we need a random variable that is the sum of random variables each in $[0, 1]$. Therefore we will define y_i and y as follows:

$$Y_i = \frac{s_i + 1}{2}$$

$$Y = Y_1 + ... + Y_k$$

Since $s_i \in [-1, 1]$, then $y_i \in [0, 1]$ for every i.

Additionally $Y = \frac{1}{2}(s_1 + ... + s_k + k) = \frac{1}{2}(k\hat{a} + k)$
and $\mu_Y = E[Y] = \frac{1}{2}(kE[\hat{a}] + k)$

Now, we will transform the inequality to be on Y:

$$|\hat{a} - \mu_{\hat{a}}|$$

$$= |\hat{a} + 1 - (\mu_{\hat{a}} + 1)$$

$$= |k\hat{a} + k - (k\mu_{\hat{a}} + k)|/k$$

$$= 2|\frac{1}{2}k\hat{a} + k - \frac{1}{2}(k\mu_{\hat{a}} + k)|/k$$

$$= 2|Y - \mu_Y|/k$$

Therefore, (2) is equivalent to::

$$Pr[|Y - \mu_Y| > \frac{k}{2}\frac{\alpha}{\mu_Y}\mu_Y] \leq \delta$$

Note that

$$Pr[|Y - \mu_Y| > \frac{k}{2}\frac{\alpha}{\mu_Y}\mu_Y] \leq Pr[|Y - \mu_Y| \geq \frac{k}{2}\frac{\alpha}{\mu_Y}\mu_Y]$$

$$\leq 2\exp(-\frac{(\frac{k}{2}\frac{\alpha}{\mu_Y})^2}{2 + \frac{k}{2}\frac{\alpha}{\mu_Y}}\mu_Y) \qquad \textit{by chernoff}$$

For a given α, δ, we want to find k such that:

$$2\exp(-\frac{(\frac{k}{2}\frac{\alpha}{\mu_Y})^2}{2 + \frac{k}{2}\frac{\alpha}{\mu_Y}}\mu_Y) \leq \delta$$

$$\frac{(\frac{k}{2}\frac{\alpha}{\mu_Y})^2}{2 + \frac{k}{2}\frac{\alpha}{\mu_Y}}\mu_Y = \frac{k^2\alpha^2}{8\mu_Y + 2k\alpha} \leq k\alpha$$

$$2exp(-k\alpha) \leq 2\exp(-\frac{(\frac{k}{2}\frac{\alpha}{\mu_Y})^2}{2+\frac{k}{2}\frac{\alpha}{\mu_Y}}\mu_Y) \leq \delta$$

$$2exp(-k\alpha) \leq \delta$$

$$\exp(k\alpha) \geq \frac{2}{\delta}$$

$$k \geq \frac{1}{\alpha}\ log(\frac{2}{\delta}) = \frac{log(2) - log(\delta)}{\alpha}$$

Side note: Alternatively, we could have computed $Var(\hat{x}) = \frac{1}{k^2}kVar(s_i) \leq \frac{1}{k}$, and then utilized Chebyshev's inequality to get $k \geq \alpha^2\delta$.

3.2 Union Bound For Multiple Queries

The setting is the same as before, except that of estimating $\frac{1}{n}\sum_{i=1}^{n} x_i$, we are trying to estimate $\frac{1}{n}\sum_{i=1}^{n} f_j(x_i)$ for several functions $f_1, ..., f_d$.

Let $X_j = \frac{1}{n}\sum_{i=1}^{n} f_j(x_i)$, and let $Y_j = \frac{1}{k}\sum_{i=1}^{k} f_j(s_i)$

Note the following:

$$\mu_{Yj} = E[Y_j] = \frac{1}{k}\sum_{i=1}^{k} E[f_j(s_i)] = \frac{1}{k}\sum_{i=1}^{k}\frac{1}{n}\sum_{m=1}^{n} f_j(x_m)$$

$$= \frac{1}{n}\sum_{m=1}^{n} f_j(x_m) = X_j$$

We want to find k for which the following holds for any given α, δ:

$$\forall j : Pr[|Y_j - X_j| \geq \alpha] < \delta$$

This is implied by the following stronger statement:

$$Pr[\bigcup_{j=1}^{d}(|Y_j - X_j| \geq \alpha)] < \delta$$

By the union bound, we know:

$$Pr[\bigcup_{j=1}^{d}(|Y_j - X_j| \geq \alpha)] \leq \sum_{j=1}^{d} Pr[|Y_j - X_j| \geq \alpha]$$

$$= \sum_{j=1}^{d} Pr[|kY_j - k\mu_{Yj}| \geq \frac{\alpha}{\mu_{Yj}}k\mu_{Yj}]$$

Now we can apply the chernoff bound, since kY_j is the sum of $f_j(s_i)$ each in $\{0,1\}$, and since $k\mu_{Yj}$ is the mean of kY_j.

$$Pr[\bigcup_{j=1}^{d}(|Y_j - X_j| \geq \alpha)] \leq \sum_{j=1}^{d}(2\exp(-\frac{(\frac{\alpha}{\mu_{Yj}})^2}{2+\frac{\alpha}{\mu_{Yj}}}k\mu Yj))$$

$$\leq \sum_{j=1}^{d} 2\exp(-\frac{\alpha^2 k}{2\mu_{Yj}+\alpha})$$

Now, what we want is to find k such that:

$$\sum_{j=1}^{d} 2\exp(-\frac{\alpha^2 k}{2\mu_{Yj}+\alpha}) \leq \delta$$

$$\sum_{j=1}^{d} \exp(-\frac{\alpha^2 k}{2\mu_{Yj}+\alpha}) \leq \frac{\delta}{2}$$

Choose $j \in \{1,...,d\}$ that minimizes $\exp(-\frac{\alpha^2 k}{2\mu_{Yj}+\alpha})$, we get:

$$d \times \exp(-\frac{\alpha^2 k}{2\mu_{Yj}+\alpha}) \leq \frac{\delta}{2}$$

$$\frac{2d}{\delta} \leq \exp(\frac{\alpha^2 k}{2\mu_{Yj}+\alpha}) \leq \exp(\alpha^2 k)$$

$$\alpha^2 k \geq \log(\frac{2d}{\delta})$$

$$k \geq \frac{\log(d)+\log(2)-\log(\delta)}{\alpha^2}$$

3.3 Reconstruction Difficult When Answers Based on a Sample

Now we consider input data $x = x(1),...,x(n)$ each bit.

We have a mechanism that chooses sample $y = y_1,...,y_k$ where $k = \frac{n}{3}$ without replacement, and uses sample to answer query f producing output $z = f(y)$.

An attacker sees the result of the queries and feeds it to a reconstruction algorithm A of her choosing to produce $\hat{x} = A(z) = A(f(y))$.

note: this is reasonable since we have shown that answering queries (averages of binary functions) about dataset is well approximated by answering these queries on random samples. We did that assuming replacement, while here there is no

replacement, although it probably does not change the bounds by much.

We define the error as

$$E = \frac{Hamming(\hat{x}, x)}{n}$$

We want to show:

$$Pr[E \geq \frac{1}{4}] \geq 1 - \exp(\Omega(n))$$

Alternatively, that is equivalent to:

$$Pr[E < \frac{1}{4}] \leq \exp(\Omega(n)) \qquad (1)$$

$\forall i \in \{1, ..., n\}$ define $I_i = x(i) \oplus \hat{x}(i)$ where $\oplus$ stands for XOR.
In other words, I_i is 1 if the attacker guess $x(i)$ incorrectly, and 0 otherwise.
Let $I = I_1 + ... + I_n$.
Note that $E = \frac{1}{n} \times I$

All that all entries of x are chosen uniformly and independently, which means that for any two distinct entries of x, these entries (and any function of one of them) are independent. In particular, for any $x(i)$ such that $x(i) \notin y$, we have $\hat{x}(i) = g(y)$ and $x(i)$ independent, where $g = A \circ f$.

Therefore, for all $x(i) \notin y$, $x(i) \oplus \hat{x}(i)$ is uniform since $x(i)$ is uniform and independent, which implies that $E[I_i] = \frac{1}{2}$. For $x(j) \in y$, we cannot immediately say that XOR is uniform due to dependence. However, we know that $0 \geq E[I_j] \leq 1$.

We know that $\mu_I = E[I] = E[I_1 + ... + I_n] \geq \frac{2n}{3}\frac{1}{2} + 0 = \frac{n}{3}$

Now we can go back to (1):

$$Pr[E < \frac{1}{4}] \leq Pr[E \leq \frac{1}{4}] = Pr[nE \leq \frac{n}{4}] = Pr[I \leq \frac{n}{4}] \qquad (2)$$

Let $\epsilon = 1 - \frac{n}{4\mu_I}$ so that $(1 - \epsilon)\mu_I = \frac{n}{4}$. Plug ϵ into (2):

$$\leq Pr[I \leq \frac{n}{4}] = Pr[I < (1 - \epsilon)\mu_I]$$

$$\leq \exp(-\frac{\epsilon^2}{2}\mu_I) \quad \textit{By Chernoff Bound 2}$$

$$= \exp(-\frac{(1 - \frac{n}{4\mu_I})^2}{2}\mu_I)$$

$$= \exp(-(\frac{1}{2} - \frac{n}{4\mu_I} + \frac{n^2}{16\mu_I^2})\mu_I)$$

$$\leq \exp(-(\frac{1}{2} - \frac{3}{4} + \frac{9}{16})\frac{n}{3})$$

$$= \exp(-\frac{5}{16}\frac{n}{3}) = \exp(-\frac{5}{48}n)$$

Therefore,

$$Pr[E \geq \frac{1}{4}] \geq 1 - \exp(\Omega(n))$$