Privacy Enhancing Technologies **Fall 2024**

Problem Set 4 — 18/12/2024 update 7

Due: Fri, Dec 20 at 11:59pm CET (submit via Gradescope)

Instructions: You **must** typeset your solution in LaTeX using the provided template:

`https://spylab.ai/teaching/pets-f24/hws/template.tex`

Submission Instructions: You must submit your problem set via Gradescope. Please use the course code provided in Moodle to sign up. Note that Gradescope requires that the solution to each problem starts on a **new page**.

Bugs: We make mistakes! If it looks like there might be a mistake in the statement of a problem, please ask a clarifying question on Moodle.

Problem 1: Conceptual Questions [8 points]. For each of the following statements, say whether it is TRUE or FALSE. **Justify your answer in one sentence.**

(a) Which of the following mechanisms are 1-DP (i.e., $\epsilon = 1$) for an input database $D = \{x_1, \ldots, x_n\} \in \{0,1\}^n$ of size $n \geq 1$? [6 points]

i. Release $\mathsf{Enc}(D)$, where Enc is a one-time pad encryption scheme (i.e., $\mathsf{Enc}(D)$ samples a mask r uniformly random from $\{0,1\}^n$ and returns $D \oplus r$).

ii. Release $\mathsf{Enc}(D)$, where Enc is an encryption scheme (with a fixed key) that is semantically secure for polynomial-time adversaries (e.g., AES).

iii. Let A be a 1-DP algorithm for computing the mean of a database. Let $z \leftarrow A(D)$. Release a denoised version $z' = (z + z_{\text{denoise}})/2$, where $z_{\text{denoise}} = \frac{1}{n}\sum_{i=1}^{n} x_i$.

iv. Release $f(D)$, where f is a function that is $\frac{1}{2}$-DP.

v. To release the sum of elements in D, first sample independent $\texttt{Laplace}(\mu = 0, b = 2)$ noise Y_1 and Y_2, compute $v_0 = \sum_{i=1}^{\lfloor n/2 \rfloor} x_i + Y_1$ and $v_1 = \sum_{i=\lfloor n/2 \rfloor + 1}^{n} x_i + Y_2$, and output $v_0 + v_1$.

vi. Release $\frac{1}{n}\sum_{i=1}^{n} x_i + \mathtt{U}[-1,1]$ where $\mathtt{U}[-1,1]$ is the uniform distribution over $[-1,1]$.

(b) Let $f\colon \mathcal{X} \to \mathbb{R}^k$ be a trained neural network that, given an input $x \in \mathcal{X}$, outputs a vector $f(x) \in \mathbb{R}^k$. In a *model inversion attack*, the attacker is given $f(x)$ for some unknown input $x \in \mathcal{X}$ and tries to recover x. This attack is prevented if f is trained with DP-SGD for some ϵ and δ, i.e., the probability that the attacker's guess x' is correct is $\Pr[x = x'] \leq \epsilon/|\mathcal{X}| + \delta$. [2 points]

Problem 2: Additive DP [4 points]. Let $M(D)$ be an algorithm that is $(0,\delta)$-DP for a database $D \in \mathcal{X}^n$ where $\mathcal{X}$ is some finite set. Show that:

(a) If $\delta \geq \frac{1}{2n}$, then M is not private. Specifically, give an example of a mechanism that is $(0,\delta)$-DP but leaks one individual's information $x_i \in \mathcal{X}$ in the database (for some arbitrary index i) with constant (non-zero) probability. [2 points]

(b) If $\delta < \frac{1}{2n}$, then M is not *useful*. Specifically, show that there exists a database $D \in \{0,1\}^n$, such that $M(D)$ incurs error $E = \Omega(1)$ (with constant, non-zero probability) when computing the mean of elements of D. [2 points]

Problem 3: Cheating on ML competitions [12 points]. Consider a typical ML competition, where participants submit their models and they are scored on a private test set. The model with the best score on the test set wins. Of course, in practice we typically don't care about the score on the specific test set, but rather the score on the actual underlying data distribution. So we want to make sure that participants don't overfit to the test set. Unfortunately, this is difficult if participants can submit many models, especially if they do so *adaptively*.

The setting is as follows. There is a data distribution $\mathcal{P}$ over some labeled space $\mathcal{X} \times \{-1,1\}$. The test set $D = \{(x_i, y_i)\}_{i=1}^n$ consists of labeled examples sampled iid from $\mathcal{P}$. A model is a function $\theta : \mathcal{X} \to \{-1,1\}$. The model's 0-1 loss (or *risk*) on the population is

$$\mathcal{L}(\theta) = \mathop{\mathbb{E}}_{(x,y)\sim\mathcal{P}} \left[\mathbb{1}\{\theta(x) \neq y\} \right] .$$

The model's score on the test set is

$$\mathcal{L}_D(\theta) = \frac{1}{n} \sum_{i=1}^{n} \mathbb{1}\{\theta(x_i) \neq y_i\} .$$

Note: this problem is quite challenging! But each sub-question is independent of the others, so don't worry if you get stuck on one.

(a) Let $\theta_1, \ldots, \theta_k$ be a set of k models chosen *non-adaptively* (that is, no model θ_i depends on the others or on their score on the test set). Show that, for n large enough, with probability at least 0.5 (over the choice of D), the generalization error satisfies:

$$\max_i |\mathcal{L}_D(\theta_i) - \mathcal{L}(\theta_i)| \leq O\left(\sqrt{\log(k)/n}\right) .$$

Hint: Use Hoeffding's inequality [2 points]

(b) The above result depends crucially on the fact that the models are chosen non-adaptively (and so, in particular, a model's performance on some test sample (x_i, y_i) is independent of the other test samples). Let's see what goes wrong if we choose the models adaptively.

Let $\theta_1, \ldots, \theta_k$ be fully *random* functions, for $k \leq n$. We submit all these models to the competition, each getting a test score $\mathcal{L}_D(\theta_j)$. Now, pick the set of models S that achieve a score of at most $\frac{1}{2} - 1/\sqrt{n}$ on the test set (you can assume that $|S| = \Omega(k)$ with constant probability). We build a new model *adaptively* by taking a majority vote of these good models:

$$\theta_{\text{maj}}(x) = \text{maj}_{j\in S} \theta_j(x) .$$

Show that the generalization error of θ_{maj} can be as large as $\Omega(\sqrt{k/n})$.

Note that here you are asked to prove a *lower bound*. So it suffices to show that there exists *some* distribution $\mathcal{P}$ over $\mathcal{X} \times \{-1,1\}$ such that, with constant probability over the choice of D and the θ_j,

$$\left| \mathcal{L}_D(\theta_{\text{maj}}) - \mathcal{L}(\theta_{\text{maj}}) \right| \geq \Omega(\sqrt{k/n}) .$$

A standard anti-concentration result for the Binomial distribution (see Theorem 2) may be useful. [3 points]

(c) A naive way to prevent overfitting to the test set is to use a different test set for each submitted model. Split the test set D into k disjoint sets $D_1, \ldots, D_k$, each of size n/k. For a set of k models $\theta_1, \ldots, \theta_k$ chosen *non-adaptively*, show that with constant probability:

$$\max_i \left| \mathcal{L}_{D_i}(\theta_i) - \mathcal{L}(\theta_i) \right| \leq O\left(\sqrt{k \log(k)/n}\right) .$$

Hint: Use the result of part a) [2 points]

(d) We will now show that a differentially private algorithm can prevent overfitting!

Let $M(D)$ be a differentially private algorithm that takes a dataset D as input and outputs a function θ. The algorithm M satisfies two properties:

(a) Privacy: M is (ϵ, δ)-DP.

(b) Accuracy: If D is a dataset of n iid samples from $\mathcal{P}$, then the model θ output by $M(D)$ has low error on D, i.e., $\mathcal{L}_D(\theta) \leq \alpha$.

Show that the function θ that $M(D)$ outputs has low test error, i.e., $\mathcal{L}(\theta) \leq e^{\epsilon}\alpha + \delta$. You can assume here that the probability in $\mathcal{L}(\theta)$ is also taken over the randomness of the mechanism M. [3 points]

(e) Here's a slightly stronger version of the result above, which we won't prove.

> **Theorem 1.** *Consider the mechanism M that scores k (possibly adaptive) models using the Gaussian mechanism, by adding some noise $\mathcal{N}(0, \sigma^2)$ to the test score $\mathcal{L}_D(\theta_i)$. Denote this noisy score by $\widetilde{\mathcal{L}}_D(\theta_i)$.*
> *Assume the noise level is set so that the mechanism M satisfies:*
>
> *(a) Privacy: M is (ϵ, δ)-DP.*
>
> *(b) Accuracy: $\max_i |\widetilde{\mathcal{L}}_D(\theta_i) - \mathcal{L}_D(\theta_i)| \leq \alpha$ holds with probability 1-o(1)*
>
> *Then these models generalize:*
>
> $$\max_i |\widetilde{\mathcal{L}}_D(\theta_i) - \mathcal{L}(\theta_i)| \leq \alpha + \epsilon\,.$$

Show that by setting the noise level of the Gaussian mechanism appropriately, we can get generalization error of order $\widetilde{O}(\sqrt[4]{k}/\sqrt{n})$ with constant probability. (you can ignore the δ term in the DP guarantee for simplicity. That is, you don't need to show precisely how the generalization error depends on δ, as long as the error is of order $\widetilde{O}(\sqrt[4]{k}/\sqrt{n})$). [2 points]

Problem 4: DP-SGD with Gradient Compression [12 points]. Recall that, in DP-SGD, we add noise to the batch gradient $\bar{g} = \frac{1}{m}\sum_{i=1}^{m} g_i$. For simplicity, we'll ignore clipping in this problem and just assume that all individual gradients have norms below the clipping threshold $C > 0$, i.e., $\|g_i\| \leq C$ for all i. Then, DP-SGD calculates a noisy gradient as $\bar{g}_{\text{noisy}} = \bar{g} + Z^{(d)}$, where $Z^{(d)} \sim \mathcal{N}(0, \sigma^2 I_d)$ with $\sigma^2 = 2\log(1.25/\delta)/m^2 \cdot \frac{4C^2}{\epsilon^2}$.

The expected squared error of this noisy gradient is

$$\mathbb{E}\left[\|\bar{g}_{\text{noisy}} - \bar{g}\|^2\right] = \mathbb{E}\left[\|Z^{(d)}\|^2\right] = d\sigma^2\,.$$

This error grows with the dimension d, which can be very large for deep learning models. In this problem, we will try to obtain a lower error by *compressing* gradients into $\mathbb{R}^p$ for $p \ll d$ before adding noise.

We do this by computing Pg_i for some (public and data-independent) matrix $P \in \mathbb{R}^{p\times d}$ with orthonormal rows.[1] In particular, $(P^\top P)^2 = P^\top P$, $\|P^\top x\| = \|x\|$ for all $x \in \mathbb{R}^p$, and $\|Px\| \leq \|x\|$ for all $x \in \mathbb{R}^d$. Finally, let $r(x) = P^\top P \cdot x - x$ denote the *residual* of projecting some $x \in \mathbb{R}^d$ to the p-dimensional subspace.

Consider the following variant of DP-SGD:

1. Compress individual gradients g_i to the smaller space.

[1] You can assume that you are given such a P throughout this problem.

2. Calculate a noisy mean of the compressed gradients.

3. Reconstruct a noisy gradient in the original space.

Hence, the noisy gradient is

$$\bar{g}^{\text{proj}}_{\text{noisy}} = P^\top \left(\frac{1}{m} \sum_{i=1}^{m} P g_i + Z^{(p)} \right) \quad \text{for} \quad Z^{(p)} \sim \mathcal{N}(0, \sigma^2 I_p) \,.$$

(a) Assume that the residual of the batch gradient is zero, i.e., $r(\bar{g}) = 0$. Show that the expected squared error of this DP-SGD variant satisfies

$$\mathbb{E}\left[\| \bar{g}^{\text{proj}}_{\text{noisy}} - \bar{g} \|^2 \right] = p\sigma^2 \,.$$

[2 points]

(b) Argue that this variant satisfies the same privacy guarantee as the original DP-SGD algorithm. That is, assuming that the original noisy gradient $\bar{g}_{\text{noisy}}$ is (ϵ, δ)-DP, argue that $\bar{g}^{\text{proj}}_{\text{noisy}}$ is also (ϵ, δ)-DP. [2 points]

(c) Show that if $r(\bar{g}) \neq 0$, then the expected gradient is biased, i.e.,

$$\mathbb{E}\left[\bar{g}^{\text{proj}}_{\text{noisy}} \right] \neq \bar{g} \,.$$

[1 point]

(d) Assume that the residual error of each individual gradient is small, i.e., $\|r(g_i)\| \leq \|g_i\|/10$ and that we compress gradients from dimension d to dimension $p = d/100$.

Construct an algorithm that produces a gradient G with the following properties, and show that those properties hold:

- The algorithm is $(\epsilon, 2\delta)$-DP.
- The algorithm is *unbiased*, i.e., $\mathbb{E}[G] = \bar{g}$.
- The expected squared error satisfies $\mathbb{E}[\|G - \bar{g}\|^2] \leq 8\sigma^2 p$.

Here, σ, ϵ, δ are the same as in the original DP-SGD algorithm. [7 points]

Problem 5: Feedback [0 points]. Please answer the following questions to help us design future problem sets. You are not required to answer these questions, and if you would prefer to answer anonymously, please use this form. However, we do encourage you to provide us feedback on how to improve the course experience.

(a) Roughly how long did you spend on this problem set?

(b) What was your favorite problem on this problem set?

(c) What was your least favorite problem on this problem set?

(d) Any other feedback for this problem set?

A Anti-concentration of the Binomial Distribution

Theorem 2. *Let* $0 < \epsilon \leq 1/\sqrt{m}$. *Then,*[2]

$$\Pr\left[\texttt{Binomial}\left(m, \frac{1}{2}+\epsilon\right) > m/2\right] > \frac{1}{2} + \Omega(\sqrt{m}\epsilon)\,.$$

[2]There is actually a factor $O(1/\sqrt{m})$ missing in this bound but we can ignore it for simplicity.