CS839: Differential Privacy and Learning Mar 19, 2026

18: FriendlyCore for Gaussian Mean Estimation

Instructor: Gavin Brown *Scribe:* Angus He

Disclaimer: This document is intended as an informal supplement to in-class note-taking. It has not been given the level of scrutiny expected in polished lecture notes, let alone that reserved for peer-reviewed publications.

Last lecture, we started talking about the *FriendlyCore* framework of [1]. We introduced FriendlyCore in the context of subsample-and-aggregate: given a dataset x and a non-private algorithm A, we split the data into k disjoint chunks $x^{(1)}, \ldots, x^{(k)}$ and produce values $A(x^{(1)}), \ldots, A(x^{(k)})$. We can use FriendlyCore to estimate the mean of these.

In many settings, some version of the central limit theorem says that the distribution of $A(x^{(j)})$ should be approximately a Gaussian. In this lecture we'll simply analyze what happens when FriendlyCore receives inputs $x_1, \ldots, x_n$ drawn i.i.d. from a Gaussian distribution, as this is of standalone interest, but you can keep the subsample-and-aggregate framing in the back of your mind if you want.

1 BasicFilter and FriendlyCore Recap

We have introduced the BasicFilter and FriendlyCore Algorithm from the previous lecture. The BasicFilter Algorithm performs consistency check on each data point and assigns it a weight based on how "consistent" it is with the rest of the dataset. The intuition of this algorithm is that it assigns each data point a weight based on how many other points that are close to x_i: Points that are close to all others get weight 1, points agreeing with at most half get 0, and points in between receive a linearly scaled weight between 0 and 1.

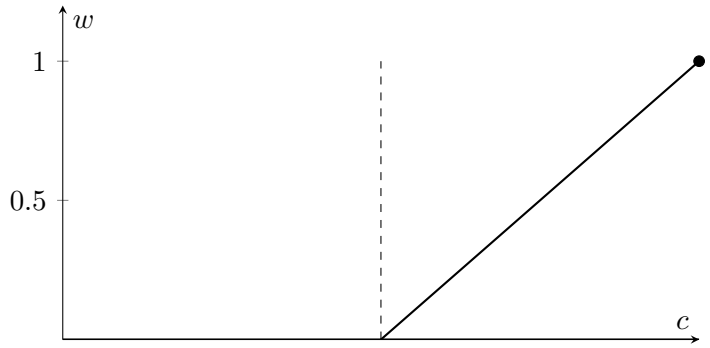
1.1 Basic Filter

Algorithm 1 Basic Filter

Input: Data $x_1, \ldots, x_n$, predicate $f : X \times X \to \{0, 1\}$
Returns: Weights $\vec{w} = (w_1, w_2, \ldots, w_n) \in [0, 1]^n$

1: **for** $i = 1$ to n **do**
2: $c_i \leftarrow \sum_{j=1}^{n} f(x_i, x_j)$
3: $w_i \leftarrow \begin{cases} 1 & \text{if } c_i = n \\ 0 & \text{if } c_i \leq \frac{n}{2} \\ \frac{c_i - \frac{n}{2}}{n} & \text{otherwise} \end{cases}$
4: Return $\vec{w}$
5: **end for**

The following plot illustrates how the output weight varies as a function of the count:



In short, BasicFilter outputs a weighted dataset which allows it to make "soft decisions" rather than using a strict include/exclude rule.

2 Friendly Core

FriendlyCore [1] is a versatile DP aggregator. It acts as a building block for DP estimation. The core idea is to take messy real-world data and turn it into something you can safely aggregate with strong privacy guarantees. We will present the framework as it is applied to Gaussian mean estimation, which is a clean theoretical assumption, but it can be applied much more widely.

Theorem 2.1. *There is an (ε, δ)-DP algorithm such that, on n i.i.d. samples from $\mathcal{N}(\mu, \mathbb{I})$, with probabity $\frac{99}{100}$ returns $\tilde{\mu}$ with $\|\tilde{\mu} - \mu\|_2 \leq \alpha$ as long as*

$$n \gtrsim \frac{d}{\alpha^2} + \frac{d\sqrt{\log\left(\frac{1}{\delta}\right)}}{\alpha\varepsilon} + \frac{\log\left(\frac{1}{\delta}\right)}{\varepsilon}$$

Remark 2.2. Some remarks of this algorithm includes:

- Known lower bounds say this task requires

  $$n \gtrsim \frac{d}{\alpha^2} + \frac{d}{\alpha\varepsilon} + \frac{\log\left(\frac{1}{\delta}\right)}{\varepsilon}.$$

  Thus, this algorithm is nearly optimal up to logarithmic factors
- There are other algorithms with similar or better accuracy, this is not the only way to achieve this result. However, as we've discussed, the approach has lots of nice ideas and applies to other problems.
- Efficiency: The algorithm we'll see today takes $O(n^2 d)$ time. A slight modification makes it nearly linear time, i.e. approximately $\tilde{O}(nd)$, ignoring logarithmic factors and the privacy parameters.
- No prior on μ: Unlike many algorithms mentioned previously in the class, this method does not require prior knowledge or bounds on the location of the true mean μ.

3 Friendly Core on Gaussian mean estimation

3.1 Defining Friendliness

In FriendlyAverage (Algorithm 2), we define a predicate that determines whether two points are friendly. Two points are considered friendly if their distance is below a specified threshold. In the

Gaussian setting, this can be defined using a norm-based condition, for example:

$$\|x_i - x_j\|_2 \leq 10\sqrt{d}.$$

The number 10 isn't critical, we want a threshold where typical Gaussian data points are below it.[1]

3.2 Friendly Average

Algorithm 2 Friendly Average

Input: Data $x_1, \ldots, x_n \in \mathbb{R}^d$, $\varepsilon, \delta \in (0,1)$
Returns: Output: $\tilde{\mu} \in \mathbb{R}^d$ or $\perp$

predicate $\text{dist}(x,y) = \mathbb{1}\left\{\|x-y\|_2 \leq 10\sqrt{d}\right\}$
$\vec{w} \leftarrow \text{BasicFilter}(x, \text{dist})$
if $\|\vec{w}\|_1 + \text{Lap}\left(\frac{3}{\varepsilon}\right) \leq n - \frac{2\log\left(\frac{1}{\delta}\right)}{\varepsilon}$ **then**
    return $\perp$
else
    return $\frac{1}{n}\sum_{i=1}^n w_i x_i + \mathcal{N}\left(0, \frac{100d}{n^2}\sigma^2_{\varepsilon,\delta}\mathbb{I}\right)$
end if
Return $\vec{w}$

Intuitively, the FriendlyAverage algorithm proceeds in three steps

1. Apply BasicFilter to assign a weight $w_i \in [0,1]$ to each data point based on how many friendly neighbors it has.
2. Compute the total weight $\|w\|_1 = \sum_i w_i$, and add Laplace noise to ensure differential privacy.
3. If the noisy total weight is too small, the algorithm fails. Otherwise, proceed to the next step.
4. Compute the weighted mean and add Gaussian noise to ensure differential privacy.

Remark 3.1. Some intuitions about this algorithm:

- The filtering step removes or down-weights outliers.
- The stability check ensures there is sufficient reliable data (we'll talk about this more below).
- The final noise addition ensures differential privacy.

3.3 Accuracy Analysis

We analyze the accuracy of the algorithm in the friendly setting, where the algorithm does not return $\perp$, which happens with high probability. In other words, when every point in the input data is friends with every other point, then nothing is removed and $\vec{w} = \vec{1}$. The following property of BasicFilter is easy to verify.

Lemma 3.2. *(Completeness) If all x_i, x_j satisfy $f(x_i, x_j) = 1$, then $\vec{w} = \vec{1}$.*

[1] To make everything formal, you would need to apply a tail bound for the norms of Gaussian random variables and then a union bound over all n examples in the data; this is not the focus of our presentation. Here's one version of the relevant claim (Laurent–Massart): if $z \sim \mathcal{N}(0, \mathbb{I}_d)$, then with probability at least $1-\beta$ we have $\|z\|_2 \leq \sqrt{d} + \sqrt{2\log(1/\beta)}$.

Then, with high probability:

$$\hat{\mu} + N(0, \frac{d}{n^2}\delta_{\varepsilon,\delta}^2 \mathbb{I}_d)$$

We know the noisy mean is usually close to the empirical mean, and the empirical mean is usually close to the true mean, so

$$\|\mu - \tilde{\mu}\|_2 \leq \|\mu - \hat{\mu}\|_2 + \|\hat{\mu} + \tilde{\mu}\|_2 \leq \sqrt{\frac{d}{n}} + \frac{\sqrt{d}}{n}\delta_{\varepsilon,\delta} \cdot \|N(0, \mathbb{I})\|_2$$

Therefore,

$$\|\mu - \tilde{\mu}\|_2 \lesssim \frac{\sqrt{d}}{n} + \frac{d}{n}\frac{\sqrt{\log\left(\frac{1}{\delta}\right)}}{\varepsilon}$$

And to ensure that the private check on $\|\vec{w}\|_1$ passes with high probability, we require

$$n \gg \frac{\log(1/\delta)}{\varepsilon}.$$

Rearranging yields the sample complexity in the theorem.

3.4 Prove that Friendly Average is DP

Proving privacy is the hard part here. In order to prove that FriendlyAverage is DP, we need to make sure that

1. Changing one datapoint cannot change the ℓ_1 norm of $\vec{w}$ by much.
2. The empirical mean cannot be changed too much.

If that's the case, FriendlyAverage will be differentially private.

Remark 3.3. The high level idea is that since this is the Gaussian Mechanism, we can try proving that on any two adjacent datasets, the weighted empirical means will be close together. That is, we need to show that $\forall x \sim x'$,

$$\|w'\|_1 - \|w\|_1 = \mathcal{O}(1)$$

and that

$$\left\| \frac{1}{n}\sum_{i=1}^{n} w_i x_i - \frac{1}{n}\sum_{i=1}^{n} w_i' x_i' \right\|_2 = \mathcal{O}\left(\frac{\sqrt{d}}{n}\right)$$

First, let's focus on the added Laplacian noise. This is the crucial part of the privacy proof: changing one point cannot change the weights very much.

Claim 3.4. *(Stability)* $\forall x \sim x'$, $\|\vec{w} - \vec{w'}\|_1 \leq 3$, *and as a consequence* $|\|\vec{w}\|_1 - \|\vec{w'}\|_1| \leq 3$.

Proof. Assume without loss of generality that $x_1 \neq x_1'$. Then we have

$$\begin{aligned}\|w - w'\|_1 = \sum_{i=1}^{n} |w_i - w_i'| &= |w_1 - w_1'| + \sum_{i=2}^{n} |w_i - w_i'| \\ &\leq 1 + \sum_{i=2}^{n} |w_i - w_i'|.\end{aligned}$$

Now consider a worst-case scenario where a data point with many "friends" is replaced by one with no friends. When this happens, every individual who was previously connected to that point loses a friend. Thus, removing a highly connected point can affect many other individuals. If an individual gains or loses a friend, their count changes by at most 1, but you can change the weight by $\frac{2}{n}$. Thus,

$$1 + \sum_{i=2}^{n} |w_i - w_i'| \le 1 + \sum_{i=2}^{n} \frac{2}{n} \le 3$$

This establishes the first part of the lemma. For the second, note that all the entries of w, w' are nonnegative so we have

$$\begin{aligned} |\|w\|_1 - \|w'\|_1| &= |\sum_i |w_i| - \sum_i |w_i'|| \\ &= |\sum_i w_i - \sum_i w_i'| \\ &= |\sum_i w_i - w_i'|. \end{aligned}$$

By the triangle inequality, this is at most $\sum_i |w_i - w_i'| = \|w - w'\|_1$. □

We need one more property of the filtering algorithm, which is that it lets no outliers pass through.

Lemma 3.5. *(Soundness) If $w_i, w_j > 0$, then*

$$\|x_i - x_j\|_2 \le 20\sqrt{d}$$

Proof. From inspecting BasicFilter, we can see that any point which receives nonzero weight has more than $n/2$ friends. Thus any two points with nonzero weight have at least one friend in common. Call this point z. Applying the triangle inequality, we have

$$\|x_i - x_j\|_2 \le \|x_i - z\|_2 + \|z - x_j\|_2 \le 20\sqrt{d}.$$

□

As with many of the proofs today, note that this is quite generic: we only used the triangle inequality, nothing specific to Gaussian mean estimation.

Then, let's first find the approximate upper bound of $\left\|\frac{1}{n}\sum_i w_i x_i - \frac{1}{n}\sum_i w_i' x_i'\right\|_2$.

Claim 3.6. *Suppose $w, w' \neq \vec{0}$. Then*

$$\left\|\frac{1}{n}\sum_i w_i x_i - \frac{1}{n}\sum_i w_i' x_i'\right\|_2 \lesssim \frac{\sqrt{d}}{n}$$

Proof. We can shift all x_i so that they are all bounded in a ball whose center is $\vec{0}$ and the radius is $10\sqrt{d}$ (from the soundness property). Then, suppose $x_1 \neq x_1'$, then we have

$$\left\|\frac{1}{n}\sum_i w_i x_i - \frac{1}{n}\sum_i w_i' x_i'\right\|_2 = \frac{1}{n}\left\|w_1 x_1 - w_1' x_1' + \sum_{i=2}^{n} w_i' x_i'\right\|_2 \le \frac{1}{n}\left(20\sqrt{d} + \left\|\sum_{i=2}^{n}(w_i - w_i')x_i\right\|_2\right)$$

Then, by the triangle inequality, we get

$$\left\|\sum_{i=2}^{n}(w_i - w_i')x_i\right\|_2 \leq \sum_{i=2}^{n}\left|w_i - w_i'\right| \cdot \|x_i\|_2 \leq 20\sqrt{d} \cdot \left\|w - w'\right\|_1 = \mathcal{O}(\sqrt{d'})$$

□

But this is a little bit messy because weights and points may change. Thus, we reinterpret this as a difference of means of distributions. The trick is that we can think of P as a distribution of putting mass $\frac{w_i}{n}$ on x_i and Q as the distribition of putting mass $\frac{w_i'}{n}$ on x_i'. Then,

$$\mu(P) = \frac{1}{n}\sum_i w_i x_i$$

$$\mu(Q) = \frac{1}{n}\sum_i w_i' x_i$$

So another way to view the problem is bounding $\|\mu(P) - \mu(Q)\|_2$.

Claim 3.7. *If P and Q are distributions whose joint subset has diameter at most $2R$, then*

$$\|\mu(P) - \mu(Q)\|_2 \leq 2 \cdot R \cdot TV(P, Q)$$

Proof. Without loss of generality, assume

$$\text{supp}(P), \text{supp}(Q) \subseteq \text{Ball}(0, R)$$

Then

$$\begin{aligned}\|\mathbb{E}_{x\sim P}[x] - \mathbb{E}_{x\sim Q}[x]\|_2 &= \|\sum_x x(p(x) - q(x))\|_2 \\ &\leq \sum_x \|x\|_2 \cdot |p(x) - q(x)| \\ &\leq R \cdot \|p - q\|_1 \\ &= 2 \cdot R \cdot TV(P, Q)\end{aligned}$$

□

But the issue is that $\frac{w_i}{n}$ is not properly distributed+ over data points. But if they did, then

$$\|\vec{w} - \vec{w}'\|_1 \Rightarrow TV(\frac{\vec{w}}{n}, \frac{\vec{w}'}{n}) \leq \frac{3}{n}$$

So

$$\|\frac{1}{n}\sum_i w_i x_i - \frac{1}{n}w_i' x_i'\|_2 \lesssim 2(2O(\sqrt{d})) \cdot \frac{3}{n} = O(\frac{\sqrt{d}}{n})$$

3.5 Overall Privacy Proof

We will just sketch the ideas. As we said above, there are two parts to the privacy proof: the check that $\|w\|_1$ is sufficiently large and the release of the mean. We proved that $|\|w\|_1 - \|w'\|_1| \leq 3$; this is a bound on the global sensitivity of the function $f(x) = \|w\|_1$, and thus adding $\text{Lap}(3/\varepsilon)$ noise preserves $(\varepsilon, 0)$-DP. The thresholding operation is just a post-processing.

Next is the tricky part. If the weighted means are at distance at most $\approx \sqrt{d}/n$ from each other, then the privacy guarantees of the Gaussian mechanism says that resulting distribution is (ε, δ)-DP. However, we only proved that the weighted means are this close under the assumption that both $w, w' \neq \vec{0}$. This is where we use the fact that we thresholded the noisy ℓ_1 norm: if one of the weight vectors is in fact all zeros, then with probability $1 - \delta$ we will fail the check and not release a noisy mean.

There are a number of ways to make this rigorous; with the right lemmas it's easy to see. However, I think the most intuitive method to see your way through the proof is a case analysis. Fix datasets x, x' which lead to weights w, w'.

- If $w, w' \neq \vec{0}$, then releasing the noisy ℓ_1 norm is (ε, δ)-DP and releasing the noisy weighted mean is (ε, δ)-DP, so overall the output is $(2\varepsilon, \delta)$-indistinguishable (by the same calculation as in basic composition).
- Now suppose $w = \vec{0}$; we know $\|w'\|_1 \leq 3$ and can show that, under either input, we fail the private ℓ_1 norm check with probability at least $1 - \delta$. If we pass the check we cannot guarantee indistinguishability, but since that happens with probability less than δ we are (ε, δ)-indistinguishable in this case as well.

4 Concluding Remarks

The intuition behind the benefit of this approach is as follows: consider a different version of the algorithm where we try to remove outliers by declaring a point an outlier if it is far from the empirical mean. This approach runs into trouble: changing a single data point can shift the empirical mean significantly, which in turn can affect all other points. In the worst case, adding or removing one individual could cause many points to suddenly be classified as outliers.

To avoid this instability, we instead base the procedure on pairwise interactions between data points. The key advantage is that pairwise quantities have much better-controlled sensitivity: changing one individual only affects the interactions involving that individual, rather than globally shifting a statistic like the mean. This allows us to bound the affect of any single data point and maintain stability under small changes to the dataset.

References

[1] Eliad Tsfadia, Edith Cohen, Haim Kaplan, Yishay Mansour, and Uri Stemmer. FriendlyCore: Practical Differentially Private Aggregation. arXiv preprint arXiv:2110.10132, 2022. `https://arxiv.org/abs/2110.10132`.