CS839: Differential Privacy and Learning March 10, 2026

Lecture 15: Privacy Amplification

Instructor: Gavin Brown *Scribe:* Rahul Choudhary

Overview

Today we cover three *privacy amplification* techniques. Each one shows that a base differentially-private mechanism can be made *more* private "for free" by composing it with a benign-looking operation:

- **Subsampling** – run the mechanism only on a random subset of the data. Used in DP-SGD, where every gradient step is taken on a small mini-batch.
- **Shuffling** – in the local DP setting, randomly permute the users' messages before release. This sits between local and central DP and eliminates the $\sqrt{n}$ accuracy gap of local DP.
- **Iteration** – in iterative algorithms (like DP-SGD), data points seen early in optimization automatically get extra privacy, because later noisy steps wash out their influence.

The first two have clean, self-contained statements that we will work out in detail. The third has a technically heavier statement, so we will focus on the intuition and applications.

Recall our working definition.

Definition 0.1 (Approximate DP)**.** An algorithm $A : \mathcal{X}^n \to \mathcal{Y}$ is (ε, δ)-*differentially private* if for all neighboring datasets $x, x' \in \mathcal{X}^n$ (differing in at most one entry) and all events $E \subseteq \mathcal{Y}$,

$$\Pr[A(x) \in E] \leq e^{\varepsilon} \Pr[A(x') \in E] + \delta.$$

1 Amplification by Subsampling

1.1 Motivation: stochastic gradient descent

In practice we always run *stochastic* gradient descent rather than full-batch GD: at each step, select a small mini-batch $S \subseteq [n]$ and compute the gradient on S. Does this help privacy?

Yes – intuitively, when a user is *not* in the mini-batch they receive perfect privacy on that step, since the algorithm never touches their record. On any given step the differing record between two neighboring datasets is usually *not* selected, so on average the mechanism sees neighboring inputs less often. Subsampling formalizes this.

1.2 The subsampling theorem

Theorem 1.1 (Amplification by subsampling [3, 5])**.** *Let A be (ε, δ)-DP. Define A' as follows: on input $x \in \mathcal{X}^n$, sample a uniformly random subset $S \subseteq [n]$ with $|S| = m$, and output $A(x_S)$. Then A' is (ε', δ')-DP for*

$$\varepsilon' = \ln\big(1 + (e^{\varepsilon} - 1)\,\tfrac{m}{n}\big), \qquad \delta' = \tfrac{m}{n}\,\delta.$$

For small ε and small subsampling rate $p = m/n$,

$$\ln\bigl(1 + (e^{\varepsilon} - 1)\tfrac{m}{n}\bigr) \approx \ln\bigl(1 + \varepsilon\tfrac{m}{n}\bigr) \approx \varepsilon\tfrac{m}{n} \;=\; \varepsilon p,$$

so up to lower-order terms, sampling at rate p *multiplies* ε *by* p. This is the key takeaway: "subsampling at rate p amplifies privacy by a factor of p."

Remark 1.2. The general proof of Theorem 1.1 is fiddly: there are many variants of "subsampling" (a uniform subset of size m, Poisson sampling with rate p, sampling with vs. without replacement) and each yields a slightly different bound. Rather than give a general proof we work through a concrete example below; [3] contains a careful, self-contained treatment.

1.3 Worked example: subsampled randomized response

Recall the binary randomized response mechanism:

$$\mathrm{RR}_{\varepsilon}(x) = \begin{cases} x & \text{w.p. } \frac{e^{\varepsilon}}{e^{\varepsilon}+1}, \\ \overline{x} & \text{w.p. } \frac{1}{e^{\varepsilon}+1}. \end{cases}$$

$\mathrm{RR}_{\varepsilon}$ is ε-DP. We want to verify the subsampling theorem on $\mathrm{RR}_{\varepsilon}$ directly. To make the algebra cleaner, introduce a closely related mechanism that approximates $\mathrm{RR}_{\varepsilon}$ for small ε:

$$\mathrm{RR}'_{\varepsilon}(x) = \begin{cases} x & \text{w.p. } \frac{1+\varepsilon}{2+\varepsilon}, \\ \overline{x} & \text{w.p. } \frac{1}{2+\varepsilon}. \end{cases}$$

where $\bar{x}$ is the flipped value of x. (Using $e^{\varepsilon} \approx 1 + \varepsilon$, $\mathrm{RR}'_{\varepsilon} \approx \mathrm{RR}_{\varepsilon}$.) Now rewrite $\mathrm{RR}'_{\varepsilon}$ as a two-step "with probability $\varepsilon/(2+\varepsilon)$ keep, else output a uniform random bit":

$$\mathrm{RR}''_{\varepsilon}(x) = \begin{cases} x & \text{w.p. } \frac{\varepsilon}{2+\varepsilon}, \\ \mathrm{Unif}(\{0,1\}) & \text{w.p. } \frac{2}{2+\varepsilon}. \end{cases}$$

These two formulations are identical as distributions:

$$\Pr[\mathrm{RR}''_{\varepsilon}(x) = x] = \frac{\varepsilon}{2+\varepsilon} + \frac{1}{2} \cdot \frac{2}{2+\varepsilon} = \frac{\varepsilon+1}{2+\varepsilon} = \Pr[\mathrm{RR}'_{\varepsilon}(x) = x].$$

Now consider the *subsampled* version: with probability p the user's record is in the mini-batch and gets $\mathrm{RR}''_{\varepsilon}$; otherwise the mechanism "doesn't see them" and emits a uniform random bit:

$$A_p(x) = \begin{cases} \mathrm{RR}''_{\varepsilon}(x) & \text{w.p. } p, \\ \mathrm{Unif}(\{0,1\}) & \text{w.p. } 1-p. \end{cases}$$

Expanding the inner mechanism, A_p outputs x with probability $p \cdot \varepsilon/(2+\varepsilon)$, and otherwise outputs a uniform random bit:

$$A_p(x) = \begin{cases} x & \text{w.p. } \dfrac{\varepsilon p}{2+\varepsilon}, \\ \mathrm{Unif}(\{0,1\}) & \text{otherwise.} \end{cases}$$

This is exactly $\mathrm{RR}''_{\varepsilon p} \equiv \mathrm{RR}'_{\varepsilon p} \approx \mathrm{RR}_{\varepsilon p}$. Subsampling at rate p turned the ε-DP mechanism $\mathrm{RR}_{\varepsilon}$ into a $(\approx \varepsilon p)$-DP one, matching Theorem 1.1. □

1.4 Aside: what about duplicates?

A natural question: if my data has duplicates – say my identical twin's record is also in the database – do I lose privacy?

This is more philosophical than mathematical. DP as defined protects each *record*, not each *person*; duplicated records do amplify information about the underlying individual, and no amount of clever analysis can hide that. The remedy is to define neighboring datasets more carefully (e.g. by the set of contributing individuals) or to bound each individual's total contribution – but those are modeling choices, not free amplification.

2 Local DP, Shuffling, and Amplification by Shuffling

2.1 The local model and its accuracy cost

In *local* differential privacy (LDP), there is no trusted curator. Each of n users i applies their own randomizer R to their record x_i and releases $y_i = R(x_i)$. The analyst only ever sees $(y_1, \dots, y_n)$.

$$\begin{array}{ccc} x_1 \longrightarrow & \boxed{R} & \longrightarrow y_1 \\ x_2 \longrightarrow & \boxed{R} & \longrightarrow y_2 \\ \vdots & \vdots & \vdots \\ x_n \longrightarrow & \boxed{R} & \longrightarrow y_n \end{array}$$

The selling point of LDP is the weaker trust assumption: each user only ever releases noisy data, so they need not trust a central curator. The price is accuracy.

Tradeoff for mean estimation. Suppose $x_1, \dots, x_n \sim \mathcal{N}(\mu, 1)$ and we want to estimate μ. Write $\hat{\mu} = \frac{1}{n}\sum_i x_i$ (the non-private MLE) and $\tilde{\mu}$ for the private estimate. By the triangle inequality,

$$|\tilde{\mu} - \mu| \le |\hat{\mu} - \mu| + |\tilde{\mu} - \hat{\mu}|,$$

where the first term is the (irreducible) statistical error and the second is the privacy cost. Comparing the two models:

- *Central DP, Laplace noise:* release $\tilde{\mu} = \hat{\mu} + \mathrm{Lap}(1/(n\varepsilon))$, giving

$$|\tilde{\mu} - \mu| \lesssim \underbrace{\tfrac{1}{\sqrt{n}}}_{\text{non-private}} + \underbrace{\tfrac{1}{n\varepsilon}}_{\text{private}}.$$

- *Local DP, randomized response per user:* averaging the debiased y_i's,

$$|\tilde{\mu} - \mu| \lesssim \tfrac{1}{\sqrt{n}} + \tfrac{1}{\sqrt{n}\,\varepsilon}.$$

The private term in LDP is a factor of $\sqrt{n}$ worse than in central DP. The reason is structural: in LDP the adversary always *knows which output corresponds to which user*, so to hide x_i each user has to inject enough noise to mask their own record on its own, rather than relying on a sum.

2.2 The shuffle model

The shuffle model interpolates between local and central DP. Each user still randomizes their record locally, but a trusted shuffler then *permutes* the messages uniformly at random before release.

$$\begin{array}{lll} x_1 \to & \boxed{R} & \to y_1 \\ x_2 \to & \boxed{R} & \to y_2 \\ \vdots & \vdots & \vdots \\ x_n \to & \boxed{R} & \to y_n \end{array} \quad \to \boxed{\text{shuffle}} \to \quad (z_1, \ldots, z_n) \text{ in random order}$$

The adversary now sees $\{z_1, \ldots, z_n\}$ but does not know which z_j came from which user. Intuitively, this should buy privacy: the user's record is "hidden among the clones" [2]. And indeed:

Theorem 2.1 (Amplification by shuffling, informal; [2])**.** *Suppose n users each apply RR_ε with $\varepsilon \le 1$ to their record, and the outputs are shuffled uniformly at random. Then for any $\delta > 0$, the shuffled protocol is (ε', δ)-DP for*

$$\varepsilon' = O\left(\varepsilon \cdot \sqrt{\frac{\log(1/\delta)}{n}}\right).$$

This $\sqrt{n}$ in the denominator is exactly what was missing from local DP: shuffling RR_ε achieves (up to log factors) the central-DP accuracy rate $1/(n\varepsilon')$, recovering essentially the central-DP error $\frac{1}{\sqrt{n}} + \frac{\sqrt{\log(1/\delta)}}{n\,\varepsilon}$ from a much weaker trust assumption (a non-colluding shuffler).

[2] give a clean proof that, in addition to being simpler than earlier analyses [4], achieves the asymptotically optimal dependence on ε and extends to general ε_0-LDP randomizers, not just RR.

3 Amplification by Iteration

The mathematical statement of amplification by iteration is technical (it is naturally phrased in Rényi DP or related divergence-based notions), so we focus on the intuition. The main reference is [1].

3.1 Setup: one-pass DP-SGD

Consider a one-pass variant of DP-SGD, where each data point is seen by the algorithm at most once and only the final iterate is returned (Algorithm 1).

Algorithm 1 One-Pass DP-SGD

Input: Dataset $\{(x_i, y_i)\}_{i=1}^n$ with $n = Tb$, batch size b, step size η, noise scale σ, initialization θ_1
Returns: Final iterate θ_{T+1}

1: **for** $t = 1, \ldots, T$ **do**
2: $\quad \tilde{g}_t \leftarrow \frac{1}{b} \sum_{i=(t-1)b+1}^{tb} \nabla \ell(\theta_t; x_i, y_i) + \mathcal{N}(0, \sigma^2 I)$
3: $\quad \theta_{t+1} \leftarrow \theta_t - \eta\, \tilde{g}_t$
4: **end for**
5: **return** θ_{T+1}

Two features will matter:

- Each example is processed in exactly one step (this is what "one pass" means).
- Only the final iterate is released; intermediate θ_t are discarded.

3.2 The key intuition: "early data is more private"

Consider four datasets, related as in Figure 1:

- x and x' are neighbors that differ in the *first* record (seen at step $t = 1$);
- y and y' are neighbors that differ in the *last* record (seen at step $t = T$).

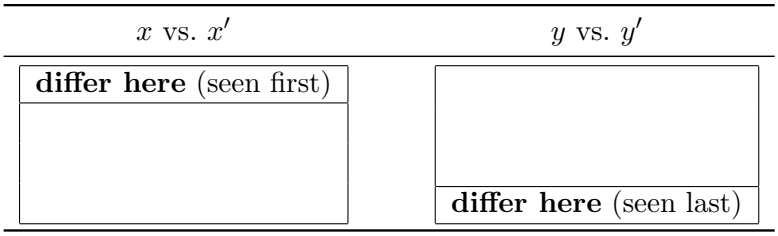

Figure 1: $x \sim x'$ differ in the first record (consumed at iteration 1); $y \sim y'$ differ in the last record (consumed at iteration T).

Intuition. It should be *harder* to distinguish $A(x)$ from $A(x')$ than to distinguish $A(y)$ from $A(y')$. Why? Because the change between x and x' enters the algorithm at iteration 1 and is then processed by $T - 1$ further noisy gradient steps before the output θ_{T+1} is released. Each of those subsequent noisy steps acts like additional "post-processing noise" on the influence of the differing record. By contrast, the change between y and y' enters at the very last step, with no further noise injected afterwards.

So if we account for privacy in the worst case (last record), we are *over-paying* for the privacy of early records. Amplification by iteration formalizes the extra privacy that early records receive.

3.3 Main result and an application

[1] study exactly this setting – the privacy loss of returning the final iterate of noisy SGD on smooth convex losses over a bounded domain. Their main result is qualitatively surprising:

Theorem 3.1 ([1], informal)**.** *For Langevin / noisy SGD on smooth convex losses over a bounded domain, the (Rényi) DP loss of releasing the final iterate θ_{T+1}* stops growing *as T increases, once T exceeds a short burn-in period. Up to constants, the long-run privacy loss matches that of a fixed number of iterations.*

Earlier analyses (e.g. via composition of per-step Gaussian mechanisms) had the privacy loss growing without bound in T. [1] show this is qualitatively wrong: once you condition on releasing only the final iterate, running longer does not cost more privacy. The proof goes through "privacy amplification by iteration" and is based on optimal transport / Wasserstein-like contraction arguments rather than Markov-chain fast mixing.

Practical takeaway. One natural way to exploit amplification by iteration is to *add less noise to early steps*. Those records will get extra privacy from the many subsequent noisy iterations, so we don't need to pay full price for them at the time they're consumed. This is a useful design lever for variants of DP-SGD that go beyond uniform per-step noise.

Summary

The three amplification techniques each give us "free privacy" in a different way:

Technique	"Free" operation	Effect on ε
Subsampling	Run on a random sub-batch of rate p	$\varepsilon \to \approx \varepsilon p$
Shuffling	Permute LDP outputs uniformly	$\varepsilon \to O(\varepsilon\sqrt{\log(1/\delta)/n})$
Iteration	Release only the final iterate of noisy SGD	Privacy stops growing in T

Subsampling is what makes DP-SGD with mini-batches privacy-efficient. Shuffling is what makes the shuffle model a usable middle ground between central and local DP. Iteration is what makes long-running DP-SGD feasible.

References

[1] Altschuler, J. M., & Talwar, K. *Privacy of noisy stochastic gradient descent: more iterations without more privacy loss.* NeurIPS, 2022. `https://arxiv.org/abs/2205.13710`

[2] Feldman, V., McMillan, A., & Talwar, K. *Hiding among the clones: a simple and nearly optimal analysis of privacy amplification by shuffling.* FOCS, 2021. `https://arxiv.org/abs/2012.12803`

[3] Steinke, T. *Composition of differential privacy & privacy amplification by subsampling.* Chapter in *Differential Privacy for Artificial Intelligence Applications*, 2022. `https://arxiv.org/abs/2210.00597`

[4] Erlingsson, Ú., Feldman, V., Mironov, I., Raghunathan, A., Talwar, K., & Thakurta, A. *Amplification by shuffling: from local to central differential privacy via anonymity.* SODA, 2019. `https://arxiv.org/abs/1811.12469`

[5] Vadhan, S. *The complexity of differential privacy.* Tutorials on the Foundations of Cryptography, 2017. `https://salil.seas.harvard.edu/sites/g/files/omnuum4266/files/salil/files/manuscript_2017.pdf`