Privacy Amplification via Iteration for Shuffled and Online PNSGD

Matteo Sordello[✉][1], Zhiqi Bu[2], and Jinshuo Dong[3]

[1] Department of Statistics, Wharton School, University of Pennsylvania
sordello@wharton.upenn.edu
[2] Graduate Group in AMCS, University of Pennsylvania
zbu@sas.upenn.edu
[3] IDEAL Institute, Northwestern University
jinshuo@northwestern.edu

Abstract. In this paper, we consider the framework of privacy amplification via iteration, which is originally proposed by Feldman et al. and subsequently simplified by Asoodeh et al. in their analysis via the contraction coefficient. This line of work focuses on the study of the privacy guarantees obtained by the projected noisy stochastic gradient descent (PNSGD) algorithm with hidden intermediate updates. A limitation in the existing literature is that only the early stopped PNSGD has been studied, while no result has been proved on the more widely-used PNSGD applied on a shuffled dataset. Moreover, no scheme has been yet proposed regarding how to decrease the injected noise when new data are received in an online fashion. In this work, we first prove a privacy guarantee for shuffled PNSGD, which is investigated asymptotically when the noise is fixed for each sample size n but reduced at a predetermined rate when n increases, in order to achieve the convergence of privacy loss. We then analyze the online setting and provide a faster decaying scheme for the magnitude of the injected noise that also guarantees the convergence of privacy loss.

Keywords: differential privacy, online learning, optimization

1 Introduction

Differential privacy (DP) [12, 11] is a strong standard to guarantee the privacy for algorithms that have been widely applied to modern machine learning [1]. It characterizes the privacy loss via statistical hypothesis testing, thus allowing the mathematically rigorous analysis of the privacy bounds. When multiple operations on the data are involvedand each intermediate step is revealed, composition theorems can be used to keep track of the privacy loss, which combines subadditively [16]. However, because such results are required to be general, their associated privacy bounds are inevitably loose. In contrast, privacy amplification provides a privacy budget for a composition of mechanisms that is less that the budget of each individual operation, which strengthens the bound the more operations

are concatenated. Classic examples of this feature are privacy amplification by subsampling [8, 4], by shuffling [14] and by iteration [15, 3]. In this paper, we focus on the setting of privacy amplification by iteration, and extend the analysis via contraction coefficient proposed by [3] to prove results that apply to an algorithm commonly used in practice, in which the entire dataset is shuffled before training a model with PNSGD. We emphasize that the shuffling is a fundamental difference compared to previous work, since it is a necessary step in training many machine learning models.

We start by laying out the definitions that are necessary for our analysis. We consider a convex function $f : \mathbb{R}^+ \to \mathbb{R}$ that satisfies $f(1) = 0$. [2] and [9] define the f-divergence between two probability distribution μ and ν is as

$$D_f(\mu\|\nu) = \mathbb{E}_\nu\left[f\left(\frac{\mathrm{d}\mu}{\mathrm{d}\nu}\right)\right] = \int f\left(\frac{\mathrm{d}\mu}{\mathrm{d}\nu}\right) d\nu$$

For a Markov kernel $K : \mathcal{W} \to \mathcal{P}(\mathcal{W})$, where $\mathcal{P}(\mathcal{W})$ is the space of probability measures over $\mathcal{W}$, we let $\eta_f(K)$ be the contraction coefficient of kernel K under the f-divergence, which is defined as

$$\eta_f(K) = \sup_{\mu,\nu: D_f(\mu\|\nu)\neq 0} \frac{D_f(\mu K\|\nu K)}{D_f(\mu\|\nu)}$$

If we now consider a sequence of Markov kernels $\{K_n\}$ and let the two sequences of measures $\{\mu_n\}$ and $\{\nu_n\}$ be generated starting from μ_0 and ν_0 by applying $\mu_n = \mu_{n-1}K_n$ and $\nu_n = \nu_{n-1}K_n$, then the strong data processing inequality [19] for the f-divergence tells us that

$$D_f(\mu_n\|\nu_n) \leq D_f(\mu_0\|\nu_0)\prod_{t=1}^{n}\eta_f(K_t)$$

Among the f-divergences, we focus on the E_γ-divergence, or hockey-stick divergence, which is the f-divergence associated with $f(t) = (t-\gamma)_+ = \max(0, t-\gamma)$. We do so because of its nice connection with the concept of (ϵ, δ) differential privacy, which is now the state-of-the-art technique to analyze the privacy loss that we incur when releasing information from a dataset. A mechanism $\mathcal{M}$ is said to be (ϵ, δ)-DP if, for every pair of neighboring datasets (datasets that differ only in one entry, for which we write $D \sim D'$) and every event $\mathcal{A}$, one has

$$\mathbb{P}(\mathcal{M}(D) \in \mathcal{A}) \leq e^\epsilon \mathbb{P}(\mathcal{M}(D') \in \mathcal{A}) + \delta \tag{1}$$

It is easy to prove that a mechanism $\mathcal{M}$ is (ϵ, δ)-DP if and only if the distributions that it generates on D and D' are close with respect to the E_γ-divergence. In particular, for $D \sim D'$ and $\mathbb{P}_D$ being the output distribution of mechanism $\mathcal{M}$ on D, then $\mathcal{M}$ is (ϵ, δ)-DP if and only if

$$E_{e^\epsilon}(\mathbb{P}_D\|\mathbb{P}_{D'}) \leq \delta. \tag{2}$$

It has been proved in [3] that the contraction coefficient of a kernel $K : \mathcal{W} \to \mathcal{P}(\mathcal{W})$ under E_γ-divergence, which we refer to as $\eta_\gamma(K)$, satisfies

$$\eta_\gamma(K) = \sup_{w_1, w_2 \in \mathcal{W}} E_\gamma(K(w_1)\|K(w_2))$$

This equality improves on a result proved by [5] and makes it easier to find an explicit form for the contraction coefficient of those distributions for which we can compute the hockey-stick divergence. Two such distributions are the Laplace and Gaussian, and [3] investigate the privacy guarantees generated by this privacy amplification mechanism in the setting of PNSGD with Laplace or Gaussian noise. As the standard stochastic gradient descent (SGD), the PNSGD is defined with respect to a loss function $\ell : \mathcal{W} \times \mathcal{X} \to \mathbb{R}$ that takes as inputs a parameter in the space $\mathbb{K} \subseteq \mathcal{W}$ and an observation $x \in \mathcal{X}$. Common assumptions made on the loss functions are the following: for each $x \in \mathcal{X}$

- $\ell(\cdot, x)$ is L-Lipschitz
- $\ell(\cdot, x)$ is ρ-strongly convex
- $\nabla_w \ell(\cdot, x)$ is β-Lipschitz.

The PNSGD algorithm works by combining three steps: (1) a stochastic gradient descent (SGD) step with learning rate η; (2) an injection of i.i.d. noise sampled from a known distribution to guarantee privacy and (3) a projection $\Pi_{\mathbb{K}} : \mathcal{W} \to \mathbb{K}$ onto the subspace $\mathbb{K}$. Combined, these steps give the following update rule

PNSGD

$$w_{t+1} = \Pi_{\mathbb{K}} \left(w_t - \eta(\nabla_w \ell(w_t, x_{t+1}) + Z_{t+1}) \right)$$

which can be defined as a Markov kernel by assuming that $w_0 \sim \mu_0$ and $w_t \sim \mu_t = \mu_0 K_{x_1} ... K_{x_t}$, where K_x is the kernel associated to a single PNSGD step when observing the data point x. The application of the PNSGD on a dataset D assumes that the entries of the dataset are passed through the algorithm in a fixed order that depends on their index, hence w_1 is defined observing the first entry x_1 and so on. With this definition, one can find an upper bound for δ by bounding the left hand side of (2). The specific bound depends on the index at which the neighboring datasets D and D' differ and the distribution of the noise injected in the PNSGD. [3] investigate the bound for both Laplace and Gaussian noise, which we report in the following theorem.

Theorem 1 (Theorem 3 and 4 in [3]). *Define*

$$Q(t) = \frac{1}{\sqrt{2\pi}} \int_t^\infty e^{-\frac{u^2}{2}} du = 1 - \Phi(t)$$

where Φ is the cumulative density function of the standard normal,

$$\theta_\gamma(r) = Q\left(\frac{\log(\gamma)}{r} - \frac{r}{2} \right) - \gamma Q\left(\frac{\log(\gamma)}{r} + \frac{r}{2} \right) \tag{3}$$

and the constant

$$M = \sqrt{1 - \frac{2\eta\beta\rho}{\beta + \rho}}$$

which depends on the parameters of the loss function and the learning rate of the SGD step. If $\mathbb{K} \subset \mathbb{R}^d$ is compact and convex with diameter $D_{\mathbb{K}}$, the PNSGD

algorithm with Gaussian noise $N(0, \sigma^2)$ *is* (ϵ, δ)*-DP for its i-th entry where* $\epsilon \geq 0$ *and*

$$\delta = \theta_{e^\epsilon}\left(\frac{2L}{\sigma}\right)\theta_{e^\epsilon}\left(\frac{MD_{\mathbb{K}}}{\eta\sigma}\right)^{n-i}$$

If instead we consider $\mathbb{K} = [a, b]$ *for* $a < b$*, then the PNSGD algorithm with Laplace noise* $\mathcal{L}(0, v)$ *is* (ϵ, δ)*-DP for its i-th entry where* $\epsilon \geq 0$ *and*

$$\delta = \left(1 - e^{\frac{\epsilon}{2} - \frac{L}{v}}\right)_+ \left(1 - e^{\frac{\epsilon}{2} - \frac{M(b-a)}{2\eta v}}\right)_+^{n-i}$$

To slightly simplify the notation, we can present the guarantees in Theorem 1 as $\delta = A \cdot B^{n-i}$ where for the Gaussian case

$$A = \theta_{e^\epsilon}\left(\frac{2L}{\sigma}\right), \quad B = \theta_{e^\epsilon}\left(\frac{MD_{\mathbb{K}}}{\eta\sigma}\right) \tag{4}$$

and for the Laplacian case

$$A = \left(1 - e^{\frac{\epsilon}{2} - \frac{L}{v}}\right)_+, \quad B = \left(1 - e^{\frac{\epsilon}{2} - \frac{M(b-a)}{2\eta v}}\right)_+ \tag{5}$$

To get a bound that does not depend on the index of the entry on which the two datasets differ, the authors later consider the randomly-stopped PNSGD, which simply consist of picking a random stopping time for the PNSGD uniformly from $\{1, ..., n\}$. The bound that they obtain for δ in the Gaussian case is $\delta = A/[n(1 - B)]$. Based on their proof, it is clear that the actual bound contains a term $(1 - B^{n-i+1})$ at the numerator and that the same result can be obtained if we consider the Laplace noise.

In Section 3 we prove that a better bound than the one obtained via randomly-stopped PNSGD can be obtained by first shuffling the dataset and then applying the simple PNSGD. In Section 4 we study the asymptotic behavior of such bound and find the appropriate decay rate for the variability of the noise level that guarantees convergence for δ to a non-zero constant.

2 Related Work

In the DP regime, (ϵ, δ)-DP (see (1)) is arguably the most popular definition, which is oftentimes achieved by an algorithm which contains Gaussian or Laplacian noises. For example, in NoisySGD and NoisyAdam in [1, 6], and PNSGD in this paper, a certain level of random noise is injected into the gradient to achieve DP. Notably, as we use more datapoints (or more iterations during the optimization) during the training procedure, the privacy loss accumulates at a rate that depends on the magnitude of the noise.

It is remarkably important to characterize, as tightly as possible, the privacy loss at each iteration. An increasing line of works have proposed to address this difficulty [10, 7, 13, 4, 18, 20, 17, 3, 1], which bring up many useful notions of DP, such as Rényi DP, Gaussian DP, f-DP and so on. Our paper extends [3] by shuffling the dataset first rather than randomly stopping the PNSGD (see Theorem 5 in [3]), in order to address the non-uniformity of privacy guarantee.

As a consequence, we obtain a strictly better privacy bound and better loss than [3], [1], and an additional online result of the privacy guarantee.

Furthermore, our results can be easily combined with composition tools in DP [16, 1, 17, 10]. In Theorem 2, Theorem 3 and Theorem 4, the (ϵ, δ) is computed based on a single pass of the entire dataset, or equivalently on one epoch. When using the shuffled PNSGD for multiple epochs, as is usual for modern machine learning, the privacy loss accumulates and is accountable by Moments accountant (using Renyi DP [18]), f-DP (using functional characterization of the type I/II errors trade-off) and other divergence approaches.

3 Shuffled PNSGD

In this section, we prove the bound on δ that we can obtain by first shuffling the dataset and then apply the PNSGD algorithm. The simple underlying idea here is that, when shuffling the dataset, the index at which the two neighboring datasets differ has equal probability to end up in each position. This is a key difference compared to the randomly-stopped PNSGD, and allows us to get a better bound that do not depend on the initial position of that index.

Theorem 2. *Let $D \sim D'$ be of size n. Then the shuffled PNSGD is (ϵ, δ)-DP with*

$$\delta = \frac{A \cdot (1 - B^n)}{n(1 - B)} \tag{6}$$

and the constants A and B are defined in (4) for Gaussian noise and (5) for Laplace noise.

Proof. Let's start by considering the simple case $n = 2$, so that $D = \{x_1, x_2\}$ and $D' = \{x'_1, x'_2\}$ and let $i \in \{1, 2\}$ be the index at which they differ. Let μ be the output distribution of the shuffled PNSGD on D, and ν be the corresponding distribution from D'. If we define $S(D)$ and $S(D')$ to be the two datasets after performing the same shuffling, then we can only have either $S(D) = \{x_1, x_2\}$ or $S(D) = \{x_2, x_1\}$, both with equal probability $1/2$. The outcomes of the shuffled PNSGD on D and D' are then

$$\mu = \frac{1}{2}\mu_0 K_{x_1} K_{x_2} + \frac{1}{2}\mu_0 K_{x_2} K_{x_1}$$

$$\nu = \frac{1}{2}\mu_0 K_{x'_1} K_{x'_2} + \frac{1}{2}\mu_0 K_{x'_2} K_{x'_1}$$

By convexity and Jensen's inequality we have that

$$E_\gamma(\mu \| v) \leq \frac{1}{2} E_\gamma \left(\mu_0 K_{x_1} K_{x_2} \| \mu_0 K_{x'_1} K_{x'_2}\right) + \frac{1}{2} E_\gamma \left(\mu_0 K_{x_2} K_{x_1} \| \mu_0 K_{x'_2} K_{x'_1}\right)$$

and now we have two options, based on where the two original datasets differ. If $i = 1$, in the first term the privacy is stronger than in the second one (because x_1 is seen earlier), and we have

$$E_\gamma(\mu \| \nu) \leq \frac{1}{2} A \cdot B + \frac{1}{2} A = \frac{1}{2} A(B + 1)$$

If $i = 2$, now the privacy is stronger in the second term, and

$$E_\gamma(\mu\|\nu) \leq \frac{1}{2}A + \frac{1}{2}A \cdot B = \frac{1}{2}A(B+1)$$

Since in both cases the bound is the same, this means that for any $i \in \{1, 2\}$ the privacy guarantee of the shuffled PNSGD algorithm is equal to $A(B+1)/2$. From here we see that, when $n > 2$, the situation is similar. Instead of just two, we have $n!$ possible permutations for the elements of D, each one happening with the same probability $1/n!$. For each fixed index i on which the two neighboring datasets differ, we have $(n-1)!$ permutations in which element x_i appears in each of the n positions. When, after the permutation, element x_i ends up in last position, the bound on $E_\gamma(\mu\|\nu)$ is the weakest and just equals A. When in ends up in first position, the bound is the strongest and is equal to $A \cdot B^{n-1}$. We then have that, irrespectively of the index i,

$$E_\gamma(\mu\|v) \leq \frac{1}{n!}(n-1)!A\sum_{j=0}^{n-1} B^j = \frac{A \cdot (1-B^n)}{n(1-B)}$$

This bound is indeed better than the one found in [3] for the randomly stopped PNSGD since it contains an extra term $(1 - B^n)$ at the numerator which does not depend on i and is smaller than 1. If n is large and B is fixed, this difference is negligible because it decays exponentially. However, we will see later that when the injected noise is reduced at the appropriate rate we can guarantee that $B \approx 1 - O(1/n)$, so that the extra term ends up having an impact in the final bound. It is also important to notice that shuffled PNSGD achieves in general better performance than randomly stopped PNSGD and it is much more commonly used in practice. We see in Figure 1 that this is the case for both linear and logistic regression, and that the variation in the result in shuffled PNSGD is less than for the early stopped case, due to the fact that we always use all the data available for each epoch. In the next section we look at the asymptotic behavior of (6) when n grows and the variance of the injected noise is properly reduced to guarantee convergence.

4 Asymptotic Analysis for δ when Using Shuffling and Fixed Noises

In this Section we investigate the behavior of the differential privacy bound in (6) when the size n of the dataset grows. In Section 4.1 we prove a results for the shuffled PNSGD with fixed Laplace noise, while in Section 4.2 we prove the same result on the shuffled PNSGD with fixed Gaussian noise.

4.1 Laplace Noise

We present first a result that holds when we consider a fixed Laplace noise $\mathcal{L}(0, v)$ injected into the PNSGD algorithm for each update. In order to get a convergence result for δ as the size n of the dataset grows, the level of noise that we use should

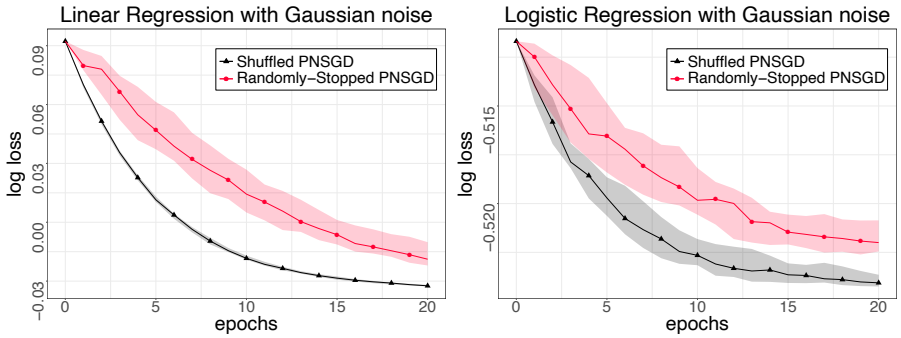

Fig. 1. Comparison between shuffled PNSGD and randomly-stopped PNSGD with Gaussian noise in linear and logistic regression. On the y-axis we report the log loss achieved. The parameters used are $n = 1000, d = 2, \sigma = 0.5, \theta^* = \Pi_{\mathbb{K}}(1, 2)$ and $\mathbb{K}$ is a ball of radius 1. The learning rate is 10^{-4} in linear regression and $5 \cdot 10^{-3}$ in logistic regression.

be targeted to the quantity n. The decay of v is regulated by two parameters, C_1 and C_2. While C_1 is set to be large, so that δ converges to a small value, the use of C_2 is simply to allow the noise level not to be too large for small n, but does not appear in the asymptotic bound.

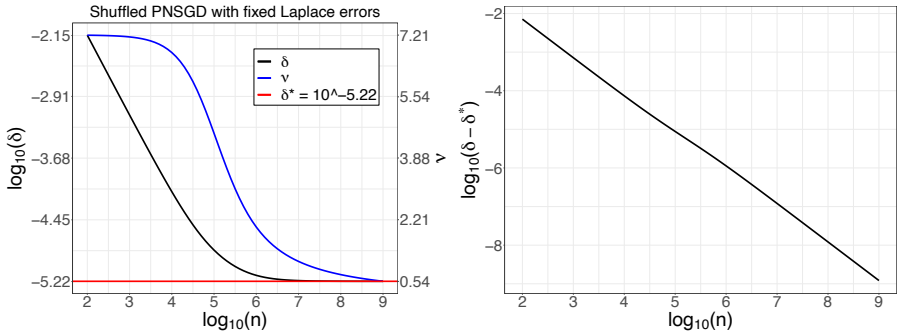

Fig. 2. (left) Convergence of δ to δ^* in (8). We plot in black the behavior of δ as a function of n, and in blue the corresponding behavior of $v(n)$ in (7). (right) We show that the convergence rate is $1/n$. The parameters used are $L = 10, \beta = 0.5, \rho = 0, \eta = 0.1, \epsilon = 1, (a, b) = (0, 1), C_1 = 10^5$ and $C_2 = 2$.

Theorem 3. *Consider the shuffled PNSGD with Laplace noise $\mathcal{L}(0, v(n))$ which is fixed for each update, where*

$$v(n) = \frac{M(b-a)}{2\eta \log\left(n/C_1 + C_2\right)}. \tag{7}$$

Then, for n sufficiently large the procedure is (ϵ, δ)-DP with $\delta = \delta^* + O(1/n)$ and

$$\delta^* = \frac{1 - e^{-C_1 \exp(\epsilon/2)}}{C_1 e^{\frac{\epsilon}{2}}} \tag{8}$$

Proof. We use the result in Theorem 2 combined with (5), and get that

$$\delta = \frac{\left(1 - e^{\frac{\epsilon}{2} - \frac{L}{v(n)}}\right)_+ \cdot \left[1 - \left(1 - e^{\frac{\epsilon}{2} - \frac{M(b-a)}{2\eta v(n)}}\right)_+^n\right]}{n \cdot e^{\frac{\epsilon}{2} - \frac{M(b-a)}{2\eta v(n)}}}$$

Once we plug in the $v(n)$ defined in (7) we have that, when n is sufficiently large,

$$\begin{aligned}
\delta &= \frac{\left(1 - e^{\frac{\epsilon}{2} - \frac{2L\eta \log(\frac{n}{C_1} + C_2)}{M(b-a)}}\right)_+ \left[1 - \left(1 - \frac{C_1 e^{\frac{\epsilon}{2}}}{n + C_1 C_2}\right)_+^n\right]}{n \cdot e^{\frac{\epsilon}{2} - \log(\frac{n}{C_1} + C_2)}} \\
&= \frac{\left[1 - \left(1 - \frac{C_1 e^{\frac{\epsilon}{2}}}{n + C_1 C_2}\right)^n\right]}{n \cdot \frac{C_1 e^{\frac{\epsilon}{2}}}{n + C_1 C_2}} \cdot \left(1 + O\left(\frac{1}{n}\right)\right) \\
&= \frac{1 - e^{-C_1 \exp(\epsilon/2)}}{C_1 e^{\frac{\epsilon}{2}}} + O\left(\frac{1}{n}\right)
\end{aligned}$$

The convergence result in Theorem 3 is confirmed by Figure 2. In the left plot we see that δ converges to the δ^* defined in (8), while in the right plot we observe that the convergence rate is indeed $1/n$.

4.2 Gaussian Noise

Similarly to what we just proved in Section 4.1 we now discuss a result for the shuffled PNSGD with Gaussian noise $N(0, \sigma^2(n))$.

Theorem 4. *Consider the shuffled PNSGD algorithm with Gaussian noise $N(0, \sigma^2(n))$ which is fixed for each update, where*

$$\sigma(n) = \frac{M D_{\mathbb{K}}}{2\eta \sqrt{W\left(\frac{n^2}{2C_1^2 \pi} + C_2\right)}} \tag{9}$$

and W is the Lambert W function. Then, for n sufficiently large, the procedure is (ϵ, δ)-DP with $\delta = \delta^* + O\left(\frac{1}{\log(n)}\right)$ and

$$\delta^* = \frac{1 - e^{-2C_1 e^{\frac{\epsilon}{2}}}}{2C_1 e^{\frac{\epsilon}{2}}} \tag{10}$$

Just like $v(n)$, the decay of the standard deviation $\sigma(n)$ is regulated by the parameters C_1 and C_2. The difference here is that, instead of a simple logarithmic decay, we now have a decay rate that depends on the Lambert W function, which is slightly harder to study analytically than the logarithm. Even though the Lambert W function is fundamentally equivalent to a logarithm when its argument

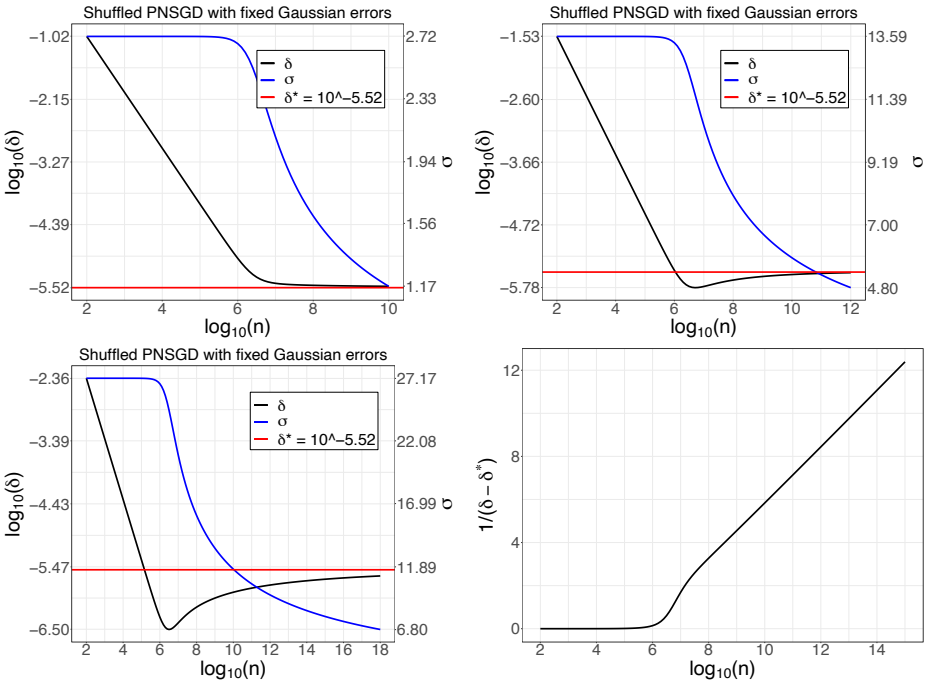

Fig. 3. Convergence of δ to δ^* defined in (10). We report in black the behavior of δ and in blue that of $\sigma(n)$ in (9). We consider $\eta \in \{0.1, 0.02, 0.01\}$ and the other parameters are $L = 10, \beta = 0.5, \rho = 0, \epsilon = 1, D_{\mathbb{K}} = 1, C_1 = 10^5$ and $C_2 = 100$. In the right-bottom panel we show that the convergence rate is $1/\log(n)$.

grows, the difference with the Laplace case is also evident in the fact that the convergence of δ to δ^* happens more slowly, at a rate of $1/\log(n)$. The proof of the theorem is in the Supplementary Material, and makes use of the following Lemma, also proved in the Supplementary Material.

Lemma 5. *For $\theta_\gamma(r)$ defined in (3), a sufficiently small σ and two constants c and ϵ, we have*

$$\theta_{e^\epsilon}\left(\frac{c}{\sigma}\right) = 1 - \frac{1}{\sqrt{2\pi}} e^{\frac{\epsilon}{2}} e^{-\frac{c^2}{8\sigma^2}} \left(\frac{4\sigma}{c} + O(\sigma^3)\right).$$

The behavior described in Theorem 4 is confirmed by what we see in Figure 3, where we can also observe that there are different patterns of convergence for δ, both from above and from below the δ^* defined in (10). In the right-bottom panel we also see a confirmation that the convergence rate is the one we expected, since $(\delta - \delta^*)^{-1}$ increase linearly with respect to $\log(n)$ when n is sufficiently large (notice that the y-axis is rescaled by a factor 10^6).

5 Multiple Epochs Composition

We now consider a simple yet important extension of the result in Theorem 2, where the shuffled PNSGD is applied for multiple epochs. In real experiments, e.g. when training deep neural networks, usually multiple passes over the data are necessary to learn the model. In such scenario, the updates are not kept secret for the whole duration of the training, but are instead released at the end of each epoch. The result proved in Theorem 2 states that for each epoch the procedure is (ϵ, δ)-DP with $\delta \leq A \cdot (1 - B^n)/[n(1 - B)]$. We can then easily combine these privacy bounds using state-of-the-art composition tools, such as the Moments Accountant [1], f-DP and Gaussian DP [10]. We present some popular ways to compute the privacy loss after E epochs.

At the high level, we migrate from (ϵ, δ) in DP to other regimes, Gaussian DP or Rényi DP, at the first epoch. Then we compose in those specific regimes until the end of training procedure. At last, we map back from the other regimes back to (ϵ, δ)-DP.

f-DP and Gaussian DP: At the first epoch, we compute the initial (ϵ, δ) and derive the four-segment curve $f_{\epsilon,\delta}$ for the type I/II errors trade-off (see Equation (5) and Proposition 2.5 in [10]). Then by Theorem 3.2 in [10], we can numerically compose this trade-off function with Fourier transform for E times, which can be accelerated by repeated squaring. When the noise is Gaussian, we can alternatively use μ in GDP to characterize the trade-off function (i.e. the mechanism is μ-GDP after the first epoch). Next, we apply Corollary 3.3 in [10] to conclude that the mechanism is $\sqrt{E}\mu$-GDP in the end. We can compute the final (ϵ, δ) reversely from GDP by Corollary 2.13 in [10].

Moments Accountant: Moments Accountant is closely related to Rényi DP (RDP), which composes easily: at the first epoch, we compute the (ϵ, δ) of our PNSGD. By Proposition 3 in [18], we can transfer from (ϵ, δ)-DP to $(\alpha, \epsilon + \frac{\log \delta}{\alpha - 1})$ RDP. After the first epoch, the initial RDP can be composed iteratively by Moments Accountant[4]. The final (α', ϵ') RDP is then mapped back to (ϵ, δ)-DP with $\epsilon = \epsilon' - \frac{\log \delta}{\alpha' - 1}$.

6 Online Results for Decaying Noises

We now go back to the original framework of [3] and consider the PNSGD algorithm applied to the non-shuffled dataset. This time, however, we want to apply a different level of noise for each update, and see if we can get a convergence result for δ when $n \to \infty$. We then need to consider values of A and B in (4) and (5) that depend on the specific index, and the privacy bound for the PNSGD with non-fixed noises and neighboring datasets that differ on index i becomes

$$\delta = A_i \cdot \prod_{t=i+1}^{n} B_t \tag{11}$$

[4] See `https://github.com/tensorflow/privacy/blob/master/tensorflow_privacy/privacy/analysis/rdp_accountant.py`

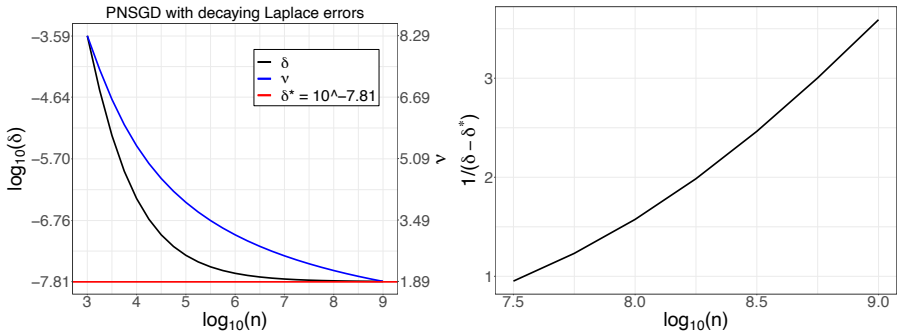

Fig. 4. (left) Convergence of δ to δ^* defined in (13). We report in black the behavior of δ and in blue that of v_n defined in (12). The parameters considered are $L = 10, \beta = 0.5, \rho = 0, \epsilon = 1, \eta = 0.01, \alpha = 1.5, (a, b) = (0, 1), i = 100, C_1 = 100$ and $C_2 = 100$. (right) The convergence rate is approximately $1/\log(n)$.

Here the definition of A_i and B_i is the same as in (4) and (5) but the noise level v and σ is now dependent on the position of each element in the dataset. In this scenario we can actually imagine adding new data to the dataset in an online fashion, without having to restart the procedure to recalibrate the noise level used for the first entries. It is clear that, in order to get convergence, the decay of the injected noise should be faster than in Theorem 3 and Theorem 4, since now the early entries receive an amount of noise that does not vanish as n becomes large. However it is interesting to notice that for both the Laplace and Gaussian noise the only difference needed with the decay rate for $v(n)$ and $\sigma(n)$ defined before is an exponent $\alpha > 1$.

6.1 Laplace Noise

We prove here the online result for the PNSGD with Laplace noise that decays for each entry. As anticipated, the decay is no longer the same for all entries and proportional to $1/\log(n)$ but now for the entry with index j we have a decay which is proportional to $1/\log(j^\alpha)$.

Theorem 6. *Consider the PNSGD where for update j we use Laplace noise $\mathcal{L}(0, v_j)$, and*

$$v_j = \frac{M(b-a)}{2\eta \log\left(j^\alpha / C_1 + C_2\right)} \tag{12}$$

for $\alpha > 1$. Then as $n \to \infty$ the procedure is (ϵ, δ^*)-DP where

$$\delta^* = \left(1 - e^{\frac{\epsilon}{2} - \frac{2L\eta \log\left(i^\alpha / C_1 + C_2\right)}{M(b-a)}}\right)_+ e^{\int_{i+1}^{\infty} \log\left(1 - \frac{C_1 e^{\frac{\epsilon}{2}}}{x^\alpha + C_1 C_2}\right) dx} \tag{13}$$

and i is the index where the neighboring datasets differ.

Proof. We show again that δ converges to a non-zero value as n goes to ∞. In fact, again following the proof of ([3] Theorem 3), we get that,

$$\delta = \left(1 - e^{\frac{\epsilon}{2} - \frac{L}{v_i}}\right)_+ \cdot \prod_{t=i+1}^{n} \left(1 - e^{\frac{\epsilon}{2} - \frac{M(b-a)}{2\eta v_t}}\right)_+$$
$$= \left(1 - e^{\frac{\epsilon}{2} - \frac{2L\eta \log(\frac{i^\alpha}{C_1} + C_2)}{M(b-a)}}\right)_+ \prod_{t=i+1}^{n} \left(1 - \frac{C_1 e^{\frac{\epsilon}{2}}}{t^\alpha + C_1 C_2}\right)_+$$

We know that, for a sequence a_t of positive values, $\prod_{t=1}^{\infty}(1-a_t)$ converges to a non-zero number if and only if $\sum_{t=1}^{\infty} a_t$ converges. Here we have that

$$\sum_{t=i+1}^{\infty} \frac{C_1 e^{\frac{\epsilon}{2}}}{t^\alpha + C_1 C_2} \leq \sum_{t=i+1}^{\infty} \frac{C_1 e^{\frac{\epsilon}{2}}}{t^\alpha}$$

and, since $\alpha > 1$ the right hand side converges, hence δ converges to a non-zero number. Let now $f(n) = \prod_{t=i+1}^{n} \left(1 - \frac{C_1 e^{\frac{\epsilon}{2}}}{t^\alpha + C_1 C_2}\right)_+$. To find the limit $f(\infty)$ we can first log-transform this function, and then upper bound the infinite sum with an integral before transforming back. Since $\log\left(1 - \frac{C_1 e^{\frac{\epsilon}{2}}}{t^\alpha + C_1 C_2}\right)$ is monotonically increasing in t, we have

$$\log(f(n)) = \sum_{t=i+1}^{n} \log\left(1 - \frac{C_1 e^{\frac{\epsilon}{2}}}{t^\alpha + C_1 C_2}\right)$$
$$< \int_{i+1}^{n} \log\left(1 - \frac{C_1 e^{\frac{\epsilon}{2}}}{t^\alpha + C_1 C_2}\right) dt \to \int_{i+1}^{\infty} \log\left(1 - \frac{C_1 e^{\frac{\epsilon}{2}}}{t^\alpha + C_1 C_2}\right) dt.$$

This integral can be written in closed form using the hypergeometric function, or approximated numerically.

The convergence result that we get is slightly conservative, since δ^* in Equation (13) is an upper bound. However, following the previous proof, we can find an easy lower bound by just noticing that $\log(f(\infty)) > \int_i^{\infty} \log\left(1 - \frac{C_1 e^{\frac{\epsilon}{2}}}{t^\alpha + C_1 C_2}\right) dt$. When i is not too small, the difference between the upper and lower bound is negligible, as it is confirmed by what we see in the left plot of Figure 4, where the convergence to the upper bound appears to be impeccable. Since the convergence is not exactly to δ^*, we cannot find an explicit convergence rate the same way we did in Section 4. However, we see in the right plot of Figure 4 that the convergence rate empirically appears to be $1/\log(n)$.

6.2 Gaussian Noise

When working with the Gaussian noises, the cumbersome form of the functions in (4) does not prevent us from finding a closed form solution for the limit δ^*. Just as in the Laplace case we can find a conservative upper bound for δ^* which is very close to the true limit, as confirmed by the left plot of Figure 5. Just as before, we notice again empirically from the right plot of Figure 5 that the convergence rate is $1/\log(n)$.

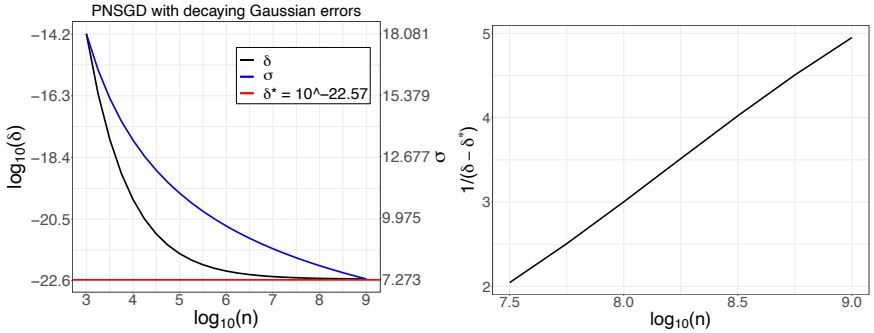

Fig. 5. (left) Convergence of δ to δ^* defined in (15). We report in black the behavior of δ and in blue that of σ_n defined in (14). The parameters considered are $L = 10, \beta = 0.5, \rho = 0, \epsilon = 1, \eta = 0.01, \alpha = 1.5, D_{\mathbb{K}} = 1, i = 100, C_1 = 100$ and $C_2 = 100$. (right) The convergence rate is approximately $1/\log(n)$.

Theorem 7. *Consider the PNSGD where for update j we use Gaussian noise $N(0, \sigma_j^2)$, and*

$$\sigma_j = \frac{MD_{\mathbb{K}}}{2\eta\sqrt{W\left(\frac{j^{2\alpha}}{2\pi C_1^2} + C_2\right)}} \tag{14}$$

for $\alpha > 1$. Then as $n \to \infty$ the procedure is (ϵ, δ^*)-DP where

$$\delta^* = \theta_{e^\epsilon}\left(\frac{2L}{\sigma_i}\right) e^{\int_{i+1}^{\infty} \log\left(\theta_{e^\epsilon}\left(2\sqrt{W\left(\frac{x^{2\alpha}}{2\pi C_1^2} + C_2\right)}\right)\right) dx} \tag{15}$$

and i is the index where the neighboring datasets differ.

The proof of this result is in the Supplementary Material, and makes use again of Lemma 5 to show that asymptotically the terms B_t in (11) behave approximately as $1 - O(1/t^\alpha)$, so that convergence is guaranteed for the same reason as in Theorem 6.

7 Conclusion

In this work, we have studied the setting of privacy amplification by iteration in the formulation proposed by [3], and proved that their analysis of PNSGD also applies to the case where the data are shuffled first. This is a much more common practice than the randomly-stopped PNSGD, originally proposed, because of a clear advantage in terms of accuracy of the algorithm. We proved two asymptotic results on the decay rate of noises that we can use, either the Laplace or the Gaussian injected noise, in order to have asymptotic convergence to a non-trivial privacy bound when the size of the dataset grows. We then showed that these practical bounds can be combined using standard tools from the composition literature. Finally we also showed two result, again for Laplace or Gaussian noise, that can be obtained in an online setting when the noise does not have to be recalibrated for the whole dataset but just decayed for the new data.

Acknowledgement

The authors would like to thank Weijie Su for his advice and encouragements.

References

1. Abadi, M., Chu, A., Goodfellow, I., McMahan, H.B., Mironov, I., Talwar, K., Zhang, L.: Deep learning with differential privacy. In: Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security. pp. 308–318 (2016)
2. Ali, S.M., Silvey, S.D.: A general class of coefficients of divergence of one distribution from another. Journal of the Royal Statistical Society: Series B (Methodological) **28**(1), 131–142 (1966)
3. Asoodeh, S., Diaz, M., Calmon, F.P.: Privacy amplification of iterative algorithms via contraction coefficients. arXiv preprint arXiv:2001.06546 (2020)
4. Balle, B., Barthe, G., Gaboardi, M.: Privacy amplification by subsampling: Tight analyses via couplings and divergences. In: Advances in Neural Information Processing Systems. pp. 6277–6287 (2018)
5. Balle, B., Barthe, G., Gaboardi, M., Geumlek, J.: Privacy amplification by mixing and diffusion mechanisms. In: Advances in Neural Information Processing Systems. pp. 13298–13308 (2019)
6. Bu, Z., Dong, J., Long, Q., Su, W.J.: Deep learning with gaussian differential privacy. Harvard data science review **2020**(23) (2020)
7. Bun, M., Steinke, T.: Concentrated differential privacy: Simplifications, extensions, and lower bounds. In: Theory of Cryptography Conference. pp. 635–658. Springer (2016)
8. Chaudhuri, K., Mishra, N.: When random sampling preserves privacy. In: Annual International Cryptology Conference. pp. 198–213. Springer (2006)
9. Csiszár, I., Shields, P.C.: Information theory and statistics: A tutorial. Now Publishers Inc (2004)
10. Dong, J., Roth, A., Su, W.J.: Gaussian differential privacy. arXiv preprint arXiv:1905.02383 (2019)
11. Dwork, C., Kenthapadi, K., McSherry, F., Mironov, I., Naor, M.: Our data, ourselves: Privacy via distributed noise generation. In: Annual International Conference on the Theory and Applications of Cryptographic Techniques. pp. 486–503. Springer (2006)
12. Dwork, C., McSherry, F., Nissim, K., Smith, A.: Calibrating noise to sensitivity in private data analysis. In: Theory of cryptography conference. pp. 265–284. Springer (2006)
13. Dwork, C., Rothblum, G.N.: Concentrated differential privacy. arXiv preprint arXiv:1603.01887 (2016)
14. Erlingsson, Ú., Feldman, V., Mironov, I., Raghunathan, A., Talwar, K., Thakurta, A.: Amplification by shuffling: From local to central differential privacy via anonymity. In: Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms. pp. 2468–2479. SIAM (2019)
15. Feldman, V., Mironov, I., Talwar, K., Thakurta, A.: Privacy amplification by iteration. In: 2018 IEEE 59th Annual Symposium on Foundations of Computer Science (FOCS). pp. 521–532. IEEE (2018)

16. Kairouz, P., Oh, S., Viswanath, P.: The composition theorem for differential privacy. In: International conference on machine learning. pp. 1376–1385. PMLR (2015)
17. Koskela, A., Jälkö, J., Honkela, A.: Computing tight differential privacy guarantees using fft. In: International Conference on Artificial Intelligence and Statistics. pp. 2560–2569. PMLR (2020)
18. Mironov, I.: Rényi differential privacy. In: 2017 IEEE 30th Computer Security Foundations Symposium (CSF). pp. 263–275. IEEE (2017)
19. Raginsky, M.: Strong data processing inequalities and ϕ-sobolev inequalities for discrete channels. IEEE Transactions on Information Theory **62**(6), 3355–3389 (2016)
20. Wang, Y.X., Balle, B., Kasiviswanathan, S.P.: Subsampled rényi differential privacy and analytical moments accountant. In: The 22nd International Conference on Artificial Intelligence and Statistics. pp. 1226–1235. PMLR (2019)