Privacy Amplification via Iteration for Shuffled and Online PNSGD

Matteo Sordello, Zhiqi Bu, Jinshuo Dong
matteo.sordello91@gmail.com

University of Pennsylvania

ECML-PKDD

Differential Privacy in Machine Learning

Differential privacy is a tool to guarantee the privacy of individuals while releasing aggregate information about a dataset.

Differential privacy is a tool to guarantee the privacy of individuals while releasing aggregate information about a dataset.

- Datasets D and D' are neighboring, we write $D \sim D'$
- Mechanism $\mathcal{M}$ acting on the datasets
- P_D is the distribution induced of $\mathcal{M}(D)$

Differential privacy is a tool to guarantee the privacy of individuals while releasing aggregate information about a dataset.

- Datasets D and D' are neighboring, we write $D \sim D'$
- Mechanism $\mathcal{M}$ acting on the datasets
- P_D is the distribution induced of $\mathcal{M}(D)$

(ϵ, δ) Differential Privacy

A mechanism $\mathcal{M}$ is (ϵ, δ)-DP if and only if for any event A

$$P_D(A) \leq e^{\epsilon} \cdot P_{D'}(A) + \delta$$

f-Divergence and Contraction Coefficient

f-Divergence

The f-divergence between two probability distribution μ and ν is

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

f-Divergence

The f-divergence between two probability distribution μ and ν is

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

Contraction Coefficient

The **contraction coefficient** of kernel K under the f-divergence

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

f-Divergence and Contraction Coefficient

f-Divergence

The f-divergence between two probability distribution μ and ν is

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

Contraction Coefficient

The **contraction coefficient** of kernel K under the f-divergence

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

- sequence of Markov kernels $\{K_n\}$
- sequence of measures $\{\mu_n\}$ generated starting from μ_0 by applying $\mu_n = \mu_{n-1}K_n$

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

E_γ-Divergence and Differential Privacy

E_γ-Divergence

The E_γ-divergence is an f-divergence with $f(t) = (t - \gamma)_+$

$$E_\gamma(\mu \| \nu) = \sup_A \left[\mu(A) - \gamma \cdot \nu(A)\right]$$

E_γ-Divergence and Differential Privacy

E_γ-Divergence

The E_γ-divergence is an f-divergence with $f(t) = (t - \gamma)_+$

$$E_\gamma(\mu \| \nu) = \sup_A [\mu(A) - \gamma \cdot \nu(A)]$$

Connection with (ϵ, δ)-DP:

$$\mathcal{M} \text{ is } (\epsilon, \delta)\text{-DP if and only if } E_{e^\epsilon}(P_D \| P_{D'}) \leq \delta$$

E_γ-Divergence and Differential Privacy

E_γ-Divergence

The E_γ-divergence is an f-divergence with $f(t) = (t - \gamma)_+$

$$E_\gamma(\mu \| \nu) = \sup_A \left[\mu(A) - \gamma \cdot \nu(A)\right]$$

Connection with (ϵ, δ)-DP:

$$\mathcal{M} \text{ is } (\epsilon, \delta)\text{-DP if and only if } E_{e^\epsilon}(P_D \| P_{D'}) \leq \delta$$

- $\mathcal{M}$ is (ϵ, δ)-DP means that $P_D(A) \leq e^\epsilon \cdot P_{D'}(A) + \delta$ for any A
- $E_{e^\epsilon}(P_D \| P_{D'}) = \sup_A \left[P_D(A) - e^\epsilon \cdot P_{D'}(A)\right]$

Projected Noisy Stochastic Gradient Descent

Consider 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 $x \in \mathcal{X}$ and is "well behaved" (L-Lipschitz, ρ-strongly convex and with gradient β-Lipschitz).

Projected Noisy Stochastic Gradient Descent

Consider 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 $x \in \mathcal{X}$ and is "well behaved" (L-Lipschitz, ρ-strongly convex and with gradient β-Lipschitz).

1. SGD step with learning rate η

PNSGD

$$w_{t+1} = \quad w_t - \eta \nabla_w \ell(w_t, x_{t+1})$$

Projected Noisy Stochastic Gradient Descent

Consider 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 $x \in \mathcal{X}$ and is "well behaved" (L-Lipschitz, ρ-strongly convex and with gradient β-Lipschitz).

1. SGD step with learning rate η
2. injection of i.i.d. noise sampled from a known distribution to guarantee privacy

PNSGD

$$w_{t+1} = \quad w_t - \eta \nabla_w \ell(w_t, x_{t+1}) + \eta Z_{t+1}$$

Projected Noisy Stochastic Gradient Descent

Consider 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 $x \in \mathcal{X}$ and is "well behaved" (L-Lipschitz, ρ-strongly convex and with gradient β-Lipschitz).

1. SGD step with learning rate η
2. injection of i.i.d. noise sampled from a known distribution to guarantee privacy
3. projection $\Pi_{\mathbb{K}} : \mathcal{W} \to \mathbb{K}$ onto the subspace $\mathbb{K}$

PNSGD

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

Projected Noisy Stochastic Gradient Descent

Consider 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 $x \in \mathcal{X}$ and is "well behaved" (L-Lipschitz, ρ-strongly convex and with gradient β-Lipschitz).

1. SGD step with learning rate η
2. injection of i.i.d. noise sampled from a known distribution to guarantee privacy
3. projection $\Pi_{\mathbb{K}} : \mathcal{W} \to \mathbb{K}$ onto the subspace $\mathbb{K}$

PNSGD

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

Each PNSGD update can be written as a composition of Markov kernels by assuming that $w_0 \sim \mu_0$ and $w_t \sim \mu_t = \mu_0 K_{x_1} \dots K_{x_t}$

Projected Noisy Stochastic Gradient Descent

Consider 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 $x \in \mathcal{X}$ and is "well behaved" (L-Lipschitz, ρ-strongly convex and with gradient β-Lipschitz).

1. SGD step with learning rate η
2. injection of i.i.d. noise sampled from a known distribution to guarantee privacy
3. projection $\Pi_{\mathbb{K}} : \mathcal{W} \to \mathbb{K}$ onto the subspace $\mathbb{K}$

PNSGD

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

Each PNSGD update can be written as a composition of Markov kernels by assuming that $w_0 \sim \mu_0$ and $w_t \sim \mu_t = \mu_0 K_{x_1} \dots K_{x_t}$

Let $\{x_1, \dots, x_n\}$ and $\{x'_1, \dots, x'_n\}$ be equal except for index i where $x_i \neq x'_i$

$$D_f(\mu_0 K_{x_1} \dots K_{x_n} \| \mu_0 K_{x'_1} \dots K_{x'_n}) \leq \underbrace{D_f(\mu_0 K_{x_1} \dots K_{x_i} \| \mu_0 K_{x'_1} \dots K_{x'_i})}_{\leq A} \prod_{t=i+1}^{n} \underbrace{\eta_f(K_{x_t})}_{\leq B} = A \cdot B^{n-i}$$

Privacy Bounds for Shuffled PNSGD with Fixed Noise

$$Q(t) = 1 - \Phi(t) \quad \text{and} \quad \theta_\gamma(r) = Q\left(\frac{\log(\gamma)}{r} - \frac{r}{2}\right) - \gamma Q\left(\frac{\log(\gamma)}{r} + \frac{r}{2}\right)$$

Privacy Bounds for Shuffled PNSGD with Fixed Noise

$$Q(t) = 1 - \Phi(t) \quad \text{and} \quad \theta_\gamma(r) = Q\left(\frac{\log(\gamma)}{r} - \frac{r}{2}\right) - \gamma Q\left(\frac{\log(\gamma)}{r} + \frac{r}{2}\right)$$

Theorem:

Let $D \sim D'$ be of size n. Then the shuffled PNSGD with fixed level of injected noise for all updates is (ϵ, δ)-DP with

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

Gaussian noise $N(0, \sigma^2)$ **on** $\mathbb{K} \subset \mathbb{R}^d$**:**

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

Laplace noise $L(0, v)$ **on** $\mathbb{K} = [a, b]$**:**

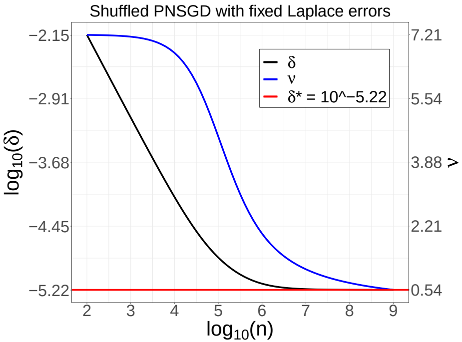
$$A = \left(1 - e^{\frac{\epsilon}{2} - \frac{L}{v}}\right)_+ \quad \text{and} \quad B = \left(1 - e^{\frac{\epsilon}{2} - \frac{M(b-a)}{2\eta v}}\right)_+$$

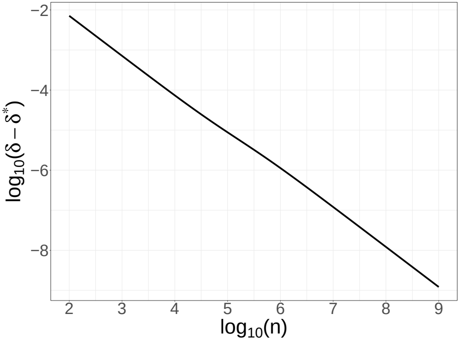
Shuffled PNSGD with Laplace Noise

$$v(n) = O\left(\frac{1}{\log\left(n/C_1\right)}\right) \quad \Rightarrow \quad \delta = \frac{1 - e^{-C_1 \exp(\epsilon/2)}}{C_1 e^{\frac{\epsilon}{2}}} + O\left(\frac{1}{n}\right)$$

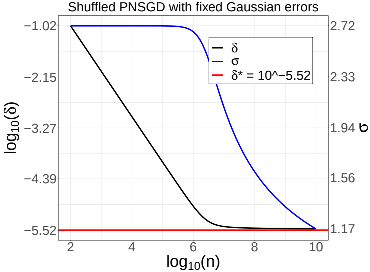
Shuffled PNSGD with Laplace Noise

$$v(n) = O\left(\frac{1}{\log\left(n/C_1\right)}\right) \quad \Rightarrow \quad \delta = \frac{1 - e^{-C_1 \exp(\epsilon/2)}}{C_1 e^{\frac{\epsilon}{2}}} + O\left(\frac{1}{n}\right)$$



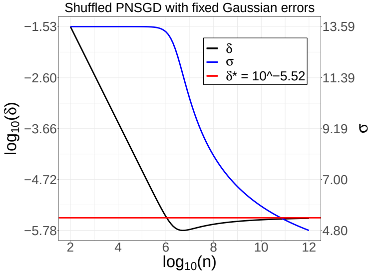


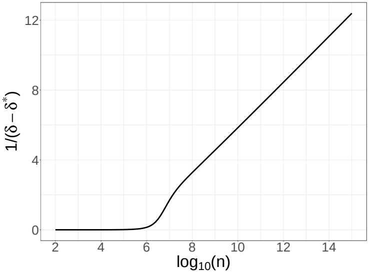
Shuffled PNSGD with Gaussian Noise



$$\sigma(n) = O\left(W\left(\frac{n^2}{2\pi C_1^2}\right)^{-1/2}\right)$$

$$\delta = \frac{1 - e^{-2C_1 e^{\frac{\epsilon}{2}}}}{2C_1 e^{\frac{\epsilon}{2}}} + O\left(\frac{1}{\log(n)}\right)$$





Online Results with Decaying Noise

- Online setting where data are added sequentially to dataset D
- Allow the noise level to be different for different entries, no need to re-calibrate for all x_i when n grows

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

Online Results with Decaying Noise

- Online setting where data are added sequentially to dataset D
- Allow the noise level to be different for different entries, no need to re-calibrate for all x_i when n grows

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

Gaussian noise $N(0, \sigma_j^2)$ for individual x_j on $\mathbb{K} \subset \mathbb{R}^d$:

$$A_j = \theta_{e^\epsilon}\left(\frac{2L}{\sigma_j}\right) \quad \text{and} \quad B_j = \theta_{e^\epsilon}\left(\frac{MD_{\mathbb{K}}}{\eta\sigma_j}\right)$$

Laplace noise $L(0, v_j)$ for individual x_j on $\mathbb{K} = [a, b]$:

$$A_j = \left(1 - e^{\frac{\epsilon}{2} - \frac{L}{v_j}}\right)_+ \quad \text{and} \quad B_j = \left(1 - e^{\frac{\epsilon}{2} - \frac{M(b-a)}{2\eta v_j}}\right)_+$$

Online PNSGD with Laplace Noise

For shuffled PNSGD: $\quad v(n) = O\left(\frac{1}{\log\left(n/C_1\right)}\right)$

Online PNSGD with Laplace Noise

For shuffled PNSGD: $$v(n) = O\left(\frac{1}{\log\left(n/C_1\right)}\right)$$

For online PNSGD, $\alpha > 1$

$$v_j = O\left(\frac{1}{\log\left(j^{\alpha}/C_1\right)}\right)$$

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

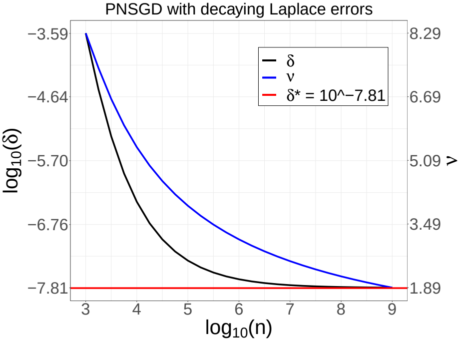
Online PNSGD with Laplace Noise

For shuffled PNSGD: $$v(n) = O\left(\frac{1}{\log\left(n/C_1\right)}\right)$$

For online PNSGD, $\alpha > 1$

$$v_j = O\left(\frac{1}{\log\left(j^\alpha/C_1\right)}\right)$$

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



Online PNSGD with Gaussian Noise

For shuffled PNSGD

$$\sigma(n) = O\left(W\left(\frac{n^2}{2\pi C_1^2}\right)^{-1/2}\right)$$

For online PNSGD, $\alpha > 1$

$$\sigma_j = O\left(W\left(\frac{j^{2\alpha}}{2\pi C_1^2}\right)^{-1/2}\right)$$

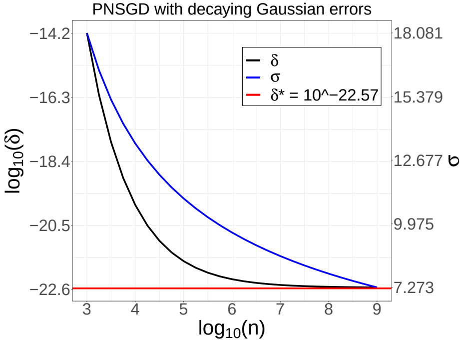
Online PNSGD with Gaussian Noise

For shuffled PNSGD

$$\sigma(n) = O\left(W\left(\frac{n^2}{2\pi C_1^2}\right)^{-1/2}\right)$$

For online PNSGD, $\alpha > 1$

$$\sigma_j = O\left(W\left(\frac{j^{2\alpha}}{2\pi C_1^2}\right)^{-1/2}\right)$$



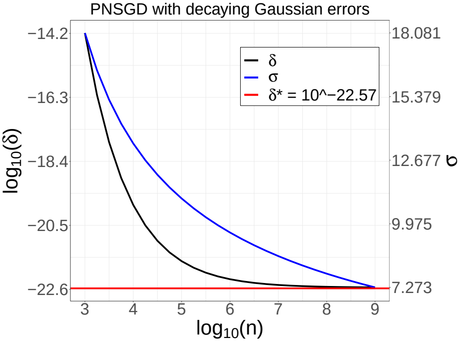
Online PNSGD with Gaussian Noise

For shuffled PNSGD

$$\sigma(n) = O\left(W\left(\frac{n^2}{2\pi C_1^2}\right)^{-1/2}\right)$$

For online PNSGD, $\alpha > 1$

$$\sigma_j = O\left(W\left(\frac{j^{2\alpha}}{2\pi C_1^2}\right)^{-1/2}\right)$$



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

Future Directions

- Extend the theory to consider PNSGD updates on mini-batches
- Larger simulations to investigate the practical usefulness of the results

Thank You!