Trustworthy Machine Learning

Differential Privacy 2

Sangdon Park

POSTECH

Contents from

A preliminary version of this paper appears in the proceedings of the *23rd ACM Conference on Computer and Communications Security (CCS 2016).* This is a full version.

Deep Learning with Differential Privacy

October 25, 2016

Martín Abadi[*]
H. Brendan McMahan[*]
Andy Chu[*]
Ilya Mironov[*]
Li Zhang[*]
Ian Goodfellow[†]
Kunal Talwar[*]

- (I guess) The first DP paper for deep learning
- We have seen the algorithm of this paper multiple times!

Difference?

- Convex loss
  - Add noise on the final model
  - Add noise before learning
  - Strategies in convex loss treat learning process as a block box
- Non-convex loss
  - Consider learning process as a white box for the careful(?) characterization of parameter updates.

Definition: Differential Privacy

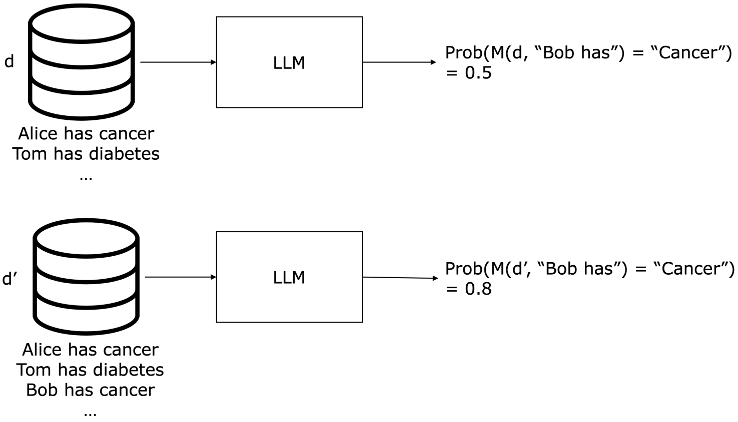
Definition

A randomized mechanism $\mathcal{M} : \mathcal{D} \to \mathcal{R}$ with domain $\mathcal{D}$ and range $\mathcal{R}$ satisfies (ε, δ)-differential privacy if for any two "adjacent" inputs $d, d' \in \mathcal{D}$ and for any subset of outputs $S \subseteq \mathcal{R}$ it holds that

$$\mathbb{P}\left\{\mathcal{M}(d) \in S\right\} \leq e^{\varepsilon} \mathbb{P}\left\{\mathcal{M}(d') \in S\right\} + \delta.$$

- "adjacent" inputs: two inputs differ in a single labeled example.

A Toy Example



- Here, the mechanism $\mathcal{M}$ includes training a LLM over a dataset and querying a question.
- At least we know that d' has Bob's information (and he likely has cancer due to the high confidence).

Differentially Private SGD (DP-SGD)

Algorithm 1 Differentially private SGD (Outline)

Input: Examples $\{x_1, \ldots, x_N\}$, loss function $\mathcal{L}(\theta) = \frac{1}{N}\sum_i \mathcal{L}(\theta, x_i)$. Parameters: learning rate η_t, noise scale σ, group size L, gradient norm bound C.

Initialize θ_0 randomly

for $t \in [T]$ **do**

Take a random sample L_t with sampling probability L/N

Compute gradient

For each $i \in L_t$, compute $\mathbf{g}_t(x_i) \leftarrow \nabla_{\theta_t}\mathcal{L}(\theta_t, x_i)$

Clip gradient

$\bar{\mathbf{g}}_t(x_i) \leftarrow \mathbf{g}_t(x_i) / \max\left(1, \frac{\|\mathbf{g}_t(x_i)\|_2}{C}\right)$

Add noise

$\tilde{\mathbf{g}}_t \leftarrow \frac{1}{L}\left(\sum_i \bar{\mathbf{g}}_t(x_i) + \mathcal{N}(0, \sigma^2 C^2 \mathbf{I})\right)$

Descent

$\theta_{t+1} \leftarrow \theta_t - \eta_t \tilde{\mathbf{g}}_t$

Output θ_T and compute the overall privacy cost (ε, δ) using a privacy accounting method.

Differentially Private SGD (DP-SGD)

Algorithm 1 Differentially private SGD (Outline)

Input: Examples $\{x_1, \ldots, x_N\}$, loss function $\mathcal{L}(\theta) = \frac{1}{N}\sum_i \mathcal{L}(\theta, x_i)$. Parameters: learning rate η_t, noise scale σ, group size L, gradient norm bound C.

Initialize θ_0 randomly

for $t \in [T]$ **do**

Take a random sample L_t with sampling probability L/N

Compute gradient

For each $i \in L_t$, compute $\mathbf{g}_t(x_i) \leftarrow \nabla_{\theta_t}\mathcal{L}(\theta_t, x_i)$

Clip gradient

$\bar{\mathbf{g}}_t(x_i) \leftarrow \mathbf{g}_t(x_i) / \max\left(1, \frac{\|\mathbf{g}_t(x_i)\|_2}{C}\right)$

Add noise

$\tilde{\mathbf{g}}_t \leftarrow \frac{1}{L}\left(\sum_i \bar{\mathbf{g}}_t(x_i) + \mathcal{N}(0, \sigma^2 C^2 \mathbf{I})\right)$

Descent

$\theta_{t+1} \leftarrow \theta_t - \eta_t \tilde{\mathbf{g}}_t$

Output θ_T and compute the overall privacy cost (ε, δ) using a privacy accounting method.

- $\mathcal{M}_t(d) \coloneqq \sum_{i \in L_t} \tilde{\mathbf{g}}_t(x_i) + \mathcal{N}(0, \sigma^2 C^2 I)$: the Gaussian mechanism with sampling

Differentially Private SGD (DP-SGD)

Algorithm 1 Differentially private SGD (Outline)

Input: Examples $\{x_1, \ldots, x_N\}$, loss function $\mathcal{L}(\theta) = \frac{1}{N}\sum_i \mathcal{L}(\theta, x_i)$. Parameters: learning rate η_t, noise scale σ, group size L, gradient norm bound C.

Initialize θ_0 randomly

for $t \in [T]$ **do**

Take a random sample L_t with sampling probability L/N

Compute gradient

For each $i \in L_t$, compute $\mathbf{g}_t(x_i) \leftarrow \nabla_{\theta_t}\mathcal{L}(\theta_t, x_i)$

Clip gradient

$\bar{\mathbf{g}}_t(x_i) \leftarrow \mathbf{g}_t(x_i) / \max\left(1, \frac{\|\mathbf{g}_t(x_i)\|_2}{C}\right)$

Add noise

$\tilde{\mathbf{g}}_t \leftarrow \frac{1}{L}\left(\sum_i \bar{\mathbf{g}}_t(x_i) + \mathcal{N}(0, \sigma^2 C^2 \mathbf{I})\right)$

Descent

$\theta_{t+1} \leftarrow \theta_t - \eta_t \tilde{\mathbf{g}}_t$

Output θ_T and compute the overall privacy cost (ε, δ) using a privacy accounting method.

- $\mathcal{M}_t(d) \coloneqq \sum_{i \in L_t} \tilde{\mathbf{g}}_t(x_i) + \mathcal{N}(0, \sigma^2 C^2 I)$: the Gaussian mechanism with sampling
- Why clipping?

Differentially Private SGD (DP-SGD)

Algorithm 1 Differentially private SGD (Outline)

Input: Examples $\{x_1, \ldots, x_N\}$, loss function $\mathcal{L}(\theta) = \frac{1}{N}\sum_i \mathcal{L}(\theta, x_i)$. Parameters: learning rate η_t, noise scale σ, group size L, gradient norm bound C.

Initialize θ_0 randomly

for $t \in [T]$ **do**

Take a random sample L_t with sampling probability L/N

Compute gradient

For each $i \in L_t$, compute $\mathbf{g}_t(x_i) \leftarrow \nabla_{\theta_t}\mathcal{L}(\theta_t, x_i)$

Clip gradient

$\bar{\mathbf{g}}_t(x_i) \leftarrow \mathbf{g}_t(x_i) / \max\left(1, \frac{\|\mathbf{g}_t(x_i)\|_2}{C}\right)$

Add noise

$\tilde{\mathbf{g}}_t \leftarrow \frac{1}{L}\left(\sum_i \bar{\mathbf{g}}_t(x_i) + \mathcal{N}(0, \sigma^2 C^2 \mathbf{I})\right)$

Descent

$\theta_{t+1} \leftarrow \theta_t - \eta_t \tilde{\mathbf{g}}_t$

Output θ_T and compute the overall privacy cost (ε, δ) using a privacy accounting method.

- $\mathcal{M}_t(d) \coloneqq \sum_{i \in L_t} \tilde{\mathbf{g}}_t(x_i) + \mathcal{N}(0, \sigma^2 C^2 I)$: the Gaussian mechanism with sampling
- Why clipping?
- How to determine the noise level?

Differentially Private SGD (DP-SGD)

Algorithm 1 Differentially private SGD (Outline)

Input: Examples $\{x_1, \ldots, x_N\}$, loss function $\mathcal{L}(\theta) = \frac{1}{N}\sum_i \mathcal{L}(\theta, x_i)$. Parameters: learning rate η_t, noise scale σ, group size L, gradient norm bound C.

Initialize θ_0 randomly

for $t \in [T]$ **do**

Take a random sample L_t with sampling probability L/N

Compute gradient

For each $i \in L_t$, compute $\mathbf{g}_t(x_i) \leftarrow \nabla_{\theta_t}\mathcal{L}(\theta_t, x_i)$

Clip gradient

$\bar{\mathbf{g}}_t(x_i) \leftarrow \mathbf{g}_t(x_i) / \max\left(1, \frac{\|\mathbf{g}_t(x_i)\|_2}{C}\right)$

Add noise

$\tilde{\mathbf{g}}_t \leftarrow \frac{1}{L}\left(\sum_i \bar{\mathbf{g}}_t(x_i) + \mathcal{N}(0, \sigma^2 C^2 \mathbf{I})\right)$

Descent

$\theta_{t+1} \leftarrow \theta_t - \eta_t \tilde{\mathbf{g}}_t$

Output θ_T and compute the overall privacy cost (ε, δ) using a privacy accounting method.

- $\mathcal{M}_t(d) := \sum_{i \in L_t} \tilde{\mathbf{g}}_t(x_i) + \mathcal{N}(0, \sigma^2 C^2 I)$: the Gaussian mechanism with sampling
- Why clipping?
- How to determine the noise level?
- How to compute privacy cost?

Main Ingredient: Norm Clipping

$$\tilde{\mathbf{g}}_t(x_i) \leftarrow \frac{\mathbf{g}_t(x_i)}{\max\left(1, \frac{\|\mathbf{g}_t(x_i)\|_2}{C}\right)}$$

- Maintain the norm of gradients to be at most C, *i.e.*,

$$\frac{\mathbf{g}}{\max\left(1, \frac{\|\mathbf{g}\|_2}{C}\right)} = \begin{cases} \mathbf{g} & \text{if } \|\mathbf{g}\|_2 \leq C \\ \frac{C}{\|\mathbf{g}\|_2}\mathbf{g} & \text{if } \|\mathbf{g}\|_2 > C \end{cases}$$

Main Ingredient: Norm Clipping

$$\tilde{\mathbf{g}}_t(x_i) \leftarrow \frac{\mathbf{g}_t(x_i)}{\max\left(1, \frac{\|\mathbf{g}_t(x_i)\|_2}{C}\right)}$$

- Maintain the norm of gradients to be at most C, *i.e.*,

$$\frac{\mathbf{g}}{\max\left(1, \frac{\|\mathbf{g}\|_2}{C}\right)} = \begin{cases} \mathbf{g} & \text{if } \|\mathbf{g}\|_2 \leq C \\ \frac{C}{\|\mathbf{g}\|_2}\mathbf{g} & \text{if } \|\mathbf{g}\|_2 > C \end{cases}$$

- Limit “privacy loss” at each learning iteration for a tighter the DP guarantee

Main Ingredient: Norm Clipping

$$\tilde{\mathbf{g}}_t(x_i) \leftarrow \frac{\mathbf{g}_t(x_i)}{\max\left(1, \frac{\|\mathbf{g}_t(x_i)\|_2}{C}\right)}$$

- Maintain the norm of gradients to be at most C, *i.e.*,

$$\frac{\mathbf{g}}{\max\left(1, \frac{\|\mathbf{g}\|_2}{C}\right)} = \begin{cases} \mathbf{g} & \text{if } \|\mathbf{g}\|_2 \leq C \\ \frac{C}{\|\mathbf{g}\|_2}\mathbf{g} & \text{if } \|\mathbf{g}\|_2 > C \end{cases}$$

- Limit "privacy loss" at each learning iteration for a tighter the DP guarantee
  - If the norm of gradients is "large", we need to add "large" noise to cover them (otherwise, privacy leaking)

Main Ingredient: Norm Clipping

$$\tilde{\mathbf{g}}_t(x_i) \leftarrow \frac{\mathbf{g}_t(x_i)}{\max\left(1, \frac{\|\mathbf{g}_t(x_i)\|_2}{C}\right)}$$

- Maintain the norm of gradients to be at most C, *i.e.*,

$$\frac{\mathbf{g}}{\max\left(1, \frac{\|\mathbf{g}\|_2}{C}\right)} = \begin{cases} \mathbf{g} & \text{if } \|\mathbf{g}\|_2 \leq C \\ \frac{C}{\|\mathbf{g}\|_2}\mathbf{g} & \text{if } \|\mathbf{g}\|_2 > C \end{cases}$$

- Limit "privacy loss" at each learning iteration for a tighter the DP guarantee
  - If the norm of gradients is "large", we need to add "large" noise to cover them (otherwise, privacy leaking)
  - Without clipping, we need to add noise proportional to the largest norm of gradients.

Main Ingredient: Norm Clipping

$$\tilde{\mathbf{g}}_t(x_i) \leftarrow \frac{\mathbf{g}_t(x_i)}{\max\left(1, \frac{\|\mathbf{g}_t(x_i)\|_2}{C}\right)}$$

- Maintain the norm of gradients to be at most C, *i.e.*,

$$\frac{\mathbf{g}}{\max\left(1, \frac{\|\mathbf{g}\|_2}{C}\right)} = \begin{cases} \mathbf{g} & \text{if } \|\mathbf{g}\|_2 \leq C \\ \frac{C}{\|\mathbf{g}\|_2}\mathbf{g} & \text{if } \|\mathbf{g}\|_2 > C \end{cases}$$

- Limit "privacy loss" at each learning iteration for a tighter the DP guarantee
  - If the norm of gradients is "large", we need to add "large" noise to cover them (otherwise, privacy leaking)
  - Without clipping, we need to add noise proportional to the largest norm of gradients.
  - With clipping, (as we control the maximum of the norm) we can choose a smaller noise level.

Main Ingredient: Norm Clipping

$$\tilde{\mathbf{g}}_t(x_i) \leftarrow \frac{\mathbf{g}_t(x_i)}{\max\left(1, \frac{\|\mathbf{g}_t(x_i)\|_2}{C}\right)}$$

- Maintain the norm of gradients to be at most C, *i.e.*,

$$\frac{\mathbf{g}}{\max\left(1, \frac{\|\mathbf{g}\|_2}{C}\right)} = \begin{cases} \mathbf{g} & \text{if } \|\mathbf{g}\|_2 \leq C \\ \frac{C}{\|\mathbf{g}\|_2}\mathbf{g} & \text{if } \|\mathbf{g}\|_2 > C \end{cases}$$

- Limit "privacy loss" at each learning iteration for a tighter the DP guarantee
  - If the norm of gradients is "large", we need to add "large" noise to cover them (otherwise, privacy leaking)
  - Without clipping, we need to add noise proportional to the largest norm of gradients.
  - With clipping, (as we control the maximum of the norm) we can choose a smaller noise level.
- Clipping before averaging

Main Ingredient: Norm Clipping

$$\tilde{\mathbf{g}}_t(x_i) \leftarrow \frac{\mathbf{g}_t(x_i)}{\max\left(1, \frac{\|\mathbf{g}_t(x_i)\|_2}{C}\right)}$$

- Maintain the norm of gradients to be at most C, *i.e.*,

$$\frac{\mathbf{g}}{\max\left(1, \frac{\|\mathbf{g}\|_2}{C}\right)} = \begin{cases} \mathbf{g} & \text{if } \|\mathbf{g}\|_2 \leq C \\ \frac{C}{\|\mathbf{g}\|_2}\mathbf{g} & \text{if } \|\mathbf{g}\|_2 > C \end{cases}$$

- Limit "privacy loss" at each learning iteration for a tighter the DP guarantee
  - If the norm of gradients is "large", we need to add "large" noise to cover them (otherwise, privacy leaking)
  - Without clipping, we need to add noise proportional to the largest norm of gradients.
  - With clipping, (as we control the maximum of the norm) we can choose a smaller noise level.
- Clipping before averaging
  - Probably provides a tighter DP guarantee.

Privacy Analysis

Finding a DP algorithm $\rightarrow$ bounding some quantity, called moments of privacy loss!

1. Bounding the quantity for each learning iteration
2. Bounding the quantity for all learning iterations

Measuring DP: Privacy Loss

$$\ell(o; \mathcal{M}, \mathbf{aux}, d, d') := \log \frac{\mathbb{P}\left\{\mathcal{M}(\mathbf{aux}, d) = o\right\}}{\mathbb{P}\left\{\mathcal{M}(\mathbf{aux}, d') = o\right\}}$$

- $d, d' \in \mathcal{D}$: neighboring datasets
- $\mathcal{M}$: a mechanism
- $\mathbf{aux}$: an auxiliary input, *e.g.*, previous gradients
- $o \in \mathcal{R}$: an outcome

Measuring DP: Privacy Loss

$$\ell(o; \mathcal{M}, \mathbf{aux}, d, d') := \log \frac{\mathbb{P}\left\{\mathcal{M}(\mathbf{aux}, d) = o\right\}}{\mathbb{P}\left\{\mathcal{M}(\mathbf{aux}, d') = o\right\}}$$

- $d, d' \in \mathcal{D}$: neighboring datasets
- $\mathcal{M}$: a mechanism
- $\mathbf{aux}$: an auxiliary input, *e.g.*, previous gradients
- $o \in \mathcal{R}$: an outcome
- We consider o and $\ell(o; \cdot)$ are random variables (where $o \sim \mathcal{M}(\mathbf{aux}, d)$).

Measuring DP: Privacy Loss

$$\ell(o; \mathcal{M}, \mathbf{aux}, d, d') := \log \frac{\mathbb{P}\left\{\mathcal{M}(\mathbf{aux}, d) = o\right\}}{\mathbb{P}\left\{\mathcal{M}(\mathbf{aux}, d') = o\right\}}$$

- $d, d' \in \mathcal{D}$: neighboring datasets
- $\mathcal{M}$: a mechanism
- $\mathbf{aux}$: an auxiliary input, *e.g.*, previous gradients
- $o \in \mathcal{R}$: an outcome
- We consider o and $\ell(o; \cdot)$ are random variables (where $o \sim \mathcal{M}(\mathbf{aux}, d)$).
- How to capture the properties of the privacy loss?

Measruing DP: Moments

$$\alpha_{\mathcal{M}}(\lambda) = \max_{\texttt{aux},d,d'} \alpha_{\mathcal{M}}(\lambda; \texttt{aux}, d, d') \quad \text{where}$$

$$\alpha_{\mathcal{M}}(\lambda; \texttt{aux}, d, d') := \log \mathbb{E}_{o \sim \mathcal{M}(\texttt{aux},d)} \ell(o; \mathcal{M}, \texttt{aux}, d, d')$$

Measruing DP: Moments

$$\alpha_{\mathcal{M}}(\lambda) = \max_{\textbf{aux},d,d'} \alpha_{\mathcal{M}}(\lambda; \textbf{aux}, d, d') \quad \text{where}$$

$$\alpha_{\mathcal{M}}(\lambda; \textbf{aux}, d, d') := \log \mathbb{E}_{o \sim \mathcal{M}(\textbf{aux},d)} \ell(o; \mathcal{M}, \textbf{aux}, d, d')$$

- The *moment-generating function* (or moments) of a real-valued random variable X, denoted by $M_X(\lambda)$, captures the useful properties of the corresponding distribution.

$$\begin{aligned} M_X(\lambda) &= \mathbb{E}\{e^{\lambda X}\} \\ &= \mathbb{E}\left\{1 + \lambda X + \frac{\lambda^2 X^2}{2!} + \frac{\lambda^3 X^3}{3!} + \cdots\right\} \\ &= 1 + \lambda \mathbb{E}\{X\} + \frac{\lambda^2 \mathbb{E}\{X^2\}}{2!} + \frac{\lambda^3 \mathbb{E}\{X^3\}}{3!} + \cdots \end{aligned}$$

  - To obtain mean, differentiating $M_X(\lambda)$ once with respect to λ and setting $\lambda = 0$.

From the Moments to the DP Guarantee

Theorem

For any $\varepsilon > 0$, the mechanism $\mathcal{M}$ is (ε, δ)-DP for

$$\delta = \min_{\lambda} e^{\alpha_{\mathcal{M}}(\lambda) - \lambda\varepsilon}.$$

- Connect (ε, δ)-DP to $\alpha_{\mathcal{M}}(\lambda)$
- Given δ, if we know the moments $\alpha_{\mathcal{M}}(\lambda)$, the privacy parameter ε is determined.
- How to compute or bound $\alpha_{\mathcal{M}}(\lambda)$?

From the Moment Bound to the DP Guarantee: A Proof Sketch

- Recall the privacy loss ℓ

$$\ell(o; \mathcal{M}, \mathbf{aux}, d, d') := \log \frac{\mathbb{P}\left\{\mathcal{M}(\mathbf{aux}, d) = o\right\}}{\mathbb{P}\left\{\mathcal{M}(\mathbf{aux}, d') = o\right\}}$$

- An event $B := \ell(o; \cdot) \geq \varepsilon$
- For any S,

$$\begin{aligned}
\mathbb{P}\left\{\mathcal{M}(d) \in S\right\} &= \mathbb{P}\left\{\mathcal{M}(d) \in S \cap B^c\right\} + \mathbb{P}\left\{\mathcal{M}(d) \in S \cap B\right\} \\
&\leq e^{\varepsilon}\mathbb{P}\left\{\mathcal{M}(d') \in S \cap B^c\right\} + \mathbb{P}\left\{\mathcal{M}(d) \in S \cap B\right\} \\
&\leq e^{\varepsilon}\mathbb{P}\left\{\mathcal{M}(d') \in S\right\} + \mathbb{P}\left\{\mathcal{M}(d) \in B\right\} \\
&\leq e^{\varepsilon}\mathbb{P}\left\{\mathcal{M}(d') \in S\right\} + e^{\alpha_{\mathcal{M}}(\lambda) - \lambda\varepsilon}
\end{aligned}$$

- The last inequality holds since

$$\mathbb{P}_{o \sim \mathcal{M}(d)}\left\{\ell(o) \geq \varepsilon\right\} = \mathbb{P}_{o \sim \mathcal{M}(d)}\left\{e^{\lambda \ell(o)} \geq e^{\lambda\varepsilon}\right\} \leq \frac{\mathbb{E}_{o \sim \mathcal{M}(d)}\{e^{\lambda \ell(o)}\}}{e^{\lambda\varepsilon}} \leq e^{\alpha_{\mathcal{M}}(\lambda) - \lambda\varepsilon},$$

where the first inequality holds due to the Markov's inequality.

Mechanism

One-step Mechanism

$$\mathcal{M}_t(d) := \sum_{i \in L_t} \tilde{\mathbf{g}}_t(x_i) + \mathcal{N}(0, \sigma^2 C^2 I)$$

- This is (almost) the Gaussian mechanism.
- We know it is DP.

Multi-step Mechanism

$$\mathcal{M}(d) \propto \sum_{t=1}^{T} (-\eta_t) \mathcal{M}_t(d)$$

- This is the composition of the Gaussian mechanisms.
- Is it DP?

Compobility Theorem

Theorem

Suppose that a mechanism $\mathcal{M}$ consists of a sequence of adaptive mechanisms, i.e., $\mathcal{M} := (\mathcal{M}_1, \ldots, \mathcal{M}_T)$, where $\mathcal{M}_t : \mathcal{R}_1 \times \cdots \times \mathcal{R}_{t-1} \times \mathcal{D} \to \mathcal{R}_t$. Then, for any $\lambda > 0$

$$\alpha_{\mathcal{M}}(\lambda) \leq \sum_{t=1}^{T} \alpha_{\mathcal{M}_t}(\lambda)$$

- "Adaptive" mechanism: a mechanism that depends on all previous mechanisms

$$\begin{aligned} \mathbf{aux}_2 &= \mathcal{M}_1(\mathbf{aux}_1, d) \\ \mathbf{aux}_3 &= \mathcal{M}_2(\mathbf{aux}_2, d) = \mathcal{M}_2(\mathcal{M}_1(\mathbf{aux}_1, d), d) \\ &\ldots \end{aligned}$$

- $\mathcal{M}$: *e.g.*, T-step gradient aggregation
- $\mathcal{M}_t$: *e.g.*, one-step gradient aggregation

Composibility Theorem: A Proof Sketch 1/2

- $\mathcal{M}_{1:t} \coloneqq (\mathcal{M}_1, \ldots, \mathcal{M}_t)$
- $o_{1:t} \coloneqq (o_1, \ldots, o_t)$
- For any neighboring datasets $d, d' \in \mathcal{D}$ and outputs $o_{1:T}$, we have

$$\begin{aligned}
\ell(o_{1:T}; \mathcal{M}_{1:T}, o_{1:T-1}, d, d') &= \log \frac{\mathbb{P}\left\{\mathcal{M}_{1:T}(o_{1:T-1}, d) = o_{1:T}\right\}}{\mathbb{P}\left\{\mathcal{M}_{1:T}(o_{1:T-1}, d') = o_{1:T}\right\}} \\
&= \log \prod_{t=1}^{T} \frac{\mathbb{P}\left\{\mathcal{M}_t(o_{1:t-1}, d) = o_t \mid \mathcal{M}_{1:t-1}(o_{1:t-2}, d) = o_{1:t-1}\right\}}{\mathbb{P}\left\{\mathcal{M}_t(o_{1:t-1}, d') = o_t \mid \mathcal{M}_{1:t-1}(o_{1:t-2}, d') = o_{1:t-1}\right\}} \\
&= \sum_{t=1}^{T} \log \frac{\mathbb{P}\left\{\mathcal{M}_t(o_{1:t-1}, d) = o_t \mid \mathcal{M}_{1:t-1}(o_{1:t-2}, d) = o_{1:t-1}\right\}}{\mathbb{P}\left\{\mathcal{M}_t(o_{1:t-1}, d') = o_t \mid \mathcal{M}_{1:t-1}(o_{1:t-2}, d') = o_{1:t-1}\right\}} \\
&= \sum_{t=1}^{T} \ell(o_t; \mathcal{M}_t, o_{1:t-1}, d, d').
\end{aligned}$$

Composibility Theorem: A Proof Sketch (2/2)

- Bound $\alpha_{\mathcal{M}}(\lambda) = \alpha_{\mathcal{M}_{1:T}}(\lambda)$ as follows

$$
\begin{aligned}
\log \mathop{\mathbb{E}}_{o'_{1:T} \sim \mathcal{M}_{1:T}(\cdot)} \left\{ e^{\lambda \ell(o'_{1:T}; \mathcal{M}_{1:T}, d, d')} \right\} &= \log \mathop{\mathbb{E}}_{o'_{1:T} \sim \mathcal{M}_{1:T}(\cdot)} \left\{ e^{\lambda \sum_{t=1}^{T} \ell(o'_t; \mathcal{M}_t, o_{1:t-1}, d, d')} \right\} \\
&= \log \mathop{\mathbb{E}}_{o'_{1:T} \sim \mathcal{M}_{1:T}(\cdot)} \left\{ \prod_{t=1}^{T} e^{\lambda \ell(o'_t; \mathcal{M}_t, o_{1:t-1}, d, d')} \right\} \\
&= \log \prod_{t=1}^{T} \mathop{\mathbb{E}}_{o'_t \sim \mathcal{M}_t(\cdot)} \left\{ e^{\lambda \ell(o'_t; \mathcal{M}_t, o_{1:t-1}, d, d')} \right\} \\
&= \log \prod_{t=1}^{T} e^{\alpha_{\mathcal{M}_t}(\lambda; o_{1:t-1}, d, d')} \\
&= \sum_{t=1}^{T} \alpha_{\mathcal{M}_t}(\lambda; o_{1:t-1}, d, d').
\end{aligned}
$$

Main Theorem

Theorem

There exist constants c_1 and c_2 so that given the sampling probability $q = L/N$ and the number of steps T, for any $\varepsilon < c_2 q^2 T$, Algorithm 1 is (ε, δ)-differentially private for any $\delta > 0$ if we choose

$$\sigma \geq c_2 \frac{q\sqrt{T \log 1/\delta}}{\varepsilon}$$

- Provide intuition on tuning nobs.
- $\varepsilon \propto T$: privacy-accuracy trade-off
- With the standard composition, we need

$$\sigma = \Omega\left(\frac{q\sqrt{T \log(1/\delta) \log(T/\delta)}}{\varepsilon}\right)$$

  - Clipping makes the difference?

Practical Summary

- The moments bound:

$$\alpha_{\mathcal{M}}(\lambda) \le \sum_{i=1}^{T} \alpha_{\mathcal{M}_i}(\lambda)$$

- For the Gaussian mechanism with random sampling

$$\alpha_{\mathcal{M}_i}(\lambda) = \log \max \left(\mathbb{E}_{z\sim\mu_0} \left(\frac{\mu_0(z)}{(1-q)\mu_0(z) + q\mu_1(z)} \right)^{\lambda}, \mathbb{E}_{z\sim\mu} \left(\frac{\mu(z)}{\mu_0(z)} \right)^{\lambda} \right),$$

where $\mu_0 \coloneqq \mathcal{N}(0,\sigma^2)$, $\mu_1 \coloneqq \mathcal{N}(1,\sigma^2)$, and $\mu(z) \coloneqq (1-q)\mu_0(z) + q\mu_1(z)$.

- From the "Moment-DP" theorem, $\mathcal{M}$ is (ε,δ)-DP if

$$\min_{\lambda} e^{\alpha_{\mathcal{M}}(\lambda) - \lambda\varepsilon} \le \min_{\lambda} e^{\sum_{i=1}^{T} \alpha_{\mathcal{M}_i}(\lambda) - \lambda\varepsilon} = \delta.$$

  - *e.g.*, if T, q, σ, and δ are given, we can compute ε.
  - greedy search over $\lambda \le 32$

(Proposed) Moments Accountant v.s. (Standard) Strong Composition

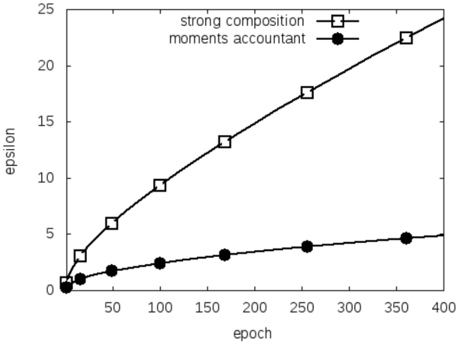


Figure 2: The ε value as a function of epoch E for $q = 0.01$, $\sigma = 4$, $\delta = 10^{-5}$, using the strong composition theorem and the moments accountant respectively.

(Proposed) Moments Accountant v.s. (Standard) Strong Composition

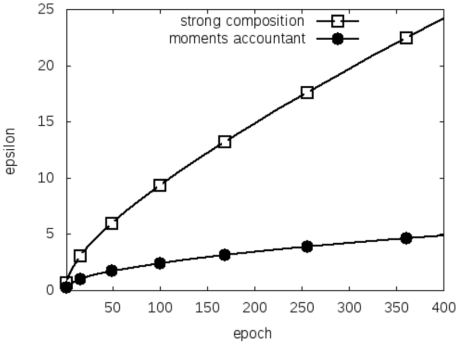


Figure 2: The ε value as a function of epoch E for $q = 0.01$, $\sigma = 4$, $\delta = 10^{-5}$, using the strong composition theorem and the moments accountant respectively.

- How about the comparison of model accuracy? Clipping may hurt accuracy.

Conclusion

- The proposed "Moments Accountant" has a stronger DP guarantee.
  - Why? practical treatments on clipping
- Nice connection between a moments bound and the DP guarantee.