Trustworthy Machine Learning From an Optimization Lens

Meisam Razaviyayn

Lecture 11: DP-SGD, Privacy Amplification, and Privacy Accountant

razaviya@usc.edu

Recap: (Approximate) differential privacy

Definition: Let $\epsilon, \delta > 0$ and X be the set of possible datasets. A randomized algorithm $M(\cdot): X \to O$ is said to be $(\epsilon, \delta) -$ differentially private if

$\Pr(M(D) \in \Omega) \leq e^{\epsilon} \Pr(M(D') \in \Omega) + \delta$ for all $\Omega \subseteq O$ and all neighboring datasets $D, D' \in X$

Connection to pure DP

- **Definition**: Consider two fixed datasets D, D', and a randomized mechanism M. The privacy loss random variable (PLRV) draws an outcome o from $M(D)$ and outputs $\ln\left(\frac{P(M(D)=o)}{P(M(D')=o)}\right)$. In other words, the random variable takes the value of $\ln\left(\frac{P(M(D)=o)}{P(M(D')=o)}\right)$ with probability $P(M(D) = o)$.

- **Theorem:** If privacy loss is bounded by ϵ with probability $\geq 1 - \delta$, then the algorithm is $(\epsilon, \delta) -$ DP

Converse? Proof

Recap: Gaussian Mechanism

Definition: Let $f: X \to R^k$. The $\ell_2 -$ sensitivity of f is defined as

$$\Delta_2 = \sup_{D,D'} \| f(D) - f(D') \|_2$$

where the supremum is taken over all neighboring datasets D and D'

Recall: Zero-mean Gaussian distribution with variance σ^2: $p(x) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{x^2}{2\sigma^2}\right)$

Theorem: Let $f: X \to R^k$ have the $\ell_2 -$ sensitivity Δ_2 and

$$M(D) = f(D) + (Z_1, .., Z_k)$$

Where $Z_1, \ldots, Z_k$ are iid Gaussian with variance $\sigma = \frac{\Delta_2 \sqrt{2 \ln \frac{1.25}{\delta}}}{\epsilon}$. Then $M(\cdot)$ is $(\epsilon, \delta) -$ DP.

Example: *Mean estimation*: If data is bounded ($\left|\left|x_i\right|\right|_2 \leq c$), the sensitivity of the mean (assuming neighboring datasets defined by replacement) is $\Delta_2 = 2c/n$

Recap: Properties of approximate DP

Post-Processing: Let $M: X \to O$ be $(\epsilon, \delta) -$ DP and let $G: O \to T$ be an arbitrary (potentially randomized) mapping. Then, $G(M(\cdot))$ is also $(\epsilon, \delta) -$ DP.

Use cases: Integer optimization/decisions, projections involved, etc.

Group Privacy: Let $M: X \to O$ be $(\epsilon, \delta) -$ DP and let D and D' be two datasets that differ in k entries. Then, for any $\Omega \in O$, we have $\Pr(M(D) \in \Omega) \leq \mathrm{e}^{k\epsilon} \Pr(M(D') \in \Omega) + \delta \frac{e^{k\epsilon}-1}{e^{\epsilon}-1}$.

Proof?

Recap: Remarks on composition

Basic Adaptive Composition: Let $M = (M_1, .., M_T)$ be a sequence of algorithms where M_i is $(\epsilon_i, \delta_i) -$ DP. The algorithms may be chosen adaptively. Then, $M(\cdot)$ is $(\sum_i \epsilon_i, \sum_i \delta_i) -$ DP.

Advanced Composition: Let $M = (M_1, .., M_T)$ be a sequence of $(\epsilon, \delta) -$ DP algorithms (may be chosen adaptively). Then, for any $\delta' > 0$, $M(\cdot)$ is $(\epsilon', T\delta + \delta') -$ DP where $\epsilon' = \epsilon\sqrt{2T \ln\left(\frac{1}{\delta'}\right)} + \frac{T\epsilon(e^{\epsilon}-1)}{e^{\epsilon}+1}$.

- See [Kairouz et al., 2015] for slightly tighter results that also hold when we have $(\varepsilon_i, \delta_i)$-DP
- BUT these composition results are typically not tight in practice (Why?)
- Some variants of (ε, δ)-DP, such as Rényi DP [Mironov, 2017] and zero-concentrated DP (zCDP) [Bun and Steinke, 2016], can enable tighter bounds for specific mechanisms (such as Gaussian mechanism)

Applying DP to practical ML models

- Privatizing the model
  - Privatizing training data
  - **DP training/optimization**
  - Synthetic data
  - …
- Privatizing the output

Recap: Differentially private optimization

- Assume we want to train a model by solve

$$\min_{w} \left(L(\mathbf{w}, \mathbf{X}) \triangleq \frac{1}{n} \sum_{i=1}^{n} \ell(\mathbf{w}, \mathbf{x}_i) + R(\mathbf{w}) \right)$$

- Our algorithm returns w^* that may reveal sensitive data.
- How can we solve this optimization problem in a DP fashion?
  - Output perturbation
  - Exponential mechanism
  - Objective perturbation
  - Privatizing the algorithm: DP-SGD

Recap: Output Perturbation: Utility Bound

- Goal: $\min_w \left(L(w) = \frac{1}{n} \sum_{i=1}^{n} \ell(w, x_i) \right)$

Theorem: Let $\ell(\cdot, x)$ be $L-$ Lipschitz and convex and $diam(W) = D$. Then, there exists $\mu > 0$ s.t.

$$E\left[L\left(w_{priv}\right)\right] - \min_w L(w) = \tilde{O}\left(LD \left(\frac{\sqrt{d}}{\epsilon n} \right)^{1/2} \right)$$

- We ignored $\log\left(\frac{1}{\delta}\right)$ term
- Trivial algorithm provides the bound $L\left(w_{priv}\right) - \min_w L(w) \leq LD$
- Excess risk for output perturbation in the convex setting: $\tilde{O}\left(LD \min\left\{ 1, \left(\frac{\sqrt{d}}{\epsilon n} \right)^{\frac{1}{2}} \right\} \right)$

Recap: Limitations of output perturbation

- Only applicable to strongly convex setting or problems with similar behaviors (why?)
- Requires Lipschitz loss functions
- We may want to maintain privacy during training
- Another idea: instead of adding noise to the solution, can we add noise to the objective?

Recap: Objective perturbation

- Introduced by Chaudhuri, Monteleoni, and Sarwate (2011).
- Idea: Instead of adding noise to the solution, we perturb the goal of the training itself
- Standard ERM Objective: $J(w, \mathrm{D}) = \frac{1}{n}\sum_{i=1}^{n} \ell(w, x_i) + R(w)$
- Perturbed ERM Objective: $\widetilde{J}(w, \mathrm{D}) = \frac{1}{n}\sum_{i=1}^{n} \ell(w, x_i) + R(w) + \frac{1}{n} b^T w$ where $b \sim N(0, \sigma^2 I)$
- Added noise effectively "tilts" the loss landscape
- Excess risk for objective perturbation in the convex setting: $\tilde{O}\left(LD \min\left\{1, \frac{\sqrt{d}}{\epsilon n}\right\}\right)$

Recap: Limitations of objective perturbation

- Requires strong convexity
- Is the strong convexity necessary?
  - Example: median estimation $J(w, \mathcal{D}) = \frac{1}{n}\sum_{i=1}^{n}|w - x_i|$
  - $\mathcal{D} = \{0,100\}$ and $\mathcal{D} = \{0,101\}$

With probability 1/2
catastrophic privacy failure

- Requires Lipschitz loss functions
- We may want to maintain privacy during training
- Another idea: instead of perturbing the solution or objective, **perturb the steps of the algorithm**

Perturbing the algorithm:

- Training Goal: $\min_{w \in W} \left(L(w) = \frac{1}{n} \sum_{i=1}^{n} \ell(w, x_i) \right)$

- (Projected) gradient descent algorithm

For $t = 0,1, \dots, T-1$:

Update $w^{t+1} \leftarrow Proj_W\left(w^t - \alpha_t \nabla L(w^t)\right)$

Return w^T

- The only part that depends on data is the gradient. Let's privatize the gradient

For $t = 0,1, \dots, T-1$:

Draw a random variable $z^t \sim N(0, \sigma^2 I)$

Update $w^{t+1} \leftarrow Proj_W\left(w^t - \alpha_t(\nabla L(w^t) + z^t)\right)$

Return w^T

Privacy analysis: how much noise should we add?

Recall: Let $M = (M_1, .., M_T)$ be a sequence of $(\epsilon', \delta') -$ DP algorithms (may be chosen adaptively). To guarantee target privacy level $(\epsilon, T\delta' + \delta_0)$ with $0 < \epsilon < 1$ and $\delta_0 > 0$, it suffices to choose $\epsilon' \leq \frac{\epsilon}{\sqrt{8T \ln(1/\delta_0)}}$

.

- Each step of the DP-GD algorithm should be $(\epsilon', \delta') -$ DP with $\epsilon' = \frac{\epsilon}{\sqrt{8T \log\left(\frac{1}{\delta}\right)}}$ and $\delta' = \frac{\delta}{T+1}$.

For $t = 0, 1, \ldots, T-1$:

Draw a random variable $z^t \sim N(0, \sigma^2 I)$

Update $w^{t+1} \leftarrow Proj_W\left(w^t - \alpha_t(\nabla L(w^t) + z^t)\right)$

Return w^T

- **Recall** post-processing and Gaussian mechanism: we need a noise with $\sigma = \frac{\Delta_2}{\epsilon'}\sqrt{2\log(1.25/\delta')}$
- Sensitivity of $\nabla L(w^t) \rightarrow \Delta_2 = \frac{2L}{n}$
- Noise variance: $\sigma = \frac{8L\sqrt{T}}{\epsilon n}\left(\log\left(\frac{1.25}{\delta}\right) + \log(T+1)\right)$

How close to optimal solution can we get?

DP-GD: excess risk

Lemma [Shamir & Zhang 2013]: Consider $\min_{w \in W} F(w)$ with convex $F(\cdot)$ and bounded closed convex domain with $diam(W) \leq D$. Assume we run SGD $w^{t+1} \leftarrow Proj(w^t - \eta_t g(w^t))$ where $E[\| g(w^t) \|^2] \leq G^2, E[g(w^t)] \in \partial \mathrm{F}(\mathrm{w}^{\mathrm{t}})$, and $\eta_t = \frac{c}{\sqrt{t}}$ with $c > 0$. Then, for any iteration T, we have $E[F(w^T)] - F(w^*) \leq \left(\frac{D^2}{c} + cG^2\right)(2 + \log T)/\sqrt{T}$

- Optimizing c, we get $E[F(w^T)] - F(w^*) \leq 2DG(2 + \log t)/\sqrt{T}$
- Let's use this lemma in our DP-GD context:
- In DP-GD, we have $G^2 \leq L^2 + d\sigma^2$
- From privacy analysis: $\sigma = \frac{8L\sqrt{T}}{\epsilon n}\left(\log\left(\frac{1.25}{\delta}\right) + \log(T+1)\right)$
- Optimizing T, for DP-GD, we obtain $E[L(w^T)] - \min_w L(w) = \tilde{O}\left(\frac{LD\sqrt{d}}{\epsilon n}\right)$

DP-GD

- Goal: $\min_w \left(L(w) = \frac{1}{n} \sum_{i=1}^{n} \ell(w, x_i) \right)$

- DP-GD Algorithm:

For $t = 0,1, \dots, T-1$:

Draw a random variable $z^t \sim N(0, \sigma^2 I)$

Update $w^{t+1} \leftarrow Proj_W\left(w^t - \alpha_t(\nabla L(w^t) + z^t)\right)$

Return w^T

Theorem: Let $\ell(\cdot, x)$ be $L-$ Lipschitz and convex and $diam(W) = D$. Assume $\epsilon < 1$. Then, by choosing T, α_t, σ^2 appropriately, w^T is $(\epsilon, \delta) -$ DP and

$$E[L(w^T)] - \min_w L(w) = \tilde{O}\left(\frac{LD\sqrt{d}}{\epsilon n}\right)$$

- We ignored $\log\left(\frac{1}{\delta}\right)$ and other logarithmic terms
- Trivial algorithm provides the bound $L(w_{priv}) - \min_w L(w) \leq LD$
- Excess risk for DP-GD in the convex setting: $\tilde{O}\left(LD \min\left\{1, \frac{\sqrt{d}}{\epsilon n}\right\}\right)$

From DP-GD to DP-SGD

- Goal: $\min_w \left(L(w) = \frac{1}{n}\sum_{i=1}^{n} \ell(w, x_i)\right)$

- Computing the entire gradient can be costly

- How about if we add noise to the gradient obtained from a batch

DP-GD Algorithm:

For $t = 0,1, \dots, T-1$:

Draw a random variable $z^t \sim N(0, \sigma^2 I)$

Update $w^{t+1} \leftarrow Proj_W\left(w^t - \alpha_t(\nabla L(w^t) + z^t)\right)$

Return w^T

DP-SGD Algorithm:

For $t = 0,1, \dots, T-1$:

Draw a random variable $z^t \sim N(0, \sigma^2 I)$

Draw a batch of data B and let $g^t = \frac{1}{|B|}\sum_{i \in B} \nabla \ell(w, x_i)$

Update $w^{t+1} \leftarrow Proj_W\left(w^t - \alpha_t(g^t + z^t)\right)$

Return w^T

- Which parts of the analysis would change?

DP-GD to DP-SGD: Privacy amplification by subsampling

Lemma [Balle et al. 2018]: Let $S: X^n \to X^m$ be a sampling mechanism that samples m samples out of n samples uniformly at random without replacement. Let A be an $(\epsilon, \delta) -$ DP mechanism. Then the mechanism $A(S(\cdot))$ is $\left(\ln\left(1 + \frac{m}{n}(e^{\epsilon} - 1)\right), \frac{m}{n}\delta\right) -$DP.

- The amplification is because the sampling procedure has randomness and is secrete.

- When $\epsilon < 1$, we can use the approximation $\ln\left(1 + \frac{m}{n}(e^{\epsilon} - 1)\right) \leq \frac{2m}{n}\epsilon$

- We can also get a similar result using Poisson subsampling

Proof of Privacy Amplification By Subsampling

- Proof **assuming Poisson subsampling** with parameter γ
  - Each data point is selected with probability γ independent of other samples
- Let $D' = D \cup \{x\}$. Then,

$$P\big(A(S(D') \in \Omega)\big) = P\big(A\big(S(D')\big) \in \Omega \mid x \in S(D')\big)\, P(x \in S(D')) + P\big(A\big(S(D')\big) \in \Omega \mid x \notin S(D')\big)\, P(x \notin S(D'))$$

$$\leq \big(e^{\epsilon} P\big(A\big(S(D)\big) \in \Omega\big) + \delta\big)\, \gamma + P\big(A\big(S(D)\big) \in \Omega\big)\, (1 - \gamma)$$

$$\leq (1 - \gamma + \gamma e^{\epsilon})\, P\big(A\big(S(D)\big) \in \Omega\big) + \gamma\delta$$

- Similar argument for the opposite direction
- Exercise: do the analysis for privacy amplification for uniform sampling without replacement
  - Notion of neighboring datasets is different

ERM: DP-SGD vs Output Perturbation

- So far, we studied approximate DP ($(\epsilon, \delta) - \text{DP}$)
- With output perturbation mechanism, we obtain $$E\left[L\left(w_{priv}\right)\right] - \min_{w} L(w) = \tilde{O}\left(LD \min\left\{1, \left(\frac{\sqrt{d}}{\epsilon n}\right)^{\frac{1}{2}}\right\}\right)$$
- With DP-SGD, we have $$E[L(w^T)] - \min_{w} L(w) = \tilde{O}\left(LD \min\left\{1, \frac{\sqrt{d}}{\epsilon n}\right\}\right)$$
- Which bound is better?
- DP-SGD is almost tight in this case. Moreover, it can also be used in non-convex optimization problems
  - How to do the analysis?

Assumptions and extensions

- Goal: $\min_w \left(L(w) = \frac{1}{n} \sum_{i=1}^{n} \ell(w, x_i) \right)$

- Do we need the loss function to be Lipschitz?

For $t = 0,1, \ldots, T-1$:

Draw a mini-batch of data B_t from training data D

Compute clipped gradient $g_t = \frac{1}{|B_t|} \sum_{i \in B_t} \text{clip}(\nabla \ell(w, x_i), C)$

Draw a random variable $z_t \sim N(0, \sigma^2 I)$

Privatize the (mini-batch) gradient $\tilde{g}_t = g_t + z_t$

Update $w_t = \text{Optimizer}(w_t, s_t, \tilde{g}_t)$

Return w^T

- Extension to other (gradient-based) optimizers is straightforward
  - DP-Adam, DP-AdamW, DP-SVRG, DP-Adafactor, …

Discussions on DP-SGD

- Goal: $\min_w \left(L(w) = \frac{1}{n} \sum_{i=1}^{n} \ell(w, x_i) \right)$

- No assumption on the convexity/strong convexity

- No assumption on Lipschitzness of the loss function → Practical

- Drawback
  - Relies on (advanced) composition theorems (which are loose)
  - How can we fix the issue?
  - Why are composition theorems are loose in practice?

Analysis of PLRV for Composition

- **Recall: Privacy Loss Random Variable (PLRV):** $L = \log\left(\frac{P(M(D)=o}{P(M(D')=o)}\right)$ with probability $P(M(D) = o)$
- **Connection to DP**: $M(\cdot)$ is (ϵ, δ)-DP if $P(L > \epsilon) \leq \delta$
- **Composition as a summation**:
  - The total privacy loss of the composed mechanism: $L_{tot} = \sum_{i=1}^{T} L_i$
- In advanced composition, we used Chernoff bound to bound the tail of the distribution of L_{tot}
- Can we do better?
  - We can numerically compute the probability of the tail → Numerical Accountant

Numerical Accountant and the Fourier Accountant

- Need to compute the distribution of $L_{tot} = \sum_{i=1}^{T} L\_i$
- Therefore, we need to (Numerically) compute the convolution of the distributions
- Assuming discretization with k bins, it requires $O(k^2)$ computational complexity
- **The Fourier Accountant (Koskela et al., 2020):**
  - **Key Idea**: Convolution in the time/loss domain is multiplication in the frequency domain.
  - **Workflow:**
    - Discretize the PLRV distribution of a single step.
    - Use Fast Fourier Transform (FFT).
    - Compose by element-wise multiplication in the frequency domain.
    - Inverse FFT to get the composed PLRV distribution.
- Complexity of Fourier Transformation: $O(k \log k)$