DS-GA 3001: Modern Topics in Statistical Learning Theory Fall 2024

Lecture 13 Differential Privacy (Cont'd) - Apr. 25, 2024

Prof. Qi Lei *Scribe: Aoran Chen*

1 Overview

In the previous lecture, we introduced the concept of differential privacy (DP).

This lecture will focus on the implementation of DP through specific mechanisms and their theoretical underpinnings.

2 Recap on Differential Privacy (DP)

Differential privacy [Dwo06] primarily addresses the security of an algorithm $M : \mathcal{X}^n \to \mathcal{Y}$, which is designed to operate on datasets differing by a single entry, thereby safeguarding the privacy of the data subjects.

Definition 1 ($\epsilon-$DP)**.** *An algorithm M achieves ε-differential privacy if, for any two adjacent datasets $X, X' \in \mathcal{X}^n$ and for all subsets $T \subseteq \mathcal{Y}$, the following inequality is satisfied:*

$$\mathbb{P}[M(X) \in T] \leq e^{\varepsilon} \cdot \mathbb{P}[M(X') \in T] \tag{2.1}$$

- The essence of randomness in this context is derived from the mechanism M, which incorporates stochastic processes to ensure that DP guarantees are met.
- The core principle of differential privacy involves utilizing a randomized response technique to obscure individual data points, thus preserving privacy.

One effective method to implement this is through the Laplace mechanism, which we will explore next.

3 Laplace Mechanism

The Laplace mechanism is central to achieving differential privacy through the injection of randomness. It allows for the privacy-preserving release of statistical queries.

Definition 2 (Laplace Mechanism)**.** *Let $f : \mathbb{R}^n \to \mathbb{R}^k$ be a query function and $\varepsilon > 0$ be a privacy parameter. The Laplace mechanism $M(x)$ is defined as follows:*

$$M(x) = f(x) + (Y_1, Y_2, \ldots, Y_k), \tag{3.1}$$

where $Y_1, Y_2, \ldots, Y_k$ are independent random variables drawn from the Laplace distribution with scale parameter $b = \frac{\Delta f}{\varepsilon}$. The probability density function of each Y_i is given by the Laplace distribution:

$$p(x) = \frac{1}{2b} \exp\left(-\frac{|x-\mu|}{b}\right).$$

The term Δf denotes the ℓ_1-sensitivity of the function f, which is defined as:

$$\Delta f = \max_{x,x' \in \mathbb{R}^n : |x-x'|_1 \leq 1} |f(x) - f(x')|_1,$$

where $|\cdot|_1$ represents the ℓ_1-norm. The ℓ_1-sensitivity quantifies the maximum change in the output of f when a single data point in the input dataset is modified while keeping all other data points unchanged.

3.1 Laplace Distribution

The Laplace distribution is used within the Laplace mechanism to ensure that the probability of output does not significantly change when a single data point in the input dataset is altered.

Definition 3 (Laplace Distribution)**.** *The probability density function (pdf) for the Laplace distribution, centered at μ with scale b, is given by:*

$$p(x) = \frac{1}{2b} \exp\left(-\frac{|x-\mu|}{b}\right) \tag{3.2}$$

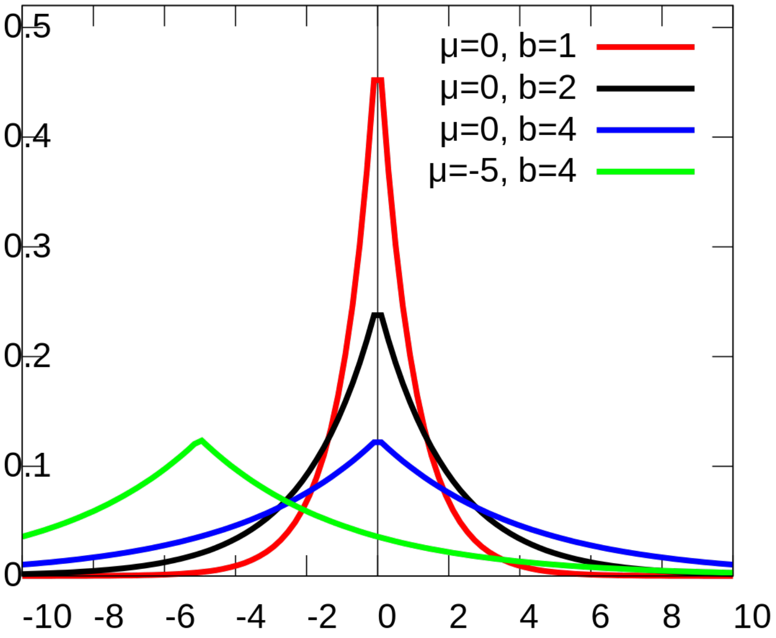

Figure 1: Laplace Distribution

This distribution is characterized by a variance of $2b^2$.

Characteristics of the scale parameter b:

- A larger b results in a flatter distribution curve, indicating higher variance and thus more privacy.
- As b decreases towards zero, the curve becomes increasingly peaked, indicating lower variance and reduced noise.

3.2 ℓ_1-Sensitivity

The ℓ_1-sensitivity of a function quantifies the maximum impact that a single individual's data can have on the function's output, which is essential for determining the noise required in differentially private mechanisms.

Definition 4 (ℓ_1-Sensitivity of f)**.** *For a function $f : \mathcal{X}^n \to \mathbb{R}^k$, the ℓ_1-sensitivity, denoted by Δf, is:*

$$\Delta f = \max_{x,x' \in \mathcal{X}^n : \|x-x'\|_1 \le 1} \|f(x) - f(x')\|_1, \tag{3.3}$$

where x and x' are neighboring datasets differing in at most one element.

The ℓ_1-sensitivity directly influences the scale of the noise distribution in differentially private mechanisms, such as the Laplace mechanism, to ensure a desired level of privacy protection.

With a clear understanding of the Laplace distribution and ℓ_1-sensitivity, we can now explore the theoretical guarantee that the Laplace mechanism provides in terms of ε-differential privacy.

3.3 Theoretical Guarantee of the Laplace Mechanism

Theorem 5. *The Laplace mechanism satisfies ε-differential privacy.*

Proof. Let X and X' be two neighboring datasets differing in at most one element, and let $M(X)$ and $M(X')$ be the probability density functions (pdfs) of the Laplace mechanism applied to X and X', respectively. For any output $z \in \mathbb{R}^k$, the ratio of the probabilities under X and X' is given by:

$$
\begin{aligned}
\frac{P_X(z)}{P_{X'}(z)} &= \frac{\prod_{i=1}^{k} \exp\left(-\frac{\varepsilon|f(X)_i - z_i|}{\Delta f}\right)}{\prod_{i=1}^{k} \exp\left(-\frac{\varepsilon|f(X')_i - z_i|}{\Delta f}\right)} && (3.4)\\
&= \prod_{i=1}^{k} \exp\left(\frac{\varepsilon(|f(X')_i - z_i| - |f(X)_i - z_i|)}{\Delta f}\right) && (3.5)\\
&\leq \prod_{i=1}^{k} \exp\left(\frac{\varepsilon|f(X')_i - f(X)_i|}{\Delta f}\right) && (3.6)\\
&= \exp\left(\frac{\varepsilon}{\Delta f}\sum_{i=1}^{k} |f(X')_i - f(X)_i|\right) && (3.7)\\
&= \exp\left(\frac{\varepsilon\|f(X') - f(X)\|_1}{\Delta f}\right) && (3.8)\\
&\leq \exp(\varepsilon), && (3.9)
\end{aligned}
$$

where the last inequality follows from the definition of ℓ_1-sensitivity, which ensures that $\|f(X') - f(X)\|_1 \leq \Delta f$. This result demonstrates that the Laplace mechanism satisfies ε-differential privacy. □

The proof relies on the properties of the Laplace distribution and the ℓ_1-sensitivity of the query function f. By bounding the ratio of the probabilities of any output under neighboring datasets, we show that the Laplace mechanism provides a strong privacy guarantee, limiting the ability of an adversary to distinguish between the presence or absence of a single individual's data in the dataset.

3.4 Recap of Query Functions

Query functions in differential privacy can range from simple aggregations to complex computational models. Let $f : \mathcal{X}^n \to \mathbb{R}^k$ be a query function, where $\mathcal{X}$ is the input domain and n is the dataset size. Some examples include:

- $f(x)$ calculating aggregate statistics like sum of ages or average income in dataset x.
- In medical research, $f(x)$ counting disease incidence across different populations.
- In federated learning, $f(x)$ computing gradients or model updates based on distributed data.

Having explored query functions and the Laplace mechanism, we will now examine their application to high-dimensional data, particularly in federated learning with deep neural networks.

4 Limitations of Differential Privacy in High-dimensional Settings

4.1 Federated Learning and the Cloud Server

- Federated learning is a distributed machine learning approach that allows edge devices to collaboratively train a model without sharing their local data directly. In this setup, devices compute gradients or model updates based on their local data and send these updates to a central cloud server.
- The server then aggregates the updates to improve the global model. Notably, only the gradients or model updates are transmitted to the server, not the raw data itself.

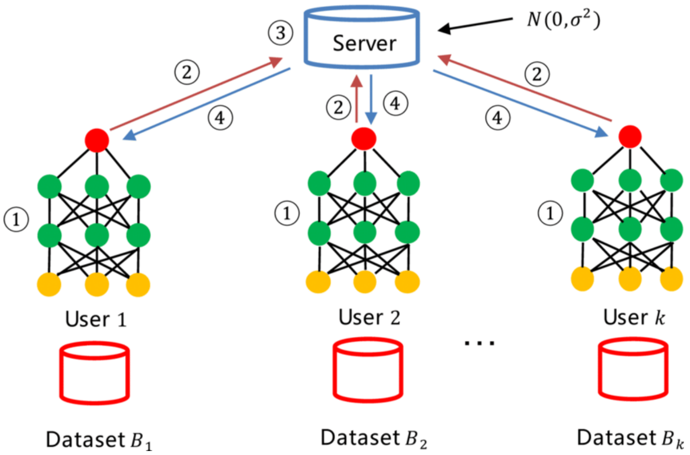

Figure 2: Federated Learning Framework

The aggregation process in federated learning can be mathematically represented as:

$$G = \sum_{x_i \in E} \nabla L(\theta; x_i) \tag{4.1}$$

- Here E represents the ensemble of local data points on each device, and $\nabla L(\theta; x_i)$ denotes the gradient of the loss function with respect to the global model parameters θ.
- However, the gradients, being derived from the local data, may still contain sensitive information that could potentially be exploited to reconstruct the original data.

To evaluate the risk of privacy breaches, it is crucial to assess the ℓ_1-sensitivity of the aggregation function f, which maps the input data $X = [x_1, x_2, \ldots, x_\beta]$ to the gradients:

$$f : X \mapsto \sum_{i=1}^{\beta} \nabla L(\Theta; x_i) \tag{4.2}$$

- The ℓ_1-sensitivity, denoted $\Delta(f)$, measures the maximum change in the output of f when one data point in X is modified slightly. To ensure privacy, the Laplace mechanism can be employed, adding noise scaled to $\Delta(f)/\varepsilon$ to achieve ε-differential privacy.

- However, applying differential privacy techniques to high-dimensional gradients can be challenging without significantly impacting the gradient information's utility.

4.2 Example: Two-layer Neural Network

To illustrate the challenges of preserving privacy in high-dimensional settings, let's consider an example of a two-layer neural network used in federated learning. The network consists of an input layer, a hidden layer with m neurons, and an output layer. The loss function for this network, calculated for a single data point (x_i, y_i), can be expressed as:

$$L(\Theta; x_i, y_i) = \left(\sum_{j=1}^{B} \sum_{w_j \in W} a_j \sigma(w_j^T x_i) - y_i \right)^2 \tag{4.3}$$

where Θ represents the model parameters, B is the batch size, a_j is the output of each neuron in the hidden layer, $W \in \mathbb{R}^m$ is the weight matrix, and σ is the activation function.

The gradient of the loss function with respect to the model parameters is a key component in the training process. For each neuron in the hidden layer, the gradient with respect to its output a_j is given by:

$$\nabla_{a_j} L(\Theta; x_i, y_i) = \left[\sum_{i=1}^{B} r_i \sigma(w_j^T x_i) \right] \in \mathbb{R}^{(m+1)} \tag{4.4}$$

where r_i is the residual for each data point, defined as:

$$r_i = \sum_{j=1}^{B} a_j \sigma(w_j^T x_i) - y_i \tag{4.5}$$

Similarly, the gradient with respect to each weight w_j is:

$$\nabla_{w_j} L(\Theta; x_i, y_i) = \left[\sum_{i=1}^{B} r_i \sigma'(w_j^T x_i) x_i \right] \in \mathbb{R}^m \tag{4.6}$$

These gradients, computed on the local data, are then sent to the cloud server for aggregation.

Assessing the Laplace Mechanism for High-dimensional Gradients

Main takeaways:

- To address the privacy concerns in federated learning, we aim to answer a crucial question: Is it possible to reconstruct a local user's data from the shared gradients?

- To answer this question, we need to evaluate the ℓ_1-sensitivity of the gradient and determine the magnitude of noise required to achieve ε-differential privacy using the Laplace mechanism. If the ℓ_1-sensitivity is high, a significant amount of noise must be added to the gradients to maintain the desired level of privacy.
- However, adding excessive noise can severely degrade the utility of the model by obscuring the informative gradients, making it difficult to learn effectively from the data.

In the following subsection, we will compute the ℓ_1-sensitivity for the two-layer neural network example and analyze the implications of applying the Laplace mechanism in high-dimensional settings.

4.3 Differential Privacy in Neural Networks

To understand the challenges of applying differential privacy in neural networks, let's consider a random network model with the following assumptions:

- The outputs y_i are normally distributed, $y_i \sim \mathcal{N}(0, I_d)$, and each a_j is an output of a neuron.
- The neural network output remains consistent for normalized inputs $|x_i| = 1$ and binary labels $y_i \in \{+1, -1\}$.

Given neighboring datasets X and X' that differ by at most one element, the difference in the residuals r and r' for input pairs (x_i, x_i') is represented by:

$$r - r' = \frac{1}{m} \sum_{j=1}^{m} \left(\sigma(w_j^\top x_i) - \sigma(w_j^\top x_i') \right), \tag{4.7}$$

where σ denotes the activation function. This captures the sensitivity of the network to input perturbations.

In particular, for the first m coordinates, the change in the gradient of the activation outputs for neighboring inputs x_i and x_i' can be bounded as:

$$|G(a_j) - G(a_j')| \leq |r_i| \left(|\sigma(w_j^\top x_i)| + |\sigma(w_j^\top x_i')| \right), \tag{4.8}$$

where G denotes the gradient and r_i the residual component. The bound follows from the Lipschitz continuity of σ, with σ being 1-Lipschitz, and the expected value of $|\sigma(w_j^\top x_i)|$ being capped by $C \log\left(\frac{n}{s}\right)$ for a chosen confidence level s and dimension d.

For the last m vectors in $\mathbb{R}^d$, we estimate the gradient difference by:

$$|G(w_j) - G(w_j')| \leq C \left(|x_i|_1 + |x_i'|_1 \right) = 2C\sqrt{d}, \tag{4.9}$$

where $|x_i|_1$ denotes the ℓ_1-norm, and we use the inequality $|x_i|_1 \leq \sqrt{d}|x_i|_2$ given the normalized input condition $|x_i|_2 = 1$.

The overall sensitivity Δ for the network, taking into account the sum of the individual sensitivities, is thus bounded by:

$$\Delta \leq n \cdot C \log\left(\frac{n}{s}\right) + Cm \cdot \sqrt{d} = \mathcal{O}(m\sqrt{d}). \tag{4.10}$$

To maintain differential privacy, Laplacian noise proportional to $\frac{\Delta}{\varepsilon}$ is added, leading to a noise variance of $2b^2$.

It is not hard to observe that such an addition is highly impractical:

- The scale of the noise can be excessive, particularly for high dimensions d and small privacy parameter ε, with a noise magnitude on the order of $\frac{d\sqrt{m}}{\varepsilon^2}$, which may overwhelm the utility of the output.
- In practice, Differential Privacy is too strong a guarantee and not super appropriate in some scenarios.

4.4 Scenarios Where Differential Privacy May Not Be Necessary

In certain scenarios, differential privacy might not be strictly necessary. Consider, for example, a 2-layer neural network with a small number of layers. If external noise is inherent in the system, the gradients can be expressed as:

$$G(a) = W\left(\sum_{i=1}^{B} r_i x_i^T\right)$$
$$G(w) = a\left(\sum_{i=1}^{B} r_i x_i^T\right) \in \mathbb{R}^{m\times d}$$

In such cases, one can only recover a linear combination of the data rather than the individual data points unless strong prior information is assumed. This simple example demonstrates that a scenario exists where data is safe even without DP guarantees. DP might not be the panacea for data privacy, and sometimes, it is important to analyze the problem on a case-by-case basis.

5 Reconstruction Attacks

Given the challenges and potential weaknesses in preserving privacy through traditional differential privacy mechanisms in high-dimensional settings, as explored in Section 4, it is crucial to consider the robustness of these mechanisms against sophisticated data reconstruction techniques.

In this section, we discuss two types of reconstruction attacks that demonstrate the potential for adversaries to approximate private data, despite the application of differential privacy.

These attacks underscore the need for continuous evaluation and enhancement of privacy-preserving methods in the face of evolving threats.

5.1 Gradient Inversion Attack

This type of attack demonstrates a potential vulnerability in differential privacy by attempting to approximate the input data. The attack involves solving the following optimization problem:

$$\min_{\hat{x}_i,\hat{y}_i} \quad d\left(\frac{1}{B}\sum_{i=1}^{B}\nabla L(\hat{\theta};\hat{x}_i,\hat{y}_i), G_T\right), \tag{5.1}$$

where d represents a distance metric such as the ℓ_2-norm or cosine similarity, and G_T is the target gradient. Notably, this approach challenges the sufficiency of noise addition, as detailed in Lei et al.[Lei+19], since the problem becomes NP-hard.

5.2 Tensor-based Method on Two-layer Networks

Building on the theme of reconstructing inputs, Wang et al.[WLL23] introduced a tensor-based method specifically for two-layer networks. This method allows for the recovery of inputs $x_1, \ldots, x_B$ with a certain degree of accuracy, constrained by:

$$\sqrt{\frac{1}{B}\sum_{i}\|\hat{x}_i - x_i\|_2^2} \leq \mathcal{O}\left(\frac{B\sqrt{d}}{\sqrt{m}}\right) \tag{5.2}$$

This inequality provides a bound on the error of the recovered data when using their method.

5.3 Defense Mechanisms Evaluation

The evaluation of defense mechanisms against such attacks has been comprehensively analyzed by Liu et al. (2024) [LWL24]. The following table summarizes the complexities associated with various defense strategies:

Defense	**Complexity**
No defense	$B\sqrt{d/m}$
k-Local aggregation	$kB\sqrt{d/m}$
ℓ_2-Gradient noise	$(B+\sigma)\sqrt{d/m}$
Gradient clipping	$B\sqrt{d/m}$
DP-SGD	$\left(B + \sigma \max\{1, \frac{\|g\|}{s}\}\right)\sqrt{d/m}$
p-Dropout	$B\sqrt{d/mp}$
Gradient pruning	Not applied

Table 1: Evaluation of different defense mechanisms based on complexity, as explored by Liu et al. (2024) [LWL24].

These complexity measures serve as upper bounds; discussions of lower bounds are proposed for future research.

References

[Dwo06] Cynthia Dwork. “Differential privacy”. In: *International colloquium on automata, languages, and programming*. Springer. 2006, pp. 1–12.

[Lei+19] Qi Lei et al. “Inverting deep generative models, one layer at a time”. In: *Advances in neural information processing systems* 32 (2019).

[WLL23] Zihan Wang, Jason Lee, and Qi Lei. “Reconstructing training data from model gradient, provably”. In: *International Conference on Artificial Intelligence and Statistics*. PMLR. 2023, pp. 6595–6612.

[LWL24] Sheng Liu, Zihan Wang, and Qi Lei. “Data Reconstruction Attacks and Defenses: A Systematic Evaluation”. In: *arXiv preprint arXiv:2402.09478* (2024).