Differential Privacy and Distributed Learning

Chen Chen

Department of Computer Science
University of Georgia

3/25/2019

Outline

Differential privacy

- Privacy issues in data analysis
- Differential Privacy - Definition
- Sensitivity
- Protecting privacy

Differential privacy in distributed learning

- Model Aggregation with Output perturbation
- Iterative Learning with Gradient perturbation

Privacy issues in data analysis

Problem setting:

Release some information (statistics) learned from a dataset D.

- Descriptive statistics:
  - Average, variance, maximum, minimum, etc.
- Machine learning model:
  - Linear models, SVM, unsupervised learning, etc.

They are all queries made on D:

- Releasing answer $a = \mathcal{Q}(D)$.

The dataset D, contributed by n individuals, may contain sensitive information of each individual, which we want to protect.

- Example: medical records, financial records, academic records, etc.

Then, releasing a might leak sensitive information of the user.

Differential Privacy (DP)

Differential privacy (DP) [Dwork et al., 2006] has become a rigorous and *de facto* standard for privacy preserving machine learning (PPML), due to its mathematical guarantee that model output is *not significantly influenced* by the presence or absence of an individual datum.

Intuitively, if the largest effect of one individual to the released statistic(s) is limited, then the confidence of the information an adversary could infer from that individual is also limited. Differential privacy provides us a way to quantify the influence and make trade off between privacy and utility.

Differential Privacy (DP)

Definitions

- **Neighboring databases** Databases D and D' are *neighboring* if they differ in one element, denoted as $D \sim D'$.
- **ϵ-differential privacy (ϵ-DP)** A mechanism $\mathcal{M}$ satisfies ϵ-DP if for any $D \sim D'$, and for all events S in the output space of $\mathcal{M}$ $\big($denoted as $\forall S \subseteq Range(\mathcal{M})\big)$, $\mathcal{M}$ guarantees

$$\Pr[\mathcal{M}(D) \in S] \leq e^{\epsilon} \Pr[\mathcal{M}(D') \in S]$$

Intuitively, privacy is guaranteed by *limiting the impact* of every individual element of the dataset to the output of the machine learning model.

The parameter $\epsilon \geq 0$ is usually called "privacy budget", meaning how large privacy we want to pay off for the learning.

Differential Privacy (DP)

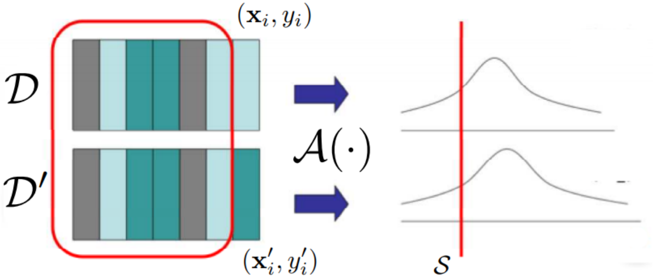

Figure 1: A differentially private algorithm [Chaudhuri et al., 2011]. When neighboring datasets D and D' are input to the algorithm $\mathcal{A}$, the two distributions of the algorithm's output $\mathcal{A}(D)$ and $\mathcal{A}(D')$ are close:
For any fixed measureable set $\mathcal{S}$, the ratio of the densities is bounded.

- A strong adversary, who knows all but one entry of the dataset, cannot gain *much* information about the unknown entry from the output of the mechanism $\mathcal{M}$

Differential Privacy (DP) - Relaxations

Relaxation: **Approximate differential privacy,** (ϵ, δ)**-DP** A mechanism $\mathcal{M}$ satisfies (ϵ, δ)-DP if for any $D \sim D'$, and for all events $S \subseteq Range(\mathcal{M})$, $\mathcal{M}$ guarantees

$$\Pr[\mathcal{M}(D) \in S] \leq e^{\epsilon} \Pr[\mathcal{M}(D') \in S] + \delta$$

- The parameter $\delta \in [0, 1]$ accounts for the likelihood that "bad events" happened ($\mathcal{M}$ result in a high privacy loss):
- Usual choice of δ: $\Theta(n^{-1})$, $\Theta(n^{-2})$, or $\Theta(n^{-\log n})$.
- When $\delta = 0$, it is pure ϵ-DP.

Sensitivity

To transform a non-private algorithm to satisfy DP, one usually sample some random noises injecting to (intermediate) results. The magnitude of noise depends on the *sensitivity* of the mechanism.

Definition

- The L_1 (resp. L_2) sensitivity of a algorithm $\mathcal{M}$ (denoted as $\Delta_{\mathcal{M}}$) is the maximum distance of output of $\mathcal{M}$ applying on all possible neighboring datasets D and D':

$$\Delta_{\mathcal{M}} = \sup_{D \sim D'} \|\mathcal{M}(D) - \mathcal{M}(D')\|$$

  where $\|\cdot\|$ denotes L_1 (resp. L_2) norm.

The sensitivity $\Delta_{\mathcal{M}}$ only depends on the algorithm, not on any specific training data D.

Protecting privacy

Laplace Mechanism [Dwork et al., 2006]

Theorem Let $\mathcal{M} : \mathcal{D}^n \to \mathbb{R}^p$ be a non-private mechanism of L_1 sensitivity $\Delta_{\mathcal{M}}$, and let $b \sim Lap(\Delta_{\mathcal{M}}/\epsilon)^p$. Then, the mechanism

$$\tilde{\mathcal{M}}(D) = \mathcal{M}(D) + b$$

provides ϵ-differential privacy.

Laplace distribution $Lap(\beta)$ has pdf as $f(x|\beta) = \frac{1}{2\beta} \exp\big(-\frac{|x|}{\beta}\big)$

Gaussian Mechanism

Theorem Let $\mathcal{M} : \mathcal{D}^n \to \mathbb{R}^p$ be a non-private mechanism of L_2 sensitivity $\Delta_{\mathcal{M}}$, and let $z \sim \mathcal{N}(0, \Delta_{\mathcal{M}}^2 \sigma^2 I_p)$. Then, for $\epsilon, \delta \in (0,1)$, the mechanism

$$\tilde{\mathcal{M}}(D) = \mathcal{M}(D) + z$$

provides $(\sqrt{2\log(1.25/\delta)}/\sigma, \delta)$-differential privacy for any $\delta > 0$.

Why Laplace mechanism provides DP

Let $s(D)$ be a statistic of database D: $s : \{\text{database}\} \to \mathbb{R}^k$.
$Z \sim Lap(\Delta_s/\epsilon)^k$, where Δ_s, the sensitivity, is defined as
$\Delta_s = \max_{D_1 \sim D_2} \|s(D_1) - s(D_2)\|_1 = \max_{D_1 \sim D_2} \sum_{i=1}^k |s_i(D_1) - s_i(D_2)|$.

Claim: $\mathcal{M}(D) = s(D) + Z$ is ϵ-DP.

Proof: $\mathcal{M}(D) \sim Lap(s(D), \Delta_s/\epsilon)$, therefore

$$\frac{\Pr[\mathcal{M}(D_1) \in [r, r+dr]]}{\Pr[\mathcal{M}(D_2) \in [r, r+dr]]} = \frac{d\Pr[\mathcal{M}(D_1) = r]}{d\Pr[\mathcal{M}(D_2) = r]}$$

$$= \prod_{i=1}^k \frac{d\Pr[\mathcal{M}_i(D_1) = r_i]}{d\Pr[\mathcal{M}_i(D_2) = r_i]} = \prod_{i=1}^k \frac{\exp(-\frac{\epsilon|s_i(D_1) - r_i|}{\Delta_s})}{\exp(-\frac{\epsilon|s_i(D_2) - r_i|}{\Delta_s})}$$

$$= \prod_{i=1}^k \exp\left(\frac{\epsilon}{\Delta_s}(|s_i(D_2) - r_i| - |s_i(D_1) - r_i|)\right)$$

$$\le \prod_{i=1}^k \exp\left(\frac{\epsilon}{\Delta_s}|s_i(D_2) - s_i(D_1)|\right)$$

$$= \exp\left(\frac{\epsilon}{\Delta_s}\sum_{i=1}^k |s_i(D_2) - s_i(D_1)|\right) \le \exp\left(\frac{\epsilon}{\Delta_s}\Delta_s\right) = \exp(\epsilon)$$

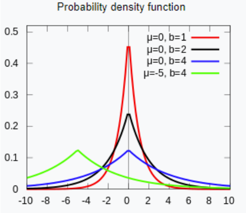

Figure 2: PDF of Laplace distribution.

PDF: $f(x|\mu, \beta) = \frac{1}{2\beta}\exp(-\frac{|x-\mu|}{\beta})$

Mean, median, mode: μ
Variance: $2\beta^2$

Machine Learning as an Empirical Risk Minimization (ERM)

Problem setting:

- A training data set $D = \{d_1, d_2, ..., d_n\} \in \mathcal{D}^n$, consists of n observations.
  - For classification problem, for $i \in [n]$: $d_i = \langle x_i, y_i \rangle$.
    $x_i \in \mathcal{X} \subseteq \mathbb{R}^p$, $y_i \in \mathcal{Y} = \{+1, -1\}$.
- An empirical risk function $J(\theta, D) = \frac{1}{n} \sum_{i=1}^{n} \ell(\theta, d_i)$
  - It may contain regularization terms.
  - Sometimes we may denote regularizer explicitly: $\ell(\theta, d_i) = f(\theta, d_i) + \lambda N(\theta)$.
- Denote $J_D(\theta) = J(\theta, D)$ and $\ell_i(\theta) = \ell(\theta, d_i)$

Example:

- Logistic regression:

$$\ell(\theta, d_i) = \ell(\theta, \langle x_i, y_i \rangle) = \log(1 + \exp(-y_i \theta^T x_i)) + \frac{\lambda}{2} \|\theta\|_2^2$$

For Empirical Risk Minimization (ERM), the goal is to find (train) a model θ which minimizes $J(\theta, D)$.

Multi-Party Machine Learning

I will present two algorithms presented in [Jayaraman et al., 2018].
Consider the following empirical risk minimization (ERM) objective:

$$J_D(\theta) = \frac{1}{n}\sum_{i=1}^{n} f(\theta, x_i, y_i) + \lambda N(\theta)$$

Assumptions:

- $f(\theta)$ is convex, G-Lipschitz, and L-smooth over $\theta \in \mathcal{R}^d$.
- $N(\cdot)$ is regularization term, which is 1-strongly convex. So $J(\cdot)$ is λ-strongly convex.
- Each data $(x_i, y_i) \in D$'s features x_i lies in a unit ball.

Now the dataset D is stored in m parties. For each party $j = 1, ..., m$, with data set D_j of size n_j, we denote its data instance as $(x_i^{(j)}, y_i^{(j)})$.

Denote $n_{(1)}$ as the size of the smallest data set among the m parties.

Output Perturbation - non-distributed setting

The output perturbation method is a direct utilization of the sensitivity method. The algorithm

- first solve the problem non-privately,
- then perturbs the non-private solution by a noise scaled by ϵ and sensitivity.

Algorithm 1 ERM with output perturbation [Chaudhuri et al., 2011]

```
Input: Data $D = \{d_1, d_2, ..., d_n\}$, loss function $J(\theta, D) = \frac{1}{n}\sum_{i=1}^{n} f(\theta, d_i) + \lambda N(\theta)$, and privacy parameter $\epsilon$.
Train a non-private model $\theta^* = \arg\min_\theta \hat{L}(\theta, D)$.
Draw a random vector $b \sim v(b) \propto \exp\{-\frac{n\lambda\epsilon|b|}{2}\}$.
Output: Compute and return $\tilde{\theta} = \theta^* + b$.
```

The privacy guarantee of the algorithm is based on that θ^* has a L_2 sensitivity of $\frac{2}{n\lambda}$.

Proof of Chaudhuri's algorithm

Claim: the sensitivity of $\theta^* = \arg\min_\theta \hat{L}(\theta, D)$ is at most $\frac{2}{n\lambda}$.

Proof: Let $D = \{(x_1, y_1), ..., (x_n, y_n)\}$ and $D' = \{(x_1, y_1), ..., (x'_n, y'_n)\}$ which differs at the last individual.
Let $G(\theta) = \hat{L}(\theta, D)$, $g(\theta) = \hat{L}(\theta, D') - \hat{L}(\theta, D)$, so $\hat{L}(\theta, D') = G(\theta) + g(\theta)$.

One can observe that $g(\theta) = \frac{1}{n}[\ell_n(\theta) - \ell_{n'}(\theta)]$.
So $\nabla g(\theta) = \frac{1}{n}(y_n \nabla \ell_n(\theta) x_n - y'_n \nabla \ell_{n'}(\theta) x'_n)$. Since $|\nabla \ell_i(\theta)| \leq 1$, $y_i \in \{+1, -1\}$,
$\|\nabla g(\theta)\| \leq \frac{1}{n}(\|x_n - x'_n\|) \leq \frac{1}{n}(\|x_n\| + \|x'_n\|) \leq \frac{2}{n}$.

Let $\theta_1 = \arg\min_\theta \hat{L}(\theta, D)$ and $\theta_2 = \arg\min_\theta \hat{L}(\theta, D')$.
Then $\nabla G(\theta_1) = 0$, $\nabla G(\theta_2) + \nabla g(\theta_2) = 0$, so $\nabla g(\theta_2) = \nabla G(\theta_1) - \nabla G(\theta_2)$.
Since $G(\theta)$ is λ-strongly convex,
$\nabla g(\theta_2)^T(\theta_1 - \theta_2) = \left(\nabla G(\theta_1) - \nabla G(\theta_2)\right)^T(\theta_1 - \theta_2) \geq \lambda\|\theta_1 - \theta_2\|^2$
By Cauchy-Schwartz inequality,
$\|\nabla g(\theta_2)\|\|\theta_1 - \theta_2\| \geq \nabla g(\theta_2)^T(\theta_1 - \theta_2) \geq \lambda\|\theta_1 - \theta_2\|^2$.

Divide both sides by $\lambda\|\theta_1 - \theta_2\|$, one have $\|\theta_1 - \theta_2\| \leq \frac{1}{\lambda}\|\nabla g(\theta_2)\| \leq \frac{1}{\lambda} \times \frac{2}{n} = \frac{2}{n\lambda}$.

Model Aggregation with Output Perturbation

In [Jayaraman et al., 2018], they extend the differential privacy bound of [Chaudhuri et al., 2011] to the multi-party setting, ensuring sufficient noise to preserve the privacy of each participant's data throughout the multi-party computation.

Given m parties, each having a data set D_j of size n_j, and the corresponding local model estimator $\hat{\theta}^{(j)}$, obtained by minimizing the local objective function:

$$J_{D_j}(\theta) = \frac{1}{n_j} \sum_{i=1}^{n_j} \ell(\theta, x_i^{(j)}, y_i^{(j)}) + \lambda N(\theta)$$

The perturbed aggregate model estimator is given as:

$$\hat{\theta}^{\mathsf{priv}} = \frac{1}{m} \sum_{j=1}^{m} \hat{\theta}^{(j)} + b$$

where $\hat{\theta}^{(j)} = \arg\min_\theta J_{D_j}(\theta)$, b is the Laplace noise added to the aggregate model estimator to preserve differential privacy.

Secure model aggregation can be performed using a secure multi-party computation (MPC) protocol.

Model Aggregation with Output Perturbation

Theorem Given a perturbed aggregate model estimator $\hat{\theta}^{\mathsf{priv}}$, the data lie in a unit ball, $\ell(\cdot)$ is G-Lipschitz, then $\hat{\theta}^{\mathsf{priv}}$ is ϵ-DP if

$$b = Lap(\frac{2G}{mn_{(1)}\lambda\epsilon})$$

Proof: Consider the output perturbation algorithm from [Chaudhuri et al., 2011]. For a dataset D of size n, where loss function ℓ is λ-strongly convex, the optimal model $\theta^* = \arg\min_\theta \frac{1}{n}\ell(\theta, d_i)$ has sensitivity of $\frac{2G}{n\lambda}$. Therefore, $\hat{\theta}$, the average of m models, has a sensitivity of $\frac{2G}{mn_{(1)}\lambda}$.

Gradient Perturbation - non-distributed setting

Gradient descent (GD), and its variants like stochastic gradient descent (SGD), are the most often method used for training an ERM model.

- Iterative algorithm.
- Data were accessed only through gradient evaluation.

A gradient purturbation method usually works as:

- Initialize θ_0.
- Iterate T times:
  - Evaluate the gradient of $\ell(\theta, d)$ on current parameter θ_t on (a portion of) data.
  - Inject Gaussian noise (sampled from $\mathcal{N}(0, \sigma^2 I)$) into the gradients.
  - Use the noise gradients to update the parameter to acquire θ_{t+1}.
- Output θ_T.

Zero concentrated differential privacy (zCDP)

zCDP [Bun and Steinke, 2016] is a relaxed version of ϵ-DP, which imposes a bound on the moment generating function (mgf) of the privacy loss random variable

$$Z = \log \frac{\Pr[\mathcal{M}(D) = o]}{\Pr[\mathcal{M}(D') = o]}$$

for output $o \in range(\mathcal{M})$. It has a parameter ρ to quantify privacy, and has these relationships with ϵ-DP and (ϵ, δ)-DP:

- If $\mathcal{M}$ satisfies ϵ-DP, then $\mathcal{M}$ satisfies $(\frac{1}{2}\epsilon^2)$-zCDP.
- If $\mathcal{M}$ satisfies ρ-zCDP, then $\mathcal{M}$ satisfies $(\rho + 2\sqrt{\rho \log(1/\delta)}, \delta)$-DP for any $\delta > 0$.

ρ-zCDP can be achieved by Gaussian mechanism, and has composition rules:

- **Perturbation** The Gaussian mechanism, which returns $q(D) + \mathcal{N}(0, \sigma^2)$, satisfies $\Delta_2(q)^2/(2\sigma^2)$-zCDP.
- **Composition** Suppose two mechanisms satisfy ρ_1-zCDP and ρ_2-zCDP, then their compositoin satisfies $(\rho_1 + \rho_2)$-zCDP.

Iterative Learning with Gradient Perturbation

Consider the centralized ERM objective for m parties, each with a data set D_j of size n_j:

$$J_D(\theta) = \frac{1}{m}\sum_{j=1}^{m}\frac{1}{n_j}\sum_{i=1}^{n_j}\ell(\theta, x_i^{(j)}, y_i^{(j)}) + \lambda N(\theta)$$

The parties can collaboratively learn a DP model via iterative learning by adding noise to the aggregated gradients within the MPC in each iteration with the folowing noise bound:

Theorem Given a centralized model estimator θ_T obtained by minimizing $J_D(\theta)$ after T iterations of gradient descent, executed jointly by m parties, if the learning rate is $1/L$, and the gradients are perturbed with noise $z \in \mathcal{N}(0, \sigma^2 I_d)$, then θ_T is (ϵ, δ)-DP if

$$\sigma^2 = \frac{8G^2 T \log(1/\delta)}{m^2 n_{(1)}^2 \epsilon^2}$$

Proof: By zCDP lemmas.

Additionally, we observe that an adversary cannot obtain additional information from the intermediate computations.
Corollary Intermediate model estimator θ_t at each iteration is $(\sqrt{t/T\epsilon}, \delta)$-DP.

References I

Bun, M. and Steinke, T. (2016).
Concentrated differential privacy: Simplifications, extensions, and lower bounds.
In *Theory of Cryptography Conference*, pages 635–658. Springer.

Chaudhuri, K., Monteleoni, C., and Sarwate, A. D. (2011).
Differentially private empirical risk minimization.
Journal of Machine Learning Research, 12(Mar):1069–1109.

Dwork, C., McSherry, F., Nissim, K., and Smith, A. (2006).
Calibrating noise to sensitivity in private data analysis.
In *Theory of cryptography conference*, pages 265–284. Springer.

Jayaraman, B., Wang, L., Evans, D., and Gu, Q. (2018).
Distributed learning without distress: Privacy-preserving empirical risk minimization.
In *Advances in Neural Information Processing Systems*, pages 6346–6357.

Backup slide: Definition of zCDP

Privacy loss random variable:

$$Z = \log \frac{\Pr[\mathcal{M}(D) = o]}{\Pr[\mathcal{M}(D') = o]}$$

ρ-zCDP imposes a bound on the *moment generating function* of the privacy loss Z, it needs to satisfy:

$$\mathbb{E}[e^{(\alpha-1)Z}] \leq e^{(\alpha-1)\alpha\rho}, \forall\alpha \in (1, \infty)$$