Locally Differentially Private Bandits Learning

Weiran Huang
Huawei Noah's Ark Lab

Joint work with Kai Zheng, Tianle Cai, Zhenguo Li and Liwei Wang

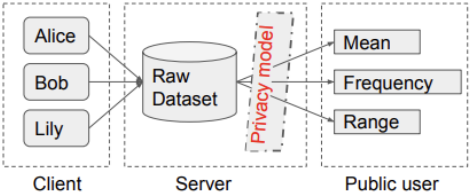
NeurIPS 2020

NOAH'S ARK LAB

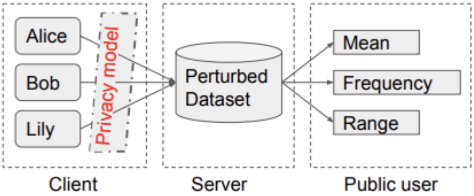
Your Privacy is Important

- Machine learning is based on data, but data usually contain user's private information.
- Direct implementation would leak users' privacy.
- Examples:
  - 2010 Netflix's recommendation contest
  - 2016 Deepmind's NHS data-sharing deal
- Recently, people pay more and more attention to their privacy.
- There are also regulations for privacy protection, e.g., GDPR.
- Privacy is an important part of trustworthy AI.

How to Define Privacy?



(a) Centralized differential privacy



(b) Local differential privacy

Local Differential Privacy (LDP)

Definition (LDP)

A mechanism $\mathcal{Q} : \mathcal{C} \to \mathcal{Z}$ is said to protect (ε, δ)-LDP, if for any two data $x, x' \in \mathcal{C}$, and any (measurable) subset $U \subset \mathcal{Z}$, there is

$$\Pr[Q(x) \in U] \leqslant e^{\varepsilon} \Pr[Q(x') \in U] + \delta.$$

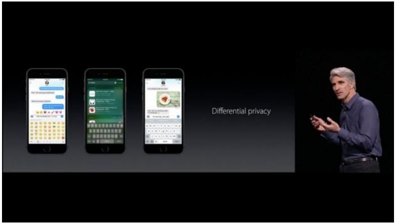
In particular, if $\mathcal{Q}$ preserves $(\varepsilon, 0)$-LDP, we call it ε-LDP.

- Goal of Privacy Preserving Machine Learning: To design algorithms/models with nearly optimal performance while protecting data privacy.
- Trade-off between privacy and accuracy.
- Difficulty: what we have collected is noisy data!

LDP Applications

LDP has been used in many real applications:

- Apple
- Google
- Uber
- Microsoft
- ...



Most of them are about statistical estimation, rather than learning.

We consider a common learning problem (i.e., bandit learning) with privacy preserving.

Bandit Learning

Offline Learning v.s. Online Learning
Denote dataset as $\{(x_t, y_t) | t \in [T]\}$:

1. data collection: non-interactive; interactive
2. data assumption: i.i.d.; adversary or i.i.d.

A general model: Bandit Convex Optimization (BCO)

BCO

For round $t = 1, 2, ..., T$, the server:
- chooses a prediction $x_t \in \mathcal{X}$ based on previous collected losses
- suffers and observes a loss value $f_t(x_t)$.

Performance measurement: regret $\max_{x \in \mathcal{X}} \mathbb{E}[\sum_{t=1}^{T} f_t(x_t) - f_t(x)]$.
Goal: Design algorithm with small regret.

Application: Movie Recommendation

Movie Recommendation

For round $t = 1, 2, ..., T$,

1. a user u_t comes.
2. the server chooses an movie $x_t \in \mathcal{X}$ based on previous collected rating scores.
3. the user rates the movie x_t and sends the score $f_t(x_t)$ to the server.

Now, what if we want to protect users' privacy in the above scenario? More generally, **how can we add privacy preserving to BCO problems**?

Basic Building Blocks in LDP

We introduce a basic mechanism in LDP literature – Gaussian Mechanism.
Given any function $h : \mathcal{C} \to \mathbb{R}^d$.
Define sensitivity $\Delta := \max_{x,x' \in \mathcal{C}} \|h(x) - h(x')\|_2$.

Gaussian Mechanism

Gaussian Mechanism is defined as $h(x) + Y$, where random vector Y is sampled from Gaussian distribution $\mathcal{N}(0, \sigma^2 \mathrm{I}_d)$ with $\sigma = \frac{\Delta\sqrt{2\ln(1.25/\delta)}}{\varepsilon}$.

One can prove Gaussian Mechanism preserves (ε, δ)-LDP.

One Point Feedback Private Learning

Algorithm 1: One-Point Bandits Learning-LDP

Input: non-private algorithm $\mathcal{A}$, privacy parameters ε, δ

Initialize: set $\sigma = \frac{2B\sqrt{2\ln(1.25/\delta)}}{\varepsilon}$

For $t = 1, 2, \ldots$

Server plays $x_t \in \mathcal{X}$ returned by $\mathcal{A}$;

User u_t suffers loss $f_t(x_t)$ and sends $f_t(x_t) + Z_t$ to the server, where $Z_t \sim \mathcal{N}(0, \sigma^2)$;

The server receives $f_t(x_t) + Z_t$ and calculates x_{t+1}.

According to the Gaussian mechanism, the guarantee of (ϵ, δ)-LDP is trivial.

Performance Guarantee

Theorem

Suppose non-private algorithm $\mathcal{A}$ achieves regret $\mathrm{Reg}_{\mathcal{A}}^{T}$ for BCO. We have the following guarantee for Algorithm 1: for any $x \in \mathcal{X}$, there is

$$\mathbb{E}\left[\sum_{t=1}^{T} f_t(x_t) - f_t(x)\right] \leqslant \tilde{\mathcal{O}}\left(\frac{\ln(T/\delta)}{\varepsilon} \cdot \mathrm{Reg}_{\mathcal{A}}^{T}\right) \tag{1}$$

where expectation is taken over the randomness of non-private algorithm $\mathcal{A}$ and all injected noise.

With the above theorem, by plugging different non-private optimal algorithms under variant cases, we can obtain corresponding regret bounds with LDP guarantee.

Bounds for Some BCO Cases

Problem		Our Regret	Previous Best
BCO	Convex	$\tilde{\mathcal{O}}\left(T^{3/4}/\varepsilon\right)$	$\tilde{\mathcal{O}}\left(T^{3/4}/\varepsilon\right)$
	Convex + Smooth	$\tilde{\mathcal{O}}\left(T^{2/3}/\varepsilon\right)$	$\tilde{\mathcal{O}}\left(T^{3/4}/\varepsilon\right)$
	S.C	$\tilde{\mathcal{O}}\left(T^{2/3}/\varepsilon\right)$	$\tilde{\mathcal{O}}\left(T^{2/3}/\varepsilon\right)$
	S.C + Smooth	$\tilde{\mathcal{O}}\left(T^{1/2}/\varepsilon\right)$	$\tilde{\mathcal{O}}\left(T^{2/3}/\varepsilon\right)$

Advantages of our approach:

1. Nearly optimal performance;
2. Strict privacy guarantee;
3. Black-box reduction;
4. A unified framework and analysis.

Extend to Multi-Point Bandit

In some cases, the server can observe multiple-point feedbacks.
For example: same user, recommend multiple items.

Suppose we are permitted to query K points $x_{t,1}, \ldots, x_{t,K}$ per round, and we observe $f_t(x_{t,1}), \ldots, f_t(x_{t,K})$. The expected regret is defined as

$$\mathbb{E}\left[\frac{1}{K}\sum_{t=1}^{T}\sum_{k=1}^{K} f_t(x_{t,k})\right] - \min_{x\in\mathcal{X}} \mathbb{E}\left[\sum_{t=1}^{T} f_t(x)\right] \tag{2}$$

where the expectation is taken over the randomness of algorithm.

We consider that $\{f_t(x)\}$ are G-Lipschitz convex functions.

Multi-Point Bandit

There are some gaps between multi-point and one-point bandit:

One-point: $\Theta(\sqrt{T})$ for convex, even for strongly convex.

Multi-point: $\Theta(\sqrt{T})$ for convex, $\Theta(\log T)$ for strongly convex.

There is not much difference between $K = 2$ and $K \geq 2$.

Algorithm 2: Two-Point Feedback Private BCO

Input: set $\mathcal{A}$ as the algorithm in [Agarwal et al., 2010] with parameters η, ρ, ξ, privacy parameters ε, δ

Initialize: set $\sigma = \frac{2G\sqrt{2\ln(1.25/\delta)}}{\varepsilon}, \eta = \frac{1}{\sqrt{T}}, \rho = \frac{\log T}{T}, \xi = \frac{\rho}{r}$

For $t = 1, 2, \ldots$

1. Server plays $x_{t,1}, x_{t,2} \in \mathcal{X}$ received from $\mathcal{A}$
2. User suffers $f_t(x_{t,1}), f_t(x_{t,2})$
3. User passes $f_t(x_{t,1}) - f_t(x_{t,2}) + n_t^\top(x_{t,1} - x_{t,2})$ to $\mathcal{A}$ in the server, where $n_t \sim \mathcal{N}(0, \sigma^2 \mathrm{I}_d)$.

Theoretical Guarantee

Algorithm 2 guarantees (ε, δ)-LDP.

Theorem

For any $x \in \mathcal{X}$, Algorithm 2 guarantees

$$\mathbb{E}\left[\frac{1}{2}\sum_{t=1}^{T}(f_t(x_{t,1}) + f_t(x_{t,2})) - f_t(x)\right] \leqslant \tilde{\mathcal{O}}\left(\frac{d^3\sqrt{T}}{\varepsilon^2}\right) \tag{3}$$

If $\{f_t\}$ are further μ strongly convex, set $\eta = \frac{1}{\mu t}, \rho = \frac{\log T}{T}, \xi = \frac{\rho}{r}$, then for any $x \in \mathcal{X}$, we have

$$\mathbb{E}\left[\frac{1}{2}\sum_{t=1}^{T}(f_t(x_{t,1}) + f_t(x_{t,2})) - f_t(x)\right] \leqslant \tilde{\mathcal{O}}\left(\frac{d^3 \log T}{\mu\varepsilon^2}\right) \tag{4}$$

LDP Generalized Linear Bandits

In the end, we consider a more practical contextual bandits learning.

At each round t, the learner chooses an action $x_t \in \mathcal{X}_t$ in the local side, where $\mathcal{X}_t$ contains the features about underlying arms. Then the user generates a reward $y_t = g(x_t^\top \theta^*) + \eta_t$, where θ^* is the unknown true parameter to be learned, g is a known function, and η_t is a random noise in $[-1, 1]$ with mean 0.

We define the regret as $\mathrm{Reg}_T^{\mathcal{A}} := \sum_{t=1}^{T} g(x_{t,*}^\top \theta^*) - g(x_t^\top \theta^*)$ where $x_{t,*} := \mathrm{argmax}_{x \in \mathcal{X}_t} g(x^\top \theta^*)$.

Main idea of Algorithm 3: The parameters that users send to the server are $x_t x_t^\top$ and $x_t^\top \hat{\theta}_t x_t$. Thus, perturbing these parameters using Gaussian noise can guarantee (ϵ, δ)-LDP.

Performance Guarantee

Theorem

With probability at least $1-\alpha$, the regret of Algorithm 3 satisfies the following bound:

$$\mathrm{Reg}_T \leqslant \tilde{\mathcal{O}}\left(\sqrt{\log\frac{1}{\delta}\log\frac{1}{\alpha}\log\frac{T}{d}}\frac{(dT)^{3/4}}{\varepsilon}\right) \tag{5}$$

Note that our upper bound is in order $\tilde{\mathcal{O}}\left(T^{3/4}\right)$, which differs from common $\mathcal{O}(\sqrt{T})$ regret bound in corresponding non-private settings. We conjecture this order is nearly the best one can achieve in LDP setting, mainly because we need to protect more information, i.e., both contexts and corresponding rewards.

Summary

In summary,

- We propose simple black-box reduction frameworks that can solve a large family of context-free bandits learning problems with LDP guarantee.
- We improve previous best results for private bandits learning with one-point feedback and give the first result for BCO with multi-point feedback under LDP.
- We extend our (ϵ, δ)-LDP algorithm to Generalized Linear Bandits and gives a sub-linear regret $\tilde{\mathcal{O}}(T^{3/4}/\varepsilon)$ which is conjectured to be nearly optimal.

Thank you!

We are looking for research interns (Contact me for details).