Principled Private Data Release with Deep Learning

Michael Fine Harvard University
Cambridge MA, 02138
mfine@college.harvard.edu

1 Background

1.1 Query Release Problem

We study the problem of privately generating synthetic data to answer statistical queries over a data universe $\mathcal{X}$. Formally, a statistical query over $\mathcal{X}$ is a function $q : \mathcal{X} \to \{0, 1\}$. Given a a dataset $S \in \mathcal{X}^n$, we define $q(S) = \sum_{s \in S} q(s)$.

Our goal is to produce a synthetic dataset that, for every query in some family of queries, takes approximately the same value as the true dataset.

Definition 1. *$\alpha-$approximate: We say a synthetic dataset S $\alpha-$approximates a true dataset $\hat{S}$ w.r.t a family of statistical queries $\mathcal{Q}$ if*

$$\forall q \in \mathcal{Q} : \quad |q(S) - q(\hat{S})| \leqslant \alpha \tag{1}$$

[TODO finish up]

1.2 Game Theoretic Formulation

One can formulate the problem of producing an $\alpha-$approximate dataset as a two-player, zero sum game [3] between a discriminator D and a generator G. The generator has an action set $\mathcal{X}$, while the discriminator has an action set $\mathcal{Q}$. The generator aims to output a dataset $S \in \mathcal{X}$ that maximally agrees with $\hat{S}$, while the discriminator aims to find queries $q \in \mathcal{Q}$ that distinguish $\hat{S}$ and S.

Formally, given a play $S \in \mathcal{X}$ and $q \in Q$, the discriminator gets payoff $V(S, q)$ and the generator gets payoff $V(S, q)$, where $V(S, q)$ denotes:

Definition 2. *Payoff*

$$V(S, q) := |q(S) - q(\hat{S})| \tag{2}$$

The goal of both G and D is to maximize their worst case payoffs, thus

$$\max_{q \in \mathcal{Q}} \min_{S \in X} V(S, q) \ \text{(Goal of } D) \quad \textit{and} \quad \min_{S \in X} \max_{q \in \mathcal{Q}} V(S, q) \ \text{(Goal of } G) \tag{3}$$

If there exists a point (S^*, q^*) such that neither G nor D can improve their payoffs by playing a different move, we call that a *Pure Nash Equilibrium*. Unfortunately, a pure equilibrium is not always guaranteed to exist (and likely does not in the case of synthetic data generation).

However, the seminal work of Nash et. al showed that there always exists a *Mixed Nash Equilibrium (MNE)*, where the players play *probability distributions* over their action sets, instead of fixed actions.

Let $\Delta(\mathcal{X})$ and $\Delta(\mathcal{Q})$ denote the set of probability distribution over $\mathcal{X}$ and $\mathcal{Q}$. Formally, if G plays a strategy $u \in \Delta(\mathcal{X})$ and D plays $w \in \Delta(\mathcal{Q})$, we define the payoff to be the expected value of a single draw:

Preprint. Under review.

$$V(u, w) := \mathbb{E}_{S \sim u, q \sim w} V(S, q) \tag{4}$$

Thus, a pair of strategies $u^* \in \Delta(\mathcal{X})$ and $w^* \in \Delta(\mathcal{Q})$ forms an $\alpha-$approximate mixed nash equilibrium if for all strategies $u \in \Delta(\mathcal{X})$ and $w \in \Delta(\mathcal{Q})$

$$V(u^*, w) \leqslant V(u, w) + \alpha \quad and \quad V(u, w^*) \leqslant V(u, w) - \alpha \tag{5}$$

Moreover, Gaboardi et. al showed how to reduce the problem of finding an $\alpha-$approximate dataset to the problem of finding an an $\alpha-$equilibrium in the query release game:

Theorem 1. *Let (u, w) be the $\alpha-$approximate MNE in a query release game for a dataset $\hat{S} \in \mathcal{X}$ and a query universe $\mathcal{Q}$. If $\mathcal{Q}$ is closed under negation, then the dataset S sampled from u α-approximates $\hat{S}$ over Q. [1]*

Hence, our task is to provide an algorithm to private reach an $\alpha-$MNE in the query release game. In the following section, we will provide the background for how this can be done with GANs.

1.3 Generative Adversarial Networks

Generative Adversarial Networks (GANs) TODO:

1.4 Online Learning

TODO: Freund and Shcapire TODO: Convex Concave

2 Results

Unfortunately, the value of the game V is not convex-concave, and therefore TODO: Theorem on no regret games solving convex concave does not apply directly. However, Grnarova et al showed that when the discriminator D is a single-layer neural network, V becomes concave with respect to D, or *semi-concave* [2].

- Replace D with multiplicative weights
- Replace D with

References

[1] Marco Gaboardi, Emilio Jesús Gallego Arias, Justin Hsu, Aaron Roth, and Zhiwei Steven Wu. Dual Query: Practical Private Query Release for High Dimensional Data. *arXiv:1402.1526 [cs]*, February 2014.

[2] Paulina Grnarova, Kfir Y. Levy, Aurelien Lucchi, Thomas Hofmann, and Andreas Krause. An Online Learning Approach to Generative Adversarial Networks. *arXiv:1706.03269 [cs, stat]*, June 2017.

[3] Justin Hsu, Aaron Roth, and Jonathan Ullman. Differential Privacy for the Analyst via Private Equilibrium Computation. *Proceedings of the 45th annual ACM symposium on Symposium on theory of computing - STOC '13*, page 341, 2013.