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 $x \in \mathcal{X}^n$, we define $q(x) = \sum_{x_i \in x} q(x_i)$.

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 x $\alpha-$approximates a true dataset $\hat{x}$ w.r.t a family of statistical queries $\mathcal{Q}$ if*

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

[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 [8] 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 $x \in \mathcal{X}$ that maximally agrees with $\hat{x}$, while the discriminator aims to find queries $q \in \mathcal{Q}$ that distinguish $\hat{x}$ and x.

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

Definition 2 (Payoff)**.**

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

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

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

If there exists a point (x^*, 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 $g \in \Delta(\mathcal{X})$ and D plays $d \in \Delta(\mathcal{Q})$, we define the payoff to be the expected value of a single draw:

Preprint. Under review.

$$V(g,d) := \mathbb{E}_{x\sim g, q\sim d} V(x,q) \tag{4}$$

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

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

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 $\hat{x} \in \mathcal{X}$ and a query universe $\mathcal{Q}$. If $\mathcal{Q}$ is closed under negation, then the dataset S sampled from u α-approximates $\hat{x}$ over Q. [4]*

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)

1.4 Online Learning

To efficiently find equilibrium in the zero-sum GAN game, we draw on results from online learning. In the online learning setting, in each of T rounds a player is given a loss function f_t , possibly adversarial chosen. The players goal is to chose an action $x_{t+1} \in \mathcal{X}$ in order to minimize the cumulative *regret*.

Definition 3 (Regret)**.** *The regret measures the cumulative loss of the player, compared to the best fixed decision in hindsight.*

$$Regret_T(f_1, ..., f_T) = \sum_{t=1}^{T} f_t(x_t) - min_{x\in\mathcal{X}} \sum_{t=1}^{T} f_t(x^*) \tag{6}$$

When a strategy provably leads to regret is sublinear in T, we call that *no-regret*, as regret $\to 0$ as $T \to \infty$.

One approach to regret minimization is to choose the action x_{t+1} that minimizes the cumulative loss over all past loss functions

$$x_{t+1} = \arg\min_{x\in\mathcal{X}} \sum_{t=1}^{T} f_t(x) \tag{7}$$

This approach is known as *Follow-The-Leader*. While natural, this approach is easily exploitable by an adversary. At a high level, this is because it *overfits* to past outcomes, allowing it to optimize between suboptimal strategies. To rectify this, a powerful strategy is *Follow-the-Regularized-Leader*

Definition 4 (Follow-The-Reguralized-Leader (FTRL))**.** *Given a reguralization function $R(x)$ and a regularization weight η_T, at each step choose $x_{t+1} \in \mathcal{X}$ to minimize the regularized cumulative loss:*

$$x_{t+1} = \arg\min_{x\in\mathcal{X}} \sum_{t=1}^{T} +\eta_T R(X) \tag{8}$$

One common regularization function is the l_2 norm $R(x) = ||x||_2$.

When the loss function f_T is convex, FTRL can be shown to be no-regret [7]. Unfortunately, in the GAN setting, where the loss function f_T is defined by a highly non-convex deep network, we don't have that guarantee.

1.4.1 Online Learning for Non-convex losses

In general, finding the minimum of a sum of non-convex functions is hard. However, in practice gradient descent over neural networks has proven to be remarkably effective at approximately solving non-convex loss functions. This leads to the natural question: assuming we have an offline non-convex optimization oracle $\mathcal{O}$, can we use that to find a no-regret strategy in the online setting? Agarwal et al showed that we can, using a Follow-The-Leader variant known as Follow-The-Perturbed-Leader [1]. Formally:

Definition 5 (Offline optimization oracle)**.** *Let $\mathcal{O}$ take a sequence of (possibly non-convex) loss functions $(f_1...f_T) \in \mathcal{L}^T$ and a d-dimensional vector d, and output $x^* \in \mathcal{X}$*

$$x^* \in \arg\min_{x \in \mathcal{X}} \sum_{t=1}^{T} f_t(x) + \sigma^T x \tag{9}$$

If we relax the requirement to allow $\mathcal{O}$ to output an approximate minimizer x^*

$$x^* \leqslant \left(\arg\min_{x \in \mathcal{X}} \sum_{t=1}^{T} f_t(x) + \sigma^T x \right) + \alpha \tag{10}$$

we then call $\mathcal{O}$ an $\alpha-$approximate offline oracle.

Extend [1] defn of Ø to be within α

We can use this offline oracle $\mathcal{O}$ to minimize regret in the online case:

Definition 6 (Follow-The-Perturbed-Leader (FTPL))**.** *Given an offline oracle $\mathcal{O}$ and a parameter η, at each step t FTPL draws a random vector $\sigma_t \sim Exp(\eta)^d$. It then outputs*

$$x^* \in \arg\min_{x \in \mathcal{X}} \sum_{t=1}^{T} f_t(x) + \sigma^T x \tag{11}$$

Theorem 2. *FTPL has sublinear regret. [1]*

Building on the work of Freund and Schapire, we can show that if G and D both play FTPL for T rounds, they will converge to an $\alpha-$approximate equilibrium. Formally:

Theorem 3. *Suppose that G and D play according to FTPL. We can choose $T \in poly(d)/\alpha^3$ such that the expected average regret of FTPL is at most α. Then, G_T and D_T produce strategies in an $\alpha-$approximate equilibrium [1].*

2 Results

summarize approach

Theorem 3 requires the existence of an actual oracle $\mathcal{O}$ that can find the minimum of a perturbed sum of non-convex functions. In this case, we need an oracle that can minimize the sub of deep neural networks. While SGD is remarkably effective in practice, we are unable to provide guarantees (probabilistic or otherwise) about how close the convergent solution SGD outputs is to the global minima. Recent work suggests that this is not a problem in practice, as spurious local minima (local minima significantly worse than the global minima) get exponentially rarer as the network gets larger [2]. However, these results still rely on too many impractical assumptions to make them relevant in practice.

Show that GAN objective is concave wrt discriminator paramters ala [6]

However, while we cannot guarantee (or even certify) convergence to an approximate global minima in general, we can take advantage of the specific structure of the query release problem.

Theorem 4. *For any discriminator, the optimal generator G* has payoff $-V(g, d) = 0$. One such generator draws a uniform sample from the true dataset $\hat{x}$.*

Make clearer the distinction between queries over single rows and queries over all rows,

Proof. Recall that the generators payoff is $-V(g, d)$, where the value V is defined as $V(g, d) := \mathbb{E}_{x \sim g, q \sim d} |q(x) - q(\hat{x})|$, where $\hat{x}$ is the true, sensitive dataset. Clearly, the generator's payoff is at most 0. This is trivially attaiable if g is the uniform distribution over the rows of $\hat{x}$. □

Maybe another line proving this

Theorem 4 allows us to track how close to optimal the generator G_i is at each step. Assuming D_i is optimal, the regret at each step is simply the generator loss $V(G_i, D_i)$. The cumulative regret is simply the sum of the generator loss at each step.

This gives us a way to track *generator* optimality, assuming the discriminator is optimal. Unlike the generator, there is no obvious maximum payoff for the discriminator in general. However, if we restrict the class of discriminators to single layer neural networks parametrized by θ:

$$D_\theta(x) = D_\theta(x) = \sigma(\theta^T x + b) \tag{12}$$

then $V(G, D_\theta)$ becomes convex with respect to θ, and we can use standard gradient descent methods to guarantee convergence to a global optimum.

Talk about how this relies on discrimiantor optimality

Algorithm 1: QueryGAN

Input: one-layer discriminator D_θ, deep generator G_ϕ, offline optimization oracle $\mathcal{O}$, Rounds T, noise η, output dimension d, game objective V

Result: Trained generator G_T, Accuracy α

for $t \in 1...T$ **do**

Draw discriminator and generator perturbations

$\sigma_1 \sim Exp(\eta)^d \;\; and \;\; \sigma_2 \sim Exp(\eta)^d$

Update D and G according to FTPL:

$\theta_{t+1} \leftarrow \theta_t - \nabla_{\theta_t}\left(\sum_{t=1}^{T} f_t + \sigma_1^T x\right)$ and $\phi_{t+1} \leftarrow \phi_t - \nabla_{\phi_t}\left(\sum_{t=1}^{T} g_t + \sigma_2^T x\right)$

Update losses:

$f_{t+1}(\cdot) = V(\cdot, D_{\phi_{t+1}}) \;\; and \;\; g_{t+1}(\cdot) = V(G_{\theta_{t+1}}, \cdot)$

Calculate cumulative regret: $R \leftarrow \sum_{t \in T} f_t(G_{\phi_t})$

return *Mixed strategy:* $G \sim Unif\{G_{\theta_1}, G_{\theta_T}\}$, *Regret:* R

Theorem 5. *Let G, α be the results of running QueryGAN with inputs . Let x be the dataset sampled from G*

What inputs

$$x := \{G(z_1), ..., G(z_n)\} \;\; where \; z \sim P_z^n \tag{13}$$

what is P_z

Then x is $\alpha-$approximate with respect to all queries representable by D.

Proof. □

prove

Explain context, tracking performance

2.1 QueryGAN privacy

Privacy is ensured by the addition of exponentially distributed noise σ_1, σ_2. Interestingly, the original purpose of this noise is not privacy, but rather to ensure convergence of the online algorithm. Because of the deep connections between differential privacy and online learning [9] [5], however, this noise also perfectly ensures ϵ-differential privacy.

2.1.1 Tracking privacy loss with moments accountant

2.1.2 Reporting Regret Bounds

-
-
-

GAN privacy

Talk about privately reporting α with report noisy max

Maybe PATE-GAN

2.2 Extensions

2.2.1 Marginals

While constraining D to be a one layer linear discriminator is restricting, it still is capable of representing a number of query families of interest. Specifically, D contains all k-way marginals.

Definition 7 (Marginal)**.** *A marginal $m : \mathcal{X} \to \{0,1\}$ over a row $x \in \{0,1\}^m$ is a monotone conjunction, parametrized by some subset S of the input features.*

$$m_S(x) = \prod_{i \in S} x_i \tag{14}$$

We extend this to a dataset X of n rows by defining $m(X) = \sum_{x \in X} m(x)$. A k-way marginal is a marginal restricted to k features.

[3]

A k-way marginal can be thought of as counting the number of rows with the same value in the chosen k features. Marginals are a useful way of providing a synopsis of a dataset that still captures complex relationships between features. Producing a differentially private synthetic dataset that agrees with all $k-$way marginals of the true dataset is an extremely well studied problem in the field . .

However, we can show that if $QueryGAN$ succeeds, it is able to match all $k-way$ marginals. This follows from the fact that a linear discriminator can contain all marginals.

Theorem 6. *Let D be single layer discriminator parametrized by θ with a sigmoid activation function s.t. $D_\theta(x) = \sigma(\theta^T x + b)$. For any single-row marginal m, there exists θ, b s.t. $D_\theta(x) = m(x)$ for all x.*

Proof. This follows from the definition of a marginal. Let m_S be the marginal over the features S. Let $\theta_i =$ □

Theorem 7. *Let G, α be the output of running $QueryGAN$ with . Then G will generate a dataset with all marginal counts accurate to within α.*

The proof of this theorem follows directly from Theorem 4 and 5.

2.3 QueryGAN with alternate discriminators

While QueryGAN follows the standard GAN practice of representing both the generator and discriminator with a neural network, this is not mandatory. Indeed, given that D is represented by the almost trivially simple 1 layer network, D is best understood in more general terms than a neural network.

Definition 8 (Tractable Discriminator Set)**.** *We say a class of functions $\mathcal{F} : \mathcal{X} \to \{0,1\}$ parametrizes a set of tractable discriminators w.r.t a class of queries $\mathcal{Q}$ iff*

1. *$\mathcal{Q} \subseteq \mathcal{F}$*
2. *There exists a tractable offline oracle $\mathcal{O}$*

As shown above, the set $\mathcal{F}_{single}$ of one layer neural networks is a tractable discriminator set for all marginals (as well as all sigmoided linear functions in general).

2.3.1 Multiplicative Weights

Consider the renowned Multiplicative Weights algorithm.

2.4 TODO

- Replace D with multiplicative weights
- Local DP GAN

Survey marginal results

Impossibility results

define sigmoid

Doesn't work because sigmoid is weird, need to rething activation

Parameters, steps etc

Oracle runtime (also comment on non-oracle runtime)

Summary, contextualize empirical results

What does it do (also define tractable)

References

[1] Naman Agarwal, Alon Gonen, and Elad Hazan. Learning in Non-convex Games with an Optimization Oracle. *arXiv:1810.07362 [cs, stat]*, October 2018.

[2] Anna Choromanska, Mikael Henaff, Michael Mathieu, Gérard Ben Arous, and Yann LeCun. The Loss Surfaces of Multilayer Networks. *arXiv:1412.0233 [cs]*, November 2014.

[3] Cynthia Dwork and Aaron Roth. The Algorithmic Foundations of Differential Privacy. *Foundations and Trends® in Theoretical Computer Science*, 9(3-4):211–407, 2013.

[4] 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.

[5] Alon Gonen, Elad Hazan, and Shay Moran. Private Learning Implies Online Learning: An Efficient Reduction. *arXiv:1905.11311 [cs, stat]*, May 2019.

[6] 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.

[7] Elad Hazan. Introduction to Online Convex Optimization. *arXiv:1909.05207 [cs, math, stat]*, September 2019.

[8] 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.

[9] Seth Neel, Aaron Roth, Giuseppe Vietri, and Zhiwei Steven Wu. Differentially Private Objective Perturbation: Beyond Smoothness and Convexity. *arXiv:1909.01783 [cs, stat]*, September 2019.