New Oracle-Efficient Algorithms for Private Synthetic Data Release

Giuseppe Vietri* Grace Tian † Mark Bun‡ Thomas Steinke§
Zhiwei Steven Wu¶

Abstract

We present three new algorithms for constructing differentially private synthetic data—a sanitized version of a sensitive dataset that approximately preserves the answers to a large collection of statistical queries. All three algorithms are *oracle-efficient* in the sense that they are computationally efficient when given access to an optimization oracle. Such an oracle can be implemented using many existing (non-private) optimization tools such as sophisticated integer program solvers. While the accuracy of the synthetic data is contingent on the oracle's optimization performance, the algorithms satisfy differential privacy even in the worst case. For all three algorithms, we provide theoretical guarantees for both accuracy and privacy. Through empirical evaluation, we demonstrate that our methods scale well with both the dimensionality of the data and the number of queries. Compared to the state-of-the-art method High-Dimensional Matrix Mechanism McKenna et al. (2018), our algorithms provide better accuracy in the large workload and high privacy regime (corresponding to low privacy loss ε).

1 Introduction

The wide range of personal data collected from individuals has facilitated many studies and data analyses that inform decisions related to science, commerce, and government policy. Since many of these rich datasets also contain highly sensitive personal information, there is a tension between releasing useful information about the population and compromising the privacy of individuals. In this work, we consider the problem of answering a large collection of statistical (or linear) queries subject to the constraint of differential privacy. Formally, we consider a data domain $\mathcal{X} = \{0,1\}^d$ of dimension d and a dataset $D \in \mathcal{X}^n$ consisting of the data of n individuals. Our goal is to approximately answer a large class of statistical queries $\mathcal{Q}$ about D. A statistical query is defined by a predicate $\phi\colon \mathcal{X} \to [0,1]$, and the query $q_\phi : \mathcal{X}^n \to [0,1]$ is given by $q_\phi(D) = \frac{1}{n}\sum_{i=1}^n \phi(D_i)$ and an approximate answer $a \in [0,1]$ must satisfy $|a - q_\phi(D)| \le \alpha$ for some accuracy parameter $\alpha > 0$. To preserve privacy we work under the constraint of differential privacy (Dwork et al., 2006). Privately answering statistical queries is at the heart of the 2020 US Census release (Abowd, 2018) and provides the basis for a wide range of private data analysis tasks. For example, many machine learning algorithms can be simulated using statistical queries (Kearns, 1998).

An especially compelling way to perform private query release is to release *private synthetic data* – a sanitized version of the dataset that approximates all of the queries in the class $\mathcal{Q}$. Notable examples of private synthetic data algorithms are the SmallDB algorithm (Blum et al., 2008) and the private multiplicative

*Department of Computer Science and Engineering, University of Minnesota. Supported by the GAANN fellowship from the U.S. Department of Education.
†Harvard University.
‡Boston University. Supported by NSF grant CCF-1947889. Part of this work was done at the Simons Institute for the Theory of Computing, supported by a Google Research Fellowship.
§IBM, Almaden Research Center
¶Department of Computer Science and Engineering, University of Minnesota. Supported in part by a Google Faculty Research Award, a J.P. Morgan Faculty Award, a Mozilla research grant, and a Facebook Research Award.

weights (PMW) mechanism (Hardt & Rothblum, 2010) (and its more practical variant the multiplicative weights exponential mechanism MWEM (Hardt et al., 2012)), which can answer exponentially many queries and achieves nearly optimal sample complexity (Bun et al., 2018). Unfortunately, both algorithms involve maintaining a probability distribution over the data domain $\mathcal{X} = \{0,1\}^d$, and hence suffer exponential (in d) running time. Moreover, under standard cryptographic assumptions, this running time is necessary in the worst case (Ullman, 2016; Ullman & Vadhan, 2011). However, there is hope that these worst-case intractability results do not apply to real-world datasets.

To build more efficient solutions for constructing private synthetic data, we consider *oracle efficient* algorithms that rely on a black-box optimization subroutine. The optimization problem is NP-hard in the worst case. However, we invoke practical optimization heuristics for this subroutine (namely integer program solvers such as CPLEX and Gurobi). These heuristics work well on many real-world instances. Thus the algorithms we present are more practical than the worst-case hardness would suggest is possible. While the efficiency and accuracy of our algorithms are contingent on the solver's performance, differential privacy is guaranteed even if the solver runs forever or fails to optimize correctly.

Overview of our results. To describe our algorithms, we will first revisit a formulation of the query release problem as a zero-sum game between a data player who maintains a distribution $\widehat{D}$ over $\mathcal{X}$ and a a query player who selects queries from $\mathcal{Q}$ (Hsu et al., 2013; Gaboardi et al., 2014). Intuitively, the data player aims to approximate the private dataset D with $\widehat{D}$, while the query player tries to identify a query which distinguishes between D and $\widehat{D}$. Prior work Hsu et al. (2013); Gaboardi et al. (2014) showed that any (approximate) equilibrium for this game gives rise to an accurate synthetic dataset. To study the private equilibrium computation within this game, we consider a *primal* framework and a *dual* framework that enable us to unify and improve on existing algorithms.

In the primal framework, we perform the equilibrium computation via the following *no-regret dynamics*: over rounds, the data player updates its distribution $\widehat{D}$ using a no-regret online learning algorithm, while the query player plays an approximate best response. The algorithm MWEM in prior work falls under the primal framework with the data player running the multiplicative weights (MW) method as the no-regret algorithm, and the query player privately responding using the exponential mechanism (McSherry & Talwar, 2007). However, since the MW method maintains an entire distribution over the domain $\mathcal{X}$, MWEM runs in exponential time even in the best case. To overcome this intractability, we propose two new algorithms FEM and sepFEM that follow the same no-regret dynamics, but importantly replace the MW method with two variants of the follow-the-perturbed-leader (FTPL) algorithm (Kalai & Vempala, 2005)—Non-Convex-FTPL (Suggala & Netrapalli, 2019) and Separator-FTPL (Syrgkanis et al., 2016)—both of which solve a perturbed optimization problem instead of maintaining an exponential-sized distribution. FEM achieves an error rate of

$$\alpha = \widetilde{O}\left(d^{3/4} \log^{1/2} |\mathcal{Q}|/n^{1/2}\right),$$

and sepFEM achieves a slightly better rate of

$$\alpha = \widetilde{O}\left(d^{5/8} \log^{1/2} |\mathcal{Q}|/n^{1/2}\right),$$

although the latter requires the query class $\mathcal{Q}$ to have a structure called a small separator set. In contrast, MWEM attains the error rate $\alpha = \widetilde{O}\left(d^{1/4} \log^{1/2} |\mathcal{Q}|/n^{1/2}\right)$. Although the accuracy analysis requires repeated sampling from the FTPL distribution (and thus repeatedly solving perturbed integer programs), our experiments show that the algorithms remain accurate even with a much lower number of samples, which allows much more practical running time.

We then consider the *dual* formulation and improve upon the existing algorithm DualQuery (Gaboardi et al., 2014). Unlike MWEM, DualQuery has the query player running MW over the query class $\mathcal{Q}$, which is often significantly smaller than the data domain $\mathcal{X}$, and has the data player playing best response, which can be computed non-privately by solving an integer program. Since the query player's MW distribution is a function of the private data, DualQuery privately approximates this distribution with a collection of

Table 1: Error bound Comparison

Algorithm	α
MWEM	$O\left(\frac{d^{1/4}\log^{1/2}\lvert\mathcal{Q}\rvert\log^{1/2}(1/\delta)}{n^{1/2}\varepsilon^{1/2}}\right)$
DualQuery	$O\left(\frac{d^{1/6}\log^{1/2}\lvert\mathcal{Q}\rvert\log^{1/6}(1/\delta)}{n^{1/3}\varepsilon^{1/3}}\right)$
FEM	$O\left(\frac{d^{3/4}\log^{1/2}\lvert\mathcal{Q}\rvert\log^{1/2}(1/\delta)}{n^{1/2}\varepsilon^{1/2}}\right)$
sepFEM	$O\left(\frac{d^{5/8}\log^{1/2}\lvert\mathcal{Q}\rvert\log^{1/2}(1/\delta)}{n^{1/2}\varepsilon^{1/2}}\right)$
DQRS	$O\left(\frac{d^{1/5}\log^{3/5}\lvert\mathcal{Q}\rvert\log^{1/5}(1/\delta)}{n^{2/5}\varepsilon^{2/5}}\right)$

Parameters: (ε, δ)-differential privacy, n data points of dimension d, query class $\mathcal{Q}$, accuracy α.

samples drawn from it. Each draw from the MW distribution can be viewed as a single instantiation of the exponential mechanism, which provides a bound on the privacy loss. We improve DualQuery by leveraging the observation that the MW distribution changes slowly between rounds in the no-regret dynamics. Thus can reuse previously drawn queries to approximate the current MW distribution via rejection sampling. By using this technique, our algorithm DQRS (DualQuery with rejection sampling) reduces the number of times we draw new samples from the MW distribution and also the privacy loss, and hence improves the privacy-utility trade-off. We theoretically demonstrate that DQRS improves the accuracy guarantee of DualQuery. Specifically DQRS attains accuracy

$$\alpha = \widetilde{O}\left(\frac{\log(|\mathcal{X}|/\beta)\cdot\log^3(|\mathcal{Q}|)}{n^2}\right)^{1/5}$$

whereas DualQuery attains accuracy $\alpha = \widetilde{O}\left(\frac{\log(|\mathcal{X}|/\beta)\cdot\log^3(|\mathcal{Q}|)}{n^2}\right)^{1/6}$. Even though the dual algorithms DualQuery and DQRS have worse accuracy performance than the primal algorithms FEM and sepFEM, the dual algorithms run substantially faster, since they make many fewer oracle calls. Thus we observe a tradeoff not only between privacy and utility but also with computational resources.

In addition to our theoretical guarantees, we perform a comprehensive experimental evaluation of our algorithms. As a benchmark, we use the state-of-the-art High-Dimensional Matrix Mechanism (HDMM) (McKenna et al., 2018); HDMM is being deployed in practice by the US Census Bureau (Kifer, 2019). We perform our experiments with the standard ADULT and LOANS datasets and use k-way conjunctions as a query workload. We compare both algorithms on different workload sizes and different privacy levels. Our experiments show that as we increase the workload size FEM performs better compared to HDMM. Similarly, FEM does better when we increase the privacy level. These results support our theoretical analysis.

1.1 Additional related work

Aside from the aforementioned DualQuery algorithm (Gaboardi et al., 2014), several works on differentially private query release and synthetic data generation are described in, or can be placed in, the framework of oracle-efficient algorithms. One example is the Projection Mechanism (Nikolov et al., 2013) and extensions thereof (Nikolov, 2015; Dwork et al., 2015; Błasiok et al., 2019) in which each projection step can be approximately implemented via a non-private optimization subroutine. This line of work focuses on the average error over the queries, rather than the maximum error as we do.

The notion of oracle-efficiency for differential privacy was formalized in a recent work of Neel et al. (2019) who introduced techniques for oracle-efficient private synthetic data generation even for exponentially large classes of queries. A more recent work by Neel et al. (2020) provides oracle-efficient methods for privately solving certain classes of non-convex optimization problems. In both Neel et al. (2019) and Neel et al. (2020), the privacy guarantees of their algorithms either rely on the exact optimality or certifiability of the oracle. All of our algorithms satisfy differential privacy even if we implement the optimization oracles with a heuristic that satisfies neither condition.

In Section 6, we compare the performance of our algorithms against other practical algorithms for synthetic data generation. The benchmark we use is the High-Dimensional Matrix Mechanism (McKenna et al., 2018) which itself builds on the Matrix Mechanism (Li et al., 2015) but is more efficient and scalable. Given a workload of queries $\mathcal{Q}$, this algorithm uses optimization routines (in a significantly different way than ours) to select a different set of "strategy queries" which can be answered with Laplace noise. Answers to the original queries in $\mathcal{Q}$ can then be reconstructed by combining the noisy answers to these strategy queries.

The study of oracle-efficiency also has a rich history in machine learning and optimization outside of differential privacy (Beygelzimer et al., 2005; Balcan et al., 2008; Beygelzimer et al., 2016; Ben-Tal et al., 2015; Hazan & Koren, 2016). In particular, a number of works have sought to design oracle-efficient fair algorithms (Agarwal et al., 2018; Alabi et al., 2018; Kearns et al., 2018).

2 Preliminaries

Definition 2.1 (Differential Privacy (DP)). A randomized algorithm $\mathcal{M} : \mathcal{X}^* \to \mathcal{R}$ satisfies (ε, δ)-differential privacy (DP) if for all databases x, x' differing in at most one entry, and every measurable subset $S \subseteq \mathcal{R}$, we have

$$\Pr[\mathcal{M}(x) \in S] \leq e^{\varepsilon} \Pr[\mathcal{M}(x') \in S] + \delta.$$

If $\delta = 0$, we say that $\mathcal{M}$ satisifies ε-diffrential privacy.

To facilitate our privacy analysis, we will rely on the privacy notion of *zero-concentrated differential privacy* (zCDP), which provides a simpler composition theorem.

Definition 2.2 (Zero Concentrated Differential Privacy(zCDP) Bun & Steinke (2016)). A mechanism $\mathcal{M} : \mathcal{X} \to R$ is (ρ)-zero-concentrated differentially private if for all neighboring datasets $x, x' \in \mathcal{X}^*$, and all $\alpha \in (0, \infty)$ the following holds

$$\mathbb{D}_{\alpha} \left(M(x) || M(x') \right) \leq \rho\alpha$$

where $\mathbb{D}_{\alpha}$ is the α-Rényi divergence between the distribution $M(x)$ and the distribution $M(x')$.

We can relate guarantees of DP and zCDP using the following lemmas.

Lemma 1 (DP to zCDP Bun & Steinke (2016)). *If $\mathcal{M}$ satisfies ε-differential privacy, then $\mathcal{M}$ satisfies $\left(\frac{1}{2}\varepsilon^2\right)$-zCDP.*

Lemma 2 (zCDP to DP Bun & Steinke (2016)). *If $\mathcal{M}$ provides ρ-zCDP, then $\mathcal{M}$ is $\left(\rho + 2\sqrt{\rho \log(1/\delta)}, \delta\right)$-DP for $\delta > 0$.*

Lemma 3 (zCDP composition Bun & Steinke (2016)). *Let $\mathcal{M} : \mathcal{X}^* \to \mathcal{Y}$ and $M' : \mathcal{X}^* \to \mathcal{Z}$ be randomized algorithm. Suppose that $\mathcal{M}$ satisfies ρ-zCDP and $\mathcal{M}'$ satisfies ρ'-zCDP. Define $\mathcal{M}'' : \mathcal{X} \to \mathcal{Y} \times \mathcal{Z}$ by $\mathcal{M}''(x) = (\mathcal{M}(x), \mathcal{M}'(x))$. Then $\mathcal{M}''$ satisfies $(\rho + \rho')$-zCDP.*

We will use the exponential mechanism as a key component in our design of private algorithms.

Definition 2.3 (Exponential Mechanism McSherry & Talwar (2007)). Given some database x, arbitrary range $\mathcal{R}$, and score function $S : \mathcal{X}^* \times \mathcal{R} \to \mathcal{R}$, the exponential mechanism $\mathcal{M}_E(x, S, \mathcal{R}, \rho)$ selects and outputs an element $r \in \mathcal{R}$ with probability proportional to

$$\exp\left(\frac{\rho S(x, r)}{2\Delta_S}\right),$$

where Δ_S is the sensitivity of S, defined as

$$\Delta_S = \max_{D,D':|D\triangle D'|=1, r\in R} |S(D,r) - S(D',r)|.$$

Lemma 4 (McSherry & Talwar (2007)). *The exponential mechanism $\mathcal{M}_E(x, S, \mathcal{R}, \rho)$ is $\left(\frac{\rho^2}{2}\right)$-zCDP.*

Theorem 5 (Exponential Mechanism Utility McSherry & Talwar (2007)). *. Fixing a database x, let $OPT_S(x)$ denote the max score of function S. Then, with probability $1-\beta$ the error is bounded by:*

$$OPT_S(x) - S(x, \mathcal{M}_E(x, u, \mathcal{R}, \rho)) \leq \frac{2\Delta_S}{\rho} \left(\ln |\mathcal{R}|/\beta\right)$$

We are interested in privately releasing *statistical linear queries*, formally defined as follows.

Definition 2.4 (Statistical linear queries). Given as predicate a linear threshold function ϕ, the linear query $q_\phi : \mathcal{X}^n \to [0,1]$ is defined by

$$q_\phi(D) = \frac{\sum_{x\in D} \phi(x)}{|D|}$$

The main query class we consider in our empirical evaluations is 3-way marginals and 5-way marginals. We give the definition here

Definition 2.5. Let the data universe with d categorical features be $\mathcal{X} = (\mathcal{X}_1 \times \ldots \times \mathcal{X}_d)$, where each $\mathcal{X}_i$ is the discrete domain of the ith feature. We write $x_i \in \mathcal{X}_i$ to mean the ith feature of record $x \in \mathcal{X}$. A 3-way marginal query is a linear query specified by 3 features $a \neq b \neq c \in [d]$, and a target $y \in (\mathcal{X}_a \times \mathcal{X}_b \times \mathcal{X}_c)$, given by

$$q_{\mathrm{abc,y}}(x) = \begin{cases} 1 & : x_a = y_1 \wedge x_b = y_2 \wedge x_c = y_3 \\ 0 & : \text{otherwise.} \end{cases}$$

Furthermore, its negation is given by

$$\bar{q}_{\mathrm{abc,y}}(x) = \begin{cases} 0 & : x_a = y_1 \wedge x_b = y_2 \wedge x_c = y_3 \\ 1 & : \text{otherwise.} \end{cases}$$

Note that for each marginal (a, b, c) there are $|\mathcal{X}_a||\mathcal{X}_b||\mathcal{X}_c|$ queries.

Finally, our algorithm will be using the following form of linear optimization oracle. In our experiments, we implement this oracle via an integer program solver.

Definition 2.6 (Linear Optimization Oracle). Given as input a set of n statistical linear queries $\{q_i\}$ and a d-dimensional vector σ, a linear optimization oracle outputs

$$\hat{x} \in \arg\min_{x\in\{0,1\}^d} \left\{ \sum_{i=1}^{n} q_i(x) - \langle x, \sigma \rangle \right\}$$

3 Query Release Game

Given a class of queries $\mathcal{Q}$ over a database D, we want to output a differentially private synthetic dataset $\widehat{D}$ such that for any query $q \in \mathcal{Q}$ we have low error:

$$\mathrm{error}(\widehat{D}) = \max_{q\in\mathcal{Q}} |q(D) - q(\widehat{D})| \leq \alpha.$$

We revisit a zero-sum game formulation between a data-player and a query player for this problem Hsu et al. (2013); Gaboardi et al. (2014). The data player has action set equal to the data universe $\mathcal{X}$ and the query player has action set equal to the query class $\mathcal{Q}$. We make the assumption that $\mathcal{Q}$ is closed under negation. That is, for every query $q \in \mathcal{Q}$ there is a *negated query* $\bar{q} \in \mathcal{Q}$ where $\bar{q}(D) = 1 - q(D)$. If $\mathcal{Q}$ is not closed under negation, we can simply add negated queries to $\mathcal{Q}$. Since $\mathcal{Q}$ is closed under negations, we can write the error as

$$|q(D) - q(\widehat{D})| = \max\{q(D) - q(\widehat{D}), \neg q(D) - \neg q(\widehat{D})\}$$

This allows us to define a payoff function that captures the error of $\widehat{D}$ without the absolute value. In particular, the payoff for actions $x \in \mathcal{X}$ and $q \in \mathcal{Q}$ is given by:

$$A(x, q) := q(D) - q(x) \tag{1}$$

The data player wants minimizes the payoff $A(x, q)$ while the query player maximizes it. Intuitively, the data player would like to find a distribution with low error, while the query player is trying to identify the query with the worst error. Each player chooses a mixed strategy, that is a distribution over their action set. Let $\Delta(\mathcal{X})$ and $\Delta(\mathcal{Q})$ denote the sets of distributions ove $\mathcal{X}$ and $\mathcal{Q}$. For any $\widehat{D} \in \Delta(\mathcal{X})$ and $\widehat{Q} \in \Delta(\mathcal{Q})$, the payoff is defined as

$$A(\widehat{D}, \cdot) = \mathbb{E}_{x \sim \widehat{D}}\left[A(x, \cdot)\right], \quad A(\cdot, \widehat{Q}) = \mathbb{E}_{q \sim \widehat{Q}}\left[A(\cdot, q)\right].$$

A pair of mixed strategies $(\widehat{D}, \widehat{Q}) \in \Delta(\mathcal{X}) \times \Delta(\mathcal{Q})$ forms an α-approximate equilibrium of the game if

$$\max_{q \in \mathcal{Q}} A(\widehat{D}, q) - \alpha \leq A(\widehat{D}, \widehat{Q}) \leq \min_{x \in \mathcal{X}} A(x, \widehat{Q}) + \alpha, \tag{2}$$

The following result allows us to reduce the problem of query release to the problem of computing an equilibrium in the game.

Theorem 6 (Gaboardi et al. (2014))**.** *Let $(\widehat{D}, \widehat{Q})$ be any α-approximate equilibrium of the query release game, then the data player's strategy $\widehat{D}$ is 2α-accurate, error$(\widehat{D}) = \max_{q \in \mathcal{Q}} |q(D) - q(\widehat{D})| \leq 2\alpha$.*

3.1 No-Regret Dynamics

To compute such an equilibrium privately, we will simulate no-regret dynamics between the two players. Over rounds $t = 1, \ldots, T$, the two players will generate a sequence of plays $(D^1, Q^1), \ldots, (D^T, Q^T) \in \Delta(\mathcal{X}) \times \Delta(\mathcal{Q})$. The regrets of the two players are defined as

$$R_{\text{data}}(T) = \frac{1}{T}\left(\sum_{t=1}^{T} A(D^t, Q^t) - \min_{x \in \mathcal{X}} \sum_{t=1}^{T} A(x, Q^t)\right)$$
$$R_{\text{qry}}(T) = \frac{1}{T}\left(\max_{q \in \mathcal{Q}} \sum_{t=1}^{T} A(D^t, q) - \sum_{t=1}^{T} A(D^t, Q^t)\right)$$

Theorem 7 (Follows from Freund & Schapire (1997))**.** *The average play $(\overline{D}, \overline{Q})$ given by $\overline{D} = \frac{1}{T}\sum_{t=1}^{T} D^t$ and $\overline{Q} = \frac{1}{T}\sum_{t=1}^{T} Q^t$ from the no-regret dynamics above is an α-approximate equilibrium with*

$$\alpha = R_{\text{data}}(T) + R_{\text{qry}}(T).$$

We will now provide two frameworks to obtain regret bounds for the two players.

4 Primal Oracle-Efficient Framework

In the primal framework, we will have the data player run a online learning algorithm to update the distributions $D^1, \ldots, D^T$ over rounds and have the query player play an approximate best response Q^t against D^t in each round. The algorithm MWEM falls under this framework, but the no-regret algorithm (MW) runs in exponential time even in the best case since it maintains a distribution over the entire domain $\mathcal{X}$.

We replace the MW method with two variants of the follow-the-perturbed-leader (FTPL) algorithm Kalai & Vempala (2005)—Non-Convex-FTPL Suggala & Netrapalli (2019) and Separator-FTPL Syrgkanis et al. (2016). Both of these algorithms can generate a sample from their FTPL distributions by relying an oracle to solve a perturbed optimization problem. (In our experiments, we instantiate this oracle with an integer program solver.) For both algorithms, the query player selects a query $q_t \in \mathcal{Q}$ (that is Q_t is point mass distribution on q_t) using the exponential mechanism, denoted by $\mathcal{M}_E$. We present this primal framework in Algorithm 1.

Algorithm 1: Primal Framework of No-Regret Dynamics

Require: FTPL algorithm $\mathcal{A}$
input A dataset $D \in \mathcal{X}^n$, query class $\mathcal{Q}$, number of rounds T, target privacy ρ.
  Initialize $\rho_0 = \rho/T$. Get initial sample $q_0 \in \mathcal{Q}$ uniformly at random.
  for $t = 1$ **to** T **do**
    Data Player Generate $\widehat{D^t}$ with online learner $\mathcal{A}$ with queries $q_0, \ldots, q_{t-1}$.
    Query player: Define score function S_t. For each query $q \in \mathcal{Q}$, set $S_t(D, q) = q(D) - q(\widehat{D^t})$.
    Sample $q_t \sim \mathcal{M}_E(D, S_t, \mathcal{Q}, \sqrt{2\rho_0})$ {such that EM satisfies ρ_0-zCDP}
  end for
output $\frac{1}{T}\sum_{t=1}^{T} \widehat{D^t}$

Now we instantiate the primal framework above with two no-regret learners, which yield two algorithms FEM ((Non-Convex)-FTPL with Exponential Mechanism) and sepFEM (Separator-FTPL with Exponential Mechanism). First, the FEM algorithm at each round t computes a distribution D_t by solving a perturbed linear optimization problem polynomially many times. The optimization objective is given by the payoff against the previous queries and a linear perturbation

$$\arg\min_{x \in \mathcal{X}} \sum_{i=0}^{t-1} A(x, q_i) + \langle x, \sigma \rangle$$

where σ is a random vector drawn from the exponential distribution. Observe that the first term $q_i(D)$ in $A(x, q_i) = q_i(D) - q_i(x)$ does not depend on x. Thus, we can further simplify the objective as

$$\arg\max_{x \in \mathcal{X}} \left\{ \sum_{i=0}^{t-1} q_i(x) - \langle x, \sigma \rangle \right\}$$

To solve this problem above, we will use an linear optimization oracle (definition 2.6), which we will implement using an integer program solver.

The second algorithm is less general, but as we will show it achieves a better error rate for important classes of queries. Algorithm sepFEM relies on the assumption that the query class $\mathcal{Q}$ has a small separator set $\mathrm{sep}(\mathcal{Q})$.

Definition 4.1 (Separator Set)**.** A queries class $\mathcal{Q}$ has a small separator set $\mathrm{sep}(\mathcal{Q})$ if for any two distinct records $x, x' \in \mathcal{X}$, there exist a query $q : \mathcal{X} \to \{0, 1\}$ in $\mathrm{sep}(\mathcal{Q})$ such that $q(x) \neq q(x')$.

Many classes of statistical queries defined over the boolean hypercube have separator sets of size proportional to their VC-dimension or the dimension of the input data. For example, boolean conjunctions, disjunctions, halfspaces defined over the $\{0, 1\}^d$, and parity functions all have separator sets of size d.

Algorithm sepFEM then perturbs the data player's optimization problem by inserting "fake" queries from the separator set:

$$\arg\max_{x \in \mathcal{X}} \left\{ \sum_{i=1}^{t-1} q_i(x) + \sum_{\tilde{q}_j \in \mathrm{sep}(\mathcal{Q})} \sigma_j \tilde{q}_j(x) \right\},$$

Algorithm 2: Data player update in FEM

input Queries $q_0, \ldots, q_{t-1} \in \mathcal{Q}$, exponential distribution scale η, number of samples s.

for $j \leftarrow 1$ **to** s **do**

Let $\sigma_j \in \mathbb{R}^d$ be a random vector such that each coordinate of σ_j is drawn from the exponential distribution $\mathrm{Exp}(\eta)$. Obtain a FTPL sample x_j^t by solving

$$x_j^t \in \arg\max_{x \in \mathcal{X}} \left\{ \sum_{i=0}^{t-1} q_i(x) - \langle x, \sigma_j \rangle \right\}$$

end for

output $\widehat{D_t}$ as the uniform distribution over $\{x_1^t, \ldots, x_s^t\}$

where each $\sigma_j \in \mathbb{R}$ is sampled from the Laplace distribution. This problem can be viewed as a simple special case of the linear optimization problem in definition 2.6 with no linear perturbation term.

Algorithm 3: Data player update in sepFEM

input Queries $q_0, \ldots, q_{t-1} \in \mathcal{Q}$, Laplace noise scale η, number of samples s.

Let $\mathrm{sep}(\mathcal{Q}) = \{\tilde{q}_1, \ldots, \tilde{q}_M\}$ be the serparator set for $\mathcal{Q}$.

for $j = 1$ **to** s **do**

Let $\sigma_j \in \mathbb{R}^M$ be a fresh random vector such that each coordinate of σ_j is drawn from the Laplace distribution $\mathrm{Lap}(\eta)$. Obtain a FTPL sample x_j^t by solving

$$x_j^t \in \arg\max_{x \in \mathcal{X}} \left\{ \sum_{i=0}^{t-1} q_i(x) + \sum_{i=1}^{M} \sigma_{j,i} \tilde{q}_i(x) \right\}$$

end for

output $\widehat{D_t}$ be a uniform distribution over $\{x_1^t, \ldots, x_s^t\}$

To derive the privacy guarantee of these two algorithms, we observe that the data player's update does not directly use the private dataset D. Thus, the privacy guarantee directly follows from the composition of T exponential mechanisms.

Theorem 8 (Privacy)**.** *Algorithm 1 satisfies ρ-zCDP for any instantiated with any no-regret algorithm then it*

Proof. The algorithm 1 executes $T = \rho/\rho_0$ runs of of the exponential mechanism $\mathcal{M}(x, S, \mathcal{R}, \sqrt{2\rho_0})$ with parameter $\sqrt{2\rho_0}$. Then by Lemma 4, we have that $\mathcal{M}(x, S, \mathcal{R}, \sqrt{2\rho_0})$ satisfies ρ_0-zCDP. Finally Lemma 3 states that the composition of $T = \frac{\rho}{\rho_0}$ ρ_0-mechanims satisfies ρ-zCDP. ☐

To derive the accuracy guarantee of the two algorithms, we first bound the regret of the two players. Note that the regret guarantee of the data player follow from the regret bounds on the two FTPL algorithms Suggala & Netrapalli (2019) and Syrgkanis et al. (2016). The regret guarantee of the query player directly follows from the utility guarantee of the exponential mechanism McSherry & Talwar (2007). We defer the details to the appendix.

Corollary 8.1 (FEM Accuracy)**.** *Let $d = \log(\mathcal{X})$. For any dataset $D \in \mathcal{X}^n$, query class $\mathcal{Q}$ and privacy parameter $\rho > 0$, there exists T, η and s so that with probability at least $1 - \beta$, the algorithm* FEM *finds a* *synthetic database $\widehat{D}$ that answers all queries in $\mathcal{Q}$ with error*

$$\max_{q\in\mathcal{Q}} |q(D) - q(\widehat{D})| \leq \widetilde{O}\left(\frac{d^{3/4}\sqrt{\log\left(\frac{|\mathcal{Q}|}{\beta}\right)}}{\rho^{1/4}n^{1/2}}\right)$$

By Lemma 2, algorithm 2 satisfies (ε,δ)-differential privacy with $\varepsilon = \rho + 2\sqrt{\rho\log(1/\delta)}$. If $\varepsilon < 1$ then FEM *has error*

$$\max_{q\in\mathcal{Q}} |q(D) - q(\widehat{D})| \leq \widetilde{O}\left(\frac{d^{3/4}\log^{1/2}|\mathcal{Q}| \cdot \sqrt{\log(\frac{1}{\delta})\log(\frac{1}{\beta})}}{\varepsilon^{1/2}n^{1/2}}\right)$$

Corollary 8.2 (sepFEM Accuracy)**.** *Let $d = \log(\mathcal{X})$. For any dataset $D \in \mathcal{X}^n$ and query class $\mathcal{Q}$ with a separator set $\operatorname{sep}(\mathcal{Q})$ and privacy parameter $\rho > 0$, there exist T, η and s so that with probability at least $1-\beta$, algorithm* sepFEM *finds a synthetic database $\widehat{D}$ that answers all queries in $\mathcal{Q}$ with error*

$$\max_{q\in\mathcal{Q}} |q(D) - q(\widehat{D})| \leq \widetilde{O}\left(\frac{|\operatorname{sep}(\mathcal{Q})|^{3/8}d^{1/4}\sqrt{\log\left(\frac{|\mathcal{Q}|}{\beta}\right)}}{\rho^{1/4}n^{1/2}}\right)$$

By Lemma 2, algorithm 1 satisfies (ε,δ)-differential privacy with $\varepsilon = \rho + 2\sqrt{\rho\ln(1/\delta)}$. If $\varepsilon < 1$ then sepFEM *has error*

$$\max_{q\in\mathcal{Q}} |q(D) - q(\widehat{D})| \leq \widetilde{O}\left(\frac{|\operatorname{sep}(\mathcal{Q})|^{3/8}d^{1/4}\sqrt{\log\left(\frac{|\mathcal{Q}|}{\beta}\right)\log\left(\frac{1}{\delta}\right)}}{\varepsilon^{1/2}n^{1/2}}\right)$$

Note that if the query class $\mathcal{Q}$ has a separator set of size $O(d)$, which is the case for boolean conjunctions, disjunctions, halfspaces defined over the $\{0,1\}^d$, and parity functions, then the bound above becomes

$$\max_{q\in\mathcal{Q}} |q(D) - q(\widehat{D})| \leq \widetilde{O}\left(\frac{d^{5/8}\log^{1/2}|\mathcal{Q}| \cdot \log^{1/2}(1/\delta)\log^{1/2}(1/\beta)}{\varepsilon^{1/2}n^{1/2}}\right)$$

Remark. *Non-convex FEM and Separator FEM exhibit a better tradeoff between α and n than DualQuery, but a slightly worse dependence on d compared to DualQuery and MWEM.*

5 DQRS: DualQuery with Rejection Sampling

In this section, we present an algorithm DQRS that builds on the DualQuery algorithm Gaboardi et al. (2014) and achieves better provable sample complexity. In DualQuery, we employ the dual framework of the query release game – the query player maintains a distribution over queries using the Multiplicative Weights (MW) no-regret learning algorithm and the data player best responds. However, the query player cannot directly use the distribution $\mathcal{Q}^t$ proposed by MW during round t because it depends on the private data. Instead, for each round t, it takes s samples from $\mathcal{Q}^t$ to form an estimate distribution $\widehat{\mathcal{Q}^t}$. The data player then best-responds against $\widehat{\mathcal{Q}^t}$. Sampling from the MW distribution $\mathcal{Q}^t$ can be interpreted as a sample from the exponential mechanism. The sampling step incurs a significant privacy cost.

Our algorithm DQRS improves the sampling step of DualQuery in order to reduce the privacy cost (and the runtime). The basic idea of our algorithm DQRS is to apply the rejection sampling technique to "recycle" samples from prior rounds. Namely, we generate some samples from $\mathcal{Q}^t$ using the samples obtained from the distribution in the previous round, i.e., $\mathcal{Q}^{t-1}$. This is possible because $\mathcal{Q}^t$ is close to $\mathcal{Q}^{t-1}$. We show that by taking fewer samples from $\mathcal{Q}^t$ for each round t, we consume less of the privacy budget. The result is that the algorithm operates for more iterations and obtains lower regret (i.e., better accuracy).

Theorem 9. DualQuery *with rejection sampling (Algorithm 4) takes in a private dataset $D \in \mathcal{X}^n$ and makes $T = O\left(\frac{\log|\mathcal{Q}|}{\alpha^2}\right)$ queries to an optimization oracle and outputs a dataset $\tilde{D} = (x^1, \cdots, x^T) \in \mathcal{X}^T$ such that, with probability at least $1-\beta$, for all $q \in \mathcal{Q}$ we have $|q(\tilde{D}) - q(D)| \leq \alpha$. The algorithm is (ε, δ)-differentially private and attains accuracy*

$$\alpha = O\left(\frac{\log(|\mathcal{X}|T/\beta) \cdot \log^3(|\mathcal{Q}|) \cdot \log(1/\delta)}{n^2\varepsilon^2}\right)^{1/5}.$$

In contrast, DualQuery (*without* rejection sampling) obtains the same result except with

$$\alpha = O\left(\frac{\log(|\mathcal{X}|T/\beta) \cdot \log^3(|\mathcal{Q}|) \cdot \log(1/\delta)}{n^2\varepsilon^2}\right)^{1/6}.$$

In other words, DQRS attains strictly better accuracy than DualQuery for the same setting of other parameters.

Algorithm 4: Rejection Sampling Dualquery

Require: Target accuracy $\alpha \in (0,1)$, target failure probability $\beta \in (0,1)$
input dataset D, and linear queries $q_1, \ldots, q_k \in \mathcal{Q}$
  Set $T = \frac{16 \log|\mathcal{Q}|}{\alpha^2}$, $\eta = \frac{\alpha}{4}$
  $s = \frac{48 \log(3|\mathcal{X}|T/\beta)}{\alpha^2}$
  Construct sample S_1 of s queries $\{q_i\}$ from $\mathcal{Q}$ according to $\mathcal{Q}^1 = \mathsf{Uniform}(\mathcal{Q})$
  for $t \leftarrow 1$ **to** T **do**
    Let $\tilde{q} = \frac{1}{s}\sum_{q \in S_t} q$;
    Find x^t with $A_D(x^t, \tilde{q}) \geq \max_x A_D(x, \tilde{q}) - \alpha/4$;
    Let $\gamma_t = \frac{1}{2t^{2/3}}$
    for all $q \in \mathcal{Q}$ **do**
      $\hat{\mathcal{Q}}_q^{t+1} := e^{-\eta-\gamma_t} \cdot \exp\left(-\eta A_D(x^t, q)\right) \mathcal{Q}_q^t$;
    end for
    Normalize $\hat{\mathcal{Q}}^{t+1}$ to obtain $\mathcal{Q}^{t+1}$
    Construct S_{t+1} as follows
    Let $\tilde{s}_t = (2\gamma_t + 4\eta)s$ and add $\tilde{s}_t$ independent fresh samples from Q^{t+1} to S_{t+1}
    for all $q \in S_t$ **do**
      Add q to S_{t+1} with probability $\hat{\mathcal{Q}}_q^{\,t+1} / \mathcal{Q}_q^t$
      If $|S_{t+1}| > s$, discard elements at random so that $|S_{t+1}| = s$
    end for
  end for
output Sample $y_1, \ldots, y_s$

The analysis of DQRS largely follows that of DualQuery. The key difference is the analysis of the rejection sampling step, which is summarized by the following two lemmas. The first one shows that taking samples drawn from $Q = \mathcal{Q}^t$ and performing rejection sampling yields samples from $P = \mathcal{Q}^{t+1}$; thus S_{t+1} is distributed exactly as if it were drawn from $\mathcal{Q}^{t+1}$. The second lemma gives a bound on the privacy loss of the rejection sampling step.

Lemma 10 (Rejection Sampling Accuracy). *Let P and Q be probability distributions over $\mathcal{Q}$, and let $M \geq \max_{q \in \mathcal{Q}} P_q/Q_q$. Sample an element of $\mathcal{Q}$ as follows. Sample q according to Q, and accept it with probability $P_q/(M \cdot Q_q)$. If q is not accepted, sample q according to P. Then the resulting element is distributed according to P.*

Lemma 11 (Rejection Sampling Privacy). *The subroutine which accepts q with probability $\hat{Q}_q^{t+1}/Q_q^t = e^{-\eta-\gamma_t} \cdot \exp(-\eta A_D(x^t, q))$ is ε-differentially private for $\varepsilon = \max\{\eta/n, \eta/\gamma_t n\}$.*

6 Experiments on the Adult dataset

We evaluate the algorithms presented in this paper on two different datasets: the ADULT dataset from the UCI repository Dua & Graff (2017) and the LOANS dataset. The datasets used in our experiments are summarized in table 2. For the experiments in this section, we focus on answering 3-way marginal and 5-way marginal queries. We ran two sets of experiments. One looks into how well the algorithms scale with the privacy budget, and we test for privacy budget ε taking value in $0.1, 0.15, 0.2, 0.25, 0.5$, and 1. The second one looks into how the algorithms' performance degrades when we rapidly increase the number of marginals workload to answer. To measure the accuracy of a synthetic dataset $\widehat{D}$ produced by the algorithm, we used the max additive error over a set of queries Q: $\text{error}(\widehat{D}) = \max_{q \in Q} |q(D) - q(\widehat{D})|$.

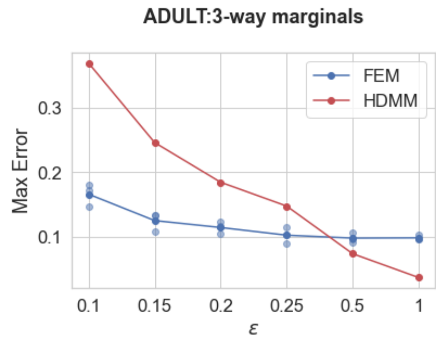
Table 2: Datasets

Data set	Records	Attributes
ADULT	48842	15
LOANS	42535	48

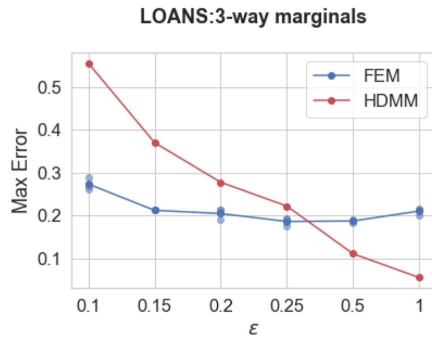
Our first set of experiments (fig. 1) fix the number of queries and evaluate the performance on different privacy levels. From the first result, we observe that FEM's max error rate increases more slowly than HDMM's as we increase the privacy level (decrease ε value). Our second set of experiments (fig. 2) fix the privacy parameters and evaluates performance on increasing workload size (or the number of marginals). The results from this section, show that FEM's max error rate increases much more slowly than HDMM's. From the experiments, we can conclude that at least of the case of k-way marginals and dataset ADULT and LOANS, FEM scales better to both the high privacy regime (low ε value) and the large workload regime (high number of queries) than the state-of-the-art HDMM method.

Hyper-Parameter Selection In our implementation, algorithm FEM has hyperparameters ε_0 and η. Both the accuracy and the run time of the algorithm depend on how we choose these hyperparameters. For FEM , we ran grid-search on different hyperparameter combinations and reported the one with the smallest error. The table 3 summarizes the range of hyperparameters used for the first set of experiments in fig. 1. Then table 4 summarizes the range of hyperparameters used for the second set of experiments in fig. 2.

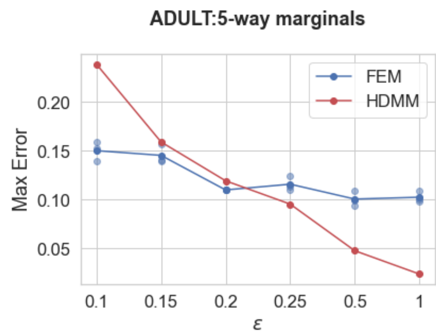
However, in real-life scenarios, we may not have access to an optimization procedure to select the best set of hyperparameters since every time we run the algorithm, we are consuming our privacy budget. Therefore, selecting the right combination of hyperparameters can be challenging. We briefly discuss how each parameter affects FEM's performance. The η parameter is the scale of the random objective perturbation term. The data player samples a synthetic dataset $\widehat{D}$ from the Follow The Perturbed Leader distribution with parameter η as in algorithm 2. The perturbation scale η controls the rate of convergence of the algorithm. Setting this value too low can make the algorithm unstable and leads to bad performance. If set too high, the solver in FTPL focuses too much on optimizing over the noise term.



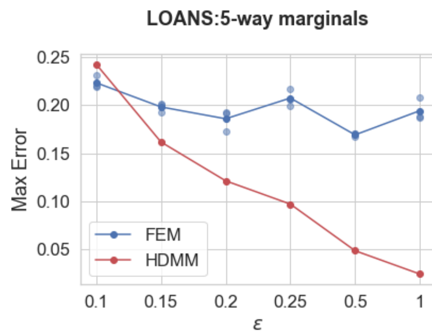
(a) ADULT dataset on 3-way marginal queries.



(b) LOANS dataset on 3-way marginal queries.



(c) ADULT dataset on 5-way marginal queries.



(d) LOANS dataset on 5-way marginal queries.

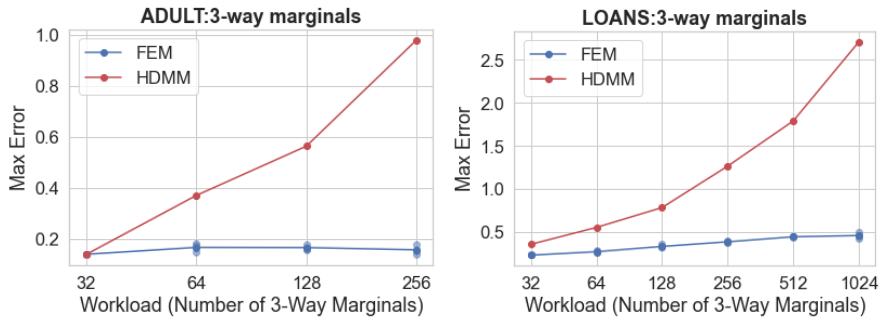
Figure 1: Max-error for 3 and 5-way marginal queries on different privacy levels. The number of marginals is fixed at 64. We enumerate all queries for each marginal.(see definition 2.5)

The parameter ε_0 corresponds to the privacy consumed on each round by the exponential mechanism parameterized with ε_0. The goal is to find a query that maximizes the error on $\widehat{D}$. Thus, the parameter ε_0 controls the number of iterations. Again we face a trade-off in choosing ε_0, since setting this value too high can lead to too few iterations giving the algorithm no chance to converge to a good solution. If ε_0 is too low, it can make the algorithm run too slow, and also it makes it hard for the query player's exponential mechanism to find queries with large errors.

Table 3: First FEM hyperparameters for fig. 1.

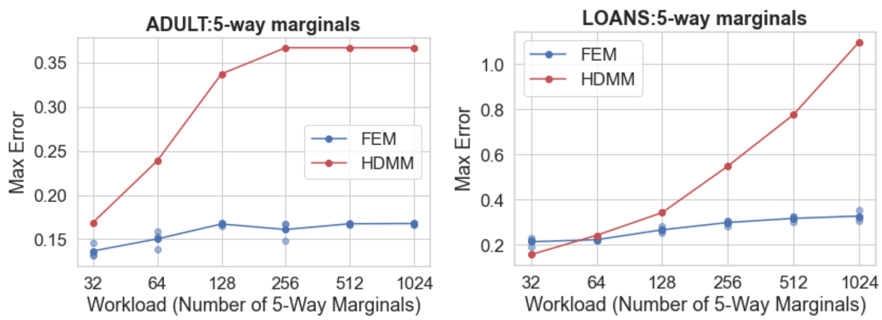
Param	Description	Range
ε_0	Privacy budget used per round	0.003, 0.005, 0.007, 0.009, 0.011, 0.015, 0.017, 0.019
η	Scale of noise for objective perturbation	1, 2, 3, 4

Data discretization We discretize ADULT and LOANS datasets into binary attributes by mapping each possible value of a discrete attribute to a new binary feature. We bucket continuous attributes, mapping each bucket to a new binary feature.



(a) ADULT dataset on 3-way marginal queries.

(b) LOANS dataset on 3-way marginal queries.



(c) ADULT dataset on 5-way marginal queries.

(d) LOANS dataset on 5-way marginal queries.

Figure 2: Max-error for increasing number of 3 and 5-way marginals. We enumerate all queries for each marginal (see definition 2.5). The privacy parameter ε is fixed at 0.1 and δ is $\frac{1}{n^2}$, where n is the size of the dataset. .

Optimizing over k-way Marginals We represent a data record by its one-hot binary encoding with dimension d, thus $\mathcal{X} = \{0,1\}^d$ is the data domain. On each round t the algorithm FEM takes as input a sequence of t queries $\left(q^{(1)}, \ldots, q^{(t)}\right)$ and a random perturbation term $\sigma \sim \mathrm{Lap}(\eta)^d$ and solves the following optimization problem

$$\arg\max_{x \in \{0,1\}^d} \left\{ \sum_{i=1}^{t-1} q^{(i)}(x) - \langle x, \sigma \rangle \right\} \tag{3}$$

Let Q_k be the set of k-way marginal queries. We can represent any k-way marginal query $q \in Q_k$ for $\mathcal{X}$ in vector form with a d-dimensional binary vector $\vec{q}$ such that $\vec{q} \in \{0,1\}^d$ and $\|\vec{q}\|_1 = k$. Then we can define $q \in Q_k$ as

$$q(x) = \begin{cases} 1 & \text{if } k = \langle x, \vec{q} \rangle \\ 0 & \text{otherwise} \end{cases}$$

Let $\bar{Q}_k$ be the set of negated k-way marginals. Then for any $q \in \bar{Q}_k$

$$q(x) = \begin{cases} 0 & \text{if } k = \langle x, \vec{q} \rangle \\ 1 & \text{otherwise} \end{cases}$$

Table 4: Second FEM hyperparameters for fig. 2.

Param	Description	Range
ε_0	Privacy budget used per round	0.0025, 0.003, 0.0035
η	Scale of noise for objective perturbation	0.75, 1, 1.25

Next we formulate the optimization problem eq. (3) as an integer program. Given a sequence of t queries $\left(q^{(1)}, \ldots, q^{(t)}\right)$ and a random perturbation term $\sigma \sim \mathrm{Lap}(\eta)^d$. Let $c_i \in \{0,1\}$ be a binary variable encoding whether the query $q^{(i)}$ is satisfied.

$$
\begin{aligned}
&\max_{x \in \{0,1\}^d} \sum_{i=1}^{t} c_i - \langle x, \sigma \rangle && \\
&\text{s.t. for all } i \in \{1, \ldots . t\} && \\
&\left\langle x, \vec{q}^{(i)} \right\rangle \geq k c_i && \text{if } q^{(i)} \in Q_k \\
&\left\langle \vec{1}_d - x, \vec{q}^{(i)} \right\rangle \geq c_i && \text{if } q^{(i)} \in \bar{Q}_k
\end{aligned}
$$

Finally, we used the Gurobi solver for mixed-integer-programming to implement FEM's optimization oracle.

The implementation We ran the experiments on a machine with a 4-core Opteron processor and 192 Gb of ram. We made publicly available the see the exact implementations used for these experiments via GitHub. For HDMM's implementation see `https://github.com/ryan112358/private-pgm/blob/master/examples/hdmm.py` and for FEM's implementation see `https://github.com/giusevtr/fem`.

7 Conclusion and Future Work

In this paper, we have studied the pressing problem of efficiently generating private synthetic data. We have presented three new algorithms for this task that sidestep known worst-case hardness results by using heuristic solvers for NP-complete subroutines. All of our algorithms are equipped with formal privacy and utility guarantees and they are oracle-efficient – i.e., our algorithms are efficient as long as the heuristic solvers are efficient.

There is a very real need for practical private synthetic data generation tools and a dearth of solutions available; the scientific literature offers mostly exponential-time algorithms and negative intractability results. This work explores one avenue for solving this conundrum and we hope that there is further work both extending this line of work and exploring entirely new approaches. Our experimental evaluation demonstrates that our algorithms are promising and supports our theoretical results. However, our experiments are relatively rudimentary. In particular, we invested most time into optimizing the most promising algorithm FEM. An immediate question is whether further optimization of the other two algorithms could yield better results.

References

Abowd, J. M. The U.S. census bureau adopts differential privacy. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD 2018, London, UK, August 19-23, 2018*, pp. 2867, 2018. doi: 10.1145/3219819.3226070. URL `https://doi.org/10.1145/3219819.3226070`.

Agarwal, A., Beygelzimer, A., Dudík, M., Langford, J., and Wallach, H. M. A reductions approach to fair classification. In Dy, J. G. and Krause, A. (eds.), *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, volume 80 of *JMLR Workshop and Conference Proceedings*, pp. 60–69. JMLR.org, 2018.

Alabi, D., Immorlica, N., and Kalai, A. Unleashing linear optimizers for group-fair learning and optimization. In *Conference On Learning Theory*, pp. 2043–2066, 2018.

Balcan, M., Bansal, N., Beygelzimer, A., Coppersmith, D., Langford, J., and Sorkin, G. B. Robust reductions from ranking to classification. *Machine Learning*, 72(1-2):139–153, 2008. doi: 10.1007/s10994-008-5058-6. URL `https://doi.org/10.1007/s10994-008-5058-6`.

Ben-Tal, A., Hazan, E., Koren, T., and Mannor, S. Oracle-based robust optimization via online learning. *Operations Research*, 63(3):628–638, 2015.

Beygelzimer, A., Dani, V., Hayes, T. P., Langford, J., and Zadrozny, B. Error limiting reductions between classification tasks. In Raedt, L. D. and Wrobel, S. (eds.), *Machine Learning, Proceedings of the Twenty-Second International Conference (ICML 2005), Bonn, Germany, August 7-11, 2005*, volume 119 of *ACM International Conference Proceeding Series*, pp. 49–56. ACM, 2005. doi: 10.1145/1102351.1102358. URL `http://doi.acm.org/10.1145/1102351.1102358`.

Beygelzimer, A., III, H. D., Langford, J., and Mineiro, P. Learning reductions that really work. *Proceedings of the IEEE*, 104(1):136–147, 2016.

Błasiok, J., Bun, M., Nikolov, A., and Steinke, T. Towards instance-optimal private query release. In *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms*, pp. 2480–2497. SIAM, 2019.

Blum, A., Ligett, K., and Roth, A. A learning theory approach to non-interactive database privacy. In *Proceedings of the Fortieth Annual ACM Symposium on Theory of Computing*, STOC '08, pp. 609–618, New York, NY, USA, 2008. ACM. ISBN 978-1-60558-047-0. doi: 10.1145/1374376.1374464. URL `http://doi.acm.org/10.1145/1374376.1374464`.

Bun, M. and Steinke, T. Concentrated differential privacy: Simplifications, extensions, and lower bounds. In *Proceedings of the 14th Conference on Theory of Cryptography*, TCC '16-B, pp. 635–658, Berlin, Heidelberg, 2016. Springer.

Bun, M., Ullman, J., and Vadhan, S. P. Fingerprinting codes and the price of approximate differential privacy. *SIAM J. Comput.*, 47(5):1888–1938, 2018. doi: 10.1137/15M1033587. URL `https://doi.org/10.1137/15M1033587`.

Dua, D. and Graff, C. UCI machine learning repository, 2017. URL `http://archive.ics.uci.edu/ml`.

Dwork, C., McSherry, F., Nissim, K., and Smith, A. Calibrating noise to sensitivity in private data analysis. In *Proceedings of the 3rd Conference on Theory of Cryptography*, TCC '06, pp. 265–284, Berlin, Heidelberg, 2006. Springer.

Dwork, C., Nikolov, A., and Talwar, K. Efficient algorithms for privately releasing marginals via convex relaxations. *Discrete & Computational Geometry*, 53(3):650–673, 2015.

Freund, Y. and Schapire, R. E. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 55(1):119 – 139, 1997. ISSN 0022-0000. doi: https://doi.org/10.1006/jcss.1997.1504. URL `http://www.sciencedirect.com/science/article/pii/S002200009791504X`.

Gaboardi, M., Arias, E. J. G., Hsu, J., Roth, A., and Wu, Z. S. Dual query: Practical private query release for high dimensional data. In *International Conference on Machine Learning*, pp. 1170–1178, 2014.

Hardt, M. and Rothblum, G. N. A multiplicative weights mechanism for privacy-preserving data analysis. In *2010 IEEE 51st Annual Symposium on Foundations of Computer Science*, pp. 61–70. IEEE, 2010.

Hardt, M., Ligett, K., and McSherry, F. A simple and practical algorithm for differentially private data release. In *Advances in Neural Information Processing Systems*, pp. 2339–2347, 2012.

Hazan, E. and Koren, T. The computational power of optimization in online learning. In *Proceedings of the 48th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2016, Cambridge, MA, USA, June 18-21, 2016*, pp. 128–141, 2016. doi: 10.1145/2897518.2897536. URL `http://doi.acm.org/10.1145/2897518.2897536`.

Hsu, J., Roth, A., and Ullman, J. Differential privacy for the analyst via private equilibrium computation. In *Proceedings of the forty-fifth annual ACM symposium on Theory of computing*, pp. 341–350. ACM, 2013.

Kalai, A. T. and Vempala, S. Efficient algorithms for online decision problems. *J. Comput. Syst. Sci.*, 71(3): 291–307, 2005. doi: 10.1016/j.jcss.2004.10.016. URL `https://doi.org/10.1016/j.jcss.2004.10.016`.

Kearns, M. J. Efficient noise-tolerant learning from statistical queries. *J. ACM*, 45(6):983–1006, 1998. doi: 10.1145/293347.293351. URL `https://doi.org/10.1145/293347.293351`.

Kearns, M. J., Neel, S., Roth, A., and Wu, Z. S. Preventing fairness gerrymandering: Auditing and learning for subgroup fairness. In Dy, J. G. and Krause, A. (eds.), *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, volume 80 of *JMLR Workshop and Conference Proceedings*, pp. 2569–2577. JMLR.org, 2018.

Kifer, D. Consistency with external knowledge: The topdown algorithm, 2019. `http://www.cse.psu.edu/~duk17/papers/topdown.pdf`.

Li, C., Miklau, G., Hay, M., McGregor, A., and Rastogi, V. The matrix mechanism: optimizing linear counting queries under differential privacy. *VLDB J.*, 24(6):757–781, 2015.

McKenna, R., Miklau, G., Hay, M., and Machanavajjhala, A. Optimizing error of high-dimensional statistical queries under differential privacy. *PVLDB*, 11(10):1206–1219, 2018.

McSherry, F. and Talwar, K. Mechanism design via differential privacy. In *FOCS*, volume 7, pp. 94–103, 2007.

Neel, S., Roth, A., and Wu, Z. S. How to use heuristics for differential privacy. In Zuckerman, D. (ed.), *60th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2019, Baltimore, Maryland, USA, November 9-12, 2019*, pp. 72–93. IEEE Computer Society, 2019.

Neel, S., Roth, A., Vietri, G., and Wu, Z. S. Oracle efficient private non-convex optimization. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020*, 2020.

Nikolov, A. An improved private mechanism for small databases. In *International Colloquium on Automata, Languages, and Programming*, pp. 1010–1021. Springer, 2015.

Nikolov, A., Talwar, K., and Zhang, L. The geometry of differential privacy: the sparse and approximate cases. In *Proceedings of the forty-fifth annual ACM symposium on Theory of computing*, pp. 351–360, 2013.

Suggala, A. S. and Netrapalli, P. Online non-convex learning: Following the perturbed leader is optimal. *CoRR*, abs/1903.08110, 2019. URL `http://arxiv.org/abs/1903.08110`.

Syrgkanis, V., Krishnamurthy, A., and Schapire, R. E. Efficient algorithms for adversarial contextual learning. In *Proceedings of the 33rd International Conference on International Conference on Machine Learning - Volume 48*, ICML'16, pp. 2159–2168. JMLR.org, 2016. URL `http://dl.acm.org/citation.cfm?id=3045390.3045618`.

Ullman, J. Answering $n^{2+o(1)}$ counting queries with differential privacy is hard. *SIAM J. Comput.*, 45(2): 473–496, 2016. doi: 10.1137/130928121. URL `https://doi.org/10.1137/130928121`.

Ullman, J. and Vadhan, S. Pcps and the hardness of generating private synthetic data. In *Theory of Cryptography Conference*, pp. 400–416. Springer, 2011.

A Missing Proofs in Section 4

.

This section describes the accuracy analysis of FEM and sepFEM in detail. The accuracy proof proceeds in two steps. First we show that the sample distribution $\widehat{D^t}$ played by the data player is close the true distribution D^t. Then we show that both the query player and and data player are following no-regret strategies. Then, by Theorem 7, we show that algorithms FEM and sepFEM find an approximate equilibrium of the game dynamics described in section 3.

To bound the deviation error in our sampling from the FTPL distribution, we use the following Chernoff bound.

Lemma 12 (Chernoff Bound)**.** *Let $X_1, \ldots, X_m$ be i.i.d random variables such that $0 \le X_i \le 1$ for all i. Let $S = \frac{1}{m}\sum_{i=1}^m X_i$ denote their mean and let $\mu = \mathbb{E}[S]$ denote their expected mean. Then,*

$$Pr\left[|S-\mu| > t\right] \le 2\exp\left(-2mt^2\right)\right]$$

Lemma 13. *Let $\beta \in (0,1)$ and let D^t be the true distribution over $\mathcal{X}$. Suppose we draw*

$$s = \frac{8\log\left(4T|\mathcal{Q}|/\beta\right)}{\alpha^2}$$

samples $\{x_i^t\}$ from D^t to form $\widehat{D^t}$. Then for all $q \in \mathcal{Q}$, with probability at least $1-\beta/2$, we have

$$\left|\frac{1}{s}\sum_{i=1}^s q(x_i^t) - q(D^t)\right| < \frac{\alpha}{4} \text{ for all } 0 \le t \le T$$

Proof. For any fixed t, note that $\frac{1}{s}\sum_{i=1}^s q(x_i^t)$ is the average of the random variables $q(x_1^t), q(x_2^t), \ldots, q(x_s^t)$. Also $\mathbb{E}[q(x^t)] = q(D^t)$ for all $0 \le t \le T$. Thus by the Chernoff bound and our choice of s,

$$\Pr\left[\left|\frac{1}{s}\sum_{i=1}^s q(x_i^t) - q(D^t)\right| > \frac{\alpha}{4}\right] \le 2\exp\left(-s\alpha^2/8\right) = \frac{\beta}{2T|\mathcal{Q}|}$$

A union bound over all T rounds and all $|\mathcal{Q}|$ queries gives a total fail probability of at most $\beta/2$ as desired. □

The query player following the Exponential Mechanism has bounded regret with high probability.

Lemma 14 (Query Player's Regret)**.** *Let n be the dataset size. For any $\rho > 0$, query class $\mathcal{Q}$, round T, and any sequence of actions $D_1, \ldots, D_T$ by the data player, with probability $1-\beta/2$ the query player from algorithm 1 achieves an average regret bound of*

$$R_{\text{qry}}(T) \le \frac{1}{n}\sqrt{\frac{2T}{\rho}}\log\left(\frac{2T|\mathcal{Q}|}{\beta}\right)$$

Proof. On each round the query player calls the exponential mechanism with parameter $\sqrt{2\rho_0}$. Since the sensitivity of the query player's score function Δ_S is $1/n$, then with probability $1-\beta/2T$ the error for each is round is at most $\frac{2/n}{\sqrt{2\rho_0}}\log\left(2T|\mathcal{Q}|/\beta\right)$ by theorem 5. Applying union bound over T rounds, with probability $1-\beta/2$ the query player's average regret for T rounds is

$$\max_{q\in\mathcal{Q}}\frac{1}{T}\sum_{t=1}^T A(\widehat{D^t}, q) - \frac{1}{T}\sum_{t=1}^T A(\widehat{D^t}, q^t) \le \frac{1}{T}\sum_{t=1}^T \frac{2/n}{\sqrt{2\rho_0}}\log\left(2T|\mathcal{Q}|/\beta\right) \le \frac{1}{n}\sqrt{\frac{2T}{\rho}}\log\left(2T|\mathcal{Q}|/\beta\right)$$

where the last inequality follows from $\rho_0 = \frac{\rho}{T}$. □

Now we will provide the accuracy guarantees for FEM and sepFEM by analyzing data player's regret in the two algorithms.

Lemma 15 (Data Player's Regret in FEM)**.** *Let $d = \log(\mathcal{X})$. For any round T and target accuracy $\alpha > 0$, there exist a parameters η and s such that if data player from algorithm* FEM *(2) plays the sequence of distributions approximations $\widehat{D^1}, \widehat{D^2} \ldots \widehat{D^T}$, and the query player plays any adversarially chosen sequence of queries $q_1, \ldots, q_T \in \mathcal{Q}$, then the data player, with probability at least $1 - \beta/2$, achieves an average regret bound of*

$$R_{\text{data}}^{\text{FEM}}(T) \leq \tfrac{\alpha}{4} + \tfrac{5}{2} d^{3/2} \sqrt{\frac{1}{T}}$$

Proof. For the data player, we use the Non-Convex-FTPL algorithm for non-convex losses due to Suggala & Netrapalli (2019). Recall that, given a sequence of queries $q_1, \ldots, q_T$ the data player in algorithm 1 wants to choose actions $x_1, \ldots, x_T$ to maximize the objective

$$\sum_{t=1}^{T} q_i(x_t)$$

thus, the regret of the data player can be writen as

$$R_{\text{data}}^{\text{FEM}}(T) = \frac{1}{T} \max_{x \in \mathcal{X}} \sum_{t=1}^{T} q_i(x) - \frac{1}{T} \sum_{t=1}^{T} q_i(x_t)$$

The results from Suggala & Netrapalli (2019) say that if an online learner chooses an action from some decision space with ℓ_∞ diameter D, the loss functions are L-Lipschitz for ℓ_1 norm, and the learner has access to an (α, β)-approximate optimization oracle then the learner has expected average regret of the learner bounded by

$$\mathbb{E}\left[R(T)\right] = 125 \eta L d^2 D + \frac{\beta d}{20 \eta L} + 2 \beta d + \frac{\alpha}{20 L} \tag{4}$$

Suppose that the data player chooses one action on each round by solving the following optimization problem

$$x_t \in \underset{x \in \{0,1\}^d}{\arg\min} \left\{ \sum_{i=1}^{t-1} q_i(x) - \langle x, \sigma_t \rangle \right\} \tag{5}$$

where each $\sigma_t \in \mathbb{R}^d$ is sampled from the exponential distribution, and each $q_i \in \mathcal{Q}$ is chosen by adversarially. We assume that on each round t, the data player plays a single record $x_t \in \mathcal{X}$ from the data space $\mathcal{X} = \{0,1\}^d$ which as ℓ_∞ diameter of 1. Furthermore, each q_i is 1-Lipschitz, this follows because each query is bounded in $[0,1]$ and the input are 0-1 vectors from the set $\{0,1\}^d$. Therefore if $x \neq y$ then at last one coordinate in x and y differ by one, hence $\|x - y\|_1 \geq 1$. Then the following holds for all $x, y \in \mathcal{X}$ such that $x \neq y$ and all $q \in \mathcal{Q}$:

$$|q(x) - q(y)| \leq 1 \leq \|x - y\|_1$$

We assume that our oracle is a perfect optimizer so $\alpha' = 0$ and $\beta = 0$. Therefore, we replace constants $D = 1$ and $L = 1$ in equation 4 and expected regret of the data player is bounded by

$$\mathbb{E}\left[R(x_1, \ldots, x_T)\right] = \mathbb{E}_{x_1, \ldots, x_T} \left[\frac{1}{T} \max_{x \in \mathcal{X}} \sum_{t=1}^{T} q_i(x) - \frac{1}{T} \sum_{t=1}^{T} q_i(x_t) \right] \leq 125 \eta d^2 + \frac{d}{20 \eta T}$$

Each x_t is a random variable sampled from its *true distribution* $D^t \in \Delta\mathcal{X}$, which is given by eq. (5). Now suppose that on each round we could play the true distribution D^t instead of x_t, then we can write the regret without the expectation

$$\frac{1}{T}\max_{D\in\Delta\mathcal{X}}\sum_{t=1}^{T} q_i(D) - \frac{1}{T}\sum_{t=1}^{T} q_i(D^t) \leq 125\eta d^2 + \frac{d}{20\eta T}$$

We want to approximate D^t. To that end, the algorithm creates a set $\widehat{D}^t$ of s samples from the distribution D^t by repeatedly calling the optimization oracle with different perturbation values sampled from the exponential distribution with parameter η. From Lemma 13, we know that there exist a sample size s such that with probability at least $1-\beta/2$, the average error per round of sample $\widehat{D^t}$ from the true distribution D^t is $\alpha/4$. Hence, with probability at least $1-\beta/2$, the average regret per round for the data player playing the sample distribution $\widehat{D^t}$ is

$$\frac{1}{T}\max_{D\in\Delta\mathcal{X}}\sum_{t=1}^{T} q_i(D) - \frac{1}{T}\sum_{t=1}^{T} q_i(D^t) \leq \tfrac{\alpha}{4} + 125\eta d^2 + \frac{d}{20\eta T}$$

Setting $\eta = \sqrt{\frac{1}{2500Td}}$, we have

$$\frac{1}{T}\max_{D\in\Delta\mathcal{X}}\sum_{t=1}^{T} q_i(D) - \frac{1}{T}\sum_{t=1}^{T} q_i(D^t) \leq \tfrac{\alpha}{4} + \sqrt{(125d^2)\left(\frac{d}{20T}\right)} = \tfrac{\alpha}{4} + d^{3/2}\sqrt{\frac{125}{20}\frac{1}{T}}$$

□

Lemma 16 (Data Player's Regret in sepFEM)**.** *Let $d = \log(\mathcal{X})$ and $M = |\operatorname{sep}(\mathcal{Q})|$. For any round T and target accuracy $\alpha > 0$, there exist a parameters η and s such that if data player from algorithm* sepFEM *3 plays the sequence of distributions approximations $\widehat{D^1}, \widehat{D^2} \ldots \widehat{D^T}$, and the query player plays any adversarially chosen sequence of queries $q_1, \ldots, q_T \in \mathcal{Q}$, then the data player, with probability at least $1-\beta/2$, achieves an average expected regret bound of*

$$R_{\text{data}}^{\text{sepFEM}}(T) \leq \tfrac{\alpha}{4} + M^{3/4}d^{1/2}\sqrt{\frac{40}{T}}$$

Proof. Let $M = |\operatorname{sep}(\mathcal{Q})|$ be the size of the separator set of the query class and $d = \log(|\mathcal{X}|)$ is the dimension of the data domain. We use the contextual bandits algorithm on the small separator setting from Syrgkanis et al. (2016) which achieves expected regret

$$4\eta M + \tfrac{10}{\eta}M^{1/2}\log(N)\frac{1}{T}$$

where N is the size of the policy space of the learner.

Suppose that the data player chooses x_t on each round t, following algorithm 3 due to Syrgkanis et al. (2016). In our setting we regard any datum $x_t \in \mathcal{X} = \{0,1\}^d$ as the policy played by the data player which maps queries to the set $\{0,1\}$. Therefore the policy space has size $2^d = |\mathcal{X}|$. Then according to Syrgkanis et al. (2016) and replacing N by 2^d we get that the data player achieves expected regret bounded by

$$\mathbb{E}_{x_1,\ldots,x_T}\left[\frac{1}{T}\max_{x\in\mathcal{X}}\sum_{t=1}^{T} q_i(x) - \frac{1}{T}\sum_{t=1}^{T} q_i(x_t)\right] \leq 4\eta M + \tfrac{10}{\eta}M^{1/2}\frac{d}{T}$$

Each x_t is a random variable sampled from its *true distribution* $D^t \in \Delta\mathcal{X}$. Now suppose that on each round we could play the true distribution D^t instead of x_t, then we can write the regret without the expectation

$$\frac{1}{T}\max_{D\in\Delta\mathcal{X}}\sum_{t=1}^{T} q_i(D) - \frac{1}{T}\sum_{t=1}^{T} q_i(D^t) \leq 4\eta M + \tfrac{10}{\eta}M^{1/2}\frac{d}{T}$$

We want to approximate D^t. To that end, the algorithm creates a set $\widehat{D}^t$ of s samples from the distribution D^t by repeatedly calling the optimization oracle with different perturbation values.

From Lemma 13, we know that with probability at least $1-\beta/2$, the average error per round of sample distribution $\widehat{D^t}$ from the true distribution D^t is $\alpha/4$. Hence, with probability at least $1-\beta/2$, the average regret per round for the data player playing the sample distribution $\widehat{D^t}$ is

$$\frac{1}{T}\max_{D\in\Delta\mathcal{X}}\sum_{t=1}^{T} q_i(D) - \frac{1}{T}\sum_{t=1}^{T} q_i(D^t) \leq \tfrac{\alpha}{4} + 4\eta M + \tfrac{10}{\eta}M^{1/2}d\frac{1}{T}$$

Setting $\eta = \sqrt{\frac{5d}{2M^{1/2}T}}$. Then the regret of the data player is

$$R_{\text{data}}^{\text{sepFEM}}(T) = \frac{\alpha}{4} + M^{3/4}d^{1/2}\sqrt{\frac{40}{T}}$$

□

Proof of Corollary 8.1.

Proof. From Lemma 15 and Lemma 14, let $R_{\text{data}}^{\text{FEM}}(T)$ and $R_{\text{qry}}(T)$ be the upper bounds for the average error of the data and query player respectively with probability at least $1-\beta/2$. Then, with probability at least $1-\beta$ due to the union bound over 2 events, α is the average regret for all rounds by Theorem 7:

$$\begin{aligned}\alpha &= R_{\text{data}}^{\text{FEM}}(T) + R_{\text{qry}}(T)\\ &= \frac{\alpha}{4} + \tfrac{5}{2}d^{3/2}\sqrt{\frac{1}{T}} + \frac{1}{n}\sqrt{\frac{2T}{\rho}}\log\left(2T|\mathcal{Q}|/\beta\right)\end{aligned}$$

To solve for α we first move the first term from the right hand side. Then we minimize the expression on the left side by setting the two terms equal to each other. We ignore the $\log(T)$ term and minimize $\frac{5}{2}d^{3/2}\sqrt{\frac{1}{T}} + \frac{1}{n}\sqrt{\frac{2T}{\rho}}\log(|\mathcal{Q}|)$ by selecting the correct choice of $\sqrt{T}$. That is, setting $T = \frac{\frac{5d^{3/2}}{2}}{\sqrt{\frac{2}{\rho n^2}\log(|\mathcal{Q}|)}}$ we get

$$\begin{aligned}\tfrac{3\alpha}{4} \leq& \sqrt{\left(\tfrac{5d^{3/2}}{2}\right)\left(\sqrt{\frac{2}{\rho n^2}\log(|\mathcal{Q}|)}\right)}\log(2T/\beta)\\ =&\frac{d^{3/4}}{\rho^{1/4}n^{1/2}}\sqrt{\tfrac{5}{2}}\sqrt[4]{2}\log(2T|\mathcal{Q}|/\beta)\end{aligned}$$

□

Proof of Corollary 8.2.

Proof. From lemma 16 we have that the data player's average regret for round T is $R_{\text{data}}^{\text{sepFEM}}(T) = M^{3/4}d^{1/2}\sqrt{40}T^{-1/2}$ and the average regret for the query player is $R_{\text{qry}}(T)$ given by lemma 14. Then, by union bound and by Theorem 7, with probability at least $1-\beta$, the accuracy of sepFEM is:

$$\begin{aligned}\alpha &= R_{\text{data}}^{\text{sepFEM}}(T) + R_{\text{qry}}(T)\\ &= \frac{\alpha}{4} + M^{3/4}d^{1/2}\sqrt{40}T^{-1/2} + \frac{1}{n}\sqrt{\frac{2T}{\rho}}\log\left(2T|\mathcal{Q}|/\beta\right)\end{aligned}$$

Now to choose T optimally we ignore the log term and set $T = \frac{M^{3/4} d^{1/2} \sqrt{40}}{\sqrt{2/\rho n^2 \log(|\mathcal{Q}|)}}$ to get

$$\frac{3}{4}\alpha \leq 2\sqrt[4]{5}\frac{M^{3/8} d^{1/4} \sqrt{\log(|\mathcal{Q}|)}}{n^{1/2}\rho^{1/4}}$$

□

B DQRS: DualQuery with Rejection Sampling

Theorem 17. DualQuery *with rejection sampling (Algorithm 4) takes in a private dataset $D \in \mathcal{X}^n$ and makes $T = O\left(\frac{\log|\mathcal{Q}|}{\alpha^2}\right)$ queries to an optimization oracle and outputs a dataset $\tilde{D} = (x^1, \cdots, x^T) \in \mathcal{X}^T$ such that, with probability at least $1-\beta$, for all $q \in \mathcal{Q}$ we have $|q(\tilde{D}) - q(D)| \leq \alpha$. The algorithm is ρ-CDP for*

$$\rho = O\left(\frac{\log(|\mathcal{X}|T/\beta) \cdot \log^3(|\mathcal{Q}|)}{n^2\alpha^5}\right).$$

In contrast, DualQuery (*without* rejection sampling) obtains the same result except with

$$\rho = O\left(\frac{\log(|\mathcal{X}|T/\beta) \cdot \log^3(|\mathcal{Q}|)}{n^2\alpha^7}\right).$$

To obtain (ε, δ)-differential privacy, it suffices to have ρ-CDP for $\rho = \Theta(\varepsilon^2/\log(1/\delta)$. Thus the guarantee of Theorem 17 can be rephrased as the sample complexity bound

$$n = O\left(\frac{\log^{1.5}(|\mathcal{Q}|) \cdot \sqrt{\log(|\mathcal{X}|T/\beta) \cdot \log(1/\delta)}}{\alpha^{2.5}\varepsilon}\right)$$

to obtain α-accurate synthetic data with probability $1-\beta$ under (ε, δ)-differential privacy.

Lemma 18. *The subroutine which accepts q with probability $\hat{Q}_q^{t+1}/Q_q^t = e^{-\eta-\gamma_t} \cdot \exp(-\eta A_D(x^t, q))$ is ε-differentially private for $\varepsilon = \max\{\eta/n, \eta/\gamma_t n\}$.*

Proof. Note that $0 < p := \hat{Q}_q^{t+1}/Q_q^t = e^{-\eta-\gamma_t} \cdot \exp(-\eta A_D(x^t, q)) \leq e^{-\gamma_t} < 1$. In particular, the probability is well-defined.

We compute the ratio between the probabilities that q is accepted under executions of the algorithm on neighboring datasets D, D' for fixed choices of the best responses $x^1, \ldots, x^t$. This ratio is given by

$$\frac{p}{p'} = \frac{\hat{Q}_q^{t+1}[D]}{Q_q^t[D]} \cdot \frac{Q_q^t[D']}{\hat{Q}_q^{t+1}[D']} = \frac{\exp(-\eta A_D(x^t, q))}{\exp(-\eta A_{D'}(x^t, q))} \leq e^{\eta/n}.$$

Similarly, we evaluate the ratio of the probabilities that q is *not* accepted under executions of the algorithm on D and D': Since $p' \leq e^{-\gamma_t}$ and $p/p' \geq e^{-\eta/n}$, we have

$$\frac{1-p}{1-p'} = 1 + \frac{1}{1/p' - 1}\left(1 - \frac{p}{p'}\right) \leq 1 + \frac{1 - e^{-\eta/n}}{e^{\gamma_t} - 1} \leq 1 + \frac{\eta/n}{\gamma_t} \leq e^{\eta/\gamma_t n},$$

as required. □

Bad samples also incur privacy loss from sampling from the distribution Q^t. Just as in Gaboardi et al. (2014), we use the fact that this step can be viewed as an instantiation of the exponential mechanism with score function $\sum_{i=1}^{t-1}(q(D) - q(x^i))$ to obtain:

Lemma 19. *Sampling from Q^t is ε-differentially private for $\varepsilon = 2\eta(t-1)/n$.*

Proof of Privacy for Theorem 17.

Proof. Each round t incurs privacy loss from s invocations of a $(\eta/\gamma_t n)$-differentially private algorithm (rejection sampling, Lemma 18), and $\tilde{s}_t$ invocations of a $(2\eta(t-1)/n)$-differentially private algorithm (Lemma 19). Since ε-differential privacy implies $\frac{1}{2}\varepsilon^2$-CDP Bun & Steinke (2016), we have (by composition) that round t is ρ_t-CDP for

$$\rho_t = \frac{\eta^2}{2\gamma_t^2 n^2} s + \frac{2\eta^2(t-1)^2}{n^2}\tilde{s}_t = \frac{\eta^2 s}{n^2}\left(\frac{1}{2\gamma_t^2} + 2(t-1)^2 \cdot (2\gamma_t + 4\eta)\right) \leq \frac{\eta^2 s}{n^2}\left(4t^{4/3} + 8\eta t^2\right).$$

Composing over rounds $t = 1 \cdots T$ yields $\rho = O\left(\frac{\log(|\mathcal{X}|T/\beta)\cdot \log^{2+1/3}(|\mathcal{Q}|)}{n^2\alpha^{4+2/3}} + \frac{\log(|\mathcal{X}|T/\beta)\cdot\log^3(|\mathcal{Q}|)}{n^2\alpha^5}\right)$, as required. $\square$

Accuracy

The accuracy analysis follows that of of DualQuery, together with the following claims showing that the rejection sampling process simulates the collection of independent samples in the DualQuery algorithm.

Lemma 20. *Let P and Q be probability distributions over $\mathcal{Q}$, and let $M \geq \max_{q\in\mathcal{Q}} P_q/Q_q$. Sample an element of $\mathcal{Q}$ as follows. Sample q according to Q, and accept it with probability $P_q/(M\cdot Q_q)$. If q is not accepted, sample q according to P. Then the resulting element is distributed according to P.*

Proof. The total probability of sampling q according to this procedure is given by

$$\begin{aligned} Q_q \cdot \frac{P_q}{M\cdot Q_q} + P_q \cdot \sum_{q'\in\mathcal{Q}} Q_{q'} \cdot \left(1 - \frac{P_{q'}}{M\cdot Q_{q'}}\right) &= P_q \cdot \left(\frac{1}{M} + \sum_{q'\in\mathcal{Q}}\left(Q_{q'} - \frac{P_{q'}}{M}\right)\right) \\ &= P_q \cdot \left(\frac{1}{M} + \left(1 - \frac{1}{M}\right)\right) \\ &= P_q. \end{aligned}$$

$\square$

Lemma 21. *For any given round t, the probability that more than $\tilde{s}_t$ samples are rejected is at most $(e/4)^{\tilde{s}_t} \leq \frac{\beta}{3T}$.*

Proof. The probability that any given sample is rejected is $1 - \hat{Q}_q^{t+1}/Q_q^t = 1 - e^{-\eta-\gamma_t}\cdot\exp(-\eta A_D(x^t,q)) \leq 1 - e^{-2\eta-\gamma_t} \leq 2\eta + \gamma_t = \frac{\tilde{s}_t}{2s}$. (In particular, $\tilde{s}_t$ is at least twice the expected number of rejected samples.) The set of s samples is rejected independently. By a multiplicative Chernoff bound, the probability that more than $\tilde{s}_t$ samples are rejected is at most $(e/4)^{\tilde{s}_t}$. Note that $\tilde{s}_t \geq 4\eta s = \frac{48}{\alpha}\log\left(\frac{3|\mathcal{X}|T}{\beta}\right)$. Thus $(e/4)^{\tilde{s}_t} \leq \left(\frac{\beta}{3|\mathcal{X}|T}\right)^{18/\alpha} \leq \frac{\beta}{3T}$. $\square$

Together Lemmas 20 and 21 show that, with high probability, at each round t, the set S_t is distributed as s independent samples from $\mathcal{Q}^t$. Given this, the rest of the proof follows that of the original DualQuery.

Proof of Accuracy for Theorem 17.

Proof. For each round t, by Hoeffding's bound and Lemma 21 and a union bound over $\mathcal{X}$, with probability at least $1 - \frac{\beta}{T}$, we have

$$\forall x \in \mathcal{X} \qquad \left|\frac{1}{s}\sum_{q\in S_t} q(x) - \mathop{\mathbb{E}}_{q\leftarrow\mathcal{Q}^t}[q(x)]\right| \leq \frac{\alpha}{4}.$$

By a union bound over the T rounds we have that the above holds for all $t \in [T]$ with probability at least $1 - \beta$.

By assumption, in each round t, our oracle returns x^t that is an $\alpha/4$-approximate best response to the uniform distribution over S_t. Thus, with high probability, the sequence $x^1, \cdots, x^T$ are $\alpha/2$-approximate best responses to the distributions $Q^1, \cdots, Q^t$. Since the distributions are generated by multiplicative weights, we have that this is an α-approximate equilibrium. Hence the uniform distribution over $x^1, \cdots, x^T$ is an α-accurate synthetic database for D. $\square$