Bounding the Excess Risk for Linear Models Trained on Marginal-Preserving, Differentially-Private, Synthetic Data

Yvonne Zhou [1] Mingyu Liang [1] Ivan Brugere [2] Danial Dervovic [2] Antigoni Polychroniadou [3 2] Min Wu [1]
Dana Dachman-Soled [1]

Abstract

The growing use of machine learning (ML) has raised concerns that an ML model may reveal private information about an individual who has contributed to the training dataset. To prevent leakage of sensitive data, we consider using differentially-private (DP), synthetic training data instead of real training data to train an ML model. A key desirable property of synthetic data is its ability to preserve the low-order marginals of the original distribution. Our main contribution comprises novel upper and lower bounds on the excess empirical risk of linear models trained on such synthetic data, for continuous and Lipschitz loss functions. We perform extensive experimentation alongside our theoretical results.

1. Introduction

Machine learning (ML) is extensively utilized at present, but a major concern is that the trained ML model may reveal private information about an individual who has contributed to the training dataset (Fredrikson et al., 2014; Shokri et al., 2017; Wang et al., 2021). In response, various differentially-private (DP) machine learning methods, which typically add noise during the *training* process, have been proposed in the literature (Abadi et al., 2016; Bassily et al., 2014a; Papernot et al., 2016; 2018; Jayaraman et al., 2018; Yu et al., 2021). We refer to these methods as Training-Based Differentially-Private Machine Learning (Training-DPML). In contrast, in this work, we consider using differentially-private, synthetic training data instead of real training data to train the machine learning model. By doing so, one automatically achieves the guarantee that any models trained on the synthetic data are themselves differentially-private–i.e. the weights associated with the trained models do not leak information about any single individual in the dataset–without adding any additional noise during training. We therefore refer to the methods we study in this work as Preprocessing-Based Differentially-Private Machine Learning (Pre-DPML).

Pre-DPML techniques are an attractive option as opposed to Training-DPML techniques for several reasons. First, Training-DPML algorithms require significant trust since the original sensitive data must be stored and handled throughout the training process and can only be discarded once all training has completed. Second, in Training-DPML techniques the privacy budget must grow with the total number of models trained and when the budget is depleted no further computations may be performed on the data.

In contrast, when Pre-DPML via DP synthetic data generation is employed, the synthetic data is generated once and for all and the original sensitive data can be immediately discarded. Subsequently, one can perform any downstream task any number of times without requiring an increased privacy budget. Further, one can safely use *any* optimization algorithm out-of-the-box for training on the synthetic data (e.g. second order methods or built-in Python optimization algorithms). Given the benefits of the Pre-DPML approach, our goal is to understand whether it is information-theoretically possible to generate synthetic data that achieves differential privacy and yields low excess risk in ML tasks. To answer this question, we first highlight a desirable property of DP synthetic data from the literature, known as *marginal-preserving* synthetic data. The main results of this work, which we summarize in Section 1.1, provide novel upper and lower bounds on the excess empirical risk when training linear models on real versus marginal-preserving, synthetic data. To obtain a complete end-to-end analysis, we prove that DP and marginal-preserving synthetic data is attainable, whereas the marginal-preserving properties of prior DP mechanisms were heuristic.

Marginal-preserving synthetic data generation. A d-th order marginal of a distribution is the joint probability distribution of a subset of d attributes. Similarly, a d-th

[1] University of Maryland, College Park, MD 20742 [2] J.P. Morgan AI Research, New York, NY, 10017 [3] AlgoCRYPT CoE . Correspondence to: Yvonne Zhou <skyzhou@umd.edu>.

Proceedings of the 41^{st} International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024. Copyright 2024 by the author(s).

order marginal of a *dataset* captures all possible statistics of the dataset for a subset of d attributes. Specifically, given a dataset, a marginal for a set of d attributes is a vector that counts the number of occurrences of each combination of possible values of the attributes in the set.

The goal of synthetic data generation algorithms is to produce a synthetic dataset that closely matches the statistics of the original dataset. In marginal-preserving synthetic data generation, the synthetic data preserves the statistics of a target set of marginals, as closely as possible.

Marginal-preserving approach for DP synthetic data. Various marginal-preserving and *differentially private* synthetic data generation algorithms have been proposed in the literature, such as PrivBayes(Zhang et al., 2017a), PrivSyn(Zhang et al., 2021), PrivMRF(Cai et al., 2021), PEP and GEM (Liu et al., 2021), Private-PGM (McKenna et al., 2019), and AIM(McKenna et al., 2022). Typically, the quality of the synthetic data has been measured in terms of the ability to accurately respond to statistical queries, even if the queries involve sets of attributes that were not contained in the target set of marginals. For example, in prior work, the synthetic data was evaluated by comparing its marginals with the marginals of the true data for random triples of attributes, or by examining how well the synthetic data preserved random high-order conjunctions (McKenna et al., 2021).

However, to our knowledge, research on the utility error of downstream tasks trained on synthetic data remains limited. While (Li et al., 2023)'s work made some initial strides in analyzing the utility of downstream tasks, it relied on certain strong assumptions. For instance, they assumed that the data distribution can be represented as a Bayesian network with a degree no greater than k, in which case the variation distance stemming from high-order terms can be omitted. Alternatively, they were able to remove this assumption, but in this case the error grows exponentially to the dimension. In contrast, our bound applies to any data distribution by using a polynomial approximation of the loss function in the analysis. This essentially allows us to bound the excess risk stemming from high-order marginals, *without imposing assumptions on the data distribution*. Additionally, they focused on training ML models with norm-bounded loss functions and utilized a specific marginal-based mechanism (PrivBayes). In contrast, our goal is to assess the quality of ML models trained on any continuous and Lipschitz loss function, and employs any marginal-based mechanisms.

1.1. Our Contributions

Our paper focuses on investigating the excess empirical risk (measured w.r.t. the real dataset) of training linear models on marginal-preserving synthetic data that approximately preserves the d-th order marginals of the real dataset. We present both theoretical and experimental results.

In Section 3.1, we upper bound the excess empirical risk, as long as the low-order marginals of the synthetic data are sufficiently close to the real marginals. We consider the setting where the dataset is scaled so that all m-dimensional datapoints lie in the m-dimensional unit ball and where we optimize the weights $\mathbf{w}$ over the unit ball. In Theorem 3.1 we demonstrate that if the ℓ_1 distance of all marginals up to order d of the real and synthetic data is at most ν, then for any continuous and $O(1)$-Lipschitz loss function, the difference in cost is upper-bounded by $O(1/\sqrt{d-1} + (3m)^{d-1}\nu/n)$, where n is the number of samples in both datasets. Additionally, in Theorem 3.2, we show that for logistic regression specifically, we achieve a tighter upper bound of $O(1/(d-1) + (3m)^{d-1}\nu/n)$.

In Section 3.2, we give an outline of an information-theoretic mechanism that generates (ϵ, δ)-differentially private synthetic data with a bounded ℓ_1 difference of $\frac{4m^{d/2}l^d\sqrt{2\ln(1.25/\delta)(\ln(2)(1+\lambda)+d\ln(ml))}}{\epsilon}$ except for $2^{-\lambda}$ probability, where l is the maximum domain size of any attribute. Substituting this bound into ν in the aforementioned Theorems, implies that as the size of the database n goes to infinity, the excess empirical risk is dominated by $O(\frac{1}{\sqrt{d-1}})$ for general continuous and $O(1)$-Lipschitz loss functions, and dominated by $O(\frac{1}{d-1})$ for logistic regression. In practice, various efficient DP algorithms can heuristically preserve the marginals. However, there is a lack of conclusive proof regarding the attainability of a specific ℓ_1 bound for all input datasets. We conduct experiments and report the average ℓ_1 distance, over selected queries, achieved in practice for multiple datasets in Section 5.5.

In Section 4, we lower bound the excess empirical risk and demonstrate that for a specific range of parameter choices, we obtain a nearly tight match to the upper bound: $\Omega(\frac{1}{\ln^3(n)})$ versus $O(\sqrt{\frac{\ln(\ln(n))}{\ln(n)}})$. Our lower bound asserts the existence of a particular data distribution for which no marginal-preserving synthetic data algorithm, even if inefficient, can significantly outperform the upper bound. This, however, does not eliminate the possibility of better performance for real-life data distributions. Indeed, in Section 5, our experimental results surpass the outcomes predicted by our lower bound. Exploring reasonable assumptions on data distributions that allow bypassing the lower bound and obtaining improved upper bounds is an interesting future direction.

We performed extensive experimentation, and the results can be found in Section 5. To summarize our findings, we observed that, when with $(2, \frac{1}{n^2})$-DP, the accuracy of the model trained on the marginal-preserving, DP synthetic data drops by less than 1% compared to the real data, and the excess empirical risk is less than 0.02. The exception is the

Heart dataset, which exhibits a 2.2% drop in accuracy and 0.032 excess empirical risk, likely due to its considerably smaller dataset size.

1.2. Related Work

Private stochastic gradient descent (SGD) was first introduced by Song et al. (Song et al., 2013), and was subsequently enhanced in (Bassily et al., 2014b) and (Abadi et al., 2016). DP-SGD modifies stochastic gradient descent by clipping per-sample gradients for sensitivity control and by injecting noise to aggregated batch gradients at each intermediate update. Researchers have explored the application of DP-SGD and its variants (Jayaraman et al., 2018) to various tasks (McMahan et al., 2017; Dupuy et al., 2022; De et al., 2022; Malekzadeh et al., 2021), and frameworks such as Distributed/Federated Learning (McMahan et al., 2017; Adnan et al., 2022; Lyu et al., 2020). In contrast to DP-GD/DP-SGD, recent studies (Avella-Medina et al., 2023; Ganesh et al., 2024) suggest introducing noise to the *Hessian* of the loss function rather than the gradient. This technique allows the realization of differentially private optimization via second-order methods, which demonstrate a faster convergence rate than first-order methods such as gradient descent. Another noteworthy DP-ML method is Private Aggregation of Teacher Ensembles(or PATE) (Papernot et al., 2016; 2018). PATE proposes training an ensemble of non-private models (teachers), obtaining their predictions on a small set of unlabeled public data, and central aggregating predictions with noise. The labeled public data points are then used to train a student model. It is apparent that deploying PATE would consume computational overhead for training multiple teacher models in order to train a single student model. Moreover, it crucially presupposes the availability of public, unlabeled data.

2. Notation and Background

We use $[n]$ to denote $\{1, 2, \ldots, n\}$ and boldface variable to represent a vector, e.g., $\mathbf{v}$ and $\mathbf{h}$. Moreover, we use $\mathbf{v}[i]$ to denote the i^{th} entry of the vector, and $\mathbf{v}[q] = (\mathbf{v}[j])_{j\in q}$ to denote the subvectors containing entries in set q.

2.1. Data and Marginals

Data. A dataset D is a multiset of n samples, each can be represented as $\mathbf{v} = (\mathbf{x}, y) \in V$, where $\mathbf{x} = (x_1, \ldots, x_m)$ is a vector of m features and y is the corresponding label/class for the sample. For convenience, we may also refer to y as the $(m+1)^{th}$ feature. For $j \in [m+1]$, let Ω_j denote the domain of possible values for j^{th} feature and $l = \max_j |\Omega_j|$. Also, we set $y \in \{-1, 1\}$. Finally, let $q \subseteq [m+1]$ be a subset of attributes, and $\Omega_q = \Pi_{j\in q}\Omega_j$.

Definition 2.1 (Marginal of Dataset)**.** The marginal of dataset D on a subset of attributes q is a vector $\mathbf{h}_q \in \mathbb{R}^{|\Omega_q|}$, indexed by domain element $\mathbf{t} \in \Omega_q$, such that each entry is a count, i.e., $\mathbf{h}_q[\mathbf{t}] = \sum_{\mathbf{v}\in D} \mathbb{I}[\mathbf{v}[q] = \mathbf{t}]$. We let $M_q : V^n \to \mathbb{R}^{|\Omega_q|}$ denote the function that computes the marginal on q, i.e., $\mathbf{h}_q = M_q(D)$.

Given that a marginal is specified by an attribute set q, we also refer to q as a marginal query. Moreover, for $d \leq m+1$, let $Q^m_{\leq d}$ consist of all $q \subseteq [m+1]$ with size at most d. Furthermore, we say a set of marginals $\{\mathbf{h}_q\}_{q\in Q^m_{\leq d}}$ is *consistent*, if there exists a dataset D, such that $M_q(D) = \mathbf{h}_q$ for all $q \in Q^m_{\leq d}$.

2.2. Learning Linear Models with a Convex Loss

We consider learning linear models for binary classification. Specifically, let $L(\mathbf{w}, D)$ be the empirical risk of dataset D on model $\mathbf{w}$ defined as $L(\mathbf{w}, D) \triangleq \frac{1}{n}\sum_{(\mathbf{x},y)\in D} \varphi(\langle \mathbf{w}, \mathbf{x}\rangle y))$, where $\varphi(\langle \mathbf{w}, \mathbf{x}\rangle y)) : \mathbb{R} \to \mathbb{R}$ is the loss of linear model $\mathbf{w}$ for sample $(\mathbf{x}, y)$. Throughout the paper, we consider φ that is convex and Lipschitz.

Logistic Regression Logistic regression is a prominent representative model in learning linear models. We denote its empirical risk of dataset D on model $\mathbf{w}$ as $\hat{L}(\mathbf{w}, D) \triangleq \frac{1}{n}\sum_{(\mathbf{x},y)\in D} \hat{\varphi}(\langle \mathbf{w}, \mathbf{x}\rangle y)$, where $\hat{\varphi}(\langle \mathbf{w}, \mathbf{x}\rangle y) = -\ln\left(\frac{1}{1+e^{-\langle \mathbf{w},\mathbf{x}\rangle y}}\right)$.

2.3. Polynomial Approximation

Our proof of the upper bound relies on the technique of approximating the loss function with a bounded degree polynomial. Specifically, we consider the Bernstein polynomial (Bernstein, 1912; Roulier, 1970; Guan, 2009), which provides a theoretic analysis of its approximation error and the absolute values of its coefficients.

Definition 2.2 (Bernstein Polynomial Approximation)**.** Let f be a function on $[a, b]$, the Bernstein polynomial approximation of degree d is defined as

$$P_d f(x) = \sum_{i=0}^{d} f\left(\frac{i}{d}\cdot(b-a)+a\right)\cdot B_{di}(x), a \leq x \leq b,$$

where $B_{di}(x) = \binom{d}{i}\cdot\left(\frac{x-a}{b-a}\right)^i\cdot\left(1-\frac{x-a}{b-a}\right)^{d-i}$.

Let $\|f\| = \max_{a\leq x\leq b}|f(x)|$ denote the maximum absolute value when the function takes value from $[a, b]$. We utilize the following two error upper-bound of Bernstein polynomial approximations.

Theorem 2.3 ((Roulier, 1970), Th. 1 and (Popoviciu, 1935), Th. 1.6.1)**.** *Suppose $a \leq 0 < 1 \leq b$, and f is a continuous* *function on* $[a, b]$*, for* $d = 1, 2, ...,$

$$\|P_d f - f\| \leq \frac{5}{4}\omega\left(f, \frac{b-a}{\sqrt{d}}\right), \qquad (1)$$

where ω *is the modulus of continuity of* f *on* $[a, b]$*. Additionally, let* $P_d f(x) = \sum_{k=0}^{d} a_{dk} x^k$*, then for* $d = 1, 2, ...,$

$$\sum_{k=0}^{d} |a_{dk}| \leq \|f\| \left(1 + \frac{2}{b-a}\right)^d. \qquad (2)$$

Theorem 2.4 ((Telyakovskii, 2009)). *Suppose* f *is a function on* $[0, 1]$ *with a continuous first-order derivative. For* $d = 1, 2, \ldots,$

$$\|P_d f - f\| \leq \frac{3}{4\sqrt{d}}\omega\left(f', \frac{1}{\sqrt{d}}\right),$$

where ω *is the modulus of continuity of* f'*, which is the first derivative of* f*.*

2.4. Differential Privacy

Differential privacy(Dwork et al., 2006) has emerged as the prevailing standard for managing the privacy risk to an individual associated with publicly sharing information about a dataset. We present the formal definition next.

Definition 2.5 ((ϵ, δ)-Differential Privacy). A randomized mechanism $\mathcal{M} : \mathcal{D} \to \mathcal{R}$ satisfies (ϵ, δ)-differential privacy if for any two adjacent inputs $x, x' \in \mathcal{D}$ and for any subset of outputs $\mathcal{S} \subseteq \mathcal{R}$ it holds that $Pr[\mathcal{M}(x) \in \mathcal{S}] \leq e^{\epsilon} Pr[\mathcal{M}(x') \in \mathcal{S}] + \delta$.

The **Gaussian Mechanism** (Dwork & Roth, 2014) adds random noise drawn from a Gaussian distribution to a query output, where the standard deviation of the noise is proportional to the sensitivity of the query.

Theorem 2.6 (Gaussian Mechanism). *Let* $\epsilon \in (0, 1)$ *and* $f : D \to R^d$*, be an arbitrary d-dimensional function. Define its* l_2 *sensitivity to be* $\Delta_2(f) = \max_{x,x'} \|f(x) - f(x')\|_2$*, where* x, x' *are any adjacent inputs in* $\mathcal{D}$*. Let* $\sigma^2 = \frac{2\Delta_2(f)^2 \log(1.25/\delta)}{\epsilon^2}$*. The Gaussian mechanism that adds noises sampled from* $\mathcal{N}(0, \sigma^2)$ *to each of the* d *components of* f*'s output is* (ϵ, δ)*-differential privacy.*

Differential privacy is immune to post-processing (Dwork & Roth, 2014): further computation on differentially private output will not further degrade the privacy guarantee.

Theorem 2.7 (Post-Processing). *Let* $\mathcal{M} : \mathcal{D} \to \mathcal{R}$ *be a randomized algorithm that is* (ϵ, δ)*-differentially private. Let* $f : \mathcal{R} \to \mathcal{R}'$ *be arbitrary randomized mapping. Then* $f \circ \mathcal{M}$ *is* (ϵ, δ)*-differentially private.*

3. Upper Bound on the Excess Empirical Risk

We present our upper bound on the excess empirical risk for learning linear models with continuous and Lipschitz losses using synthetic data. In Section 3.1, we utilize the polynomial approximation techniques to show that the risk difference between the models trained from real and synthetic datasets can be bounded using the ℓ_1 norm of marginal difference between real and synthetic datasets. In Section 3.2, we present an information-theoretic mechanism for generating synthetic data that is provably both marginal-preserving and DP, and we extend our theorems from Section 3.1 to demonstrate a trade-off between privacy and loss.

Throughout this section, we let D_r be the real dataset and D_s be the synthetic dataset. We assume the datasets are normalized, i.e., for all $(\mathbf{x}, y) \in D_r, D_s$ and for all $j \in [m]$, $\mathbf{x}[j] \in [-1, 1]$. On the other hand, the label y takes value from $\{-1, 1\}$, and we may also refer to y as the $(m+1)^{th}$ attribute. Given a set $q \in [m+1]$, let $\mathbf{h}_q^{(r)}$ and $\mathbf{h}_q^{(s)}$ denote the marginals of the real and synthetic datasets on q, i.e., $\mathbf{h}_q^{(r)} = M_q(D_r)$ and $\mathbf{h}_q^{(s)} = M_q(D_s)$. Let $Q_{\leq d}^m$ be the set of all subsets of attributes (including label) with size no more than d.

3.1. Bounding the Risk via Bounded Marginals' ℓ_1-Distance

We begin by presenting a generic result, assuming only the loss function is continuous and Lipschitz.

Theorem 3.1. *Let* $L(\mathbf{w}, D) = \sum_{(\mathbf{x},y)\in D} \frac{1}{n}\varphi(\langle \mathbf{w}, \mathbf{x} \rangle y)$ *such that* φ *is continuous and* K*-Lipschitz. Let* $\mathbf{w}_r = \mathrm{argmin}_{\mathbf{w}, \|\mathbf{w}\| \leq \tau} L(\mathbf{w}, D_r)$ *and* $\mathbf{w}_s = \mathrm{argmin}_{\mathbf{w}, \|\mathbf{w}\| \leq \tau} L(\mathbf{w}, D_s)$*. If for all* $q \in Q_{\leq d}^m$*,* $\|\mathbf{h}_q^{(r)} - \mathbf{h}_q^{(s)}\|_1 \leq \nu$*, then*

$$|L(\mathbf{w}_s, D_r) - L(\mathbf{w}_r, D_r)| \in O\Bigg(K \cdot \tau\sqrt{m/(d-1)} + \frac{1}{n} \cdot (K\tau\sqrt{m} + \varphi(0)) \cdot (3m \cdot \max\{1, \tau\})^{d-1}\nu\Bigg)$$

Note that each sample in the dataset, $\mathbf{x}$, lies in the m-dimensional ball of radius $\sqrt{m}$. If we set $\tau = \frac{1}{\sqrt{m}}$, we can view the optimization problem as consisting of datapoints contained in the m-dimensional unit ball and optimizing over linear models, $\mathbf{w}$, contained in the m-dimensional unit ball. Thus, setting $\tau = \frac{1}{\sqrt{m}}$ and $K = O(1)$, the above implies that as the size of the database n goes to infinity, the excess empirical risk of the optimization problem is dominated by $O(\frac{1}{\sqrt{d-1}})$.

Our proof relies on the two generic upper bounds for any $\mathbf{w}$ such that$\|\mathbf{w}\|_2 \leq \tau$. First, we construct an *approximated empirical risk function* L' through replacing the loss function φ with its degree $d-1$ Bernstein polynomial approximation $P_{d-1}\varphi$. Then, we argue $L'(\mathbf{w}, D) \approx L(\mathbf{w}, D)$ by invoking results on the maximum error in Bernstein polynomial approximation given in Theorem 2.3.

Second, we bound the difference in empirical risk $|L'(\mathbf{w}, D_s) - L'(\mathbf{w}, D_r)|$ between the real and synthetic datasets on *any* linear model $\mathbf{w}$, by using the approximately marginal-preserving property of the synthetic dataset. Specifically, $P_{d-1}\varphi(\langle \mathbf{w}, \mathbf{x}\rangle y)$ can be expanded to a multivariate polynomial, where each monomial contains at most d variables in $(\mathbf{x}, y)$. Next, we can upper bound the risk L', which is the average of this multivariate polynomial evaluated on each data sample, by a sum of the averages of individual monomials evaluated on each data sample. Then, this allows us to associate each average monomial with the marginal correponding to the set of attributes appearing in this monomial. Further, this average monomial value is fully determined given the corresponding marginal. Finally, we can apply the ℓ_1 norm bound between the marginals of the real and synthetic datasets to bound the difference of each average monomial.

By applying the above bounds on different linear models in a sequence of inequalities, we arrive at the theorem statement. We provide the formal proof in Appendix A.1.

Next, we give a tighter bound for logistic regression. Our theorem can also extend to any loss function whose first derivative is continuous.

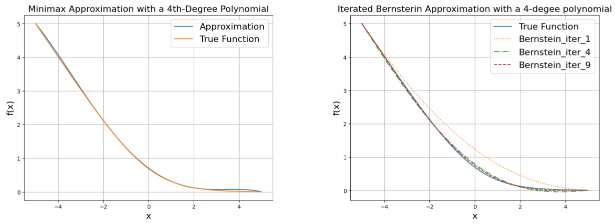
Theorem 3.2. *Let* $\hat{L}(\mathbf{w}, D) = \frac{1}{n}\sum_{(\mathbf{x},y)\in D}\hat{\varphi}(\langle \mathbf{w}, \mathbf{x}\rangle y)$*. Let* $\mathbf{w}_r = \mathsf{argmin}_{\mathbf{w}, \|\mathbf{w}\|\leq\tau}\hat{L}(\mathbf{w}, D_r)$ *and* $\mathbf{w}_s = \mathsf{argmin}_{\mathbf{w}, \|\mathbf{w}\|\leq\tau}\hat{L}(\mathbf{w}, D_s)$*. If for all* $q \in Q^m_{\leq d}$, $\|\mathbf{h}_q^{(r)} - \mathbf{h}_q^{(s)}\|_1 \leq \nu$*, then*

$$\begin{aligned}&|\hat{L}(\mathbf{w}_s, D_r) - \hat{L}(\mathbf{w}_r, D_r)|\\ \in &O\left(\tau\sqrt{m}/(d-1) + \frac{1}{n}\cdot\tau\sqrt{m}\cdot(2m\cdot\max\{1,\tau\})^{d-1}\nu\}\right).\end{aligned}$$

Setting $\tau = \frac{1}{\sqrt{m}}$, the above implies that as the size of the database n goes to infinity, the excess empirical risk is dominated by $O(\frac{1}{d-1})$ for logistic regression.

We provide the formal proof in Appendix A.2, wherein the primary difference is we apply a tighter Bernstein polynomial approximation bound from Theorem 2.4.

Polynomial Approximation Error The term $\tau\sqrt{m}/(d-1)$ in the bound in Theorem 3.2 comes from the error of the degree-$(d-1)$ Bernstein polynomial approximating the $\log(\text{sigmoid}(\cdot))$ function on the interval $[-\tau\cdot\sqrt{m}, \tau\cdot\sqrt{m}]$. For a fixed degree d, the Bernstein polynomial approximation may not yield the best error. Replacing it with an approximation with better error immediately leads to an improvement in the upper bound. We therefore investigate two alternative methods for polynomial approximation, namely the minimax approximation (Davis, 1975) and an "iterated" Bernstein approximation (Bernstein, 1912; Roulier, 1970; Guan, 2009). Refer to Figure 1 below for examples illustrating the quality of the approximations of the $\log(\text{sigmoid(x)})$ function by 4-degree polynomial functions obtained by using the minimax and iterated Bernstein approximations.



(a) Minimax Approximation (b) Bernstein Approximation

Figure 1. (a) shows Minimax approximation for $\log(\text{sigmoid(x)})$ function within interval $[-5, 5]$ in 4-degree polynomial: $\log(\text{sigmoid(x)})_{minimax} \approx 0.71 - 0.5x + 0.1096x^2 - 0.0015x^4$, with an error of 0.061. (b) shows the iterated Bernstein Approximations for $\log(\text{sigmoid(x)})$ function within interval $[-5, 5]$ in 4-degree polynomial by iterate Bernstein approximation for 1 time, 4 times, and 9 times: $\log(\text{sigmoid(x)})_{Bern_1} \approx 1.2377 - 0.5x + 0.0544x^2 - 0.0001x^4$, with an error of 0.545; $\log(\text{sigmoid(x)})_{Bern_4} \approx 0.7934 - 0.5x + 0.0812x^2 - 0.0005x^4$, with an error of 0.100; $\log(\text{sigmoid(x)})_{Bern_9} \approx 0.7504 - 0.5x + 0.0931x^2 - 0.0009x^4$, with an error of 0.057.

Through observations, three significant findings emerge. Firstly, the approximation error reduces while the polynomial degree increases for both approximation methods. Secondly, the error reduces with each successive iteration of the Bernstein approximation. Thirdly, the 9th-iterated Bernstein polynomial approximation slightly outperforms the minimax polynomial approximation in our experimental results. Nevertheless, we opt for Bernstein Approximation in our subsequent analysis, which provides a theoretic analysis of its approximation error and the absolute values of its coefficients. However, any polynomial approximation method can be used interchangeably in practical applications or in our analysis without losing generality by simply switching its approximation error bound according to the approximation method would be employed.

3.2. DP and marginal-preserving synthetic data

We present a DP synthetic data generating mechanism that preserves ℓ_1 norm of all marginals with order no more than d (with overwhelming probability), and analyze the end-to-end privacy and utility trade-off.

The differential privacy guarantee of Mechanism 1 follows directly from Theorem 2.6 and Theorem 2.7.

Lemma 3.3. $\mathsf{Gen}_{d,\sigma}$ *is* (ϵ, δ)*-DP, if* $\epsilon \in (0, 1)$ *and* $\sigma = \frac{2m^{d/2}\sqrt{\ln(1.25/\delta)}}{\epsilon}$.

We provide the formal proof in Appendix A.3.

Mechanism 1 Generating Synthetic Data $\mathsf{Gen}_{d,\sigma}$

```
Input: Real dataset $D_r$, number of samples $n$
Output: Synthetic Dataset $D_s$
Measure Noise Marginals $\mathsf{Mea}(D_r)$:
for $q \in Q^m_{\leq d}$ do
    Measuring marginal: $\mathbf{h}^{(r)}_q = M_q(D_r)$;
    Add noise: $\hat{\mathbf{h}}_q \leftarrow \mathbf{h}^{(r)}_q + \mathcal{N}(0, \sigma)$;
end for
Generate synthetic data $\mathsf{Syn}(n, \{\hat{\mathbf{h}}_q\}_{q \in Q^m_{\leq d}})$:
(Brute Force) Find $D_s$ that minimizes the maximum $\ell_1$
difference with respect to marginals in $\{\hat{\mathbf{h}}_q\}_{q \in Q^m_{\leq d}}$, i.e,
$D_s = \mathsf{argmin}_D \max_{q \in Q^m_{\leq d}} \|\hat{\mathbf{h}}_q - M_q(D)\|_1$.
```

Next, we bound the ℓ_1 difference between noisy and real marginals using Chernoff bound.

Lemma 3.4. *Let $D_s \leftarrow \mathsf{Gen}_{d,\sigma}$. Then $\|\mathbf{h}^{(r)}_q - M_q(D_s)\|_1 \leq 2l^d\sqrt{2(\ln(2)(1+\lambda) + d\ln(ml))}\sigma$ for all $q \in Q^m_{\leq d}$ with $1 - 2^{-\lambda}$ probability.*

We provide the formal proof in Appendix A.4.

Using Lemmas 3.3 and 3.4 allows us to represent the marginal difference ν in previous Theorems 3.1 and 3.2 with the expression containing the privacy parameters. In particular, it yields the following corollaries.

Corollary 3.5. *Let $L(\mathbf{w}; (\mathbf{x}, y)) = \varphi(\langle \mathbf{w}, \mathbf{x}\rangle y)$ such that φ is continuous and K-Lipschitz. Let $D_s \leftarrow \mathsf{Gen}_{d,\sigma}(D_r)$, where $\sigma = \frac{2m^{d/2}\sqrt{\ln(1.25/\delta)}}{\epsilon}$ and $\epsilon \in (0,1)$. Then D_s satisfies (ϵ,δ)-DP. Additionally, let $\mathbf{w}_r = \mathsf{argmin}_{\mathbf{w}, \|\mathbf{w}\| \leq \tau} L(\mathbf{w}, D_r)$ and $\mathbf{w}_s = \mathsf{argmin}_{\mathbf{w}, \|\mathbf{w}\| \leq \tau} L(\mathbf{w}, D_s)$. Then*

$$
\begin{aligned}
|L(\mathbf{w}_s, D_r) - L(\mathbf{w}_r, D_r)| \in O\Big(&K \cdot \tau\sqrt{m/(d-1)} \\
&+ \frac{1}{n} \cdot (K\tau\sqrt{m} + \varphi(0)) \cdot (3m \cdot \max\{1,\tau\})^{d-1} \\
&\cdot 2l^d\sqrt{2(\ln(2)(1+\lambda) + d\ln(ml))} \cdot \sigma\Big),
\end{aligned}
$$

except with $2^{-\lambda}$ probability.

Corollary 3.6. *Let $D_s \leftarrow \mathsf{Gen}_{d,\sigma}(D_r)$, where $\sigma = \frac{2m^{d/2}\sqrt{\ln(1.25/\delta)}}{\epsilon}$ and $\epsilon \in (0,1)$. Then D_s satisfies (ϵ,δ)-DP. Additionally, let $\mathbf{w}_r = \mathsf{argmin}_{\mathbf{w}, \|\mathbf{w}\| \leq \tau} \hat{L}(\mathbf{w}, D_r)$ and $\mathbf{w}_s = \mathsf{argmin}_{\mathbf{w}, \|\mathbf{w}\| \leq \tau} \hat{L}(\mathbf{w}, D_s)$. Then*

$$
\begin{aligned}
|\hat{L}(\mathbf{w}_s, D_r) - \hat{L}(\mathbf{w}_r, D_r)| \in O\Bigg(&\tau\sqrt{m}/(d-1) \\
&+ \frac{1}{n} \cdot \tau\sqrt{m} \cdot (3m \cdot \max\{1,\tau\})^{d-1} \\
&\cdot 2l^d\sqrt{2(\ln(2)(1+\lambda) + d\ln(ml))} \cdot \sigma\Bigg),
\end{aligned}
$$

except with $2^{-\lambda}$ probability.

As in the previous section, setting $\tau = \frac{1}{\sqrt{m}}$ and $K = O(1)$, the above corollaries imply that as the size of the database n goes to infinity, the excess empirical risk is dominated by $O(\frac{1}{\sqrt{d-1}})$ for general continuous and K-Lipschitz loss functions, and dominated by $O(\frac{1}{d-1})$ for logistic regression.

4. Lower Bound on the Excess Empirical Risk

We next present a theorem that shows that our upper bound in the previous section is nearly tight for certain ranges of parameter settings. Specifically, we show that there exists a distribution over datasets D_r, a convex, 2-Lipschitz cost function L, and a range of parameter settings for n, m, d, τ such that Theorem 3.1 implies the existence of a synthetic data generation algorithm with excess risk at most $O\left(\sqrt{\frac{\ln(\ln(n))}{\ln(n)}}\right) + O\left(\frac{\ln(n)}{n^{1/4}}\right)$. On the other hand, we show that for any synthetic data generation algorithm Syn (of a particular form), the excess risk is at least $\Omega\left(\frac{1}{\ln^3(n)}\right)$. Thus, both the upper and lower bounds on the difference in loss are fixed polynomials in $\frac{1}{\ln(n)}$, where n is the size of the dataset. Although existing differentially private convex optimization methods such as gradient perturbation(Bassily et al., 2014a; Yu et al., 2021), output perturbation(Zhang et al., 2017b), and objective perturbation (Chaudhuri et al., 2011) demonstrate an error of $O(1/n)$ or less, it is crucial to highlight the primary advantages of synthetic data: the ability to execute numerous downstream tasks without compromising the privacy guarantee, along with the flexibility to employ any non-private learning algorithm out-of-the-box.

Our lower bound captures synthetic data generation algorithms that obtain noisy marginals as input and then use an arbitrary (potentially computationally unbounded), randomized algorithm to construct a synthetic dataset from these noisy marginals. The synthetic data generation algorithms that we consider may not make any assumption about the distribution of the inputted noisy marginals, other than the fact that *each noisy marginal is close (within some tolerance) to the true expectation of the data distribution*. Our matching upper bound holds for synthetic data generation of this form, since Theorem 3.1 does not make any distributional assumption on $\mathbf{h}^{(s)}_q$ but only requires that $\|\mathbf{h}^{(s)}_q - \mathbf{h}^{(r)}_q\|_1 \leq \nu$.

Before presenting our Theorem and proof, we begin with some notation. For a vector $\mathbf{v} = (v[j])_{j \in q}$, let $n \cdot \mathbf{v} = (n \cdot v[j])_{j \in q}$. Let $\mathcal{D}_m$ be a distribution over $(\mathbf{x}, y)$, where input $\mathbf{x} \in \{-1,1\}^m$ and label $y \in \{-1,1\}$. In our writeup, we treat $(\mathbf{x}, y)$ as a single vector where the last entry is y. We say that a set of vectors $\{\mathbf{u}_q\}_{q \in Q^m_{\leq d}}$ has tolerance tol relative to distribution $\mathcal{D}_m$ if $\forall q \in Q^m_{\leq d}, \forall \mathbf{t} \in \Omega_q$, $\|\mathbf{u}_q(\mathbf{t}) - \mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}_m}[\mathbb{I}[(\mathbf{x}, y)[q] = \mathbf{t}]]\|_\infty \leq \mathsf{tol}$, where $\mathbb{I}[(\mathbf{x}, y)[q] = \mathbf{t}]$ is the indicator variable set to 1 if $(\mathbf{x}, y)[q] = \mathbf{t}$ and set to 0 otherwise. Let Syn be a synthetic data generation algorithm that receives as input $n \in \mathbb{N}$ and a set of vectors $\{n \cdot \mathbf{u}_q\}_{q \in Q^m_{\leq d}}$ and uses it in an arbitrary way to output a synthetic database D_s of size n with marginals $\{\mathbf{h}^s_q\}_{q \in Q^m_{\leq d}}$.

Theorem 4.1. *For sufficiently large n, $m = m(n) = O(\ln^6(n))$ and $d = d(n) = O(\frac{\ln(n)}{\ln\ln(n)})$, there exists a cost function $L(\mathbf{w}, D) := \frac{1}{n}\sum_{(\mathbf{x},y)\in D} \varphi(\langle \mathbf{w}, \mathbf{x}\rangle \cdot y)$ with φ being $\frac{1}{\ln^3(n)}$-strongly convex and 2-Lipschitz, for which the following hold:*

- *There exists a deterministic algorithm Syn such that for all distributions $\mathcal{D}_m$ and all sets of vectors $\{\mathbf{u}_q\}_{q\in Q^m_{\leq d}}$ with tolerance $\mathsf{tol} = \frac{1}{n}$ relative to $\mathcal{D}_m$, with all but negligible probability over $D_r \sim \mathcal{D}^n_m$,*

$$|L(\mathbf{w}_r, D_r) - L(\mathbf{w}_s, D_r)| \in O\left(\sqrt{\frac{\ln(\ln(n))}{\ln(n)}}\right) + O\left(\frac{\ln(n)}{n^{1/4}}\right),$$

- *For every randomized algorithm Syn, there exists a distribution $\mathcal{D}_m$ and a set of vectors $\{\mathbf{u}_q\}_{q\in Q^m_{\leq d}}$ with tolerance $\mathsf{tol} = \frac{1}{n}$ relative to $\mathcal{D}_m$, such that with all but negligible probability over $D_r \sim \mathcal{D}^n_m$,*

$$\left|L(\mathbf{w}_r, D_r) - \mathbb{E}_{D_s \leftarrow \mathsf{Syn}(n, \{n\cdot\mathbf{u}_q\}_{q\in Q^m_{\leq d}})}[L(\mathbf{w}_s, D_r)]\right| \in \Omega\left(\frac{1}{\ln^3(n)}\right),$$

where $\mathbf{w}_s = \mathsf{argmin}_{\mathbf{w}} L(\mathbf{w}, D_s)$, and $\mathbf{w}_r = \mathsf{argmin}_{\mathbf{w}} L(\mathbf{w}, D_r)$.

Our main insight to achieve the above result is that for any $\{\mathbf{u}_q\}_{q\in Q^m_{\leq d}}$ with tolerance $\mathsf{tol} = \frac{1}{n}$, and any algorithm Syn, e.g., the subroutine used in Mechanism 1, the Algorithm 2 can be viewed as a *non-adaptive statistical query* learning algorithm that makes $\sum_{q\in Q^m_{\leq d}} |\Omega_q| \leq m^d \cdot 2^d$ number of statistical queries.

Algorithm 2 A non-adaptive statistical query algorithm.

```
Let $\{\mathbf{u}_q\}_{q\in Q^m_{\leq d}}$ represent the responses of a statistical query oracle on the non-adaptive queries $\mathbf{t} \in \Omega_q$, for every $q \in Q^m_{\leq d}$;
Set $D_s \leftarrow \mathsf{Syn}(n, \{n\cdot\mathbf{u}_q\}_{q\in Q^m_{\leq d}})$;
Output $\mathbf{w}_s = \mathsf{argmin}_{\mathbf{w}} L(\mathbf{w}, D_s)$;
```

We can then invoke known lower bounds on the number of queries needed by non-adaptive statistical query algorithms to learn linear separators with statistical queries of a certain tolerance (Dagan & Feldman, 2020). Their Theorem 5 holds even for *large-margin linear separators*, where the target concept class consists of linear separators $\mathbf{w}$ such that for every $(\mathbf{x}, y)$ in the support of $\mathcal{D}$, $\frac{\langle\mathbf{x},\mathbf{w}\rangle y}{|\mathbf{x}|\cdot|\mathbf{w}|} \geq \gamma$. As we will see later, this large-margin will allow us to convert the lower bound given in Theorem B.3 which shows a gap in *accuracy*, to a result which shows a gap in *cost* (for cost function L_{D_r}) between the optimal linear separator and the linear separator outputted by the non-adaptive statistical query algorithm. We provide the formal proof of Theorem 4.1 in Appendix B.

Table 1. Summary of datasets used in experiments

Dataset	Size	#Dim	Dataset	Size	#Dim
Adult	48,842	14	Compas	7,214	9
Churn	3,859	16	Heart	303	14
Law	20,798	12	Dutch	60,420	12

5. Experimental Evaluation

We conducted experiments [1] to evaluate the performance of DP and marginal-preserving synthetic data generation on six public datasets. We select AIM (McKenna et al., 2022), the typical and notable mechanism from among the marginal-preserving methods, to generate the DP synthetic data. The assessment utilizes the "Train on Synthetic, Test on Real" (TSTR) approach (Esteban et al., 2017), where we train the real-data-model and synthetic-data-model (using the scikit-learn's(Pedregosa et al., 2011) library of logistic regression with the LBFGS solver), and evaluate both models on the real test data. Furthermore, we employ two other widely recognized DPML methods, DP-SGD (Abadi et al., 2016) and PATE learning (Papernot et al., 2016), for comparison with our proposed marginal-preserving synthetic data training.

5.1. Dataset

For our experimental evaluation, we utilized six datasets along with data preprocessing: Adult(Becker & Kohavi, 1996), Compas(Angwin et al., 2016), Churn(chu, 2020), Dutch(Centraal Bureau voor de Statistiek , CBS), Law(Wightman, 1998) and Heart(Janosi et al., 1988), refer to the Table 1 for an overview of the datasets.

5.2. Synthetic Data Generation

Marginal-based approaches are the state-of-art method for preserving key statistical properties of the ground truth data to generate synthetic data with DP guarantees. In our experiments, we examined its performance in DP-ML setting. The

[1] Our experiments code and datasets are available at `https://github.com/DPML-syn/MarginalPreserving_DP_SyntheticData`

marginal-based approaches all align with select-measure-generate framework, which, at a high level, can be divided into three steps: (1) Select sets of attributes, referred to as marginal queries, each containing at most d attributes; (2) Using the real dataset, compute the marginal for each selected query, with injected noise; (3) Generate synthetic data that matches the noisy marginals as closely as possible. We opt for one of the leading marginal-based mechanisms, AIM (McKenna et al., 2022), to validate the effectiveness of marginal preserving synthetic data. AIM is built using the core component Private-PGM (McKenna et al., 2019), wherein, Private-PGM operates for steps 2 and 3 in the framework. Additionally, AIM incorporates a greedily and iteratively algorithm to fulfill step 1. We defer a more detailed discussion of Private-PGM, AIM to Appendix C.

5.3. Data Preprocessing

The raw data we use to generate synthetic data may present various challenges, including missing values or containing continuous values that require conversion to discrete numbers. Therefore, we executed a series of data preprocessing before inputting it to the synthetic data generation mechanism: (1). Cleaning noisy data, by e.g. deleting data samples that contained missing values. (2). Converting categorical variables like gender and nationality into numerical values, to make them suitable for machine learning algorithms. (3). Converting continuous variables, such as income, into discrete values, while preserving the original ascending order of values. The quantization method employed here is a simple bucketing approach. More sophisticated quantization methods, such as those discussed in Gersho et al. (Gersho & Gray, 1992), could lead to improved handling of continuous data. (4). Feature scaling, to scale numeric features to a standard range starting from 0.

We highlight that these pre-processing steps applied to the real data do not compromise the privacy guarantee of the outputted synthetic data, since the data-preprocessing steps do not impact the sensitivity of the marginals, which determines the amount of noise added. Leveraging the post-processing theorem 2.7, we can safely perform any supplementary data-preprocessing steps, e.g. data normalization, before engaging on subsequent ML training. This augments the model's training effectiveness without degrading its privacy.

5.4. Evaluation Metrics

We evaluate the performance of our approach using metrics of accuracy and ROC-AUC score, as they are commonly used and provide a comprehensive evaluation of classification performance. Accuracy measures the proportion of correctly classified samples, while ROC-AUC score indicates how well the classifier discriminates between the positive and negative classes. Additionally, we also compare the empirical risk from both models on real testing data.

To gain better insight of marginal-preserving synthetic data, we conducted comparative experiments: We generate synthetic data for six(6) dataset with each eight(8) DP parameters, ϵ. We compared the performance of trained ML model in these synthetic data across with different ϵ. Furthermore, as a supplementary investigation, we conducted two additional experiments: (1). We compare marginal-preserving synthetic data approach with current predominant Training-DPML approaches: PATE learning and DP-SGD (refer to Appendix D.2 for details). (2) We demonstrated the model-agnostic advantage of AIM's synthetic data by evaluating its synthetic data on training in two classifiers with distinct target labels. AIM proves to be effective without requiring prior knowledge of which features specifically correspond to the downstream classification task, and consistently maintains its performance across diverse classifiers (see in Appendix D.3). This is desirable in the synthetic data setting, since the goal is to generate synthetic data once, and subsequently train many models on the same synthetic data.

5.5. Results

We assessed the generated synthetic datasets on (1) how well they preserved the marginals and (2) the performance of ML model training on the synthetic data. We utilize the normalized-ℓ_1 errors to evaluate the effectiveness of marginals preservation for different synthetic datasets generated. Here, the normalized-ℓ_1 error for a marginal query $q \in Q^m_{\leq d}$, is defined as $\frac{||\mathbf{h}_q^{(r)} - \mathbf{h}_q^{(s)}||_1}{n}$, where $\mathbf{h}_q^{(r)}$ and $\mathbf{h}_q^{(s)}$ denote the marginals of the real and synthetic datasets on q, and n is the size of the real dataset. AIM mechanism reports an average normalized-ℓ_1 error over all selected marginal queries, with the assertion that these errors serve as upper bounds for the maximum error across all marginals with at most d attributes, (including both selected and non-selected ones.) This assertion is substantiated by Theorem C.3. In our experiment, we set $d = 4$. The computed normalized-ℓ_1 errors are shown in Figure 2. It is easy to see that the higher the privacy budget, the less noise added into marginal measurements and so the smaller the normalized-ℓ_1 error in synthetic data.

In Figure 3 (see full results in Appendix D.1 Table 2,) we present our empirical results on the performance of ML models that are trained using marginal preserving synthetic datasets. The results show that the models acquired from training on the synthetic datasets with higher privacy budget exhibit higher accuracy, and lower excess empirical risk. In conjunction, we note that higher privacy budget enables us to achieve smaller ℓ_1 error synthetic data, leading to better synthetic data performance on ML training. Moreover, we observe that among all synthetic datasets, the Heart dataset has the lowest accuracy, which can likely be attributed to its

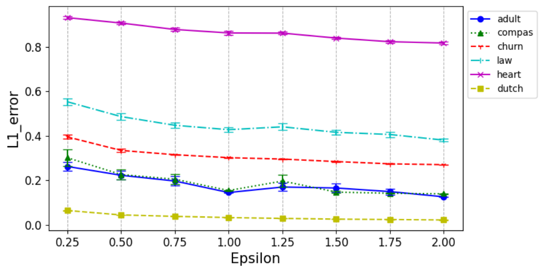


Figure 2. We compare the L_1 error of synthetic data using AIM mechanism for all six(6) datasets with different privacy budget.

relatively small sample size. Other than the Heart dataset, the accuracy of the models trained on the synthetic datasets, for $\epsilon = 2$, drops by less than 1% compared to the real data, and the excess empirical risk is less than 0.02.

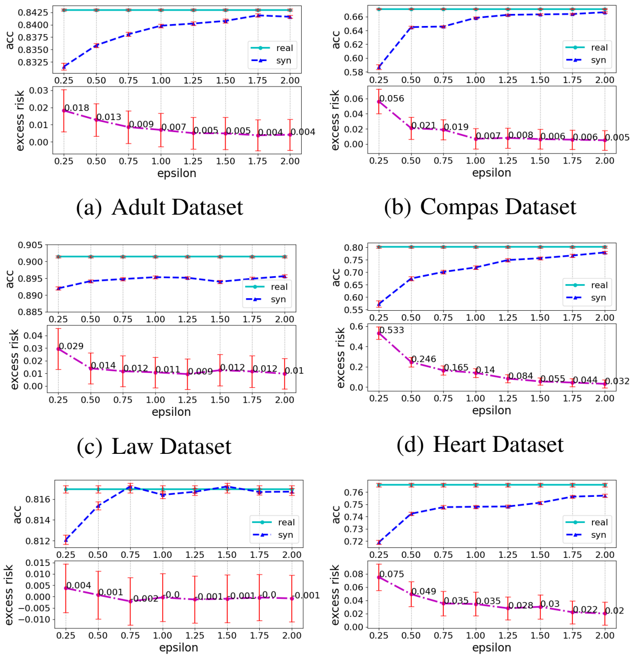
6. Conclusions and Future Work

In our study, we give both upper and lower bounds for the excess empirical risk (measured w.r.t. the real dataset) of training linear models on marginal-preserving synthetic data. Also, we show that for specific ranges of parameter choices, there exists a data distribution such that our upper and lower bounds are nearly tight (both are $1/\text{polylog}(n)$). Moreover, we give an end-to-end privacy and excess empirical risk analysis for a synthetic data generation mechanism that preserves all d-th order marginals. Finally, we supplement our theoretic results with extensive experiments using the AIM mechanism (McKenna et al., 2022) to heuristically generate marginal-preserving synthetic datasets for multiple real datasets. Our experiments show that the resulting models, with $\epsilon = 2$, reduce the accuracy by at most 2.2%, compared to that of the (non-private) real models.

Moving forward, we believe the following directions are interesting to consider: (1). Given that our experiments on real-world datasets perform significantly better than the lower bound for the worst-case data distribution, it is interesting to explore assumptions on the data distribution that are consistent with the real-world datasets, and which may allow bypassing the lower bound. (2) It will be interesting to extend our techniques to non-linear models, such as decision trees, SVM, KNN, and neural networks, etc. (3) Finally, it will be interesting to broaden our approach to handle data with continuous attributes, or with discrete attributes but very large cardinality. In both cases, the marginals are harder/costlier (in terms of privacy) to preserve, and it may be necessary to develop novel proof techniques.

Acknowledgments and Disclosure of Funding

The authors thank the anonymous reviewers for their insightful comments and valuable feedback on this work. Dana Dachman-Soled is supported in part by NSF grants CNS-2154705, CNS-1933033, and IIS2147276 is jointly provided with Min Wu. We gratefully acknowledge support from the JP Morgan Chase Faculty Research Award.



(a) Adult Dataset

(b) Compas Dataset

(c) Law Dataset

(d) Heart Dataset

(e) Dutch Dataset

(f) Churn Dataset

Figure 3. We generated synthetic data for the six(6) datasets with $\epsilon \in (\frac{1}{4}, \frac{2}{4}, \frac{3}{4}, 1, \frac{5}{4}, \frac{6}{4}, \frac{7}{4}, 2)$. We produce 10 randomized sets of synthetic data for each ϵ. We assess performance by training the machine learning model 10 times with randomly split datasets to 80% training, 20% testing. Note that some degree of minor unpredictability is inevitable due to the limited number of trials, and this causes the slight graph oscillation.

Disclaimer This paper was prepared for informational purposes in part by the Artificial Intelligence Research group of JPMorgan Chase & Co and its affiliates ("J.P. Morgan") and is not a product of the Research Department of J.P. Morgan. J.P. Morgan makes no representation and warranty whatsoever and disclaims all liability, for the completeness, accuracy or reliability of the information contained herein. This document is not intended as investment research or investment advice, or a recommendation, offer or solicitation for the purchase or sale of any security, financial instrument, financial product or service, or to be used in any way for evaluating the merits of participating in any transaction, and shall not constitute a solicitation under any jurisdiction or to any person, if such solicitation under such jurisdiction or to such person would be unlawful.

© 2024 JPMorgan Chase & Co. All rights reserved.

Impact Statement

This paper presents work whose goal is to advance the protection of an individual's privacy in Machine Learning (ML) applications. Ensuring privacy is a societal concern, and is especially crucial in the ML setting where large amounts of potentially sensitive data are required for training. Furthermore, we believe that the particular methodology put forth in this work–in which differentially private (DP) synthetic data is generated for training—allows for equitable access to training data, in comparison to standard Training-DPML techniques. Specifically, the DP synthetic data can be released publicly once generated. Further, any out-of-the box optimization algorithm can be run on the data, in contrast to Training-DP algorithms, which require specialized knowledge to properly set the parameters and to run the modified algorithms. Finally, our experiments were performed solely on publicly available data, and we anticipate no potential misuse of the outcomes derived from our research.

References

Iranian Churn. UCI Machine Learning Repository, 2020. DOI: https://doi.org/10.24432/C5JW3Z.

Abadi, M., Chu, A., Goodfellow, I., McMahan, H. B., Mironov, I., Talwar, K., and Zhang, L. Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*. ACM, oct 2016. doi: 10.1145/2976749.2978318. URL https://doi.org/10.1145%2F2976749.2978318.

Adnan, M., Kalra, S., Cresswell, J., et al. Federated learning and differential privacy for medical image analysis. *Scientific Reports*, 12:1953, 2022. doi: 10.1038/s41598-022-05539-7.

Angwin, J., Larson, J., Mattu, S., and Kirchner, L. How we analyzed the compas recidivism algorithm. ProPublica, 2016. URL https://www.propublica.org/article/how-we-analyzed-the-compas-recidivism-algorithm.

Avella-Medina, M., Bradshaw, C., and Loh, P.-L. Differentially private inference via noisy optimization. *The Annals of Statistics*, 51(5):2067–2092, 2023.

Bassily, R., Smith, A., and Thakurta, A. Private empirical risk minimization: Efficient algorithms and tight error bounds. In *2014 IEEE 55th Annual Symposium on Foundations of Computer Science*, pp. 464–473, 2014a. doi: 10.1109/FOCS.2014.56.

Bassily, R., Smith, A., and Thakurta, A. Private empirical risk minimization: Efficient algorithms and tight error bounds. In *2014 IEEE 55th annual symposium on foundations of computer science*, pp. 464–473. IEEE, 2014b.

Becker, B. and Kohavi, R. Adult. UCI Machine Learning Repository, 1996. DOI: https://doi.org/10.24432/C5XW20.

Bernstein, S. Proof of the theorem of weierstrass based on the calculus of probabilities. *Communications of the Kharkov Mathematical Society*, 13:1–2, 1912.

Cai, K., Lei, X., Wei, J., and Xiao, X. Data synthesis via differentially private markov random fields. *Proc. VLDB Endow.*, 14(11):2190–2202, jul 2021. ISSN 2150-8097. doi: 10.14778/3476249.3476272. URL https://doi.org/10.14778/3476249.3476272.

Canonne, C. L., Kamath, G., and Steinke, T. The discrete gaussian for differential privacy. *Advances in Neural Information Processing Systems*, 33:15676–15688, 2020.

Centraal Bureau voor de Statistiek (CBS) (Statistics Netherlands), M. P. C. The dutch virtual census of 2001 - ipums subset. Integrated Public Use Microdata Series (IPUMS) [dataset]. Minneapolis: University of Minnesota, 2015, 2016-04-25. URL https://microdata.worldbank.org/index.php/catalog/2102. DOI: 10.18128/D020.V6.4.

Chaudhuri, K., Monteleoni, C., and Sarwate, A. D. Differentially private empirical risk minimization. *Journal of Machine Learning Research*, 12(3), 2011.

Dagan, Y. and Feldman, V. Interaction is necessary for distributed learning with privacy or communication constraints. In *Proceedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing*, STOC 2020, pp. 450–462, New York, NY, USA, 2020. Association for Computing Machinery. ISBN 9781450369794. doi: 10.1145/3357713.3384315. URL https://doi.org/10.1145/3357713.3384315.

Davis, P. J. *Interpolation and approximation*. Courier Corporation, 1975.

De, S., Berrada, L., Hayes, J., Smith, S. L., and Balle, B. Unlocking high-accuracy differentially private image classification through scale. *arXiv preprint arXiv:2204.13650*, 2022.

Dupuy, C., Arava, R., Gupta, R., and Rumshisky, A. An efficient dp-sgd mechanism for large scale nlu models. In *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 4118–4122, 2022. doi: 10.1109/ICASSP43922.2022.9746975.

Dwork, C. and Roth, A. The algorithmic foundations of differential privacy. *Foundations and Trends in Theoretical Computer Science*, 9(3-4):211–407, 2014.

Dwork, C., McSherry, F., Nissim, K., and Smith, A. Calibrating noise to sensitivity in private data analysis. In Halevi, S. and Rabin, T. (eds.), *Theory of Cryptography*, pp. 265–284, Berlin, Heidelberg, 2006. Springer Berlin Heidelberg. ISBN 978-3-540-32732-5.

Esteban, C., Hyland, S. L., and Rätsch, G. Real-valued (medical) time series generation with recurrent conditional gans. *ArXiv*, abs/1706.02633, 2017. URL https://api.semanticscholar.org/CorpusID:29681354.

Fredrikson, M., Lantz, E., Jha, S., Lin, S., Page, D., and Ristenpart, T. Privacy in pharmacogenetics: An end-to-end case study of personalized warfarin dosing. In *USENIX Security Symposium*, pp. 17–32, 2014.

Ganesh, A., Haghifam, M., Steinke, T., and Guha Thakurta, A. Faster differentially private convex optimization via second-order methods. *Advances in Neural Information Processing Systems*, 36, 2024.

Gersho, A. and Gray, R. M. *Vector Quantization and Signal Compression*. The Springer International Series in Engineering and Computer Science. Springer, New York, NY, 1 edition, 1992. ISBN 978-0-7923-9181-4. doi: 10.1007/978-1-4615-3626-0. Published: 30 November 1991.

Guan, Z. Iterated bernstein polynomial approximations. *arXiv*, preprint arXiv:0909.0684, 2009.

Iyengar, R., Near, J. P., Song, D., Thakkar, O., Thakurta, A., and Wang, L. Towards practical differentially private convex optimization. In *2019 IEEE Symposium on Security and Privacy (SP)*, pp. 299–316. IEEE, 2019.

Janosi, A., Steinbrunn, W., Pfisterer, M., and Detrano, R. Heart Disease. UCI Machine Learning Repository, 1988. DOI: https://doi.org/10.24432/C52P4X.

Jayaraman, B., Wang, L., Evans, D., and Gu, Q. Distributed learning without distress: Privacy-preserving empirical risk minimization. In Bengio, S., Wallach, H., Larochelle, H., Grauman, K., Cesa-Bianchi, N., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018. URL https://proceedings.neurips.cc/paper_files/paper/2018/file/7221e5c8ec6b08ef6d3f9ff3ce6eb1d1-Paper.pdf.

Li, X., Wang, C., and Cheng, G. Statistical theory of differentially private marginal-based data synthesis algorithms. *arXiv preprint arXiv:2301.08844*, 2023.

Liu, T., Vietri, G., and Wu, Z. S. Iterative methods for private synthetic data: Unifying framework and new methods. *arXiv*, 2106.07153, 2021.

Lyu, L., Li, Y., Nandakumar, K., Yu, J., and Ma, X. How to democratise and protect ai: Fair and differentially private decentralised deep learning. *IEEE Transactions on Dependable and Secure Computing*, pp. 1–1, 2020. ISSN 2160-9209. doi: 10.1109/tdsc.2020.3006287. URL http://dx.doi.org/10.1109/TDSC.2020.3006287.

Malekzadeh, M., Hasircioglu, B., Mital, N., Katarya, K., Ozfatura, M. E., and Gündüz, D. Dopamine: Differentially private federated learning on medical data. *arXiv preprint arXiv:2101.11693*, 2021.

McKenna, R., Sheldon, D., and Miklau, G. Graphical-model based estimation and inference for differential privacy. *CoRR*, abs/1901.09136, 2019. URL http://arxiv.org/abs/1901.09136.

McKenna, R., Miklau, G., and Sheldon, D. Winning the NIST contest: A scalable and general approach to differentially private synthetic data. *CoRR*, abs/2108.04978, 2021. URL https://arxiv.org/abs/2108.04978.

McKenna, R., Mullins, B., Sheldon, D., and Miklau, G. Aim: An adaptive and iterative mechanism for differentially private synthetic data. *arXiv preprint arXiv:2201.12677*, 2022.

McMahan, H. B., Ramage, D., Talwar, K., and Zhang, L. Learning differentially private recurrent language models. *arXiv preprint arXiv:1710.06963*, 2017.

McSherry, F. and Talwar, K. Mechanism design via differential privacy. In *Proceedings - Annual IEEE Symposium on Foundations of Computer Science, FOCS*, pp. 94–103, 11 2007. ISBN 978-0-7695-3010-9. doi: 10.1109/FOCS.2007.66.

Papernot, N. and Steinke, T. Hyperparameter tuning with renyi differential privacy. *arXiv preprint arXiv:2110.03620*, 2021.

Papernot, N., Abadi, M., Erlingsson, U., Goodfellow, I., and Talwar, K. Semi-supervised knowledge transfer for deep learning from private training data. *arXiv preprint arXiv:1610.05755*, 2016.

Papernot, N., Song, S., Mironov, I., Raghunathan, A., Talwar, K., and Erlingsson, Ú. Scalable private learning with pate. *arXiv preprint arXiv:1802.08908*, 2018.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

Popoviciu, T. Sur l'approximation des fonctions convexes d'ordre supérieur. *Mathematica (Cluj)*, 10:49–54, 1935.

Roulier, J. A. Permissible bounds on the coefficients of approximating polynomials. *Journal of Approximation Theory*, 3(2):117–122, 1970. ISSN 0021-9045. doi: https://doi.org/10.1016/0021-9045(70)90018-3. URL https://www.sciencedirect.com/science/article/pii/0021904570900183.

Shokri, R., Stronati, M., Song, C., and Shmatikov, V. Membership inference attacks against machine learning models. In *2017 IEEE symposium on security and privacy (SP)*, pp. 3–18. IEEE, 2017.

Song, S., Chaudhuri, K., and Sarwate, A. D. Stochastic gradient descent with differentially private updates. In *2013 IEEE Global Conference on Signal and Information Processing*, pp. 245–248, 2013. doi: 10.1109/GlobalSIP.2013.6736861.

Telyakovskii, S. On the rate of approximation of functions by the bernstein polynomials. *Proc. Steklov Inst. Math.*, 264(Suppl 1):177–184, 2009. doi: 10.1134/S0081543809050150. URL https://doi.org/10.1134/S0081543809050150.

Wang, K.-C., Fu, Y., Li, K., Khisti, A., Zemel, R., and Makhzani, A. Variational model inversion attacks. *Advances in Neural Information Processing Systems*, 34: 9706–9719, 2021.

Wightman, L. Lsac national longitudinal bar passage study. LSAC Research Report Series. ERIC, 1998.

Yu, D., Zhang, H., Chen, W., Yin, J., and Liu, T.-Y. Gradient perturbation is underrated for differentially private convex optimization. In *Proceedings of the Twenty-Ninth International Conference on International Joint Conferences on Artificial Intelligence*, pp. 3117–3123, 2021.

Zhang, J., Cormode, G., Procopiuc, C. M., Srivastava, D., and Xiao, X. Privbayes: Private data release via bayesian networks. *ACM Trans. Database Syst.*, 42(4), oct 2017a. ISSN 0362-5915. doi: 10.1145/3134428. URL https://doi.org/10.1145/3134428.

Zhang, J., Zheng, K., Mou, W., and Wang, L. Efficient private erm for smooth objectives. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, pp. 3922–3928, 2017b.

Zhang, Z., Wang, T., Li, N., Honorio, J., Backes, M., He, S., Chen, J., and Zhang, Y. PrivSyn: Differentially private data synthesis. In *30th USENIX Security Symposium (USENIX Security 21)*, pp. 929–946. USENIX Association, August 2021. ISBN 978-1-939133-24-3. URL https://www.usenix.org/conference/usenixsecurity21/presentation/zhang-zhikun.

A. Proofs in Section 3

A.1. Proof of Theorem 3.1

Proof. Our proof relies on the following empirical risk function L' that approximates L:

$$L'(\mathbf{w}, D) = \frac{1}{n} \sum_{(\mathbf{x},y)\in D} P_{d-1}\varphi(\langle \mathbf{w}, \mathbf{x}\rangle y),$$

where $P_{d-1}\varphi$ is the degree-$(d-1)$ Bernstein polynomial to approximate φ within the interval $[-\tau\sqrt{m}, \tau\sqrt{m}]$ (or $[-1,-1]$ if $\tau\sqrt{m} < 1$).

Lemma A.1. *For any normalized dataset D and any $\mathbf{w}$ such that $\|\mathbf{w}\|_2 \leq \tau$,*

$$|L'(\mathbf{w}, D) - L(\mathbf{w}, D)| \in O(K \cdot \tau\sqrt{m/(d-1)}).$$

Proof. It suffices to show that for any $(\mathbf{x}, y) \in D$, $|P_{d-1}\varphi(\langle \mathbf{w}, \mathbf{x}\rangle y) - \varphi(\langle \mathbf{w}, \mathbf{x}\rangle y)| \in O(K \cdot \tau\sqrt{m/(d-1)})$.

First, we have $|\langle \mathbf{w}, \mathbf{x}\rangle y| \leq \|\mathbf{w}\|_2\|\mathbf{x}\|_2 \leq \tau\sqrt{m}$, where the first inequality follows from Cauchy–Schwarz inequality and $y \in \{-1, 1\}$, and the second inequality follows from $\|\mathbf{w}\|_2 \leq \tau$.

Next, using the approximation error of Bernstein polynomial (Theorem 2.3, Eq. 1), we have the maximum error $|P_{d-1}\varphi(\langle \mathbf{w}, \mathbf{x}\rangle y) - \varphi(\langle \mathbf{w}, \mathbf{x}\rangle y)| \in O(\omega(\varphi, \frac{\tau\sqrt{m}}{\sqrt{d-1}}))$ for any $\langle \mathbf{w}, \mathbf{x}\rangle y \in [-\tau\sqrt{m}, \tau\sqrt{m}]$. As φ is K-Lipschitz, $\omega(\varphi, \frac{\tau\sqrt{m}}{\sqrt{d-1}}) \leq K \cdot \frac{\tau\sqrt{m}}{\sqrt{d-1}}$. □

Next, we bound the empirical risk difference using L' between the real and synthetic datasets on any $\mathbf{w}$.

Lemma A.2. *For any $\mathbf{w}$ such that $\|\mathbf{w}\|_2 \leq \tau$, and any datasets D_r and D_s such that for all $q \in Q^m_{\leq d}$, $\|\mathbf{h}^{(r)}_q - \mathbf{h}^{(s)}_q\|_1 \leq \nu$, we have*

$$\begin{aligned}&|L'(\mathbf{w}, D_r) - L'(\mathbf{w}, D_s)| \\ \in& O\left(\frac{1}{n} \cdot (K\tau\sqrt{m} + \varphi(0)) \cdot (3m \cdot \max\{1, \tau\})^{d-1}\nu\right).\end{aligned}$$

Proof. We start by expressing the L' of dataset D on $\mathbf{w}$ using D's marginals with order no more than d:

$$\begin{aligned}L'(\mathbf{w}, D) =& \frac{1}{n} \sum_{(\mathbf{x},y)\in D} P_{d-1}\varphi(\langle \mathbf{w}, \mathbf{x}\rangle y) \\ =& \frac{1}{n} \sum_{(\mathbf{x},y)\in D} \sum_{k=0}^{d-1} a_k(\langle \mathbf{w}, \mathbf{x}\rangle y)^k \\ =& \frac{1}{n} \sum_{(\mathbf{x},y)\in D} \sum_{k=0}^{d-1} a_k \sum_{\mathbf{u}\in[m]^k} \prod_{u\in\mathbf{u}} (\mathbf{w}[u] \cdot \mathbf{x}[u] \cdot y) \\ =& \frac{1}{n} \sum_{k=0}^{d-1} a_k \sum_{\mathbf{u}\in[m]^k} \sum_{\mathbf{t}\in\Omega_q, q=\mathcal{S}(\mathbf{u})\cup\{m+1\}} \mathbf{h}_q(\mathbf{t}) \prod_{u\in\mathbf{u}} (\mathbf{w}[u] \cdot \mathbf{t}[u] \cdot \mathbf{t}[m+1]),\end{aligned}$$

where $\mathcal{S}(\mathbf{u})$ returns a set containing unique elements in $\mathbf{u}$ and the entries of $\mathbf{t}$ is indexed by the set q.

The above expression allows us to bound the empirical risk difference between real and synthetic datasets using the bounded difference of their marginals. Specifically,

$$
\begin{aligned}
&|L'(\mathbf{w}, D_r) - L'(\mathbf{w}, D_s)| \\
=&\left|\frac{1}{n}\sum_{k=0}^{d-1} a_k \sum_{\mathbf{u}\in[m]^k} \sum_{\mathbf{t}\in\Omega_q, q=\mathcal{S}(\mathbf{u})\cup\{m+1\}} (\mathbf{h}_q^r(\mathbf{t}) - \mathbf{h}_q^s(\mathbf{t})) \prod_{u\in\mathbf{u}} (\mathbf{w}[u]\cdot\mathbf{t}[u]\cdot\mathbf{t}[m+1])\right| \\
\leq&\left|\frac{1}{n}\sum_{k=0}^{d-1} a_k \cdot \sum_{\mathbf{u}\in[m]^k} \|\mathbf{h}_q^{(r)}(\mathbf{t}) - \mathbf{h}_q^{(s)}(\mathbf{t})\|_1 \tau^k\right| \\
\leq&\left|\frac{1}{n}\sum_{k=0}^{d-1} a_k \cdot \sum_{\mathbf{u}\in[m]^k} \nu\tau^k\right| \\
\leq&\left|\frac{1}{n}\sum_{k=0}^{d-1} a_k \cdot m^{d-1}\nu \max\{1,\tau\}^{d-1}\right| \\
\leq&\frac{1}{n}\sum_{k=0}^{d-1} |a_k| \cdot m^{d-1}\nu \max\{1,\tau\}^{d-1} \\
\in&O\left(\frac{1}{n}\cdot(K\cdot\tau\sqrt{m}+\varphi(0))\cdot 3^{d-1}\cdot(m\cdot\max\{1,\tau\})^{d-1}\nu\right),
\end{aligned}
$$

where the first inequality follows from $\|\mathbf{w}\|_2 \leq \tau$ and $\mathbf{t}[m+1] \in \{-1, 1\}$, and the last expression follows by applying Theorem 2.3, Eq. 2 to bound the sum of the absolute values of the polynomial coefficients.

□

We are ready to prove the Theorem statement by combining the results in Lemmas A.1 and A.2. Specifically, We write $A \overset{P}{\approx} B$ to denote the LHS and RHS is bounded by the error due in Lemma A.1 and write $A \overset{M}{\approx} B$ to denote the LHS and RHS is bounded by the error due in Lemma A.2.

Let $\mathbf{w}'_r = \mathsf{argmin}_{\mathbf{w}} L'(\mathbf{w}, D_r)$ and $\mathbf{w}'_s = \mathsf{argmin}_{\mathbf{w}} L'(\mathbf{w}, D_s)$. (In the case that there is more than one minimums, it suffices to use arbitrary tie-breaking.) Then we have:

$$
\begin{aligned}
L(\mathbf{w}_s, D_r) &\overset{P}{\approx} L'(\mathbf{w}_s, D_r) \overset{M}{\approx} L'(\mathbf{w}_s, D_s) \overset{P}{\approx} L(\mathbf{w}_s, D_s) \leq L(\mathbf{w}'_s, D_s) \\
&\overset{P}{\approx} L'(\mathbf{w}'_s, D_s) \leq L'(\mathbf{w}'_r, D_s) \overset{M}{\approx} L'(\mathbf{w}'_r, D_r) \leq L'(\mathbf{w}_r, D_r) \overset{P}{\approx} L(\mathbf{w}_r, D_r),
\end{aligned}
$$

where the inequalities follows the optimality of $\mathbf{w}'_s, \mathbf{w}_s, \mathbf{w}'_r, \mathbf{w}_r$. This suggests $L(\mathbf{w}_s, D_r) - L(\mathbf{w}_r, D_r) \in O\left(\frac{1}{n}\cdot(K\tau\sqrt{m}+\varphi(0))\cdot(3m\cdot\max\{1,\tau\})^{d-1}\nu\right)$. Similarly, we have:

$$
\begin{aligned}
L(\mathbf{w}_s, D_r) &\overset{P}{\approx} L'(\mathbf{w}_s, D_r) \overset{M}{\approx} L'(\mathbf{w}_s, D_s) \geq L'(\mathbf{w}'_s, D_s) \\
&\overset{M}{\approx} L'(\mathbf{w}'_s, D_r) \geq L'(\mathbf{w}'_r, D_r) \overset{P}{\approx} L(\mathbf{w}'_r, D_r) \geq L(\mathbf{w}_r, D_r),
\end{aligned}
$$

which suggests $L(\mathbf{w}_r, D_r) - L(\mathbf{w}_s, D_r) \in O\left(\frac{1}{n}\cdot(K\tau\sqrt{m}+\varphi(0))\cdot(3m\cdot\max\{1,\tau\})^{d-1}\nu\right)$. This concludes our proof of the Theorem.

□

A.2. Proof of Theorem 3.2

Proof. The majority of the proof is the same as that of Theorem 3.1. By using the additional property the first derivative of $\hat{\varphi}$ is continuous and its first derivative is $1/4$-Lipschitz, we can apply Theorem 2.4 to give a tighter bound of polynomial approximation error. Note that while Theorem 2.4 only considers functions defined over $[0, 1]$, we can shrink any function defined over $[a, b]$ into this range. In the case of $\hat{\varphi}$, this results in the Lipschitz constant of its first derivative multiplied by $(b-a) = \tau\sqrt{m}$. Therefore, we have

$$|\hat{L}'(\mathbf{w}, D) - \hat{L}(\mathbf{w}, D)| \in O(K\tau\sqrt{m}/(d-1)).$$

Finally, by plugging in $K = 1$ and $\|\hat{\varphi}\| \leq \ln(2) + \tau\sqrt{m}$ for logistic loss yields the result.

□

A.3. Proof of Lemma 3.3

Proof. Recall each marginal is a vector of counts, where altering a single data point can, at most, result in a difference of 1 in two counts. Therefore, the ℓ_2 sensitivity of the concatenated marginals is $\sqrt{2|Q^m_{\leq d}|} \leq \sqrt{2m^d}$. By Theorem 2.6, the noisy marginals satisfies (ϵ, δ)-DP. As the synthetic data is exclusively generated using these noisy marginals, it also satisfies (ϵ, δ)-DP through post-processing (Theorem 2.7).

□

A.4. Proof of Lemma 3.4

Proof. Note that for any $q \in Q^m_{\leq d}$ and any $\mathbf{t} \in \Omega_q$, $\hat{\mathbf{h}}_q(\mathbf{t}) - \mathbf{h}^{(r)}_q(\mathbf{t}) \sim \mathcal{N}(0, \sigma)$. Using Chernoff bound, $|\hat{\mathbf{h}}_q(\mathbf{t}) - \mathbf{h}^{(r)}_q(\mathbf{t})| \leq k\sigma$ with $1 - 2e^{-k^2/2}$ probability. Using Union bound, $\|\hat{\mathbf{h}}_q - \mathbf{h}^{(r)}_q\|_1 = \sum_{t \in \Omega_q} |\hat{\mathbf{h}}_q(\mathbf{t}) - \mathbf{h}^{(r)}_q(\mathbf{t})| \leq l^d k\sigma$ with $1 - 2l^d e^{-k^2/2}$ probability.

By definition of D_s, we have $\max_{q \in Q^m_{\leq d}} \|\hat{\mathbf{h}}_q - M_q(D_s)\|_1 \leq \max_{q \in Q^m_{\leq d}} \|\hat{\mathbf{h}}_q - M_q(D_r)\|_1$. Therefore, using triangle inequality, we have $\max_{q \in Q^m_{\leq d}} \|\mathbf{h}^{(r)}_q - M_q(D_s)\|_1 \leq 2l^d k\sigma$ with $1 - 2l^d e^{-k^2/2}$ probability. Finally, using union bound, we have the above inequality holds for all $q \in Q^m_{\leq d}$ with $1 - 2(ml)^d e^{-k^2/2}$.

By setting $k = \sqrt{2(\ln(2)(1+\lambda) + d\ln(ml))}$ concludes our proof. □

B. Proof of Theorem 4.1

Proof. We begin by setting parameters r, n, m, d, γ, τ as follows:

Definition B.1 (Parameter Settings)**.** We set parameters as follows:

- Set $r = 5/6$.
- Set $m > 2e$.
- Set $\gamma = (m/2)^{\frac{-5}{10-2r}}$.
- Set $d = \frac{c' \cdot \gamma^{-2r/5}}{-\ln(\gamma)}$, for $c' = \min\{\frac{1}{5}, \frac{c}{8}\}$, where c is a constant depending only on r (See Theorem B.3).
- Set $n = \exp(\gamma^{-2r/5})$.
- Set $\tau = \frac{1}{\sqrt{m}}$.

We next define the loss function which will be used for both the upper bound and the lower bound.

Consider the following convex loss function $\varphi_\gamma : [-1, 1] \to \mathbb{R}$ defined in (Dagan & Feldman, 2020):

$$\varphi_\gamma(t) = \frac{(1-t)^2}{8} + \begin{cases} 1 - 2t/\gamma & -1 \leq t \leq 0 \\ (t-\gamma)^2/\gamma^2 & 0 \leq t \leq \gamma \\ 0 & \gamma \leq t \leq 1. \end{cases} \tag{3}$$

The loss function φ from Theorem 4.1 is set to be $\varphi(t) := \gamma \cdot \varphi_\gamma(t)$.

Claim B.2. *Let $P_d\varphi(x)$ be the Bernstein polynomial of order d of φ on $[-1, 1]$. Then*

$$||P_d\varphi - \varphi|| \leq \frac{5}{\sqrt{d}}.$$

The claim follows from Theorem 2.3, Eq. 1 and the fact that φ is 2-Lipschitz. Note that $\gamma < 1$.

We now turn to the upper bound (the first item in Theorem 4.1). For the upper bound, the algorithm $\mathsf{Syn}(n, \{n \cdot \mathbf{u}_q\}_{q \in Q^m_{\leq d}})$ will return the database D_s of size n in the support of $\mathcal{D}^n_m$ that minimizes $\max_{q \in Q^m_{\leq d}} \|\mathbf{h}^{(s)}_q - n \cdot \mathbf{u}_q\|_1$, where $\{\mathbf{h}^{(s)}_q\}_{Q^m_{\leq d}}$ are the marginals computed with respect to D_s. In the following we show that if $\{\mathbf{u}_q\}_{q \in Q^m_{\leq d}}$ has tolerance $\mathsf{tol} = \frac{1}{n}$, then with all but negligible probability over choice of $D_r \sim \mathcal{D}^n_m$, $\max_{q \in Q^m_{\leq d}} \|\mathbf{h}^{(r)}_q - n \cdot \mathbf{u}_q\|_\infty \in O(\ln^2(n) \cdot \sqrt{n})$. This implies that the optimal $\{\mathbf{h}^{(s)}_q\}_{q \in Q^m_{\leq d}}$ must also satisfy $\max_{q \in Q^m_{\leq d}} \|\mathbf{h}^{(s)}_q - n \cdot \mathbf{u}_q\|_\infty \in O(\ln^2(n) \cdot \sqrt{n})$, which in turn implies that $\max_{q \in Q^m_{\leq d}} \|\mathbf{h}^{(s)}_q - \mathbf{h}^{(r)}_q\|_\infty \in O(\ln^2(n) \cdot \sqrt{n})$. Finally, the ℓ_1 norm of any marginals can be bounded $\max_{q \in Q^m_{\leq d}} \|\mathbf{h}^{(s)}_q - \mathbf{h}^{(r)}_q\|_1 \in O(2^d \cdot \ln^2(n) \cdot \sqrt{n})$.

We next show that with all but negligible probability over choice of $D_r \sim \mathcal{D}^n_m$, $\max_{q \in Q^m_{\leq d}} \|\mathbf{h}^{(r)}_q - n \cdot \mathbf{u}_q\|_\infty \in O(\ln^2(n) \cdot \sqrt{n})$. By Chernoff bounds and the tolerance guarantee, for a particular $q \in Q^m_{\leq d}$ and $\mathbf{t} \in \Omega_q$, $\Pr[|n \cdot \mathbf{u}_q[\mathbf{t}] - \mathbf{h}^{(r)}_q[\mathbf{t}]| > \beta] \leq 2 \cdot \exp(-2(\beta - 1)^2/n)$. We set $\beta = \ln^2(n) \cdot \sqrt{n}$ for this probability to be negligible in n. Since we have also set parameters such that $\sum_{q \in Q^m_{\leq d}} |\Omega_q| \leq n$, after taking a union bound over all $q \in Q^m_{\leq d}$ and $\mathbf{t} \in \Omega_q$, we have that with all but negligible probability over choice of $D_r \sim \mathcal{D}^n_m$, $\max_{q \in Q^m_{\leq d}} \|\mathbf{h}^{(r)}_q - n \cdot \mathbf{u}_q\|_\infty \in O(2^d \cdot \ln^2(n) \cdot \sqrt{n})$.

Using the parameter settings in Definition B.1 we invoke Theorem 3.1 to obtain the upper bound:

$$\begin{aligned}
|L(\mathbf{w}_s, D_r) - L(\mathbf{w}_r, D_r)| &\in O(\frac{1}{\sqrt{d}}) + O\left(\frac{(K\tau\sqrt{m} + \varphi(0)) \cdot (3m \cdot \max\{1, \tau\})^{d-1}\nu}{n}\right) \\
&\in O(\frac{1}{\sqrt{d}}) + O\left(\gamma(3m)^{d-1} \cdot \frac{2^d \cdot \ln^2(n) \cdot \sqrt{n}}{n}\right) \\
&\in O(\frac{1}{\sqrt{d}}) + O\left(\ln^{-1}(n)(6\ln^5 n)^{d-1} \cdot \frac{2^d \cdot \ln^2(n) \cdot \sqrt{n}}{n}\right) \\
&\in O\left(\sqrt{\frac{-\ln(\gamma)}{\gamma^{-2r/5}}}\right) + O\left((\ln^5 n)^d \cdot \frac{12^d \cdot \ln(n) \cdot \sqrt{n}}{n}\right) \\
&\in O\left(\sqrt{\frac{\ln(\ln(n))}{\ln(n)}}\right) + O\left(\frac{n^{1/4} \cdot \ln(n) \cdot \sqrt{n}}{n}\right) \\
&\in O\left(\sqrt{\frac{\ln(\ln(n))}{\ln(n)}}\right) + O\left(\frac{\ln(n)}{n^{1/4}}\right)
\end{aligned}$$

We now turn to the lower bound (the second item in Theorem 4.1). For the lower bound, we utilize the following lower bound on the accuracy of non-adaptive statistical query algorithms, where the accuracy is measured by the *classification error*: $\mathsf{err}_{f^*, \mathcal{D}_m}(\hat{f}) \triangleq \Pr_{(\mathbf{x}, y) \sim \mathcal{D}_m}[f^*(\mathbf{x}) \neq \hat{f}(\mathbf{x})]$. (Looking forward, we consider $\mathcal{D}_m$ being linearly separable, and f^* is one of the linear separators. Therefore, $f^*(\mathbf{x}) = y$ for any $(\mathbf{x}, y)$ in the support of $\mathcal{D}_m$.

Theorem B.3 (Theorem 5 in (Dagan & Feldman, 2020))**.** *Let $r \in (0, 1)$, $\gamma \in (0, 2^{-1/(1-r)})$, $m \geq 2 \cdot \gamma^{-2-2r/5}$ and define $\eta = \gamma^{1-r}$. Let $\mathcal{A}$ be a non-adaptive statistical query algorithm such that for any linear separator f^* and distribution $\mathcal{D}_m$ over $X = \{-1, 1\}^m$ with margin $\gamma(f^*, \mathcal{D}_m) \geq \gamma$, returns a hypothesis $\hat{f}$ with $\mathbb{E}_{\mathcal{A}}[\mathsf{err}_{f^*, \mathcal{D}_m}(\hat{f})] \leq 1/2 - \eta$. If $\mathcal{A}$ has* *access to statistical queries with tolerance* $\mathsf{tol} \geq \exp(-c\gamma^{-2r/5})$*, then* $\mathcal{A}$ *requires at least* $\exp(c\gamma^{-2r/5})$ *queries, where* $c > 0$ *is a constant depending only on* r*.*

Corollary B.4. *For the parameter settings given in Definition B.1, for any (even computationally inefficient) algorithm* Syn*, the algorithm defined in Algorithm 2 has error at least* $\mathbb{E}[\mathsf{err}_{f^*,D}(\hat{f})] > 1/4$*.*

The corollary follows by noting that for the parameter settings of r, n, m, γ, d in Definition B.1, all of the following hold: $r \in (0, 1)$, $\gamma \in (0, 2^{-1/(1-r)})$, $m \geq 2 \cdot \gamma^{-2-2r/5}$, $\eta = \gamma^{1-r} \leq \frac{1}{4}$, $\mathsf{tol} = \frac{1}{n} \geq \exp(-c\gamma^{-2r/5})$, and $\sum_{q \in Q^m_{\leq d}} |\Omega_q| \leq 2^d \cdot m^d < e^{8c'\gamma^{-2r/5}} \leq \exp(c\gamma^{-2r/5})$. Since the algorithm defined in Algorithm 2 is a non-adaptive statistical query algorithm with tolerance $\mathsf{tol} = \frac{1}{n}$ and making less than $\exp(c\gamma^{-2r/5})$ number of statistical queries, Theorem B.3 implies that its error must be at least $1/4$.

The above corollary gives a bound on the error of linear separator $\mathbf{w}_s$ outputted by Algorithm 2, whereas we need a bound on the difference in loss between $\mathbf{w}_s$ and the optimal linear separator. The following Claim allows us to relate the error and the loss.

Claim B.5. *Let the loss function* $L''(\mathbf{w}, \mathcal{D}_m) := \mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}_m}[\gamma \cdot \varphi_\gamma(y\langle \mathbf{w}, \mathbf{x}\rangle)]$*, where the expectation is taken with respect to distribution* $\mathcal{D}_m$*. Let* $\hat{\mathbf{w}}$ *be any vector of norm at most* τ*. Let* $\mathbf{w}^*$ *be the optimal linear separator with respect to* $L''(\mathbf{w}, \mathcal{D}_m)$*. Let* A *be any algorithm. If* $\mathbb{E}_{\hat{\mathbf{w}}\leftarrow A}[L''(\hat{\mathbf{w}}, \mathcal{D})] \leq L''(\mathbf{w}^*, \mathcal{D}) + \frac{\gamma}{8}$*, then* $\mathbb{E}_{\hat{\mathbf{w}}\leftarrow A}[\mathsf{err}_{\mathcal{D}_m}(\hat{\mathbf{w}})] \leq 1/4$*.*

Proof. Assume $\mathbb{E}_{\hat{\mathbf{w}}\leftarrow A}[L''(\hat{\mathbf{w}}, \mathcal{D})] \leq L''(\mathbf{w}^*, \mathcal{D}) + \frac{\gamma}{8}$. Then this implies that $\mathbb{E}_{\hat{\mathbf{w}}\leftarrow A}[L'(\hat{\mathbf{w}}, \mathcal{D}_m)] \leq L'(\mathbf{w}^*, \mathcal{D}_m) + \frac{1}{8}$, where $L'_{\mathcal{D}_m}$ is the cost function $L'(\mathbf{w}, \mathcal{D}_m) := \mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}_m}[\varphi_\gamma(y\langle \mathbf{w}, \mathbf{x}\rangle)]$. By Claim 3 in (Dagan & Feldman, 2020), this implies that $\mathbb{E}_{\hat{\mathbf{w}}\leftarrow A}[\mathsf{err}_{\mathcal{D}_m}(\hat{\mathbf{w}})] \leq 1/4$. □

Taking Corollary B.4 and Claim B.5 together, we have that for every algorithm Syn there exists a distribution $\mathcal{D}_m$ and a set of vectors $\{\mathbf{u}_q\}_{q\in Q^m_{\leq d}}$ of tolerance $\mathsf{tol} = \frac{1}{n}$ such that

$$\mathbb{E}_{D_s \leftarrow \mathsf{Syn}(n, \{n\cdot\mathbf{u}_q\}_{q\in Q^m_{\leq d}})}[L''(\mathbf{w}_s, \mathcal{D}_m)] \geq L''(\mathbf{w}^*, \mathcal{D}_m) + \frac{\gamma}{8}. \tag{4}$$

We must now convert the expected loss given above to *excess empirical risk w.r.t. the real training data*. To do so, we note that for every $(\mathbf{x}, y)$ in the support of $\mathcal{D}_m$, $\varphi(y\langle \mathbf{w}, \mathbf{x}\rangle)$ is lower bounded by 0 and upper bounded by 1 and therefore so is $\mathbb{E}_{D_s \leftarrow \mathsf{Syn}(n, \{n\cdot\mathbf{u}_q\}_{q\in Q^m_{\leq d}})}[\varphi(y\langle \mathbf{w}_s, \mathbf{x}\rangle)]$. Recall that

$$L''(\mathbf{w}^*, \mathcal{D}_m) = \mathbb{E}_{(\mathbf{x},y)\leftarrow\mathcal{D}_m}[\varphi(y\langle \mathbf{w}^*, \mathbf{x}\rangle)]. \tag{5}$$

By linearity of expectation, we also have that

$$\mathbb{E}_{D_s \leftarrow \mathsf{Syn}(n, \{n\cdot\mathbf{u}_q\}_{q\in Q^m_{\leq d}})}[L''(\mathbf{w}_s, \mathcal{D}_m)] = \mathbb{E}_{(\mathbf{x},y)\leftarrow\mathcal{D}_m}[\mathbb{E}_{D_s \leftarrow \mathsf{Syn}(n, \{n\cdot\mathbf{u}_q\}_{q\in Q^m_{\leq d}})}[\varphi(y\langle \mathbf{w}_s, \mathbf{x}\rangle)]]. \tag{6}$$

Since our setting of parameters implies that $\frac{n}{\ln^2(n)} \geq \frac{800}{\gamma^2}$, we have by (5), (6) and by standard Hoeffding bounds that with all but negligible probability over choice of D_r,

$$\mathbb{E}_{D_s \leftarrow \mathsf{Syn}(n, \{n\cdot\mathbf{u}_q\}_{q\in Q^m_{\leq d}})}[L''(\mathbf{w}_s, \mathcal{D}_m)] - \mathbb{E}_{D_s \leftarrow \mathsf{Syn}(n, \{n\cdot\mathbf{u}_q\}_{q\in Q^m_{\leq d}})}[L(\mathbf{w}_s, D_r)] \leq \frac{\gamma}{20} \text{ and } L(\mathbf{w}^*, D_r) - L''(\mathbf{w}^*, \mathcal{D}_m) \leq \frac{\gamma}{20}. \tag{7}$$

Therefore, combining (4), (7), and by the optimality of $\mathbf{w}_r$,

$$\mathbb{E}_{D_s \leftarrow \mathsf{Syn}(n, \{n\cdot\mathbf{u}_q\}_{q\in Q^m_{\leq d}})}[L(\mathbf{w}_s, D_r)] \geq L(\mathbf{w}^*, D_r) + \frac{\gamma}{16} \geq L(\mathbf{w}_r, D_r) + \frac{\gamma}{16}. \tag{8}$$

Substituting $\gamma = \frac{1}{\ln^3(n)}$ into (8) we obtain

$$|\mathbb{E}_{D_s \leftarrow \mathsf{Syn}(n, \{n\cdot\mathbf{u}_q\}_{q\in Q^m_{\leq d}})}[L(\mathbf{w}_s, D_r)] - L(\mathbf{w}_r, D_r)| \in \Omega(\frac{1}{\ln^3(n)}),$$

which concludes the proof of the theorem. □

C. More on Synthetic Data Generation

C.1. Private-PGM

The core of Private-PGM is to fit a graphical model to the sensitive data in a differentially-private way, and then use the graphical model to generate the synthetic data. The high-level steps involve computing noisy marginals of the sensitive data for selected sets of attributes of small size. Secondly, executing an optimization problem to identify a probability distribution that "best explains" these noisy marginal measurements, representing it as a probabilistic graphical model. Finally, generate synthetic data that closely matches the estimated distribution. Please refer to Algorithm 3 for the pseudocode for generating synthetic data using Private-PGM.

Algorithm 3 f_{PPGM} Generating Synthetic Data using Private PGM (McKenna et al., 2021)

```
Input: Real dataset $D_r \in \mathbb{R}^{n\times(m+1)}$, marginals queries $Q$, noise scale $\sigma$
Output: Synthetic Dataset $D_s$
for $q \in Q$ do
    Measuring marginal: $\mathbf{h}_q = M_q(D_r)$, where $M_q$ is the algorithm for measuring marginals;
    Add noise: $\hat{\mathbf{h}}_q = \mathbf{h}_q + \mathcal{N}(0,\sigma)$;
end for
Generate graphical model $P_\theta$ with weight vector $\theta$: $\mathrm{argmin}_\theta \sum_{q\in Q} \left\| M_q(P_\theta) - \hat{\mathbf{h}}_q \right\|_2^2$ ;
Generate synthetic data $D_s$ using $P_\theta$ using Algorithm 4 and Algorithm 5 (See Below);
```

The algorithm 3 above makes use of the following two subroutines to generate synthetic data from the graphical model: algorithm 4 and algorithm 5.

Algorithm 4 Synthetic data generation

```
Input: graphical model (see Algorithm 3)
Output: dataset (synthetic dataset)
Initialize the set of processed attributes to the empty set;
for each attribute $i$ do
    Let C be the set of all neighbors of $i$ in the graphical model, intersected with the set of processed attributes;
    Group data by C, and
    for each group in C do
        Calculate $\mu$ from the graphical model, the vector of fractional counts for every possible value of attribute $i$, for the
        given group of other attributes;
        Generate synthetic column for this group using Algorithm 5;
        Add this partial column to the grouped rows in the dataset;
    end for
    Add $i$ to the set of processed attributes;
end for
```

Algorithm 5 Synthetic column

```
Input: $\mu$ (vector of fractional counts), $n$ (total number of samples to generate)
Output: column (synthetic column of data)
Generate $\lfloor \mu_t \rfloor$ items with value t and add to column for each t in domain;
Calculate remainders: $p_t = \mu_t - \lfloor \mu_t \rfloor$ ;
Sample $n - \sum_t \lfloor \mu_t \rfloor$ items (without replacement) from distribution proportional to $p_t$, and add to column;
Shuffle values in column;
```

In the Private-PGM approach, differential privacy is achieved by applying a noise mechanism to the marginal measurements. In our experiments, we use the Gaussian mechanism (Dwork & Roth, 2014), as it is reliable and widely used noise mechanisms for enforcing differential privacy. For any single individual's data is altered, it can affect up to two queries by 1 in each marginal measurement. This results in an sensitivity of $\sqrt{2|Q|}$ for all measurements. Invoking Theorem 2.6, adding a Gaussian noise to each query with variance, $\sigma^2 = \frac{2\sqrt{2|Q|}^2 \log(1.25/\delta)}{\epsilon^2}$, we have that the collection of noisy marginals outputted in step *Add noise* of Algorithm 3 achieves (ϵ, δ)-differential privacy. Since the inputs to *Generate graphical model* of Algorithm 3 are differentially private, then the synthetic data finally outputted by Algorithm 3 must also be (ϵ, δ)-differentially private. Any subsequent analyses, including the model training on synthetic data, and further analysis using the trained model, are considered as post-processing. According to Theorem 2.7, these analyses will continue to uphold (ϵ, δ)-DP.

C.2. AIM

The optimal choice of marginal queries/attribute sets to be captured by the synthetic data can be difficult to determine, and can itself leak private information. Therefore, AIM uses an adaptive and iterative algorithm to "automatically" select marginal query that best reduces the distance between the real and synthetic data.

More specifically, AIM allows the user to pre-specify a privacy budget ρ and a collection Q of marginal queries to be selected from. For instance, Q can be the collection of all 3-order marginal queries. The algorithm starts with an initial synthetic data distribution $\hat{\mathcal{D}}_0$. In each iteration $i = 1, 2, \ldots$, it randomly selects a marginal query q_i from Q with probability proportional to q_i's *quality score* that captures the distance between its real marginal and its marginal evaluated from the current estimated synthetic data distribution $\hat{\mathcal{D}}_{i-1}$. This randomness in the selection process ensures differential privacy and the method is formally known as the exponential mechanism (McSherry & Talwar, 2007) in DP literature. Then, AIM uses the Gaussian mechanism to measure the marginal of the selected query, followed by using Private-PGM to estimate data distribution $\hat{\mathcal{D}}_i$ from all noisy marginals measured so far. Finally, to terminate, AIM keeps track of the privacy parameter and the junction tree size corresponding to the selected marginals and makes sure they do not exceed their limits.

To handle composition easily, AIM uses zero-concentrated differential privacy (zCDP) and formally claims the following theorem.

Theorem C.1. *For any $T \geq m$, where T is a user-specified limit on the number of iterations, and $\rho \geq 0$, AIM satisfies ρ-zCDP.*

This can be converted to the standard DP guarantee using the following proposition:

Proposition C.2 (zCDP to DP (Canonne et al., 2020))**.** *If a mechanism M satisfies ρ-zCDP, it also satisfies (ϵ, δ)-differential privacy for all $\epsilon \geq 0$ and $\delta = \min_{\alpha > 1} \frac{\exp((\alpha-1)(\alpha\rho-\epsilon)}{\alpha-1}(1-\frac{1}{\alpha})^{\alpha}$.*

While AIM algorithm may only select a small subset of marginal queries to measure before termination, it provides upper bounds of the ℓ_1 difference on both the selected marginals and non-selected marginals in Q. The former can be easily derived as the selected marginal are measured with Gaussian noise. For the latter, it utilizes the relation between the last selected marginal query and the remaining non-selected ones. In particular, as the marginal query is selected with probability proportional to the exponential of their marginal distance to the real ones, this provides a way to derive the upper bound on all remaining non-selected marginals. More formally, for a marginal query $q \in Q$, let $n_q = |\Omega_q|$, and w_q be a parameter that specifies the "importance" of q among Q, which is larger if the average intersection size of q with other sets in Q is high). At i-th iteration, let σ_i, ϵ_i be the hyperparameters that AIM automatically selected to determine the amount of noise, and q_i be the marginal query selected at this iteration, and $Q_i \subseteq Q$ is the marginal queries that can be selected from, which only includes marginal queries that can be measured without significantly increase the junction tree size for Private-PGM. AIM paper proves the following theorem:

Theorem C.3 (Confidence Bound for Non-selected Marginal Query)**.** *Let $\Delta_i = \max_{q \in Q_i} w_q$. For all $q \in Q_i$, with probability at least $1 - e^{-\lambda_1^2/2} - e^{-\lambda_2}$:*

$$\|\mathbf{h}_q^{(r)} - M_q(\hat{\mathcal{D}}_{i-1})\|_1 \leq w_q^{-1}(B_q + \lambda_1\sigma_i\sqrt{n_{q_i}} + \lambda_2\frac{2\Delta_i}{\epsilon_i}),$$

where B_q is equal to:

$$w_{q_i}\|M_q(\hat{\mathcal{D}}_{t-1}) - \mathbf{h}_{q_i}^{(r)}\|_1 + \sqrt{2/\pi}\sigma_i(w_q n_q - w_{q_i} n_{q_i}) + \frac{2\Delta_i}{\epsilon_i}\log(|Q_i|)$$

Table 2. Table presenting comprehensive performance results for various evaluation metrics across six datasets, employing varied ϵ, using AIM synthetic data generation. Refer to Figure 3 for a visual representation.

Dataset		Synthetic data with varied epsilon								Real data
		0.25	0.5	0.75	1	1.25	1.5	1.75	2	
Adult	Accuracy	0.832	0.836	0.838	0.84	0.84	0.841	0.842	0.842	0.843
	ROC score	0.879	0.883	0.886	0.887	0.889	0.888	0.889	0.889	0.891
	Empirical Risk	0.358	0.353	0.349	0.347	0.345	0.345	0.344	0.344	0.34
Churn	Accuracy	0.719	0.742	0.748	0.748	0.748	0.751	0.756	0.757	0.766
	ROC score	0.762	0.789	0.801	0.801	0.801	0.804	0.809	0.81	0.826
	Empirical Risk	0.546	0.52	0.506	0.505	0.498	0.501	0.493	0.491	0.47
Compas	Accuracy	0.587	0.645	0.646	0.658	0.663	0.663	0.664	0.666	0.671
	ROC score	0.6	0.68	0.685	0.705	0.709	0.71	0.71	0.71	0.718
	Empirical Risk	0.674	0.638	0.636	0.624	0.625	0.624	0.623	0.622	0.617
Dutch	Accuracy	0.812	0.815	0.817	0.816	0.817	0.817	0.817	0.817	0.817
	ROC score	0.884	0.885	0.887	0.886	0.886	0.886	0.886	0.886	0.886
	Empirical Risk	0.43	0.427	0.424	0.426	0.425	0.426	0.426	0.426	0.427
Heart	Accuracy	0.572	0.675	0.701	0.719	0.749	0.756	0.767	0.78	0.802
	ROC score	0.569	0.728	0.778	0.78	0.82	0.834	0.838	0.853	0.883
	Empirical Risk	0.986	0.699	0.617	0.593	0.536	0.508	0.496	0.484	0.452
Law	Accuracy	0.892	0.894	0.895	0.895	0.895	0.894	0.895	0.896	0.901
	ROC score	0.829	0.854	0.856	0.857	0.859	0.859	0.857	0.858	0.869
	Empirical Risk	0.274	0.259	0.257	0.256	0.254	0.258	0.257	0.255	0.245

In McKenna et al's empirical evaluation, it indicated that the marginal selection approach employed by AIM makes it consistently outperformed all other marginal-preserving mechanisms for preserving statistical properties. In Section D.3, we will show our experimental results that extend this advantage to consistently learning multiple models with different classifiers.

D. Additional Experimental Results

D.1. Performance of synthetic data with various privacy budgets

This section provides Table 2, serving as a supplement to Section 5.5, presenting numerical test results of performance for various evaluation metrics across six datasets, employing varied ϵ, using AIM synthetic data generation. Refer to Figure 3 for a visual representation.

D.2. Comparison with Other DPML Techniques

To appraise the performance of AIM synthetic data in comparison to prevailing DP-ML approaches, we conducted training using two DP-ML methods. The first one is Differentially-Private Stochastic Gradient Descent (DP-SGD) (Abadi et al., 2016), which ensures differential privacy by introducing carefully calibrated noise to the gradients during the training process. Refer to Algorithm 6 for details. The second method, Private Aggregation of Teacher Ensembles (PATE) learning method (Papernot et al., 2016), assumes a slightly different threat model, as we discuss next. The PATE method entails training multiple teacher models on sensitive training data and ensuring differential privacy by introducing noise to the counts of teacher predictions for each subsequent query made. In addition, a *public*, unlabeled training dataset is required, and differentially-private queries to the teachers are used to label the data. Finally, a student model is trained using the newly labeled data, and this student model can then be released as the final DP-ML model. Note that the model that is ultimately released does not preserve the privacy of the unlabeled training dataset. Thus, this mechanism crucially assumes existence of public, unlabeled training data. Therefore, in order to compare against PATE we construct 3 datasets, a private-labeled-training dataset for teacher models, a public-unlabeled-training dataset for student model, and a testing dataset to assess the performance of the student model. We provide additional details below.

Algorithm 6 Differentially-Private Stochastic Gradient Descent (DP-SGD) (Iyengar et al., 2019) Algorithm 2

```
Input: Training dataset: $(X, Y) \in D$, where features $X \in \mathbb{R}^{n \times m}$, labels $Y \in \mathbb{R}^n$, Lipschitz constant: L, privacy parameters: $(\epsilon, \delta)$, number of iterations: T, minibatch size: B, learning rate: $\eta$, gradient norm bound $C$.
Output: Logistic Regression Model with weights $\mathbf{w}$
Initialize weights $\mathbf{w} = \{0\}^m$;
$\sigma^2 = \frac{16L^2 T \log(1/\delta)}{n^2 \epsilon^2}$;
for $t \in T$ do
    Sample $B$ samples uniformly with replacement from $D$: $(x_1, y_1), ..., (x_B, y_B)$;
    Clip gradient: $\hat{\nabla} L(x_i, y_i) = \nabla L(x_i, y_i) / \max(1, \frac{\|\nabla L(x_i, y_i)\|_2}{C})$
    Add noise: $\nabla L_t(\mathbf{w}) = \frac{1}{B} \sum_{i=1}^{B} \hat{\nabla} L(x_i, y_i) + \mathcal{N}(0, \sigma)$;
    Update weights: $\mathbf{w} = \mathbf{w} - \eta \cdot \nabla L_t(\mathbf{w})$;
end for
```

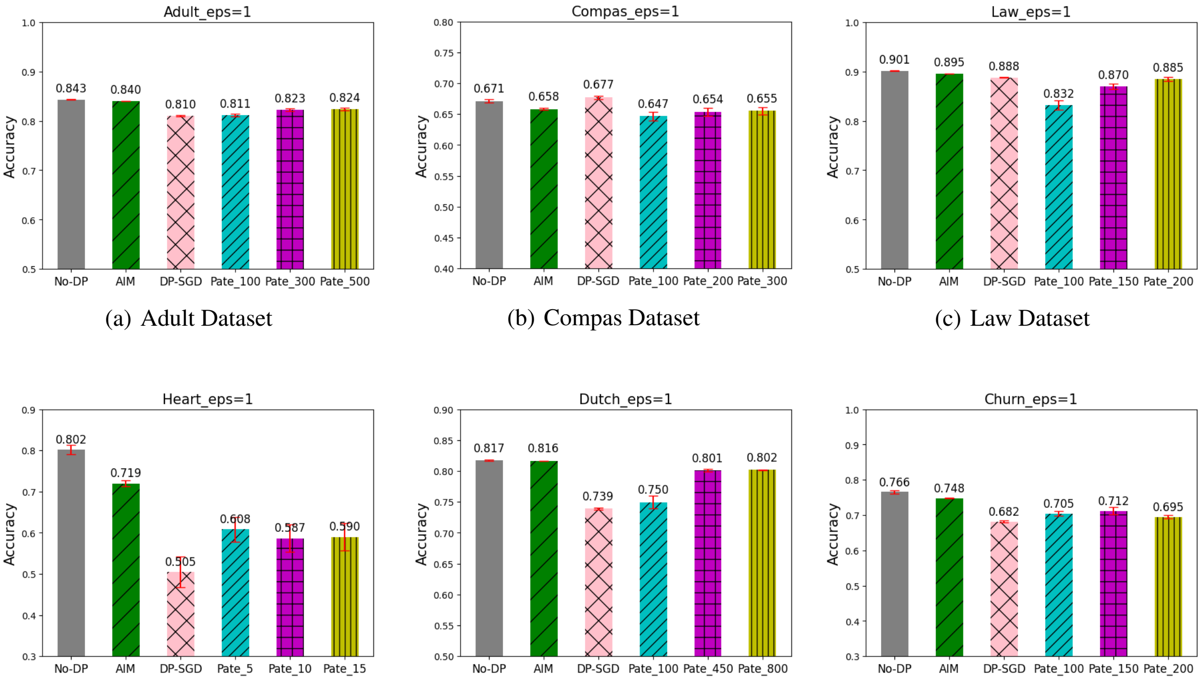
In our experiments, we retained the standard procedure of splitting the real data into 80% training set and 20% testing set, a consistent approach across all three DP-ML methods. For PATE-learning, we additionally sampled 100 data points from the training data (20 data points for the Heart Data, due to its small dataset size), corresponding to the public, unlabeled data, and set those aside for later training of the student model. The teachers models were trained on the remaining training set using the scikit-learn's logistic regression model with the LBFGS solver, the same algorithm used for training the student model. In sum, all three methods end up outputting a DP-ML model, and they all preserve the DP of training data, while PATE has 100 data points less in its training data, and we evaluate performance for all models using the testing data.

Refer to Figure 4 for the detailed parameters setup, which also displays the accuracy comparison among the three methods across six datasets. We note that the AIM and PATE models were trained using second-order methods such as Newton's method, converge faster, as opposed to gradient descent used by DP-SGD. Secondly, we notes that the quality of the model obtained from DP-SGD for some dataset, i.e. Heart and Dutch datasets, is less competitive. We believe it may be possible to further improve the quality of the model outputted by DP-SGD but it would require a considerable amount of effort in tuning its essential hyperparameters, such as learning rate, iterations and decay rate. We further note that such fine-tuning incurs its own privacy leakage resulting from either running multiple differentially-private training runs to set the hyperparameters, or from setting hyperparameters based on *non-private* training runs (Papernot & Steinke, 2021).

In summary, in our experiments, under identical privacy budgets, ϵ, the Pre-DPML approach with AIM-generated synthetic data yielded a model that performs as well as or better than the models generated via the two Training-DPML methods, with the added benefit that with the Pre-DPML approach subsequent training can be performed on the synthetic data without increasing the privacy budget.

D.3. Assess AIM for Different Classifiers

We proposed AIM as the tool to generate synthetic data. Here we would show why select smartly marginal using AIM mechanism is beneficial for generating synthetic data. Figure 5 shows the experiments we conducted on three(3) datasets. For each dataset, we generated synthetic data with $\epsilon = 1$, using AIM that using exponential mechanism to select the most useful marginals. We trained three classification models with two different target labels, $\{y_1, y_2, y_3\}$. The result reveals that, the performance of classifiers trained on real data and AIM data are comparable. This suggests that AIM is effective even without prior knowledge and maintains its performance across various classifier.



(a) Adult Dataset

(b) Compas Dataset

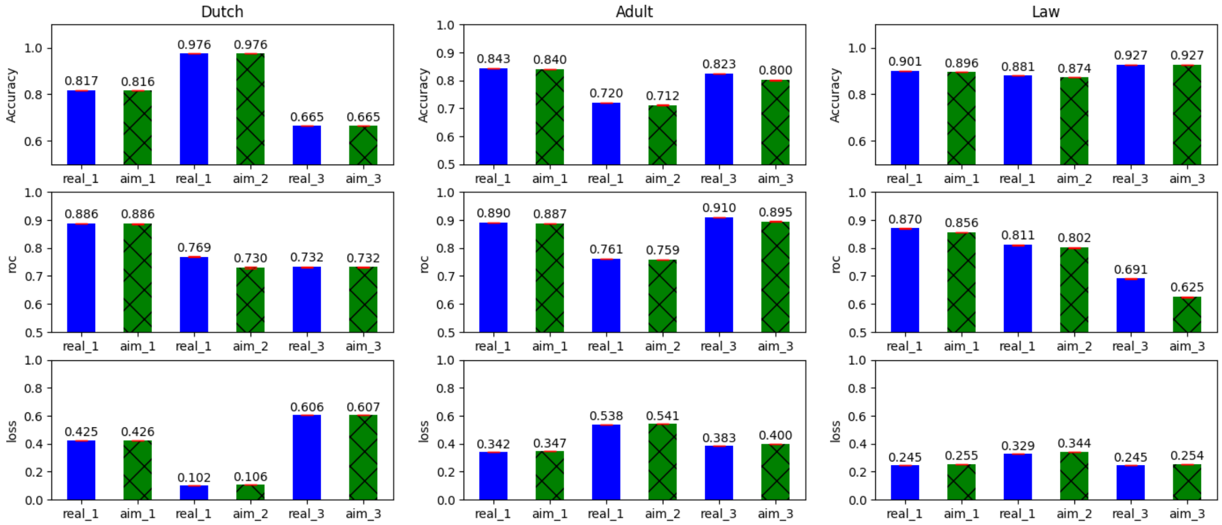
(c) Law Dataset

(d) Heart Dataset

(e) Dutch Dataset

(f) Churn Dataset

Figure 4. We train the six dataset with DP-SGD approach that was described as Algorithm 6, incorporating a gradient norm clipping threshold as 1, and differential privacy budget, epsilon=1. Specifically, we select the learning rate from {1, 5}, running step T from {300, 500, 1000}, decay rate from {0.1, 0.5}, and batch size from{20, 100, 200, 500, 1000, 3000}. Additionally, we train another DP method, PATE-learning, based on (Papernot et al., 2016). For each dataset, we consider three different teacher numbers chosen from {10, 15, 20, 100, 150, 200, 300, 450, 800}. The figure illustrates a comparison of accuracy using various differential privacy methods, which includes Non-DP, AIM (generated DP synthetic data), DP-SGD, PATE learning (with 3 teacher numbers), respectively.



(a) Dutch Dataset

(b) Adult Dataset

(c) Law Dataset

Figure 5. We train the three classifier models on each dataset and their synthetic data generated by AIM with privacy budget, epsilon=1. Dataset {Adult, Churn, Law}, three models are trained to classify three different target features: Dutch: {'occupation', 'prev_residence_place', 'sex'}, Adult: {'income>50K', 'sex', 'relationship}, Law: {'pass_bar', 'race', 'fulltime'}. real_1 and aim_1 show results when classifying the first feature, and trained on real data, synthetic data from AIM, respectively; real_2 and aim_2 show results when classifying the 2nd feature, and trained on real data, synthetic data from AIM, respectively; real_3 and aim_3 show results when classifying the 3rd feature, and trained on real data, synthetic data from AIM, respectively.