Private Regression via Data-Dependent Sufficient Statistic Perturbation

Anonymous authors
Paper under double-blind review

Abstract

Sufficient statistic perturbation (SSP) is a widely used method for differentially private linear regression. SSP adopts a data-independent approach where privacy noise from a simple distribution is added to sufficient statistics. However, sufficient statistics can often be expressed as linear queries and better approximated by data-dependent mechanisms. In this paper we introduce data-dependent SSP for linear regression based on post-processing privately released marginals, and find that it outperforms state-of-the-art data-independent SSP. We extend this result to logistic regression by developing an approximate objective that can be expressed in terms of sufficient statistics, resulting in a novel and highly competitive SSP approach for logistic regression. We also make a connection to synthetic data for machine learning: for models with sufficient statistics, training on synthetic data corresponds to data-dependent SSP, with the overall utility determined by how well the mechanism answers these linear queries.

1 Introduction

Differential privacy (DP) (Dwork et al., 2006) is an established mathematical framework for protecting user privacy while analyzing sensitive data. A differentially private algorithm injects calibrated random noise into the data analytic process to mask the membership of single records in the data, limiting the information revealed about them in the output of the privatized algorithm. The literature encompasses numerous methods for achieving differential privacy across a wide range of machine learning algorithms, including objective perturbation (Chaudhuri and Monteleoni, 2008; Chaudhuri et al., 2011; Kifer et al., 2012; Jain and Thakurta, 2013), with applications to models trained via empirical risk minimization; gradient perturbation (Bassily et al., 2014; Abadi et al., 2016), which is commonly used in deep learning and models trained via gradient descent; one-posterior sampling (Wang et al., 2015; Dimitrakakis et al., 2017) with applications in private Bayesian inference; and finally, sufficient statistic perturbation (`SSP`) (Vu and Slavkovic, 2009; McSherry and Mironov, 2009; Dwork and Smith, 2010; Zhang et al., 2016; Foulds et al., 2016; Wang, 2018; Bernstein and Sheldon, 2019; Ferrando et al., 2022), with natural applications in exponential family estimation and linear regression.

`SSP` adds calibrated random noise to the sufficient statistics of the problem of interest and uses the noisy sufficient statistics downstream to retrieve the target estimate. It is appealing for a number of reasons. Sufficient statistics are by definition an information bottleneck, in that they summarize all the information about the model parameters (Fisher, 1922). For many models, like linear regression and exponential family distributions, their sensitivity is easy to quantify or bound, simplifying the DP analysis. Finally, they can be privatized via simple additive mechanisms, like the Laplace or Gaussian mechanism (Dwork et al., 2014).

Existing `SSP` methods are *data-independent*, meaning they add noise to the sufficient statistics in a way that does not depend on the underlying data distribution. In a different branch of DP research, recent work has shown that *data-dependent* mechanisms are the most effective for query answering and synthetic data Hardt et al. (2012); Gaboardi et al. (2014); Zhang et al. (2017); Aydore et al. (2021); Liu et al. (2021); McKenna et al. (2022).

In this paper, we introduce `DD-SSP`, a data-dependent `SSP` method that leverages private linear query answering to release differentially private (DP) sufficient statistics. Its most immediate application is to linear regression, where finite sufficient statistics exist and, as we demonstrate, can be estimated privately through simple transformations of DP pairwise marginals. Furthermore, we extend the application of `DD-SSP` to models without defined finite sufficient statistics by proposing a novel framework for logistic regression, where approximate sufficient statistics are derived and released in a data-dependent way to train the model by optimizing an approximate loss function.

The proposed framework can be used with virtually any DP query answering algorithm. In this paper, we use `AIM` McKenna et al. (2022) as our primary method, but show the overall method is robust to different choices. The main advantage of `AIM` is its demonstrated accuracy at preserving marginal queries, albeit at the cost of restriction to discrete data (a requirement in `AIM`). We show experimentally that `DD-SSP` outperforms the state-of-the-art data-independent `SSP` method `AdaSSP` (Wang, 2018) for linear regression, and for logistic regression tasks, `DD-SSP` achieves better results than the widely used objective perturbation baseline. We also compare `DD-SSP` with `DP-SGD` Abadi et al. (2016), known to achieve excellent performance when hyperparameters are properly fine-tuned. Our results show that the proposed method is competitive with `DP-SGD` when the privacy cost of hyperparameter tuning is taken into account.

Finally, we elaborate on the significance of our results with respect to the increasingly popular practice of training machine learning models on DP synthetic data. Our results support the observation that for these models training on synthetic data generated by linear-query preserving mechanisms effectively corresponds to a form of data-dependent `SSP`.

2 Background

2.1 Differential privacy

Differential privacy (DP) (Dwork et al., 2006) has become the preferred standard for preserving user privacy in data analysis, and it has been widely adopted by private and governmental organizations. Differential privacy allows many data computations (including statistical summaries and aggregates, and the training of various predictive models) to be performed while provably meeting privacy constraints. The concept of neighboring datasets is integral to differential privacy, which aims to limit the influence of any one individual on the algorithmic output in order to safeguard personal privacy.

Definition 2.1 (Neighboring datasets)**.** Two datasets D and D' are considered neighbors ($D \sim D'$) if D' can be created by adding or deleting a single record from D.

Based on the concept of neighboring datasets, we can define the *sensitivity* of a function:

Definition 2.2 (L_2 sensitivity)**.** Given a vector-valued function of the data $f : \mathcal{D} \to \mathbb{R}^p$, the L_2 sensitivity of f is defined as $\Delta(f) = \max_{D \sim D'} \|f(D) - f(D')\|_2$.

Differential privacy can be achieved via different mechanisms of addition of calibrated random noise, with slightly different definitions. In this paper, we adopt (ϵ, δ)-DP, which can be achieved via the Gaussian Mechanism.

Definition 2.3 ((ϵ, δ)-Differential Privacy)**.** A randomized mechanism $\mathcal{M} : \mathcal{D} \to \mathcal{R}$ satisfies (ϵ, δ)-differential privacy if for any neighbor datasets $D \sim D' \in \mathcal{D}$, and any subset of possible outputs $S \subseteq \mathcal{R}$

$$\Pr[\mathcal{M}(D) \in S] \leq \exp(\epsilon) \Pr[\mathcal{M}(D') \in S] + \delta$$

Definition 2.4 (Gaussian mechanism)**.** Let $f : \mathcal{D} \to \mathbb{R}^p$ be a vector-valued function of the input data. The Gaussian mechanism is given by

$$\mathcal{M}(D) = f(D) + \nu$$

where ν is random noise drawn from $\mathcal{N}(0, \sigma^2 I_p)$ with variance $\sigma^2 = 2\ln(1.25/\delta) \cdot \Delta(f)^2/\epsilon^2$ and $\Delta(f)$ is the L_2-sensitivity of f. That is, the Gaussian mechanism adds i.i.d. Gaussian noise to each entry of $f(D)$ with scale σ dependent on the privacy parameters.

Definition 2.5 (Post-processing property of DP)**.** (Dwork et al., 2014) If $\mathcal{M}(D)$ is (ϵ, δ)-DP, then for any deterministic or randomized function g, $g(\mathcal{M}(D))$ satisfies (ϵ, δ)-DP.

2.2 Differentially private synthetic data

One of the most appealing applications of differential privacy is the creation of synthetic data, which is designed to be representative of the original data while maintaining privacy (Hardt et al., 2012; Zhang et al., 2017; Xie et al., 2018; Jordon et al., 2019; McKenna et al., 2019; Rosenblatt et al., 2020; Vietri et al., 2020; Liu et al., 2021; Aydore et al., 2021; McKenna et al., 2021b; Vietri et al., 2022; McKenna et al., 2022). In this line of work, instead of perturbing the data or data analytic process, a model capturing the data distribution is estimated and used to generate new surrogate data, which can be safely used for the analytic tasks of interest. Research on differentially private synthetic data is ongoing, and two dedicated competitions have been hosted by the U.S. National Institute of Standards and Technology (NIST) NIST 2018; NIST 2020, propelling advances in this area. Private synthetic data has many advantages, including fitting into existing data processing workflows and enabling users to answer multiple queries while maintaining privacy. Depending on the characteristics of the data captured in generative mechanism, the resulting synthetic data will be accurate specifically for tasks where those characteristics are relevant.

The key goal is therefore to tailor synthetic data to perform well on select classes of queries (workloads) that are relevant to the problem of interest. The subject of generative synthetic data for machine learning tasks has recently gained traction in the literature (Tao et al., 2021; Zhou et al., 2024), with the focus so far limited to benchmarking different synthetic data methods and bounding the empirical error it incurs when used for training machine learning models. However, an end-to-end analysis of the characteristics that make DP synthetic data suitable for machine learning tasks is needed. The following definitions allow us to formally elaborate on differentially private synthetic data.

Definition 2.6 (Dataset)**.** A dataset D is defined as a collection of n potentially sensitive records. Each record $\chi^{(i)} \in D$ is a d-dimensional vector $(\chi_1^{(i)}, \ldots, \chi_d^{(i)})$.

Definition 2.7 (Domain)**.** The domain of possible values for attribute $\chi_j^{(i)}$ is $\Omega_j = \{1, \ldots, m_j\}$. The full domain of possible values for $\chi^{(i)}$ is thus $\Omega = \Omega_1 \times \cdots \times \Omega_d$ which has size $\prod_j m_j = m$.

We will later talk about numerical encodings of attributes (Section 3.1).

Definition 2.8 (Marginals)**.** Let $r \subseteq [d]$ be a subset of features, $\Omega_r = \prod_{j \in r} \Omega_j, m_r = |\Omega_r|$, and $\chi_r = (\chi_j)_{j \in r}$. The marginal on r is a vector $\mu_r \in \mathbb{R}^{m_r}$ indexed by domain elements $t \in \Omega_r$ such that each entry is $\mu_r[t] = \sum_{\chi \in D} \mathbb{1}[\chi_r = t]$ (i.e., counts).

With marginal queries, one record can only contribute a count of one to a single cell of the output vector. For this reason, the L_2 sensitivity of a marginal query M_r is 1, regardless of the attributes in r. This facilitates the differential privacy analysis for marginal queries.

Definition 2.9 (Workload)**.** A marginal workload W is defined as a set of marginal queries $r_1, \ldots, r_K$ where $r_k \subseteq [d]$.

The goal of workload-based synthetic data generation models to minimize the approximation error on workload queries.

Marginal-based approaches and *select-measure-reconstruct* methods have generally emerged as effective methods for DP linear query answering and synthetic data generation (Hay et al., 2009; Li et al., 2010; Ding et al., 2011; Xiao et al., 2012; Li and Miklau, 2012; Xu et al., 2013; Yaroslavtsev et al., 2013; Li et al., 2014; Qardaji et al., 2014; Zhang et al., 2014; Li et al., 2015; McKenna et al., 2021a). The *select-measure-reconstruct* paradigm operates by selecting a set of queries to measure using a noise addition mechanism to ensure their privacy. The noisy measurements can be used to estimate a target set of queries, called the *workload*. Many differentially private synthetic data generation strategies use `Private-PGM` (McKenna et al., 2019) to post-process the noise-perturbed marginals and generate a synthetic dataset that respects them. `Private-PGM` can be used in the context of select-measure-reconstruct paradigms. Among these, `AIM` (McKenna et al., 2022) has emerged as state-of-the-art by implementing a few key features described in Appendix A.

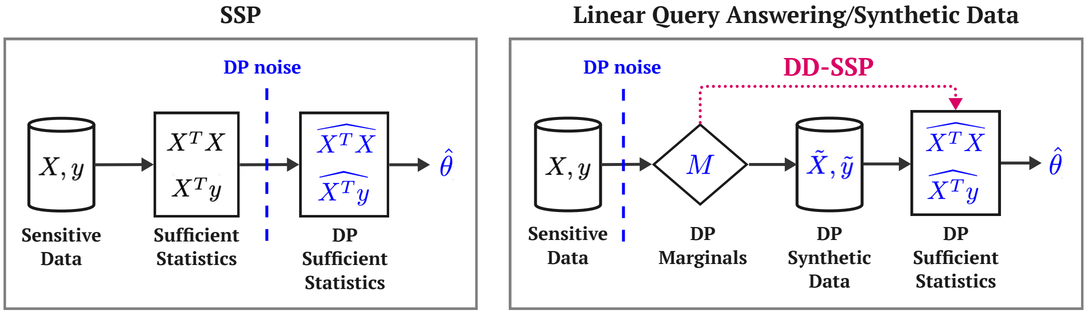

Figure 1: Diagram of the `SSP` data-independent workflow (top) vs the data-dependent linear query answering mechanism for marginal release/synthetic data workflow (bottom). Quantities indicated in blue follow privacy noise injection and are differentially private.

In this work, we use `AIM` as our query-answering algorithm of choice. Note that `AIM` uses zero-concentrated differential privacy (zCDP) Bun and Steinke (2016), an alternative privacy definition; the Gaussian mechanism as defined in Section 2.1 satisfies $\frac{1}{2\sigma^2}$-zCDP Bun and Steinke (2016). In our experiments, we work with (ϵ, δ)-DP and the conversion to zCDP is handled internally in `AIM`.

3 Methods

In this paper, we propose methods to privately estimate sufficient statistics in a data-dependent way. Specifically, our methods leverage privately released marginals computed by a data-dependent linear query answering algorithm to estimate sufficient statistics. We call this set of methods `DD-SSP`. The main advantage of `DD-SSP` is that it is *data-dependent*. Our approach stems from the insight that, while data-independent noise addition is a simple and established approach to `SSP`, many sufficient statistics can be expressed as linear queries, creating an opportunity to improve utility by using data-dependent query-answering DP mechanisms, which often achieve higher accuracy than simple additive noise mechanisms McKenna et al. (2022). In fact, using private synthetic data to train certain models *is already* a form of SSP. Figure 1 compares a standard `SSP` workflow and our proposed `DD-SSP` workflows. As seen in the figure, the pipeline of releasing synthetic data and then training a model via sufficient statistics can be viewed as a specific way of privatizing sufficient statistics for model training.

For linear regression, the application is straightforward: the problem has finite sufficient statistics and we demonstrate that two-way marginal queries are sufficient for their estimation. For other models, finite sufficient statistics are not available, but a polynomial approximation of the loss functions provides approximate sufficient statistics. This is the case for logistic regression, where finite sufficient statistics do not exist, but a Chebyshev polynomial approximation based on Huggins et al. (2017) allows us to propose an *approximate* version of the learning objective based on approximate sufficient statistics that can be expressed as linear queries, again retrievable via two-way marginal tables.

We use the synthetic data mechanism `AIM` as our private query answering algorithm and modify its implementation to output marginals directly, without the need to execute the synthetic data generation step. Depending on the input workload, `AIM` will privately release marginals that preserve certain linear queries more accurately. We find that a *two-way* marginal workload is sufficient for estimating or approximating the sufficient statistics for both linear and logistic regression. The proposed method is amenable to generalization beyond these classes of problems and can be potentially extended to others by i) identifying or approximating their sufficient statistics and ii) customizing the workload passed on as input in `AIM` accordingly. Since our proposed methods are based on post-processing DP workload query answers, the differential privacy analysis is straightforward (Definition 2.5).

3.1 Numerical encoding

We assume discrete (or discretized) input data, which is a common format for tabular data and is required by `AIM` and other marginal-based approaches. However, for machine learning, each record χ must be mapped to a numerical vector $z = (x, y)$ where $x \in \mathbb{R}^p$ is a feature vector and $y \in \mathbb{R}$ is a target. While the details of this encoding are often overlooked, they are important here for two reasons. First, they are needed to tightly bound $\|x\|$, which is used in sensitivity calculations of a number of DP ML methods, with tighter bounds leading to higher utility. Second, the encoding is a key part of recovering sufficient statistics from marginals.

Let $\psi_j(\chi_j) \in \mathbb{R}^{m_j}$ be the one-hot encoding of χ_j, i.e., the vector with entries $\psi_{j,s}(\chi_j) = \mathbf{1}[\chi_j = s]$ for each $s \in \Omega_j$. We consider any numerical encoding of the form $\chi_j \mapsto A_j\psi_j(\chi_j)$ where $A_j \in \mathbb{R}^{p_j \times m_j}$ is a fixed linear transformation applied to the one-hot vector. This covers two special cases of interest. The first is the scalar encoding with $A_j = v_j^T$ for a vector $v_j \in \mathbb{R}^{m_j}$ that specifies the numerical value for each $s \in \Omega_j$. In this case the mapping simplifies to $\chi_j \mapsto v_j(\chi_j)$. The second special case of interest is when $A_j = I_j$ is the $m_j \times m_j$ identity matrix, so the mapping simplifies to $\chi_j \mapsto A_j(\chi_j)$ to give the one-hot encoding itself. Another common variation is a reduced one-hot encoding where $A_j = \tilde{I}_j \in \mathbb{R}^{(m_j-1)\times m_j}$ is equal to I_j with one row dropped to avoid redundant information in the one-hot encoding. We include a simple example of the encoding strategy in Appendix B.

The full encoded record is $z = (z_{[j]})_{j=1}^d$ where $z_{[j]} = A_j\psi_j(\chi_j) \in \mathbb{R}^{p_j}$ is the encoding of the jth attribute and these column vectors are concatenated vertically. A single entry of z is selected as the target variable y leaving a feature vector x of dimension $p := (\sum_{j=1}^d p_j) - 1$. Later, we will also use indexing expressions like $(\cdot)_{[j]}$ and $(\cdot)_{[j,k]}$ to refer to blocks of a vector or matrix corresponding to the encoding of the jth and kth attributes.

Let $z^{(i)} = (x^{(i)}, y^{(i)})$ denote the encoding of record $\chi^{(i)}$ and let $X \in \mathbb{R}^{n\times p}$ be the matrix with ith row equal to $(x^{(i)})^T$ and $y \in \mathbb{R}^n$ be the vector with ith entry equal to $y^{(i)}$. Many DP ML methods require bounds on the magnitude of the encoded data. Let $\|\mathcal{X}\| = \sup_{x\in\mathcal{X}} \|x\|$ and $\|\mathcal{Y}\| = \sup_{y\in\mathcal{Y}} |y|$ be bounds provided by the user where $\mathcal{X} \subset \mathbb{R}^p$ and $\mathcal{Y} \subset \mathbb{R}$ are guaranteed to contain all possible encoded feature vectors x and target values y, respectively. For example, a typical bound is $\|\mathcal{X}\| = \|x^+\|$ where $x_k^+ \geq \sup |x_k|$ bounds the magnitude of a single feature. If x_k is the scalar encoding of χ_j we can take $x_k^+ = \max_{s\in\Omega_j} |v_j(s)|$. The following proposition describes how to tightly bound a feature vector that combines scalar features and one-hot encoded features.

Proposition 3.1. *Suppose $x = (u, w)$ where $u \in \mathbb{R}^a$ satisfies $\|u\| \leq \|\mathcal{U}\|$ and $w \in \mathbb{R}^b$ contains the one-hot encodings (either reduced or not reduced) of c attributes. Then $\|\mathcal{X}\| := \sqrt{\|\mathcal{U}\|^2 + c}$ is an upper bound on $\|x\|$.*

Proof. $\|x\|^2 = \|u\|^2 + \|w\|^2 \leq \|\mathcal{U}\|^2 + c$ where $\|w\|^2 \leq c$ because w is the concatenation of c vectors each with at most a single entry of 1 and all other entries equal to 0. ☐

Suppose u consists of scalar features and $\|\mathcal{U}\|$ is obtained by bounding each one separately as described above. This bound is tighter than the naive one of $\sqrt{\|\mathcal{U}\|^2 + b}$ that would be obtained by bounding each entry of the one-hot vectors separately.

3.2 Linear regression

The goal of linear regression is to minimize the sum of squared differences between the observed values y and predicted values $X\theta$ in a linear model with $\theta \in \mathbb{R}^p$. The ordinary least squares (OLS) estimator is obtained by minimizing the squared error loss function $\|y - X\theta\|^2$. Mathematically, the OLS estimator is given by $\hat{\theta} = (X^TX)^{-1}X^Ty$. In this context, the sufficient statistics are $T(X, y) = \{X^TX, X^Ty\}$. In `DD-SSP`, we approximate $T(X, y)$ using linear queries. Specifically, we show that each entry of X^TX and X^Ty can be obtained from pairwise marginals. The sufficient statistics we will consider all have the form of empirical second moments of the encoded attributes.

3.2.1 Sufficient Statistics from Pairwise Marginals

Let $Z = [X, y] \in \mathbb{R}^{n\times(p+1)}$. The matrix $Z^T Z$ has blocks that contain our sufficient statistics of interest:

$$Z^T Z = \begin{bmatrix} X^T X & X^T y \\ y^T X & y^T y \end{bmatrix}. \tag{1}$$

However, we will see that we can also construct $Z^T Z$ directly from marginals.

Proposition 3.2. *Let $(Z^T Z)_{[j,k]}$ be the block of $Z^T Z$ with rows corresponding to the jth attribute encoding $z_{[j]} = A_j \psi_j(\chi_j)$ and columns corresponding to the kth attribute encoding $z_{[k]} = A_k \psi_k(\chi_k)$. Then*

$$(Z^T Z)_{[j,k]} = A_j \langle \mu_{j,k} \rangle A_k^T$$

where $\langle \mu_{j,k} \rangle \in \mathbb{R}^{m_j \times m_k}$ is the (j,k)-marginal shaped as a matrix with (s,t) entry $\mu_{j,k}[s,t] = \sum_{i=1}^n \mathbf{1}[\chi_j^{(i)} = s, \chi_k^{(i)} = t]$. Note that according to this definition $\langle \mu_{j,j} \rangle = diag(\mu_j)$.

This shows that we can reconstruct the sufficient statistic matrix $Z^T Z$ directly from the set of all single-attribute and pairwise marginals. Note that single-attribute marginals μ_j can be constructed from any $\mu_{j,k}$ with $k \neq j$.

Proof. The sufficient statistic matrix can be written as $Z^T Z = \sum_{i=1}^n z^{(i)} (z^{(i)})^T$. Indexing by blocks gives

$$\begin{aligned}
(Z^T Z)_{[j,k]} &= \sum_{i=1}^n z_{[j]}^{(i)} (z_{[k]}^{(i)})^T \\
&= \sum_{i=1}^n A_j \psi_j(\chi_j^{(i)}) \psi_k(\chi_k^{(i)})^T A_k^T \\
&= A_j \Big(\sum_{i=1}^n \psi_j(\chi_j^{(i)}) \psi_k(\chi_k^{(i)})^T \Big) A_k^T \\
&= A_j \langle \mu_{j,k} \rangle A_k^T,
\end{aligned}$$

In the last line, we used that $\psi_j(\chi_j^{(i)}) \psi_k(\chi_k^{(i)})^T$ is a matrix with (s,t) entry equal to $\mathbf{1}[\chi_j^{(i)} = s, \chi_k^{(i)} = t]$, so summing over all i gives the matrix $\langle \mu_{j,k} \rangle$. ☐

Algorithm 1 outlines how to retrieve approximate sufficient statistics $\widetilde{X^T X}$ and $\widetilde{X^T y}$ from marginals privately estimated by `AIM`. This implies we can solve DP linear regression by i) retrieving $\widetilde{X^T X}$ and $\widetilde{X^T y}$ as outlined in Algorithm 1, and ii) finding $\hat{\theta}_{\text{DP}} = \widetilde{X^T X}^{-1} \widetilde{X^T y}$.

Algorithm 1 `DD-SSP`

```
1: $M \leftarrow$ DPQuery($\mathcal{D}, \epsilon, \delta$) is the collection of privately computed pairwise marginal tables $\mu_{j,k}$ for all attribute pairs $(j,k)$, computed by DP query release algorithm of choice DPQuery.
2: $(\widetilde{Z^T Z})_{[j,k]} \leftarrow A_j \langle \mu_{j,k} \rangle A_k^T$ for all attribute pairs $(j,k)$ (see Proposition 3.2)
3: Extract $\widetilde{X^T X}$ and $\widetilde{X^T y}$ from $\widetilde{Z^T Z}$ using the block structure of Equation (1)
```

Proposition 3.3. *`DD-SSP` is (ϵ, δ)-DP.*

The proof follows directly from the (ϵ, δ)-DP properties of the marginal-releasing algorithm (in our case, `AIM`), and the fact that all subsequent steps are post-processing of a DP result (Definition 2.5).

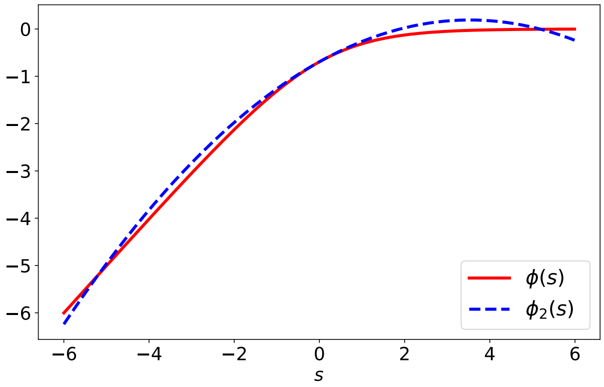

Figure 2: Degree 2 Chebyshev approximation of the logit function ϕ, where $\phi(s) := -\log(1 + e^{-s})$. The inner products $\langle y^{(i)}x^{(i)}, \theta\rangle$ tend to be concentrated in the range $[-4, 4]$ across many datasets (Huggins et al., 2017). We conservatively choose range $[-6, 6]$ based on the inner DP products of the chosen datasets.

3.3 Logistic regression

Logistic regression predicts the probability a binary label $y \in \{-1, +1\}$ takes value $+1$ as $p = 1/(1+\exp(-x\cdot\theta))$, where $\theta \in \mathbb{R}^p$ is a coefficient vector and $x \cdot \theta$ is the dot-product. The log-likelihood function is

$$\ell(\theta) = \sum_{i=1}^{n} \phi(x^{(i)} \cdot \theta y^{(i)})$$

where $\phi(s) := -\log(1 + e^{-s})$. Optimizing this log-likelihood is a convex optimization problem solvable numerically via standard optimizers. The log-likelihood does not have *finite* sufficient statistics. However, Huggins et al. (2017) offers a polynomial strategy to obtain *approximate* sufficient statistics for generalized linear models (GLMs), including logistic regression. Kulkarni et al. (2021) used a similar polynomial approximation for private Bayesian GLMs.

We propose a novel DP logistic regression method that combines two ideas: i) we use a Chebyshev approximation of the logistic regression log-likelihood based on Huggins et al. (2017), which allows us to write the objective in terms of approximate sufficient statistics, and then ii) use `AIM` privately released marginals to estimate the approximate sufficient statistics without accessing the sensitive data. This gives us the option to directly optimize an approximate log-likelihood based on privatized linear queries computed by `AIM`. The choice of the input workload for `AIM` depends on the characterization of the approximate log-likelihood. Based on our derivation below, we find that a suitable workload input for logistic regression is all pairwise marginals.

Huggins et al. (2017) propose to approximate $\ell(\theta)$ by using an degree-M polynomial approximation of the function ϕ:

$$\phi(s) \approx \phi_M(s) := \sum_{m=0}^{M} {b_m}^{(M)} s^m$$

where $b_j^{(M)}$ are constants. There are different choices for the orthogonal polynomial basis, and as in Huggins et al. (2017), we focus on Chebyshev polynomials, which provide uniform quality guarantees over a finite interval $[-R, R]$ for positive R (Figure 2). We can then write

$$\ell(\theta) \approx \sum_{i=1}^{n} \sum_{m=0}^{M} b_m^{(M)} (x^{(i)} \cdot \theta y^{(i)})^m$$

Based on the distribution of noisy inner products obtained through objective perturbation (Figure 4, Appendix C), we choose to work with a degree-2 Chebyshev approximation over the range $[-6, 6]$, leading to a precise approximation over a range that encompasses most of the observed inner product values.

Proposition 3.4. *The logistic regression log-likelihood is approximated by second order Chebyshev polynomial* $\tilde{\ell}(\theta) \approx nb_0^{(2)} + b_1^{(2)}\theta\widetilde{X^Ty} + b_2^{(2)} \cdot (\theta^T\widetilde{X^TX}\theta)$*, where* $b_0^{(2)}, b_1^{(2)}, b_2^{(2)}$ *are constants, and* $\widetilde{X^TX}$ *and* $\widetilde{X^Ty}$ *can be retrieved from pairwise marginals as in Proposition 3.2.*

The proof is provided in Appendix C. This allows us to define a logistic regression objective where $\widetilde{X^TX}$ and $\widetilde{X^Ty}$ are obtained via Algorithm 1, which is DP by post-processing (Proposition 3.3). The objective can then be optimized directly via standard (non-private) procedures.

3.4 Connections to synthetic data

Based on the insight that sufficient statistics (or their approximations) can be expressed as linear queries, the proposed framework highlights a novel connection between differentially private synthetic data generation and `SSP` (Figure 2.1). Query answering algorithms are often used in synthetic data generation procedures. Many such procedures follow the select-measure-reconstruct approach to synthetic data, where linear queries are privately estimated from noisy measurements, and integrated into a model from which synthetic data can be sampled. This process ensures that the output synthetic data supports the selected linear queries. The synthetic data can then be used downstream to compute any statistic of interest, or in the example of linear regression, we could fit the model by training on the synthetic data. This workflow differs from `DD-SSP` only in that it consolidates the noisy measurements into a model and samples synthetic data from it.

By training select-measure-reconstruct synthetic generative models to preserve the appropriate workload of queries, synthetic data can therefore be "tuned" for specific machine learning tasks; for example, based on the findings in 3.2 and 3.3, we expect synthetic data that preserves pairwise marginals to perform well on linear and logistic regression, as it implicitly computes the relevant sufficient statistics (or approximate sufficient statistics). These findings provide a new perspective that enhances the understanding and utility of private synthetic data, especially as related to synthetic data for machine learning tasks.

3.5 Generality of the approach

The proposed approach is amenable to generalization on two fronts: i) the flexibility with respect to the choice of private linear query answering method, and ii) the applicability to a variety of models.

Query-answering methods are often used in synthetic data generation algorithms. Virtually any private query answering algorithm, such as Vietri et al. (2020); McKenna et al. (2021b); Aydore et al. (2021), could be adopted in the context of `DD-SSP`. To give a concrete example, we run supplementary experiments using `MST` (McKenna et al., 2021b) instead of `AIM` within our framework (Appendix G). `Private-PGM` can also be replaced with different methods for model estimation from the private measurements. In Appendix G, we show results for experiments where `AIM` is combined with the mixture inference step from `RAP` Aydore et al. (2021).

The `DD-SSP` approach can be extended to any model where sufficient statistics can be linked to marginal queries either directly (as we demonstrate for linear regression in 3.2) or via a polynomial approximation (as in logistic regression, 3.3). More detailed extensions of the method are discussed in Section 5.

3.6 Baselines

We compare `DD-SSP` to synthetic data method `AIM-Synth`, where private measurements are used to estimate the underlying distribution, and surrogate data is sampled from it. We train `AIM-Synth` with an input workload of all pairwise marginals to match the workload utilized for `DD-SSP`. Since marginal-based synthetic data is designed to preserve the same linear queries that are sufficient to solve linear regression, or approximately sufficient for logistic regression, the expectation is that `DD-SSP` will closely match the performance of `AIM-Synth`. The difference between these approaches is whether linear queries for sufficient statistics are computed directly from marginals estimated by the mechanism, or computed from synthetic data (Figure 1).

We also compare both methods against established DP baselines. Since our methods are based on privately reconstructing sufficient statistics, for DP linear regression sufficient statistic perturbation (`SSP`) is the natural baseline choice. We choose `AdaSSP` (Wang, 2018) for its competitive performance. `AdaSSP` uses limited data-adaptivity to add a ridge penalty based on an estimated bound on the eigenvalues of $X^T X$, but then adds independent noise to each entry of the sufficient statistics, unlike fully data-adaptive query-answering mechanisms. The `AdaSSP` algorithm is detailed in Appendix D. For logistic regression, objective perturbation (`ObjPert`) is a widely adopted solution originally proposed by Chaudhuri et al. (2011), and further refined by Kifer et al. (2012) where it is extended to (ϵ, δ)-DP, with more general applicability and improved guarantees. Algorithm and details are provided in Appendix E. Both `AdaSSP` and `ObjPert` determine how much noise to add based on $\|\mathcal{X}\|$, which is the upper bound to the L_2-norm of any row of X (Section 3.1). For example, in `AdaSSP`, $X^T X$ is noise-perturbed proportionally to $\|\mathcal{X}\|^2$. From Proposition 3.1, we can set $\|\mathcal{X}\|^2 = \|\mathcal{U}\|^2 + c$ where $\|\mathcal{U}\|$ is a bound on the numerically-encoded features and c is the number of one-hot-encoded attributes to obtain a tight sensitivity bound for these baselines.

In addition to our proposed methods, we compare against `DP-SGD` Abadi et al. (2016), a widely adopted algorithm for differentially private training. `DP-SGD` is highly sensitive to the choice of hyperparameters and requires extensive tuning to achieve optimal performance. To accurately account for the privacy loss incurred during hyperparameter tuning, advanced privacy accounting techniques must be applied Ponomareva et al. (2023). To illustrate the variability in `DP-SGD`'s performance, our plots depict a shaded region representing the range between two versions of `DP-SGD`: an "optimistic" baseline that disregards the privacy cost of hyperparameter tuning, artificially inflating performance, and a more realistic version that incorporates this cost using advanced composition Steinke (2022). The effective performance of `DP-SGD` in practical scenarios, including with more advanced privacy accounting Papernot and Steinke (2021), is expected to lie within this range.

3.7 Limitations

When choosing `AIM` as the mechanism for DP marginals, we are limited to working with discrete data, which is a requirement in `AIM` itself. Thus, our comparisons to other regression methods are scoped to discrete numerical data. Future work may consider `DD-SSP` with other mechanisms that support continuous data without discretization. For the logistic regression approximation, we use Chebyshev second-order polynomials; other approximation functions and/or degrees of precision could be evaluated. As shown in section F, `DD-SSP` demonstrates overall gains in regression accuracy compared to baseline methods, however this improvement comes at the cost of increased computational time tied to the complexity involved in privately releasing private marginals for large domains with high utility. This cost can be significantly reduced by replacing `AIM` with faster marginal-releasing methods, which we demonstrate in Appendix G. More detail on the computational runtime is discussed in Appendix F, including a few accessible strategies to mitigate the computational costs in real-world applications.

4 Experiments

In our experiments, we evaluate the effectiveness of `DD-SSP` for linear and logistic regression against `AdaSSP`, `ObjPert`, and `DP-SGD`. `AIM` is used for privately releasing the marginals needed in `DD-SSP`. Additionally, we assess the similarity between the performance of `DD-SSP` and that of `AIM` synthetic data (`AIM-Synth`), suggesting that data-dependent estimation of sufficient statistics explains why marginal-based synthetic data performs well for machine learning tasks.

We compare the Mean Squared Error (MSE) of DP query-based methods `DD-SSP` and `AIM-Synth` against the DP baseline `AdaSSP` and the public baseline for $\epsilon \in \{0.05, 0.1, 0.5, 1.0, 2.0\}$, with a fixed $\delta = 10^{-5}$. Figure 3 shows that `DD-SSP` and `AIM-Synth` have nearly identical performance and both improve significantly upon `AdaSSP` on all datasets except ACSIncome, where performance is similar. For logistic regression, `DD-SSP` closely matches `AIM-Synth` and surpasses `ObjPert` in low-ϵ regimes while being competitive at higher ϵ values. Assessing `DD-SSP` vs. `DP-SGD` is more challenging due to `DP-SGD`'s reliance on hyperparameter fine-tuning. We represent this variability with a shaded region spanning the case where `DP-SGD` accounts for the privacy cost of hyperparameter tuning via advanced composition Steinke (2022), and the case where this cost is ignored. Our results show that `DD-SSP` is competitive with `DP-SGD` under advanced composition, with performance

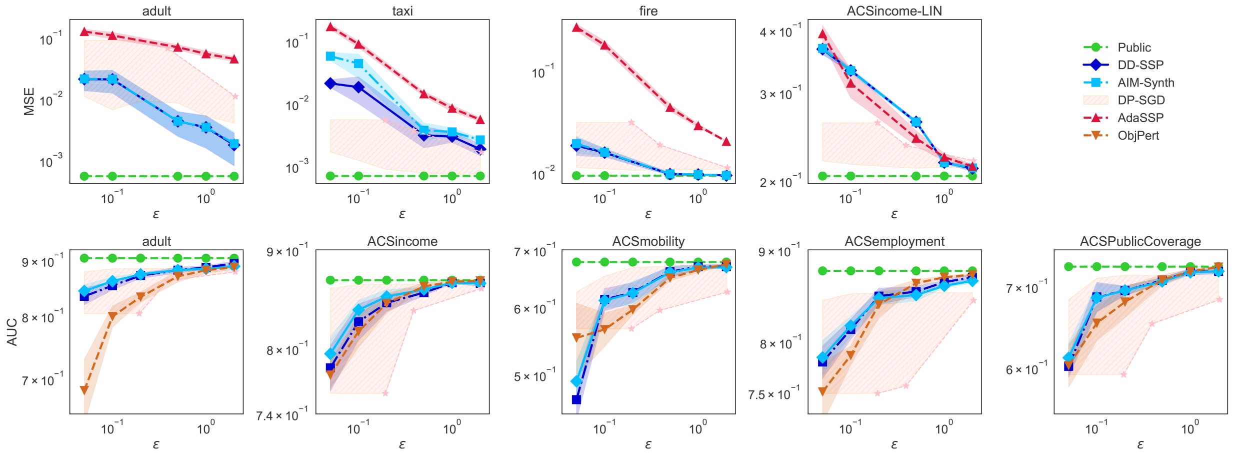

Figure 3: Linear regression MSE results (first row) and logistic regression AUC results (second row). Standard error bars are computed over 5 trials. For `DP-SGD`, the shaded region spans the range between an optimistic baseline ignoring hyperparameter tuning cost, and a realistic baseline using advanced composition to account for this cost (this line is highlighted by star markers). The advanced composition curve starts further to the right as the lowest ϵ achievable by `DP-SGD` in these cases is 0.005.

varying across datasets. Additional details on experimental setup, datasets, and implementation can be found in Appendix F.

Based on these results, we observe the following:

- `DD-SSP` is a competitive option for DP linear and logistic regression, surpassing data-independent `SSP` and `ObjPert` baselines in specific cases, and being competitive with `DP-SGD` overall.
- The performance of `AIM-Synth` suggests that estimating problem-specific data-dependent sufficient statistics explains the suitability of `AIM` synthetic data for machine learning tasks. This implies `DD-SSP` is effective whenever pairwise marginals are available.
- The approximate `DD-SSP` method for logistic regression constitutes a novel DP algorithm as an alternative to privatized ERM procedures.

5 Future work

We imagine extensions to two sets of models. iBeyond linear regression, all exponential family distributions, including graphical models like Naive Bayes, have finite sufficient statistics, and for such models we can devise similar `DD-SSP` solutions, or tailor synthetic data, by identifying a workload that supports the estimation of their sufficient statistics. Additionally, future work can focus on developing approximate loss functions with finite sufficient statistics for a broader class of other models, including generalized linear models (GLMs), where our results for logistic regression can be extended to obtain approximate sufficient statistics from k-way marginals by making a k-degree polynomial approximation to the GLM mapping function following the reasoning of Huggins et al. (2017). This will open the door to novel `DD-SSP` methods that directly minimize the approximate loss functions, and improve utility by adding privacy noise in a data-dependent way. Additionally, methods targeting encoded workload $A_j \langle \mu_{j,k} \rangle A_k^T$ instead of $\mu_{j,k}$ can be explored and combined with advanced data-independent mechanisms like the matrix mechanism, potentially leading to a new class of encoding-aware `SSP` methods.

6 Conclusions

We introduce methods for data-dependent sufficient statistic perturbation (`DD-SSP`). Our methods use privately released marginal tables to solve linear and logistic regression via sufficient statistics. We find that `DD-SSP` performs better than data-independent SSP on linear regression and objective perturbation for logistic regression, and is competitive with `DP-SGD`, known to achieve excellent results under fine-tuned hyperparameter setting.

Notably, the approximate `DD-SSP` logistic regression algorithm is the first DP logistic regression method that allows analysts to solve logistic regression via a SSP algorithm, directly minimizing the approximate loss function. Additionally, we find that the performance of `DD-SSP` is almost indistinguishable from that of `AIM` synthetic data: this suggests that with the appropriate workload, training these machine learning models on query-based DP synthetic data corresponds to data-dependent SSP.

References

M. Abadi, A. Chu, I. Goodfellow, H. B. McMahan, I. Mironov, K. Talwar, and L. Zhang. Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*, pages 308–318, 2016.

S. Aydore, W. Brown, M. Kearns, K. Kenthapadi, L. Melis, A. Roth, and A. A. Siva. Differentially private query release through adaptive projection. In *International Conference on Machine Learning*, pages 457–467. PMLR, 2021.

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

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

G. Bernstein and D. R. Sheldon. Differentially private bayesian linear regression. *Advances in Neural Information Processing Systems*, 32, 2019.

M. Bun and T. Steinke. Concentrated differential privacy: Simplifications, extensions, and lower bounds. In *Theory of Cryptography Conference*, pages 635–658. Springer, 2016.

K. Chaudhuri and C. Monteleoni. Privacy-preserving logistic regression. *Advances in neural information processing systems*, 21, 2008.

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

C. Dimitrakakis, B. Nelson, Z. Zhang, A. Mitrokotsa, and B. I. Rubinstein. Differential privacy for bayesian inference through posterior sampling. *Journal of machine learning research*, 18(11):1–39, 2017.

B. Ding, M. Winslett, J. Han, and Z. Li. Differentially private data cubes: optimizing noise sources and consistency. In *Proceedings of the 2011 ACM SIGMOD International Conference on Management of data*, pages 217–228, 2011.

F. Ding, M. Hardt, J. Miller, and L. Schmidt. Retiring adult: New datasets for fair machine learning. *Advances in neural information processing systems*, 34:6478–6490, 2021.

C. Dwork and A. Smith. Differential privacy for statistics: What we know and what we want to learn. *Journal of Privacy and Confidentiality*, 1(2), 2010.

C. Dwork, F. McSherry, K. Nissim, and A. Smith. Calibrating noise to sensitivity in private data analysis. In *Theory of Cryptography: Third Theory of Cryptography Conference, TCC 2006, New York, NY, USA, March 4-7, 2006. Proceedings 3*, pages 265–284. Springer, 2006.

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

C. Ferrando, S. Wang, and D. Sheldon. Parametric bootstrap for differentially private confidence intervals. In *International Conference on Artificial Intelligence and Statistics*, pages 1598–1618. PMLR, 2022.

R. A. Fisher. On the mathematical foundations of theoretical statistics. *Philosophical transactions of the Royal Society of London. Series A, containing papers of a mathematical or physical character*, 222(594-604): 309–368, 1922.

J. Foulds, J. Geumlek, M. Welling, and K. Chaudhuri. On the theory and practice of privacy-preserving bayesian data analysis. *arXiv preprint arXiv:1603.07294*, 2016.

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

R. M. Gower and F. Bach. Properties and examples of convexity and smoothness, February 2019. URL `https://gowerrobert.github.io/pdf/M2_statistique_optimisation/exe_convexity_smoothness_solution.pdf`.

L. Grégoire, C. Task, I. Slavitt, G. Nicolas, K. Bhagat, D. Streat, and G. Howarth. Sdnist: Benchmark data and evaluation tools for data sythesizers, Dec. 2021.

M. Hardt, K. Ligett, and F. McSherry. A simple and practical algorithm for differentially private data release. *Advances in neural information processing systems*, 25, 2012.

M. Hay, V. Rastogi, G. Miklau, and D. Suciu. Boosting the accuracy of differentially-private histograms through consistency. *arXiv preprint arXiv:0904.0942*, 2009.

J. Huggins, R. P. Adams, and T. Broderick. Pass-glm: polynomial approximate sufficient statistics for scalable bayesian glm inference. *Advances in Neural Information Processing Systems*, 30, 2017.

P. Jain and A. Thakurta. Differentially private learning with kernels. In *International conference on machine learning*, pages 118–126. PMLR, 2013.

J. Jordon, J. Yoon, and M. Van Der Schaar. Pate-gan: Generating synthetic data with differential privacy guarantees. In *International conference on learning representations*, 2019.

D. Kifer, A. Smith, and A. Thakurta. Private convex empirical risk minimization and high-dimensional regression. In *Conference on Learning Theory*, pages 25–1. JMLR Workshop and Conference Proceedings, 2012.

T. Kulkarni, J. Jälkö, A. Koskela, S. Kaski, and A. Honkela. Differentially private bayesian inference for generalized linear models. In *International Conference on Machine Learning*, pages 5838–5849. PMLR, 2021.

C. Li and G. Miklau. An adaptive mechanism for accurate query answering under differential privacy. *arXiv preprint arXiv:1202.3807*, 2012.

C. Li, M. Hay, V. Rastogi, G. Miklau, and A. McGregor. Optimizing linear counting queries under differential privacy. In *Proceedings of the twenty-ninth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, pages 123–134, 2010.

C. Li, M. Hay, G. Miklau, and Y. Wang. A data-and workload-aware algorithm for range queries under differential privacy. *arXiv preprint arXiv:1410.0265*, 2014.

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

T. Liu, G. Vietri, and S. Z. Wu. Iterative methods for private synthetic data: Unifying framework and new methods. *Advances in Neural Information Processing Systems*, 34:690–702, 2021.

R. McKenna, D. Sheldon, and G. Miklau. Graphical-model based estimation and inference for differential privacy. In *International Conference on Machine Learning*, pages 4435–4444. PMLR, 2019.

R. McKenna, G. Miklau, M. Hay, and A. Machanavajjhala. Hdmm: Optimizing error of high-dimensional statistical queries under differential privacy. *arXiv preprint arXiv:2106.12118*, 2021a.

R. McKenna, G. Miklau, and D. Sheldon. Winning the nist contest: A scalable and general approach to differentially private synthetic data. *arXiv preprint arXiv:2108.04978*, 2021b.

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

F. McSherry and I. Mironov. Differentially private recommender systems: Building privacy into the netflix prize contenders. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 627–636, 2009.

NIST 2018. Differential privacy synthetic data challenge. `https://www.nist.gov/ctl/pscr/open-innovation-prize-challenges/past-prize-challenges/2018-differential-privacy-synthetic`, 2018.

NIST 2020. Differential privacy temporal map challenge. `https://www.nist.gov/ctl/pscr/open-innovation-prize-challenges/past-prize-challenges/2020-differential-privacy-temporal`, 2020.

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

N. Ponomareva, H. Hazimeh, A. Kurakin, Z. Xu, C. Denison, H. B. McMahan, S. Vassilvitskii, S. Chien, and A. G. Thakurta. How to dp-fy ml: A practical guide to machine learning with differential privacy. *Journal of Artificial Intelligence Research*, 77:1113–1201, 2023.

W. Qardaji, W. Yang, and N. Li. Priview: practical differentially private release of marginal contingency tables. In *Proceedings of the 2014 ACM SIGMOD international conference on Management of data*, pages 1435–1446, 2014.

D. Ridgeway, M. F. Theofanos, T. W. Manley, and C. Task. Challenge design and lessons learned from the 2018 differential privacy challenges. *NIST technical note*, 2151, 2021.

L. Rosenblatt, X. Liu, S. Pouyanfar, E. de Leon, A. Desai, and J. Allen. Differentially private synthetic data: Applied evaluations and enhancements. *arXiv preprint arXiv:2011.05537*, 2020.

O. Sheffet. Differentially private ordinary least squares. In *International Conference on Machine Learning*, pages 3105–3114. PMLR, 2017.

T. Steinke. Composition of differential privacy & privacy amplification by subsampling. *arXiv preprint arXiv:2210.00597*, 2022.

Y. Tao, R. McKenna, M. Hay, A. Machanavajjhala, and G. Miklau. Benchmarking differentially private synthetic data generation algorithms. *arXiv preprint arXiv:2112.09238*, 2021.

G. Vietri, G. Tian, M. Bun, T. Steinke, and S. Wu. New oracle-efficient algorithms for private synthetic data release. In *International Conference on Machine Learning*, pages 9765–9774. PMLR, 2020.

G. Vietri, C. Archambeau, S. Aydore, W. Brown, M. Kearns, A. Roth, A. Siva, S. Tang, and Z. S. Wu. Private synthetic data for multitask learning and marginal queries. *arXiv preprint arXiv:2209.07400*, 2022.

D. Vu and A. Slavkovic. Differential privacy for clinical trial data: Preliminary evaluations. In *2009 IEEE International Conference on Data Mining Workshops*, pages 138–143. IEEE, 2009.

Y.-X. Wang. Revisiting differentially private linear regression: optimal and adaptive prediction & estimation in unbounded domain. *arXiv preprint arXiv:1803.02596*, 2018.

Y.-X. Wang, S. Fienberg, and A. Smola. Privacy for free: Posterior sampling and stochastic gradient monte carlo. In *International Conference on Machine Learning*, pages 2493–2502. PMLR, 2015.

Y. Xiao, L. Xiong, L. Fan, and S. Goryczka. Dpcube: Differentially private histogram release through multidimensional partitioning. *arXiv preprint arXiv:1202.5358*, 2012.

L. Xie, K. Lin, S. Wang, F. Wang, and J. Zhou. Differentially private generative adversarial network. *arXiv preprint arXiv:1802.06739*, 2018.

J. Xu, Z. Zhang, X. Xiao, Y. Yang, G. Yu, and M. Winslett. Differentially private histogram publication. *The VLDB journal*, 22:797–822, 2013.

G. Yaroslavtsev, G. Cormode, C. M. Procopiuc, and D. Srivastava. Accurate and efficient private release of datacubes and contingency tables. In *2013 IEEE 29th International Conference on Data Engineering (ICDE)*, pages 745–756. IEEE, 2013.

J. Zhang, G. Cormode, C. M. Procopiuc, D. Srivastava, and X. Xiao. Privbayes: Private data release via bayesian networks. *ACM Transactions on Database Systems (TODS)*, 42(4):1–41, 2017.

X. Zhang, R. Chen, J. Xu, X. Meng, and Y. Xie. Towards accurate histogram publication under differential privacy. In *Proceedings of the 2014 SIAM international conference on data mining*, pages 587–595. SIAM, 2014.

Z. Zhang, B. Rubinstein, and C. Dimitrakakis. On the differential privacy of bayesian inference. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 30,1, 2016.

Y. Zhou, M. Liang, I. Brugere, D. Dachman-Soled, D. Dervovic, A. Polychroniadou, and M. Wu. Bounding the excess risk for linear models trained on marginal-preserving, differentially-private, synthetic data. *arXiv preprint arXiv:2402.04375*, 2024.

Appendix

A AIM

For `AIM`, we follow the original algorithm in McKenna et al. (2022). `AIM` uses an intelligent initialization step to estimate one-way marginals. This results in a model where all one-way marginals are preserved well, and higher-order marginals can be estimated under an independence assumption. Additionally, `AIM` uses a carefully chosen subset of the marginal queries and leverages the observation that lower-dimensional marginals exhibit a better signal-to-noise ratio than marginals with many attributes and low counts, and at the same time they can be used to estimate higher-dimensional marginal queries in the workload. Finally, the quality score function for selecting marginals to measure ensures that the selection is "budget-adaptive", i.e. it measures larger dimensional marginals only when the available privacy budget is large enough.

Algorithm 2 AIM (McKenna et al., 2022)

```
Input: Dataset $\mathcal{D}$, workload $W$, privacy parameter $\rho$
Output: Synthetic Dataset $\tilde{\mathcal{D}}$
Hyper-Parameters: MAX-SIZE=80MB, $T = 16d$, $\alpha = 0.9$
$\sigma_0 = \sqrt{T/(2\,\alpha\,\rho)}$
$\rho_{used} = 0$
$t = 0$
Initialize $\hat{p}_t$ (using Algorithm 3)
$w_r = \sum_{s \in W} c_s \mid r \cap s \mid$
$\sigma_{t+1} \leftarrow \sigma_0 \quad \epsilon_{t+1} \leftarrow \sqrt{8(1-\alpha)\rho/T}$
while $\rho_{used} < \rho$ do
    $t = t + 1$
    $\rho_{used} \leftarrow \rho_{used} + \frac{1}{8}\epsilon_t^2 + \frac{1}{2\sigma_t^2}$
    $C_t = r_t \in W_+ \mid \text{JunctionTree-SIZE}(r_1, \dots, r_t)) \leq \frac{\rho_{used}}{\rho} \cdot \text{MAX-SIZE}$
    select $r_t \in C_t$ using the exponential mechanism with:
        $q_r(\mathcal{D}) = w_r\Big( \|M_r(\mathcal{D}) - M_r(\hat{p}_{t-1})\|_1 - \sqrt{2/\pi} \cdot \sigma_t \cdot n_r \Big)$
    measure marginal on $r_t$:
        $\tilde{y}_t = M_{r_t}(\mathcal{D}) + \mathcal{N}(0, \sigma_t^2 I)$
    estimate data distribution using Private-PGM:
        $\hat{p}_t = \underset{p \in S}{\arg\min} \sum_{i=1}^{t} \frac{1}{\sigma_i} \|M_{r_i}(p) - \tilde{y}_i\|_2^2$
    anneal $\epsilon_{t+1}$ and $\sigma_{t+1}$ using Algorithm 4
end while
generate synthetic data $\tilde{\mathcal{D}}$ from $\hat{p}_t$ using Private-PGM
return $\tilde{\mathcal{D}}$
```

Algorithm 3 Initialize p_t (Subroutine of Algorithm 2) (McKenna et al., 2022)

```
for $r \in \{r \in W_+ \mid |r| = 1\}$ do
    $t \leftarrow t + 1$
    $\sigma_t \leftarrow \sigma_0$
    $r_t \leftarrow r$
    $\tilde{y}_t \leftarrow M_r(\mathcal{D}) + \mathcal{N}(0, \sigma_t^2 \mathbb{I})$
    $\rho_{\text{used}} \leftarrow \rho_{\text{used}} + \frac{1}{2\sigma_t^2}$
end for
$\hat{p_t} \leftarrow \text{argmin}_{p \in S} \sum_{i=1}^{t} \frac{1}{\sigma_i} \|M_{r_i}(p) - \tilde{y}_i\|_2^2$
```

Algorithm 4 Budget Annealing (Subroutine of Algorithm 2) (McKenna et al., 2022)

```
if $\|M_{r_t}(\hat{p}_t) - M_{r_t}(\hat{p}_{t-1})\|_1 \leq \sqrt{\frac{2}{\pi}} \cdot \sigma_t \cdot n_{r_t}$ then
    $\epsilon_{t+1} \leftarrow 2 \cdot \epsilon_t$
    $\sigma_{t+1} \leftarrow \sigma_t / 2$
else
    $\epsilon_{t+1} \leftarrow \epsilon_t$
    $\sigma_{t+1} \leftarrow \sigma_t$
end if
if $(\rho - \rho_{\text{used}}) \leq 2\left(\frac{1}{2\sigma_{t+1}^2} + \frac{1}{8}\epsilon_{t+1}^2\right)$ then
    $\epsilon_{t+1} \leftarrow \sqrt{8 \cdot (1 - \alpha) \cdot (\rho - \rho_{\text{used}})}$
    $\sigma_{t+1} \leftarrow \sqrt{\frac{1}{2 \cdot \alpha \cdot (\rho - \rho_{\text{used}})}}$
end if
```

B Encoding example

The following example demonstrates the application of the encoding strategy in 3.1. The mapping is:

$$\chi_j \mapsto A_j \psi_j(\chi_j)$$

For a concrete example, suppose there are 5 levels (i.e., $m_j = 5$) and the feature value is $\chi_j = 3$, then the one-hot encoding vector is

$$\psi_j(\chi_j) = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \end{bmatrix}$$

For the numerical encoding we use $A_j = v_j^T$. For example, suppose $v_j^T = [1, 2, 4, 8, 16]$, then in our example we have

$$\chi_j \mapsto v_j^T \psi_j(\chi_j) = [1, 2, 4, 8, 16] \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \end{bmatrix} = 4$$

It is easy to see that the numerical value is always equal to $v_j[\chi_j]$, i.e., value in vector v_j at index χ_j. In other words, the vector v_j enumerates the numerical values for each level. For the one-hot encoding case, we use $A_j = I_j$, the identity matrix. In our example, this gives

$$\chi_j \mapsto I_{m_j} \psi_j(\chi_j) = \psi_j(\chi_j) = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \end{bmatrix}$$

It is clear that this gives the one-hot encoding. The reduced one-hot encoding is similar.

C Logistic Regression log-likelihood approximation

Proof. The log-likelihood for logistic regression can be expressed as

$$\ell(\theta) = \sum_{i=1}^{n} \phi(x^{(i)} \cdot \theta y^{(i)})$$

where $\phi(s) := -\log(1 + e^{-s})$. Based on Huggins et al. (2017), we can approximate the logistic regression log-likelihood with a Chebyshev polynomial approximation of degree M:

$$\phi(s) \approx \phi_M(s) := \sum_{m=0}^{M} {b_m}^{(M)} s^m$$

where $b_j^{(M)}$ are constants. Then,

$$\ell(\theta) \approx \sum_{i=1}^{n} \sum_{m=0}^{M} b_m^{(M)} (x^{(i)} \cdot \theta y^{(i)})^m$$

If we choose $M = 2$,

$$\ell(\theta) \approx \sum_{i=1}^{n} b_0^{(2)} + b_1^{(2)} \cdot (x^{(i)} \cdot \theta y^{(i)}) + b_2^{(2)} \cdot (x^{(i)} \cdot \theta y^{(i)})^2$$

The quadratic term is

$$\sum_{i=1}^{n} (x^{(i)} \cdot \theta y^{(i)})^2 = \sum_{i=1}^{n} {y^{(i)}}^2 \sum_{j,k=1}^{d} x_j^{(i)} x_k^{(i)} \theta_j \theta_k = \sum_{j,k=1}^{d} \theta_j \theta_k \sum_{i=1}^{n} {y^{(i)}}^2 x_j^{(i)} x_k^{(i)}$$

Therefore we can rewrite the approximate log-likelihood as

$$\begin{aligned}
\ell(\theta) &\approx n b_0^{(2)} + b_1^{(2)} \sum_{i=1}^{n} x^{(i)} \cdot \theta y^{(i)} + b_2^{(2)} \sum_{j,k=1}^{d} \theta_j \theta_k \sum_{i=1}^{n} {y^{(i)}}^2 x_j^{(i)} x_k^{(i)} \\
&\approx n b_0^{(2)} + b_1^{(2)} y^T X \theta + b_2^{(2)} \sum_{j,k=1}^{d} \theta_j \theta_k \sum_{i=1}^{n} x_j^{(i)} x_k^{(i)} \\
&\approx n b_0^{(2)} + b_1^{(2)} y^T X \theta + b_2^{(2)} \theta^T X^T X \theta
\end{aligned}$$

where the simplification in the last line follows from the fact that we work with $y^{(i)} \in \{1, -1\}$ and ${y^{(i)}}^2 = 1$ for any i.

$X^T X$ in the third term and $y^T X$ can be derived following the same proposed strategies as in linear regression (Section 3.2), obtaining marginal query-based estimates $\widetilde{X^T X}$ and $\widetilde{y^T X}$. We can then express the log-likelihood as

$$\ell(\theta) \approx n b_0^{(2)} + b_1^{(2)} \widetilde{y^T X} \theta + b_2^{(2)} \cdot (\theta^T \widetilde{X^T X} \theta)$$

□

Figure 4 shows the distribution of the inner DP products for the datasets used in the experiments, supporting the choice of $[-6, 6]$ for the range of the Chebyshev approximation.

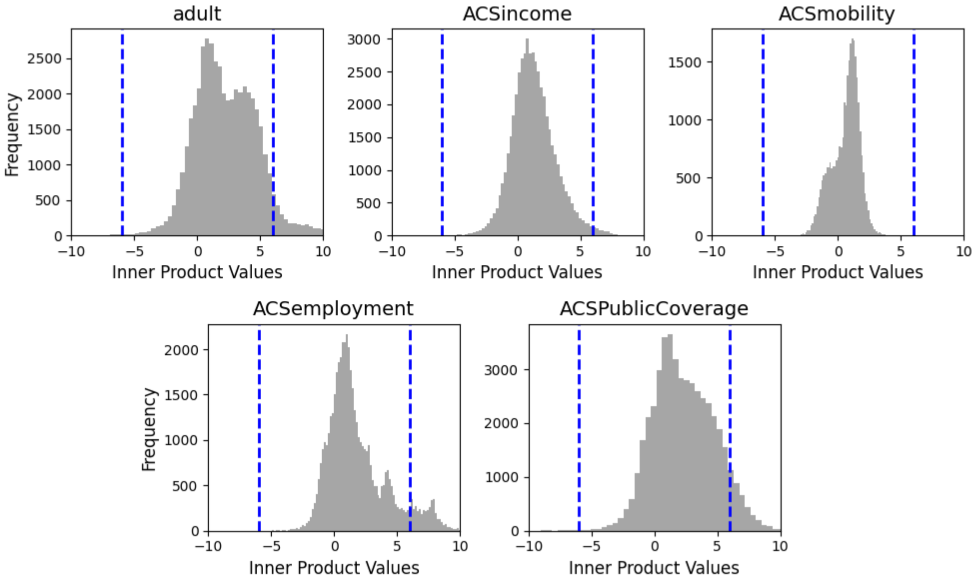

Figure 4: Distribution of the inner DP products $\langle y^{(i)}x^{(i)}, \hat{\theta}_{DP}\rangle$ for the datasets of interest. $\hat{\theta}_{DP}$ is computed using the objective perturbation method with $\epsilon = 1$ and $\delta = 10^{-5}$. The dashed vertical lines represenent the [-6, 6] bounds chosen for the Chebyshev approximation.

Algorithm 5 AdaSSP (Wang, 2018)

```
Input: Data $X$, $\boldsymbol{y}$. Privacy budget: $\epsilon$, $\delta$. Bounds: $\|\mathcal{X}\|$, $\|\mathcal{Y}\|$, $\rho \in (0,1)$ (0.05 in the paper)
1. Calculate the minimum eigenvalue $\lambda_{\min}\left(X^TX\right)$.
2. Privately release $\tilde{\lambda}_{\min} = \max\left\{\lambda_{\min} + \frac{\sqrt{\log(6/\delta)}}{\epsilon/3}\|\mathcal{X}\|^2 Z - \frac{\log(6/\delta)}{\epsilon/3}\|\mathcal{X}\|^2, 0\right\}$, where $Z \sim \mathcal{N}(0,1)$.
3. Set $\lambda = \max\left\{0, \frac{\sqrt{d\log(6/\delta)\log(2d^2/\rho)}\|\mathcal{X}\|^2}{\epsilon/3} - \tilde{\lambda}_{\min}\right\}$.
4. Privately release $\widehat{X^TX} = X^TX + \frac{\sqrt{\log(6/\delta)}\|\mathcal{X}\|^2}{\epsilon/3}Z$ for $Z \in \mathbb{R}^{d\times d}$ is a symmetric matrix and every
   element from the upper triangular matrix is sampled from $\mathcal{N}(0,1)$.
5. Privately release $\widehat{X^Ty} = X^Ty + \frac{\sqrt{\log(6/\delta)}\|\mathcal{X}\|\|\mathcal{Y}\|}{\epsilon/3}Z$ for $Z \sim \mathcal{N}\left(0, I_d\right)$.
Output: $\tilde{\theta} = \left(\widehat{X^TX} + \lambda I\right)^{-1}\widehat{X^Ty}$.
```

D Linear regression baseline

Algorithm D outlines the `AdaSSP` method for linear regression Wang (2018).

To reason about the sensitivity of X^TX, consider two neighboring datasets $X \in \mathbb{R}^{n\times d}$ and $X' \in \mathbb{R}^{(n+1)\times d}$ differing by one data entry $v \in \mathcal{X}$, where v is a $d \times 1$ vector. Then,

$$\Delta_{X^TX} = \sup_{X\sim X'}\|f(X') - f(X)\|_F$$

Since X and X' only differ by one row (v), then $f(X') - f(X) = vv^T$ (Sheffet, 2017).

So the sensitivity is maximum over v of $\|\text{vec}(vv^T)\| = \|vv^T\|_F$. We have

$$\begin{aligned}
\Delta^2_{X^TX} &= \sup_{v\in\mathcal{X}} \|vv^T\|_F^2 \\
&= \sup_{v\in\mathcal{X}} \sum_{i=1}^{d}\sum_{j=1}^{d}(v_iv_j)^2 \\
&= \sup_{v\in\mathcal{X}} \left(\sum_{i=1}^{d} v_i^2\right)\left(\sum_{j=1}^{d} v_j^2\right) \\
&= \sup_{v\in\mathcal{X}} \|v\|^4 \\
&= \|\mathcal{X}\|^4
\end{aligned}$$

where $\|\mathcal{X}\|$ is the greatest possible norm of a vector in the domain $\mathcal{X}$. Therefore,

$$\Delta_{X^TX} = \|\mathcal{X}\|^2.$$

The sensitivity of X^Ty can be similarly derived. Given neighboring datasets $X \in \mathbb{R}^{n\times d}, y \in \mathbb{R}^n$, and $X' \in \mathbb{R}^{(n+1)\times d}, y' \in \mathbb{R}^{n+1}$, where $v \in \mathcal{X} \subset \mathbb{R}^d$ is the new row, and $w \in \mathcal{Y} \subset \mathbb{R}$ is the new value in y'. Then,

$$\|f(X', y') - f(X, y)\| = \|X'^Ty' - X^Ty\| = \|wv\|$$

Since $\|\mathcal{X}\| = \sup_{x\in\mathcal{X}} \|x\|$ and $\|\mathcal{Y}\| = \sup_{y\in\mathcal{Y}} |y|$, we have

$$\begin{aligned}
\Delta_{X^Ty} &= \sup_{(X,y)\sim(X',y')} \|f(X', y') - f(X, y)\| \\
&= \sup_{w\in\mathcal{Y}, v\in\mathcal{X}} |w|\cdot\|v\| = \|\mathcal{Y}\|\cdot\|\mathcal{X}\|
\end{aligned}$$

E Logistic regression baseline

Our DP logistic regression baseline is based on the generalized objective perturbation algorithm in Kifer et al. (2012) (Algorithm E). In this section, to match the notation in Kifer et al. (2012), $\ell(\theta; z) = \log(1+\exp(-x\cdot\theta y))$ is the loss for a single datum and $\hat{\mathcal{L}}(\theta; \mathcal{D}) = \frac{1}{n}\sum_{i=1}^{n} \ell(\theta; z^{(i)})$ is the average loss over the dataset.

Algorithm 6 Generalized Objective Perturbation Mechanism (`ObjPert`) (Kifer et al., 2012)

```
Require: dataset $\mathcal{D} = \{z^{(1)}, \dots, z^{(n)}\}$, where $z^{(i)} = (x^{(i)}, y^{(i)})$, privacy parameters $\epsilon$ and $\delta$ ($\delta = 0$ for $\epsilon$-differential privacy), bound $\|\mathcal{X}\|$ on the $L_2$ norm of any $x$ entry, convex regularizer $r$, a convex domain $\mathbb{F} \subseteq \mathbb{R}^d$, convex loss function $\hat{\mathcal{L}}(\theta; \mathcal{D}) = \frac{1}{n}\sum_{i=1}^{n} \ell(\theta; z^{(i)})$, with continuous Hessian, $\|\nabla\ell(\theta; z)\| \le \|\mathcal{X}\|$ (for all $z \in \mathcal{D}$ and $\theta \in \mathbb{F}$), and the eigenvalues of $\nabla^2\ell(\theta; z)$ bounded by $\frac{\|\mathcal{X}\|^2}{4}$ (for all $z$ and for all $\theta \in \mathbb{F}$).
Set $\Delta \ge \frac{\|\mathcal{X}\|^2}{2\epsilon}$.
Sample $b \in \mathbb{R}^d$ from $\nu_2(b; \epsilon, \delta, \|\mathcal{X}\|) = \mathcal{N}\left(0, \frac{\|\mathcal{X}\|^2(8\log\frac{2}{\delta}+4\epsilon)}{\epsilon^2} I_d\right)$.
$\hat{\theta}_{\mathrm{DP}} \equiv \arg\min_{\theta\in\mathbb{F}} \hat{\mathcal{L}}(\theta; \mathcal{D}) + \frac{1}{n}r(\theta) + \frac{\Delta}{2n}\|\theta\|^2 + \frac{b^T\theta}{n}$.
```

The algorithm requires the following bounds for the gradient and Hessian of ℓ:

$$\|\nabla\ell(\theta; z)\| \le \|\mathcal{X}\|$$

$$\lambda_{\max}(\nabla^2\ell(\theta; z)) \le \frac{\|\mathcal{X}\|^2}{4}$$

To reason about the sensitivity bounds, let $\phi(s) = \log(1 + e^s)$. Then we can write $\ell(\theta; z) = \phi(-x \cdot \theta y)$. Following Gower and Bach (2019), it is straightforward to derive that

$$\begin{aligned}
\phi'(s) &= \frac{e^s}{1+e^s} \leq 1 \\
\phi''(s) &= \frac{e^s}{(1+e^s)^2} \leq \frac{1}{4}
\end{aligned}$$

and clear that both quantities are non-negative. Then the gradient of ℓ is:

$$\nabla \ell(\theta; z) = \nabla_\theta \phi(-x \cdot \theta y) = \phi'(-x \cdot \theta y) \cdot -yx$$

The norm is bounded as

$$\|\nabla \ell(\theta; z)\| = |\phi'(-x \cdot \theta y)| \cdot |y| \cdot \|x\| \leq \|\mathcal{X}\|$$

where the inequality holds since $|\phi'(s)| \leq 1$ for all s and $|y| = 1$.

By differentiating the gradient again and using the fact that $y^2 = 1$, we can derive the Hessian as:

$$\nabla^2 \ell(\theta; z) = \phi''(-x \cdot \theta y) x x^T$$

The maximum eigenvalue is

$$\begin{aligned}
\lambda_{\max}(\nabla^2 \ell(\theta; z)) &= \phi''(-x \cdot \theta y) \lambda_{\max}(x x^T) \\
&= \phi''(-x \cdot \theta y) \|x\|^2 \\
&\leq \frac{\|\mathcal{X}\|^2}{4}
\end{aligned}$$

In the second line, we used the fact that $\lambda_{\max}(xx^T) = \|x\|^2$. To see this, note that xx^T is rank one and $(xx^T)x = \|x\|^2 x$, therefore x is an eigenvector with eigenvalue $\|x\|^2$ and this is the largest eigenvalue. In the last line we used that $\|\phi''(s)\| \leq \frac{1}{4}$ for all s and that $\|x\| \leq \|\mathcal{X}\|$.

F Experiment details

F.1 Datasets and Preprocessing

We use the following datasets:

- **Adult** (Becker and Kohavi, 1996): The target variable is 'num-education' (number of education years) for linear regression and 'income>50K' for logistic regression.
- **Fire** (Ridgeway et al., 2021): The target variable is 'Priority' (of the call).
- **Taxi** (Grégoire et al., 2021): The target variable is 'totalamount' (total fare amount).
- **ACS Datasets** (Ding et al., 2021): Data is queried for California (2018). Includes binary classification tasks for 'PINCP' (income above \$50k), 'MIG' (mobility), 'ESR' (employment), and 'PUBCOV' (public coverage). **ACSincome** is also used for linear regression with the target variable 'PINCP' (income) discretized into 20 bins.

More detail on the datasets is included in Table 1. Data is shuffled and split into 1,000 test points and up to 50,000 training points. Non-numerical features are one-hot encoded, dropping the first level to avoid multi-collinearity, and numerical features are rescaled to $[-1, 1]$ for noise allocation (see Section 3.1 for more detail on data encoding).

Table 1: Dataset information

Dataset	**Size**	**# Attributes**	**Attributes**	**Target**
Adult	48,842	15	[‘age’, ‘workclass’, ‘fnlwgt’, ‘education’, ‘marital-status’, ‘occupation’, ‘relationship’, ‘race’, ‘sex’, ‘capital-gain’, ‘capital-loss’, ‘hours-per-week’, ‘native-country’, ‘income>50K’, ‘education-num’]	‘income>50K’ (logistic), ‘education-num’ (linear)
ACSIncome	195,665	9	[‘AGEP’, ‘COW’, ‘SCHL’, ‘MAR’, ‘RELP’, ‘WKHP’, ‘SEX’, ‘RAC1P’, ‘PINCP’]	‘PINCP’
Fire	305,119	15	[‘ALS Unit’, ‘Battalion’, ‘Call Final Disposition’, ‘Call Type’, ‘Call Type Group’, ‘City’, ‘Final Priority’, ‘Fire Prevention District’, ‘Neighborhooods - Analysis Boundaries’, ‘Original Priority’, ‘Station Area’, ‘Supervisor District’, ‘Unit Type’, ‘Zipcode of Incident’, ‘Priority’]	‘Priority’
Taxi	1,048,575	11	[‘VendorID’, ‘passengercount’, ‘tripdistance’, ‘RatecodeID’, ‘PULocationID’, ‘DOLocationID’, ‘paymenttype’, ‘fareamount’, ‘tipamount’, ‘tollsamount’, ‘totalamount’]	‘totalamount’
ACSmobility	29,358	20	[‘AGEP’, ‘SCHL’, ‘MAR’, ‘SEX’, ‘DIS’, ‘CIT’, ‘MIL’, ‘ANC’, ‘WKHP’, ‘NATIVITY’, ‘RELP’, ‘DEAR’, ‘DEYE’, ‘DREM’, ‘RAC1P’, ‘GCL’, ‘COW’, ‘ESR’, ‘JWMNP’, ‘PINCP’]	‘MIG’
ACSEmployment	378,817	17	[‘AGEP’, ‘SCHL’, ‘MAR’, ‘RELP’, ‘DIS’, ‘ESP’, ‘CIT’, ‘MIG’, ‘MIL’, ‘ANC’, ‘NATIVITY’, ‘DEAR’, ‘DEYE’, ‘DREM’, ‘SEX’, ‘RAC1P’, ‘ESR’]	‘ESR’
ACSPublicCoverage	138,550	19	[‘AGEP’, ‘SCHL’, ‘MAR’, ‘SEX’, ‘DIS’, ‘ESP’, ‘CIT’, ‘MIG’, ‘MIL’, ‘ANC’, ‘NATIVITY’, ‘DEAR’, ‘DEYE’, ‘DREM’, ‘PINCP’, ‘ESR’, ‘FER’, ‘RAC1P’, ‘PUBCOV’]	‘PUBCOV’

F.2 Methodology

`AIM` training: `AIM` is trained with a model size of 200MB, a maximum of 1,000 iterations, and a workload of all pairwise marginals. For `AdaSSP`, sensitivity is calibrated as described in Sections 3.1 and 3.6.

`DP-SGD` fine tuning and training: `DP-SGD`'s hyperparameters are fine-tuned by running a gridsearch for the best parameter. The search space spans the following values:

- Batch size: $[n, 1024, 256]$
- Gradient clipping norm: $[0.01, 0.1, 0.2]$
- Number or epochs: $[1, 10, 20]$
- Learning rate: $[0.001, 0.01, 0.1, 1.0]$

Advanced composition is used to account for hyperparameter tuning costs as per Theorem 22 in Steinke (2022). The optimistic baseline ignores this cost entirely.

F.3 Computational runtime

All experiments were conducted on an internal cluster equipped with Xeon Gold 6240 CPUs @ 2.60GHz, 192GB RAM, and 240GB local SSD storage. The runtime of our method is influenced by several factors, including dataset size, domain size, and the privacy parameter ϵ. Since our method relies on private marginals released by `AIM`, its runtime is inherently tied to `AIM`'s computational demands, which are significantly higher than those of the DP baselines.

For clarity, we focus the detailed runtime analysis on the Adult dataset as it is representative of general trends observed across all datasets. On this dataset, `AIM` runtime increases significantly with ϵ. For linear regression, `AIM` requires approximately 8 minutes at $\epsilon = 0.05$, scaling up to 18 hours at $\epsilon = 2.0$. In comparison, the DP baseline `AdaSSP` completes the same experiment in approximately 5 seconds regardless of ϵ. Similar trends are observed for logistic regression, with `AIM` runtime increasing from 12 minutes to 21 hours as ϵ grows, while the corresponding DP baseline `ObjPert` completes these experiments in approximately 2 seconds.

Across other datasets, we observe consistent trends in runtime scaling. For linear regression, `AIM` runtimes range from 50 minutes (ACSincome) to 29 hours (Fire), depending on dataset size and complexity. Similarly, for logistic regression, `AIM` runtime varies from 10 hours (ACSincome) to 43 hours (ACSPublicCoverage). By contrast, all DP baselines except `DP-SGD` complete these experiments in less than 10 seconds for any dataset. For `DP-SGD`, the runtime of the hyperparameter search phase varies across datasets, tasks and ϵ between around 1 hour and 17 hours; once the best hyperparameters have been determined, the runtime of `DP-SGD`, taking the case of linear regression on the Adult dataset as example, varies between 2 minutes for $\epsilon = 0.05$ and 10 minutes for $\epsilon = 2.0$.

F.4 Runtime mitigation strategies

While AIM's runtime is substantial, it is a direct consequence of the complexity involved in accurately computing private marginals for large domains and high utility. A few factors mitigate the runtime of the proposed methods: i) `AIM` is much faster for low ϵ values; ii) `AIM` model size can be reduced to 100MB for a significant runtime cut, without significantly sacrificing the accuracy of the method. In the context of this paper, we choose to prioritize accuracy, which is consistent with the motivation of synthetic data, where computation is spent up front to release a data set that can be used downstreams in many ways; iii) other faster query answering mechanisms can be used in place of `AIM`, such as `MST` (see 3.5), which runs in approximately 18 minutes for a full experiment on Adult. Various ways to improve the accuracy vs running-time trade-offs for use in specific settings can be explored, which is beyond the scope of this paper. In terms of experimentation (see Section 4), for computational viability we study the variability of our results across 5 trials.

G Additional experiments

`DD-SSP` can accommodate different DP marginal query releasing methods. For our main results we use `AIM` (see Appendix A), which uses `Private-PGM` (McKenna et al., 2019) for the “generate” step. In this section, we briefly demonstrate the use of the `DD-SSP` framework with alternative methods. In particular, we consider two alternative settings: i) replacing `AIM` with `MST` (McKenna et al., 2021b); ii) replacing the `Private-PGM` step in `AIM` with `MixtureInference`[1], a close approximation of the relaxed projection methods in (Aydore et al., 2021) and (Liu et al., 2021). Results are shown in Figure 5 and Figure 6 respectively.

From the ablation studies in (McKenna et al., 2022), we expect `MST` to perform well at low ϵ values (high privacy) and `AIM` to outperform it at higher ϵ values. `MST` performs a domain compression operation, and we hypothesize this benefits some cases (e.g. low ϵ), but hurts in others. Based on the same ablation studies, we expect `Private-PGM` to yield lower workload errors with respect to `MixtureInference`, resulting in better overall metric scores for the tasks of interest. Consistently with the findings in (McKenna et al., 2022), we find that `AIM` using `Private-PGM` has the most reliable performance overall.

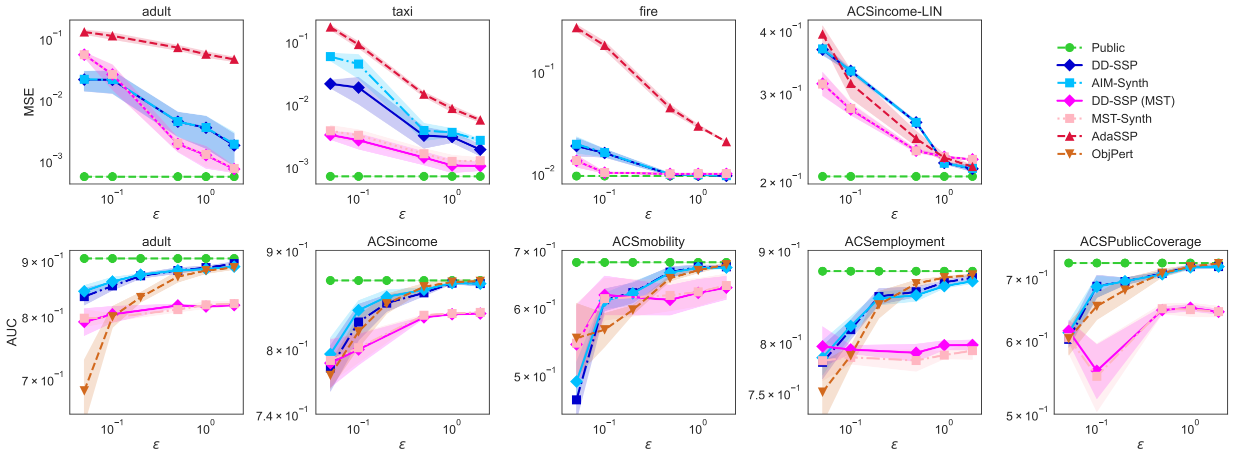

Figure 5: `MST` (McKenna et al., 2021b) vs. `AIM` (McKenna et al., 2022) as a query answering algorithm. Top: linear regression. Bottom: logistic regression. Standard error bars are computed over 5 trials.

---

[1] `https://github.com/ryan112358/private-pgm/blob/master/src/mbi/mixture_inference.py`

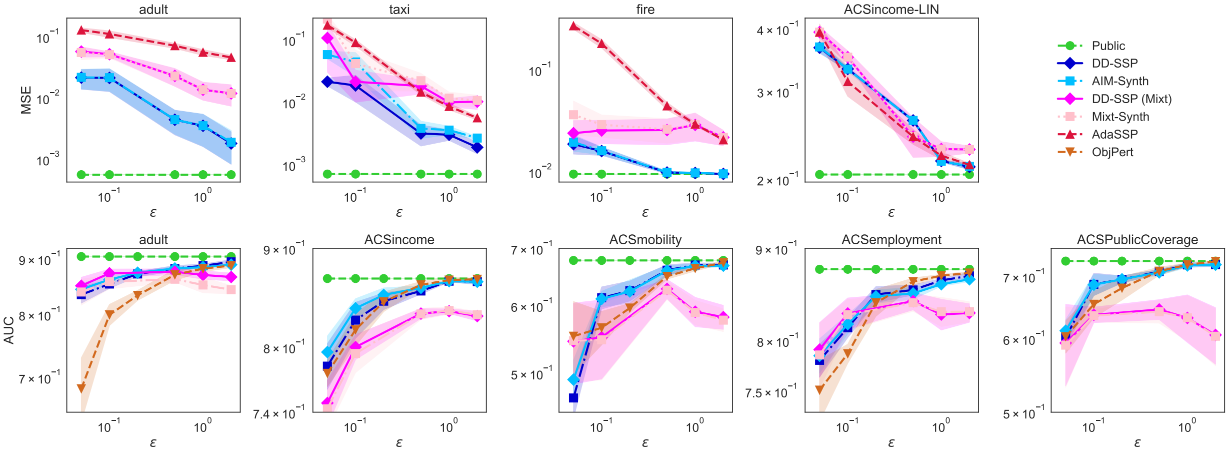

Figure 6: `AIM` using `MixtureInference` (Aydore et al., 2021; Liu et al., 2021) vs. `Private-PGM` (McKenna et al., 2019) for model estimation and synthetic data generation. Top: linear regression. Bottom: logistic regression. Standard error bars computer over 5 trials.