Collecting and Analyzing Multidimensional Data with Local Differential Privacy

Ning Wang1, Xiaokui Xiao2, Yin Yang3, Jun Zhao4, Siu Cheung Hui4, Hyejin Shin5, Junbum Shin5, Ge Yu1,6

1School of Computer Science and Engineering, Northeastern University, China
2School of Computing, National University of Singapore, Singapore
3College of Science and Engineering, Hamad Bin Khalifa University, Qatar
4School of Computer Science and Engineering, Nanyang Technological University, Singapore
5Samsung Research, Samsung Electronics, Korea

Abstract—Local differential privacy (LDP) is a strong privacy standard for collecting and analyzing data, which has been used, e.g., in the Chrome browser, iOS and macOS. In LDP, each user perturbs her information locally, and only sends the randomized version to an aggregator who performs analyses, which protects both the users and the aggregator against private information leaks. Although LDP has attracted much research attention in recent years, the majority of existing work focuses on applying LDP to complex data and/or analysis tasks. In this paper, we point out that the fundamental problem of collecting multidimensional data under LDP has not been addressed sufficiently, and there remains much room for improvement even for basic tasks such as computing the mean value over a single numeric attribute under LDP. Motivated by this, we first propose novel LDP mechanisms for collecting a numeric attribute, whose accuracy is at least no worse (and usually better) than existing solutions in terms of worst-case noise variance. Then, we extend these mechanisms to multidimensional data that can contain both numeric and categorical attributes, where our mechanisms always outperform existing solutions regarding worst-case noise variance. As a case study, we apply our solutions to build an LDP-compliant stochastic gradient descent algorithm (SGD), which powers many important machine learning tasks. Experiments using real datasets confirm the effectiveness of our methods, and their advantages over existing solutions.

Index Terms—Local differential privacy, multidimensional data, stochastic gradient descent.

I. INTRODUCTION

Local differential privacy (LDP), which has been used in well-known software systems such as Google Chrome [18], Apple iOS and macOS [34], and Microsoft Windows Insiders [12], is a strong privacy protection scheme for collecting and analyzing sensitive data from individual users. Specifically, in LDP, each user perturbs her data record locally to satisfy differential privacy [16], and sends only the randomized, differentially private version of the record to an aggregator. The latter then performs computations on the collected noisy dataset to estimate statistical analysis results on the original data. For instance, in [18], Google as an aggregator collects perturbed usage information from users of the Chrome browser, and estimates, for example, the proportion of users running a particular operating system. Compared with traditional privacy standards such as differential privacy in the centralized setting [16], which typically assume a trusted data curator who possesses a set of sensitive records, LDP provides a stronger privacy assurance to users, as the true values of private records never leave their local devices. Meanwhile, LDP also protects the aggregator against potential leakage of users’ private information (which happened to AOL[7] and Netflix[8] with serious consequences), since the aggregator never collects exact private information in the first place. In addition, LDP satisfies the strong and rigorous privacy guarantees of differential privacy; i.e., the adversary (which includes the aggregator in LDP) cannot infer sensitive information of an individual with high confidence, regardless of the adversary’s background knowledge.

Although LDP has attracted much research attention in recent years, the majority of existing solutions focus on applying LDP to complex data types and/or data analysis tasks, as reviewed in Section VII. Notably, the fundamental problem of collecting numeric data has not been addressed sufficiently. Specifically, as we explain in Section III-A, in order to release a numeric value in the range $[-1, 1]$ under LDP, currently the user has only two options: (i) the classic Laplace mechanism [16], which injects unbounded noise to the exact data value, and (ii) a recent proposal by Duchi et al. [14], which releases a perturbed value that always falls outside the original data domain, i.e., $[-1, 1]$. Further, it is non-trivial to extend these methods to handle multidimensional data. As elaborated in Section IV, a straightforward extension of a single-attribute mechanism, using the composition property of differential privacy, leads to suboptimal result accuracy. Meanwhile, the multidimensional version of [14], though asymptotically optimal in terms of worst-case error, is rather complicated and involves a large constant. Finally, to our knowledge, there is no existing solution that can perturb multidimensional data containing both numeric and categorical data with optimal worst-case error.

https://www.wired.com/2009/12/netflix-privacy-lawsuit/
This paper addresses the above challenges and makes several major contributions. First, we propose two novel mechanisms, namely Piecewise Mechanism (PM) and Hybrid Mechanism (HM), for collecting a single numeric attribute under LDP, which obtain higher result accuracy compared to existing methods. In particular, HM is built upon PM, and has a worse-case noise variance that is at least no worse (and usually better) than existing solutions. Then, we extend both PM and HM to multidimensional data with both numeric and categorical attributes with an elegant technique that achieves asymptotic optimal error, while remaining conceptually simple and easy to implement. Further, our fine-grained analysis reveals that although both PM and the proposed methods obtain asymptotically optimal error bound on multidimensional numeric data, the former involves a larger constant than our solutions. Table I summarizes the main theoretical results in this paper, which are confirmed in our experiments.

As a case study, using the proposed novel mechanisms as building blocks, we present an LDP-compliant algorithm for stochastic gradient descent (SGD), which can be applied to train a broad class of machine learning models based on empirical risk minimization, e.g., linear regression, logistic regression and SVM classification. Specifically, SGD iteratively updates the model based on gradients of the objective function, which are collected from individuals under LDP. Experiments using several real datasets confirm the high utility of the proposed methods for various types of data analysis tasks.

In the following, Section II provides the necessary background on LDP. Sections III presents the proposed fundamental mechanisms for collecting a single numeric attribute under LDP, while Section IV describes our solution for collecting and analyzing multidimensional data with both numeric and categorical attributes. Section V applies our solution to common data analytics tasks based on SGD, including linear regression, logistic regression, and support vector machines (SVM) classification. Section VI contains an extensive set of experiments. Section VII reviews related work. Finally, Section VIII concludes the paper.

### Table I: Main theoretical results comparing the proposed mechanisms PM and HM, as well as Duch et al.’s solution [14]. The terms MaxVarPM, MaxVarHM, and MaxVarDu denote the worst-case noise variance of these three methods, respectively, for perturbing a $d$-dimensional numeric tuple under $\epsilon$-local differential privacy (elaborated in Section II). In addition, $\epsilon^* = \ln\left(\frac{5+2\sqrt{6353}+405\sqrt{27}}{2^7} + 2\sqrt{6353}+405\sqrt{27}\right) \approx 0.61$, and $\epsilon^* = \ln\left(\frac{7+4\sqrt{7}+2\sqrt{70+14\sqrt{7}}}{9}\right) \approx 1.29$.

<table>
<thead>
<tr>
<th>Setting</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>$d &gt; 1$</td>
<td>$\epsilon &gt; 0$</td>
</tr>
<tr>
<td>$d = 1$</td>
<td>$\epsilon &gt; \epsilon^*$</td>
</tr>
<tr>
<td>$\epsilon = \epsilon^*$</td>
<td>MaxVarPM &lt; MaxVarPM = MaxVarDu</td>
</tr>
<tr>
<td>$\epsilon^* &lt; \epsilon &lt; \epsilon^*$</td>
<td>MaxVarPM &lt; MaxVarDu &lt; MaxVarPM</td>
</tr>
<tr>
<td>$0 &lt; \epsilon &lt; \epsilon^*$</td>
<td>MaxVarPM = MaxVarDu &lt; MaxVarPM</td>
</tr>
</tbody>
</table>

In the problem setting, an aggregator collects data from a set of $n$ users, and computes statistical models based on the collected data. The goal is to maximize the accuracy of these statistical models, while preserving the privacy of the users. Following the local differential privacy model [5], [14], [18], we assume that the aggregator already knows the identities of the users, but not their private data. Formally, let $n$ be the total number of users, and $u_i (1 \leq i \leq n)$ denote the $i$-th user. Each user $u_i$’s private data is represented by a tuple $t_i$, which contains $d$ attributes $A_1, A_2, \ldots, A_d$. These attributes can be either numeric or categorical. Without loss of generality, we assume that each numeric attribute has a domain $[-1, 1]$, and each categorical attribute with $k$ distinct values has a discrete domain $\{1, 2, \ldots, k\}$.

To protect privacy, each user $u_i$ first perturbs her tuple $t_i$ using a randomized perturbation function $f$. Then, she sends the perturbed data $f(t_i)$ to the aggregator instead of her true data record $t_i$. Given a privacy parameter $\epsilon > 0$ that controls the privacy-utility tradeoff, we require that $f$ satisfies $\epsilon$-local differential privacy ($\epsilon$-LDP) [18], defined as follows:

**Definition 1** ($\epsilon$-local differential privacy). A randomized function $f$ satisfies $\epsilon$-local differential privacy if and only if for any two input tuples $t$ and $t'$ in the domain of $f$, and for any output $f(t^*)$ of $f$, we have:

$$\Pr[f(t) = t^*] \leq \exp(\epsilon) \cdot \Pr[f(t') = t^*].$$ (1)

The notation $\Pr[\cdot]$ means probability. If $f$’s output is continuous, $\Pr[\cdot]$ in (1) is replaced by the probability density function. Basically, local differential privacy is a special case of differential privacy [17] where the random perturbation is performed by the users, not by the aggregator. According to the above definition, the aggregator, who receives the perturbed tuple $t^*$, cannot distinguish whether the true tuple is $t$ or another tuple $t''$ with high confidence (controlled by parameter $\epsilon$), regardless of the background information of the aggregator. This provides plausible deniability to the user [9].

We aim to support two types of analytics tasks under $\epsilon$-LDP: (i) mean value and frequency estimation and (ii) machine learning models based on empirical risk minimization. In the former, for each numerical attribute $A_j$, we aim to estimate the mean value of $A_j$ over all $n$ users, $\frac{1}{n}\sum_{i=1}^{n} t_i[A_j]$. For each categorical attribute $A_j$, we aim to estimate the frequency of each possible value of $A_j$. Note that value frequencies in a categorical attribute $A_j$ can be transformed to mean values once we expand $A_j$ into $k$ binary attributes using one-hot encoding. Regarding empirical risk minimization, we focus on three common analysis tasks: linear regression, logistic regression, and support vector machines (SVM) [11].

Unless otherwise specified, all expectations in this paper are taken over the random choices made by the algorithms considered. We use $\mathbb{E}[\cdot]$ and $\text{Var}[\cdot]$ to denote a random variable’s expected value and variance, respectively.
Algorithm 1: Duchi et al.’s Solution [14] for One-Dimensional Numeric Data.

input : tuple \( t_i \in [-1, 1] \) and privacy parameter \( \epsilon \).
output: tuple \( t_i^* \in \left\{ -\frac{\epsilon^*+1}{\epsilon^*-1}, \frac{\epsilon^*+1}{\epsilon^*-1} \right\} \).

1. Sample a Bernoulli variable \( u \) such that
   \[
   \Pr[u = 1] = \frac{e^\epsilon - 1}{2e^\epsilon + 2} \cdot t_i + \frac{1}{2};
   \]
2. if \( u = 1 \) then
3. \[ t_i^* = \frac{e^\epsilon - 1}{\epsilon^* - 1}; \]
4. else
5. \[ t_i^* = -\frac{e^\epsilon + 1}{\epsilon^* - 1}; \]
6. return \( t_i^* \).

III. COLLECTING A SINGLE NUMERIC ATTRIBUTE

This section focuses on the problem of estimating the mean value of a numeric attribute by collecting data from individuals under \( \epsilon \)-LDP. Section III-A reviews two existing methods, Laplace Mechanism [16] and Duchi et al.’s solution [14], and discusses their deficiencies. Then, Section III-B describes a novel solution, called Piecewise Mechanism (PM), that addresses these deficiencies and usually leads to higher (or at least comparable) accuracy than existing solutions. Section III-C presents our main proposal, called Hybrid Mechanism (HM), whose worst-case result accuracy is no worse than PM and existing methods, and is often better than all of them.

A. Existing Solutions

Laplace mechanism. A classic mechanism for enforcing differential privacy is the Laplace Mechanism [16], which can be applied to the LDP setting as follows. For simplicity, assume that each user \( u_i \)’s data record \( t_i \) contains a single numeric attribute whose value lies in range \([-1, 1]\). In the following, we abuse the notation by using \( t_i \) to denote this attribute value. Then, we define a randomized function that outputs a perturbed record \( t_i^* = t_i + \text{Lap}\left( \frac{2}{\epsilon} \right) \), where \( \text{Lap}(\lambda) \) denotes a random variable that follows a Laplace distribution of scale \( \lambda \), with the following probability density function:
   \[
   pdf(x) = \frac{1}{2\lambda} \exp\left( -\frac{|x|}{\lambda} \right). \]

Clearly, this estimate \( t_i^* \) is unbiased, since the injected Laplace noise \( \text{Lap}\left( \frac{2}{\epsilon} \right) \) in each \( t_i^* \) has zero mean. In addition, the variance in \( t_i^* \) is \( \frac{8}{\epsilon^2} \). Once the aggregator receives all perturbed tuples, it simply computes their average \( \frac{1}{n} \sum_{i=1}^{n} t_i^* \) as an estimate of the mean with error scale \( O\left( \frac{1}{\sqrt{n}} \right) \).

Duchi et al.’s solution. Duchi et al. [14] propose a method to perturb multidimensional numeric tuples under LDP. Algorithm 1 illustrates Duchi et al.’s solution [14] for the one-dimensional case. (We discuss the multidimensional case in Section IV.) In particular, given a tuple \( t_i \in [-1, 1] \), the Duchi et al.’s data record \( t_i^* \) returns either \( \frac{e^\epsilon - 1}{\epsilon^* - 1} \) or \( -\frac{e^\epsilon + 1}{\epsilon^* - 1} \), with the following probabilities:
   \[
   \Pr\left[ t_i^* = x | t_i \right] = \begin{cases} \frac{e^\epsilon - 1}{2e^\epsilon + 2} \cdot t_i + \frac{1}{2}, & \text{if } x = \frac{e^\epsilon - 1}{\epsilon^* - 1}, \\ -\frac{e^\epsilon - 1}{2e^\epsilon + 2} \cdot t_i + \frac{1}{2}, & \text{if } x = -\frac{e^\epsilon + 1}{\epsilon^* - 1}. \end{cases} \tag{2} \]

Duchi et al. prove that \( t_i^* \) is an unbiased estimator of the input value \( t_i \). In addition, the variance of \( t_i^* \) is:
   \[
   \text{Var}(t_i^*) = \mathbb{E}\left[ (t_i^*)^2 \right] - \left( \mathbb{E}[t_i^*] \right)^2 
   = \left( \frac{e^\epsilon + 1}{\epsilon^* - 1} \right)^2 \cdot t_i^* \left( \frac{e^\epsilon - 1}{\epsilon^* - 1} \right) + \left( \frac{e^\epsilon + 1}{\epsilon^* - 1} \right)^2 \cdot t_i^* \left( \frac{e^\epsilon - 1}{\epsilon^* - 1} \right) - t_i^2 
   = \left( \frac{e^\epsilon + 1}{\epsilon^* - 1} \right)^2 - t_i^2. \tag{3} \]

Therefore, the worst-case variance of \( t_i^* \) equals \( \left( \frac{e^\epsilon + 1}{\epsilon^* - 1} \right)^2 \), and it occurs when \( t_i^* = 0 \). Upon receiving the perturbed tuples output by Algorithm 1, the aggregator simply computes the average value of the attribute over all users to obtain an estimated mean.

Deficiencies of existing solutions. Figure 1 illustrates the worst-case variance of the noisy values returned by the Laplace mechanism and Duchi et al.’s solution, when \( \epsilon \) varies. Duchi et al.’s solution offers considerably smaller variance than the Laplace mechanism when \( \epsilon \leq 2 \), but is significantly outperformed by the latter when \( \epsilon \) is large. To explain, recall that Duchi et al.’s solution returns either \( t_i^* = \frac{e^\epsilon - 1}{\epsilon^* - 1} \) or \( t_i^* = -\frac{e^\epsilon + 1}{\epsilon^* - 1} \), even when the input tuple \( t_i = 0 \). As such, the noisy value \( t_i^* \) output by Duchi et al.’s solution always has an absolute value \( \frac{e^\epsilon + 1}{\epsilon^* - 1} > 1 \), due to which \( t_i^* \)’s variance is always larger than 1 when \( t_i = 0 \), regardless of how large the privacy budget \( \epsilon \) is. In contrast, the Laplace mechanism incurs a noise variance of \( 8/\epsilon^2 \), which decreases quadratically with the increase of \( \epsilon \), due to which it is preferable when \( \epsilon \) is large. However, when \( \epsilon \) is small, the relatively “fat” tail of the Laplace distribution leads to a large noise variance, whereas Duchi et al.’s solution does not suffer from this issue since it confines \( t_i^* \) within a relatively small range \( [-\frac{e^\epsilon - 1}{\epsilon^* - 1}, \frac{e^\epsilon + 1}{\epsilon^* - 1}] \).

A natural question is: can we design a perturbation method that combines the advantages of the Laplace mechanism and Duchi et al.’s solution to minimize the variance of \( t_i^* \) across a
Algorithm 2: Piecewise Mechanism for One-Dimensional Numeric Data.

input : tuple \( t_i \in [-1, 1] \) and privacy parameter \( \epsilon \).
output : tuple \( t_i^* \in [-C, C] \).
1 Sample \( x \) uniformly at random from \([0, 1] \);
2 if \( x < \frac{e^{\epsilon/2}}{e^{\epsilon/2} + 1} \) then
3 [ Sample \( t_i^* \) uniformly at random from \([\ell(t_i), r(t_i)] \);
4 else
5 [ Sample \( t_i^* \) uniformly at random from \([-C, \ell(t_i)] \) \( \cup \) \([r(t_i), C] \);
6 return \( t_i^* \).

A wide spectrum of \( \epsilon \)? Intuitively, such a method should confine \( t_i^* \) to a relatively small domain (as Duchi et al.’s solution does), and should allow \( t_i^* \) to be close to \( t_i \) with reasonably large probability (as the Laplace mechanism does). In what follows, we will present a new perturbation method based on this intuition.

B. Piecewise Mechanism

Our first proposal, referred to as the Piecewise Mechanism (PM), takes as input a value \( t_i \in [-1, 1] \), and outputs a perturbed value \( t_i^* \) in \([-C, C] \), where

\[
C = \frac{\exp(\epsilon/2) + 1}{\exp(\epsilon/2) - 1}.
\]

The probability density function (pdf) of \( t_i^* \) is a piecewise constant function as follows:

\[
\text{pdf}(t_i^* = x \mid t_i) = \begin{cases} p, & \text{if } x \in [\ell(t_i), r(t_i)], \\ \frac{p}{\exp(\epsilon)}, & \text{if } x \in [-C, \ell(t_i)] \cup [r(t_i), C]. \end{cases}
\]

where

\[
p = \frac{\exp(\epsilon) - \exp(\epsilon/2)}{2 \exp(\epsilon/2) + 2},
\]

\[
\ell(t_i) = \frac{C + 1}{2} \cdot t_i - \frac{C - 1}{2}, \quad \text{and}
\]

\[
r(t_i) = \ell(t_i) + C - 1.
\]

Let pdf\((t_i^* = x \mid t_i)\) be short for pdf\((t_i^* \mid t_i)\). Fig. 2 illustrates pdf\((t_i^*)\) for the cases of \( t_i = 0, t_i = 0.5, \) and \( t_i = 1 \). Observe that when \( t_i = 0, \) pdf\((t_i^*)\) is symmetric and consists of three “pieces”, among which the center piece (i.e., \( t_i^* \in [\ell(t_i), r(t_i)] \)) has a higher probability than the other two. When \( t_i \) increases from 0 to 1, the length of the center piece remains unchanged (since \( r(t_i) - \ell(t_i) = C - 1 \)), but the length of the rightmost piece (i.e., \( t_i^* \in (r(t_i), C] \)) decreases, and is reduced to 0 when \( t_i = 1 \). The case when \( t_i < 0 \) can be illustrated in a similar manner.

Algorithm 2 shows the pseudo-code of PM. The following lemmas establish the theoretical guarantees of Algorithm 2.

**Lemma 1.** Algorithm 2 satisfies \( \epsilon \)-local differential privacy. In addition, given an input value \( t_i \), it returns a noisy value \( t_i^* \) with \( \mathbb{E}[t_i^*] = t_i \) and

\[
\text{Var}[t_i^*] = \frac{t_i^2}{e^{\epsilon/2} - 1} + \frac{e^{\epsilon/2} + 3}{3(e^{\epsilon/2} - 1)^2}.
\]

**Proof.** By Equation 4 for any \( t^* \in [-C, C] \) and any two input values \( t_i, t_i^* \in [-1, 1] \), we have \( \text{pdf}(t_i^*) \leq \frac{p}{\exp(\epsilon)} = \exp(\epsilon) \). Thus, Algorithm 2 satisfies \( \epsilon \)-LDP. In addition,

\[
\mathbb{E}[t_i^*] = \int_{-C}^{C} \text{pdf}(t_i^*) \cdot t_i^* \cdot \text{pdf}(t_i) \cdot \text{pdf}(t_i^*) \, dt_i^*\]...

This completes the proof.

By Lemma 1, PM returns a noisy value \( t_i^* \) whose variance is at most

\[
\frac{1}{e^{\epsilon/2} - 1} + \frac{e^{\epsilon/2} + 3}{3(e^{\epsilon/2} - 1)^2} = \frac{4e^{\epsilon/2}}{3(e^{\epsilon/2} - 1)^2}.
\]

The purple line in Fig. 1 illustrates this worst-case variance of PM as a function of \( \epsilon \). Observe that PM’s worst-case variance is considerably smaller than that of Duchi et al.’s solution when \( \epsilon \geq 1.5 \), and is only slightly larger than the latter when \( \epsilon < 1.5 \). Furthermore, it can be proved that PM’s worst-case variance is strictly smaller than Laplace mechanism’s, regardless of the value of \( \epsilon \). This makes PM a more preferable choice than both the Laplace mechanism and Duchi et al.’s solution.

Furthermore, Lemma 1 also shows that the variance of \( t_i^* \) in PM monotonically decreases with the decrease of \( |t_i| \), which makes PM particularly effective when the distribution of the input data is skewed towards small-magnitude values. (In Section VI, we show that \( |t_i| \) tends to be small in a large class of applications.) In contrast, Duchi et al.’s solution inverts a noise variance that increases with the decrease of \( |t_i| \), as shown in Equation 3.

Now consider the estimator \( \frac{1}{n} \sum_{i=1}^{n} t_i^* \) used by the aggregator to infer the mean value of all \( t_i \). The variance of this estimator is \( 1/n \) of the average variance of \( t_i^* \). Based on this, the following lemma establishes the accuracy guarantee of \( \frac{1}{n} \sum_{i=1}^{n} t_i^* \).

**Lemma 2.** Let \( Z = \frac{1}{n} \sum_{i=1}^{n} t_i^* \) and \( X = \frac{1}{n} \sum_{i=1}^{n} t_i \). With at least \( 1 - \beta \) probability,

\[
|Z - X| = O \left( \frac{\sqrt{\log(1/\beta)}}{\epsilon n^{1/2}} \right).
\]
Given that $t_i \in [-1, 1]$, the maximum value of the r.h.s. of Equation 7 is:

$$
\max \left\{ \frac{\alpha}{e^{\epsilon/2} - 1} + (1 - \alpha) \left( \frac{\epsilon^3 + 1}{\epsilon^3 - 1} \right)^2 \right\}
$$

It can be proved that

(i) $\left( \frac{\epsilon^3/2 + 3}{3(e^{\epsilon/2} - 1)} - \frac{\epsilon^3 + 1}{\epsilon^3 - 1} \right)^2 > 0$ for $\epsilon > 0$;

(ii) the term $\left( \frac{\epsilon^3/2 + 3}{3(e^{\epsilon/2} - 1)} - \frac{\epsilon^3 + 1}{\epsilon^3 - 1} \right)^2$ in Equation 8 is positive for $0 < \epsilon < \epsilon^*$, negative for $\epsilon > \epsilon^*$, and $0$ for $\epsilon = \epsilon^*$, where $\epsilon^*$ is defined by Equation 5.

Combining (i) and (ii) and Equation 8, we can derive that $\max_{t_i \in [-1, 1]} \sigma_H^2(t_i, \epsilon)$ is minimized when $\alpha$ satisfies Equation 6.

By Lemma 3 when $\alpha$ satisfies Equation 6, the worst-case noise variance of HM is:

$$
\max_{t_i \in [-1, 1]} \sigma_H^2(t_i, \epsilon) = \left( \frac{\epsilon^3 + 1}{\epsilon^3 - 1} \right)^2, \quad \text{for } \epsilon \leq \epsilon^*.
$$

Based on Equation 9, Lemma 1 and Equation 5, which present $\sigma_H^2(t_i, \epsilon)$, $\sigma_P^2(t_i, \epsilon)$, and $\sigma_D^2(t_i, \epsilon)$, respectively, we can show that HM often dominates both PM and Duchi et al.’s solution in minimizing the worst-case noise variance. The detailed results are summarized under $d = 1$ (meaning one dimension) in Table II of Section IV, where $\epsilon^*$ follows from Equation 5 and $\epsilon^#$ is derived by solving $\epsilon$ which makes $\max_{t_i \in [-1, 1]} \sigma_P^2(t_i, \epsilon)$ and $\max_{t_i \in [-1, 1]} \sigma_D^2(t_i, \epsilon)$ equal. We highlight some results as follows.

**Corollary 1.** Suppose that $\alpha$ satisfies Equation 6. If $\epsilon > \epsilon^*$,

$$
\max_{t_i \in [-1, 1]} \sigma_H^2(t_i, \epsilon) < \min \left\{ \max_{t_i \in [-1, 1]} \sigma_P^2(t_i, \epsilon), \max_{t_i \in [-1, 1]} \sigma_D^2(t_i, \epsilon) \right\};
$$

otherwise,

$$
\max_{t_i \in [-1, 1]} \sigma_H^2(t_i, \epsilon) = \max_{t_i \in [-1, 1]} \sigma_D^2(t_i, \epsilon) < \max_{t_i \in [-1, 1]} \sigma_P^2(t_i, \epsilon).
$$

---

Fig. 2: The noisy output $t_i^*$'s probability density function $pdf(t_i^*)$ in the Piecewise Mechanism.
Next, we focus on the calculation of each attribute over all users, and outputs these averages as the tuples, the aggregator simply computes the average value for a constant decided by $d$. In addition, observe that PM’s accuracy is close to HM’s, which demonstrates the effectiveness of PM.

### IV. Collecting Multiple Attributes

We now consider the case where each user’s data record contains $d > 1$ attributes. In this case, a straightforward solution is to collect each attribute separately using a single-attribute perturbation algorithm, such that every attribute is given a privacy budget $\epsilon/d$. Then, by the composition theorem [17], the collection of all attributes satisfies $\epsilon$-LDP. This solution, however, offers inferior data utility. For example, suppose that all $d$ attributes are numeric, and we process each attribute using PM, setting the privacy budget to $\epsilon/d$. Then, by Lemma [2] the amount of noise in the estimated mean of each attribute is $O\left(\sqrt{d \log d} / \epsilon \right)$, which is super-linear to $d$, and hence, can be excessive when $d$ is large. To address the problem, the first and only existing solution that we are aware of is by Duchi et al. [14] for the case of multiple numeric attributes, presented in Section IV-A.

#### A. Existing Solution for Multiple Numeric Attributes

Algorithm 3 shows the pseudo-code of Duchi et al.’s solution for multidimensional numeric data. It takes as input a tuple $t_i \in [-1,1]^d$ of user $u_i$ and a privacy parameter $\epsilon$, and outputs a perturbed vector $t_i^* \in [-B,B]^d$, where $B$ is a constant decided by $d$ and $\epsilon$. Upon receiving the perturbed tuples, the aggregator simply computes the average value for each attribute over all users, and outputs these averages as the estimates of the mean values for their corresponding attributes. Next, we focus on the calculation of $B$, which is rather complicated.


**input**: tuple $t_i \in [-1,1]^d$ and privacy parameter $\epsilon$.
**output**: tuple $t_i^* \in [-B,B]^d$.
1. Generate a random tuple $v \in [-1,1]^d$ by sampling each $v[j] \in [-1,1]$ independently from the following distribution:
   \[
   \Pr[v]\{j\} = \begin{cases} 
   \frac{1}{2} + \frac{\epsilon}{4}t_i[j], & \text{if } x = 1 \\
   \frac{1}{2} - \frac{\epsilon}{4}t_i[j], & \text{if } x = -1
   \end{cases};
   \]
2. Let $T^+$ (resp. $T^-$) be the set of all tuples $t^* \in \{-B,B\}^d$ such that $t^* \cdot v \geq 0$ (resp. $t^* \cdot v \leq 0$);
3. Sample a Bernoulli variable $u$ that equals 1 with $e^\epsilon/(e^\epsilon + 1)$ probability;
4. if $u = 1$ then
   5. return a tuple uniformly at random from $T^+$;
5. else
   6. return a tuple uniformly at random from $T^-$;

Essentially, $B$ is a scaling factor to ensure that the expected value of a perturbed attribute is the same as that of the exact attribute value. First, we compute:

\[
C_d = \begin{cases} 
\frac{2^{d-1}}{(d-1)!}, & \text{if } d \text{ is odd,} \\
\frac{2^{d-1} + 1}{d^{d/2}}, & \text{otherwise.}
\end{cases}
\]  

Then, $B$ is calculated by:

\[
B = \frac{\exp(\epsilon) + 1}{\exp(\epsilon) - 1} \cdot C_d.
\]

Duchi et al. show that $\frac{1}{n}\sum_{i=1}^{n} t_i^*[A_j]$ is an unbiased estimator of the mean of $A_j$, and

\[
\mathbb{E}\left[\max_{j \in [1,d]} \left| \frac{1}{n}\sum_{i=1}^{n} t_i^*[A_j] - \frac{1}{n}\sum_{i=1}^{n} t_i[A_j]\right|\right] = O\left(\frac{\sqrt{d \log d}}{\epsilon / \sqrt{n}}\right),
\]

which is asymptotically optimal [14].

Although Duchi et al.’s method can provide strong privacy assurance and asymptotic error bound, it is rather sophisticated, and it cannot handle the case that a tuple contains numeric attributes as well as categorical attributes. To address this issue, we present extensions of PM and HM that (i) are much simpler than Duchi et al.’s solution but achieve the same privacy assurance and asymptotic error bound, and (ii) can handle any combination of numeric and categorical attributes. For ease of exposition, we first extend PM and HM for the case when each $t_i$ contains only numeric attributes in Section IV-B, and then discuss the case of arbitrary attributes in Section IV-C.

#### B. Extending PM and HM for Multiple Numeric Attributes

Algorithm 4 shows the pseudo-code of our extension of PM and HM for multidimensional numeric data. Given a tuple $t_i \in [-1,1]^d$, the algorithm returns a perturbed tuple $t_i^*$ that has non-zero value on $k$ attributes, where

\[
k = \max\left\{1, \min\left\{d, \left\lceil \frac{\epsilon}{2.5} \right\rceil\right\}\right\}.
\]
In particular, each $A_j$ of those $k$ attributes is selected uniformly at random (without replacement) from all $d$ attributes of $t_i$, and $t^*_i[A_j]$ is set to $d \cdot x_i$, where $x_i$ is generated by PM or HM given $t_i[A_j]$ and $\xi$ as input.

The intuition of Algorithm 4 is as follows. By requiring each user to submit $k$ (instead of $d$) attributes, it increases the privacy budget for each attribute from $\epsilon/d$ to $\epsilon/k$, which in turn reduces the noise variance incurred. As a trade-off, sampling $k$ out of $d$ attributes entails additional estimation error, but this trade-off can be balanced by setting $k$ to an appropriate value. We are able to derive the setting of $k$ that minimizes the worst-case noise variance of Algorithm 4 when it utilizes PM (resp. HM), but find the formula for the optimal $k$ to be rather complicated. For simplicity, we set $k$ as in Equation 13, which yields asymptotically optimal performance while offering strong differential privacy, as we show in the following lemmas.

**Lemma 4.** Algorithm 4 satisfies $\epsilon$-local differential privacy. In addition, given an input tuple $t_i$, it outputs a noisy tuples $t^*_i$, such that for any $j \in [1, d]$, $\mathbb{E}[t^*_i[A_j]] = t_i[A_j]$.

**Proof.** Since Algorithm 4 composes $k$ numbers of $\frac{x_i}{n}$-LDP operations, then by the composition theorem [17], Algorithm 4 satisfies $\epsilon$-LDP. In Algorithm 4, $t^*_i[A_j]$ equals $d \cdot x_{i,j}$ with probability $\frac{k}{d}$ and 0 with probability $1 - \frac{k}{d}$. Hence, $\mathbb{E}[t^*_i[A_j]] = \frac{k}{d} \cdot \mathbb{E}[d \cdot x_{i,j}] = \mathbb{E}[x_{i,j}] = t_i[A_j]$, where the last step uses Lemma 1.

By Lemma 4, the aggregator can use $\frac{d}{k} \sum_{i=1}^{n} t^*[A_j]$ as an unbiased estimator of the mean of $A_j$. The following lemma shows that the accuracy guarantee of this estimator matches that of Duchi et al.’s solution for multidimensional unbiased estimator of the mean of $A_j$. The following lemma shows that the accuracy guarantee of this estimator matches that of Duchi et al.’s solution for multidimensional numeric data (see Equation 12), which has been proven to be asymptotically optimal [14]. This indicates that Algorithm 4’s accuracy guarantee is also optimal in the asymptotic sense.

**Lemma 5.** For any $j \in [1, d]$, let $Z[A_j] = \frac{d}{k} \sum_{i=1}^{n} t^*[A_j]$ and $X[A_j] = \frac{d}{k} \sum_{i=1}^{n} t_i[A_j]$. With at least $1 - \beta$ probability, 

$$
\max_{j \in [1, d]} |Z[A_j] - X[A_j]| = O \left( \frac{\sqrt{d \log(d/\beta)}}{\epsilon \sqrt{n}} \right).
$$

**Proof.** For any $i \in [1, n]$, by Lemma 4, the random variable $t^*_i[A_j] - t_i[A_j]$ has zero mean. In both PM and HM, $|t^*_i[A_j] - t_i[A_j]| \leq \frac{d}{k} \frac{e^{\epsilon/(2k)} + 1}{e^{\epsilon/(2k)} - 1}$. Then by Bernstein’s inequality,

$$
\Pr \left[ \left| \sum_{i=1}^{n} \{ t^*_i[A_j] - t_i[A_j] \} \right| \geq n \lambda \right] 
\leq 2 \cdot \exp \left( - \frac{(n \lambda)^2}{2 \sum_{i=1}^{n} \text{Var}(t^*_i[A_j]) + \frac{d}{k} n \lambda \frac{e^{\epsilon/(2k)} + 1}{e^{\epsilon/(2k)} - 1} \frac{3 e^{\epsilon/(2k)} - 1}{(2 e^{\epsilon/(2k)} - 1)^2} \lambda^2} \right) \quad (14).
$$

We now evaluate $\text{Var}[t^*_i[A_j]]$ appearing in (14) above. In Algorithm 4, $t^*_i[A_j]$ equals $\frac{d}{k} x_{i,j}$ with probability $\frac{k}{d}$ and 0 with probability $1 - \frac{k}{d}$. Also, $\mathbb{E}[t^*_i[A_j]] = t_i[A_j]$ by Lemma 4. Therefore,

$$
\text{Var}[t^*_i[A_j]] = \mathbb{E}[(t^*_i[A_j])^2] - (\mathbb{E}[t^*_i[A_j]])^2 
= \frac{k}{d} \mathbb{E} \left[ \left( \frac{d}{k} x_{i,j} \right)^2 \right] - (t_i[A_j])^2 
= \frac{d}{k} \mathbb{E} \left[ (x_{i,j})^2 \right] - (t_i[A_j])^2.
$$

Note that asymptotic expressions involving $\epsilon$ are in the sense of $\epsilon \to 0$. In Algorithm 4, if Line 5 uses PM, Lemma 4 implies

$$
\mathbb{E} \left[ (x_{i,j})^2 \right] = \text{Var} [x_{i,j}] + (\mathbb{E}[x_{i,j}])^2 
= \frac{(t_i[A_j])^2}{e^{\epsilon/(2k)} - 1} + (t_i[A_j])^2 = O \left( \frac{k^2}{\epsilon^2} \right).
$$

In Algorithm 4, if Line 5 uses HM, Equation 2 implies

$$
\mathbb{E} \left[ (x_{i,j})^2 \right] = \text{Var} [x_{i,j}] + (\mathbb{E}[x_{i,j}])^2 
= \left( \frac{e^{\epsilon/(2k)} + 1}{e^{\epsilon/(2k)} - 1} \right)^2 + (t_i[A_j])^2, 
$$

for $\epsilon/k > \epsilon^*$,

$$
O \left( \frac{k^2}{\epsilon^2} \right), \quad (17)
$$

where $\epsilon^*$ is defined by Equation 5.

Substituting Equation 16 or 17 into Equation 15, no matter whether Line 5 of Algorithm 4 uses PM or HM, we always have

$$
\text{Var}[t^*_i[A_j]] = \frac{d}{k} \cdot O \left( \frac{k^2}{\epsilon^2} \right) - (t_i[A_j])^2 = O \left( \frac{dk}{\epsilon^2} \right). \quad (18)
$$

Applying Equation 18 and $\frac{e^{\epsilon/(2k)} + 1}{e^{\epsilon/(2k)} - 1} \cdot \frac{d}{k} = O \left( \frac{k}{d} \right) = O \left( \frac{d}{\epsilon} \right)$ to Inequality 14, we obtain

$$
\Pr \left[ |Z[A_j] - X[A_j]| \geq \lambda \right] \leq 2 \cdot \exp \left( - \frac{(n \lambda)^2}{d \log(d/\beta) \sqrt{n}} \right).
$$

By the union bound, there exists $\lambda = O \left( \frac{\sqrt{d \log(d/\beta)}}{\epsilon \sqrt{n}} \right)$ such that $\max_{j \in [1, d]} |Z[A_j] - X[A_j]| < \lambda$ holds with at least $1 - \beta$ probability. By Equation 13, $k = 1$ for $\epsilon < 5$. Since asymptotic expressions involving $\epsilon$ consider $\epsilon \to 0$, $\lambda$ can also be written as $O \left( \frac{\sqrt{d \log(d/\beta)}}{\epsilon \sqrt{n}} \right)$.

Applying Equations 16 and 17 to Equation 15, we present the noise variances of our PM and HM in Equations 20 and 21 of Lemma 6 below. Also, in Duchi et al.’s solution given by Algorithm 3, since $(\frac{d}{k} - 1)^2$ always equals $B^2$ (i.e., $(\frac{d}{k} - 1)^2 C_d^2$), it follows that $\mathbb{E}[t^*_i[A_j]]^2 = (\frac{d}{k} - 1)^2 C_d^2$, which along with Duchi et al.’s result $\mathbb{E}[t^*_i[A_j]] = t_i[A_j]$ implies Equation 19 below.

**Lemma 6.** For a $d$-dimensional numeric tuple $t_i$ which is perturbed as $t_i^*$ under $\epsilon$-LDP, and for each $A_j$ of the $d$ attributes, the variance of $t^*_i[A_j]$ induced by Duchi et al.’s solution is

$$
\text{Var}_D [t^*_i[A_j]] = \left( \frac{d}{k} - 1 \right)^2 C_d^2 - (t_i[A_j])^2, \quad (19)
$$
where \(C_d\) is defined by Equation 10. Meanwhile, the variance of \(t_i^*\) induced by PM is
\[
\text{Var}_{i}[t_i^*] = \frac{d(e^r/(e^r+1)+3)}{3k(e^r/(e^r+1)-1)^2} + \left[\frac{d(e^r/(e^r+1))}{k(e^r/(e^r+1)-1)} - 1\right] \left(t_i[A_j]\right)^2;
\]
and the variance of \(t_i^*\) induced by HM is
\[
\text{Var}_{i}[t_i^*] = \begin{cases} 
\frac{d}{k} \left(\frac{e^r/(e^r+1)}{k(e^r/(e^r+1)-1)}\right)^2 + \left(\frac{d}{k} - 1\right) \left(t_i[A_j]\right)^2, & \text{for } \epsilon/k > \epsilon^*, \\
\text{Var}_{i}[t_i^*] + \frac{d}{k} \left(\frac{e^r/(e^r+1)}{k(e^r/(e^r+1)-1)}\right)^2 + \left(\frac{d}{k} - 1\right) \left(t_i[A_j]\right)^2, & \text{for } \epsilon/k \leq \epsilon^*,
\end{cases}
\]
where \(\epsilon^*\) is as defined in Equation 5.

From Equations 19, 20, and 21, we can prove the following:

**Corollary 2.** For any \(d > 1\) and \(\epsilon > 0\), both PM and HM outperform Duchi et al.'s solution in minimizing the worst-case noise variance; more specifically, for any \(d > 1\) and \(\epsilon > 0\),
\[
\max_{t_i[A_j] \in [-1,1]} \text{Var}_{H}[t_i^*] < \max_{t_i[A_j] \in [-1,1]} \text{Var}_{P}[t_i^*] < \max_{t_i[A_j] \in [-1,1]} \text{Var}_{D}[t_i^*].
\]

To illustrate Corollary 2, Figure 3 shows the worst-case variance of PM (resp. HM) as a fraction of the worst-case variance of Duchi et al.'s solution, for various \(d\) and privacy budget \(\epsilon\). Observe that for \(d = 5, 10, 20, 40\), the worst-case variance of HM is at most 77% of that of Duchi et al.'s solution, and PM's worst-case variance is also smaller than the latter. In our experiments, we demonstrate that both HM and PM outperform Duchi et al.'s solution in terms of the empirical accuracy for multidimensional numeric data.

C. **Handling Categorical Attributes**

So far our discussion is limited to numeric attributes. Next we extend Algorithm 4 to handle data with both numeric and categorical attributes. Recall from Section III that for each categorical attribute \(A\), our objective is to estimate the frequency of each value \(v\) in \(A\) over all users. We note that most existing LDP algorithms (e.g., [5], [18], [39]) for categorical data are designed for this purpose, albeit limited to a single categorical attribute.

Formally, we assume that we are given an algorithm \(f\) that takes an input a privacy budget \(\epsilon\) and a one-dimensional tuple \(t_i\) with a categorical attribute \(A\), and outputs a perturbed tuple \(t_i^*\) while ensuring \(\epsilon\)-LDP. In addition, for any value \(v\) in \(A\), we assume there is an estimator \(g\), such that for any random subset \(S \subseteq \{1, 2, \ldots, n\}\), \(d \sum_{i \in S} g(t_i^*; v)\) is an estimator of the frequency of \(v\) over all users. Then, for the general case when \(t_i\) contains \(d\) (numeric or categorical) attributes \(A_1, A_2, \ldots, A_d\), the extended version of Algorithm 4 would request each user to perform the following:

1) Sample \(k\) values uniformly at random from \([1, 2, \ldots, d]\), where \(k\) is as defined in Equation 13.

2) For each sampled \(j\), if \(A_j\) is a numerical attribute, then submit a noisy version of \(f(t_i[A_j])\) computed as in Lines 3-5 of Algorithm 4; otherwise (i.e., \(A_j\) is a categorical attribute), submit \(f(t_i[A_j], \epsilon)\), where \(f\) can be any existing solution for perturbing a single categorical attribute under \(\epsilon\)-LDP.

Once the aggregator collects data from all users, she can estimate the mean of each numeric attribute \(A\) in the same way as in Algorithm 4. In addition, for any categorical attribute \(A'\) and any value \(v\) in the domain of \(A\), she can estimate the frequency of \(v\) among all users as \(d \sum_{i \in S} g(v^*, v)\), where \(V^*\) denotes the set of perturbed \(A'\) values submitted by users. The accuracy of this estimator depends on both \(d\) and the accuracy of single-attribute perturbation algorithm used for \(A'\). In our experiments, we apply the optimized local hashing (OH) protocol of Wang et al. [39] to perturb a single categorical attribute, which is the current state of the art to our knowledge.

V. **STOCHASTIC GRADIENT DESCENT UNDER LOCAL DIFFERENTIAL PRIVACY**

This section investigates building a class of machine learning models under \(\epsilon\)-LDP that can be expressed as empirical risk minimization, and solved by stochastic gradient descent (SGD). In particular, we focus on three common types of learning tasks: linear regression, logistic regression, and support vector machines (SVM) classification.

Suppose that each user \(u_i\) has a pair \((x_i, y_i)\), where \(x_i \in [-1, 1]^d\) and \(y_i \in [-1, 1]\) (for linear regression) or \(y_i \in \{0, 1\}\) (for logistic regression and SVM classification). Let \(\ell(\cdot)\) be a loss function that maps a \(d\)-dimensional parameter...
vector $\beta$ into a real number, and is parameterized by $x_i$ and $y_i$. We aim to identify a parameter vector $\beta^*$ such that

$$\beta^* = \arg \min_\beta \left[ \frac{1}{n} \left( \sum_{i=1}^{n} \ell(\beta; x_i, y_i) \right) + \frac{\lambda}{2} \|\beta\|^2 \right],$$

where $\lambda > 0$ is a regularization parameter. We consider three specific loss functions:

1. Linear regression: $\ell(\beta; x_i, y_i) = (x_i^T \beta - y_i)^2$;
2. Logistic regression: $\ell(\beta; x_i, y_i) = \log (1 + e^{-y_i x_i^T \beta})$;
3. SVM (hinge loss): $\ell(\beta; x_i, y_i) = \max \{0, 1 - y_i x_i^T \beta\}$.

For convenience, we define

$$\ell'(\beta; x, y) = \ell(\beta; x, y) + \frac{\lambda}{2} \|\beta\|^2.$$

The proposed approach solves $\beta^*$ using SGD, which starts from an initial parameter vector $\beta_0$, and iteratively updates it into $\beta_1, \beta_2, \ldots$ based on the following equation:

$$\beta_{t+1} = \beta_t - \gamma_t \cdot \nabla \ell'(\beta_t; x, y),$$

where $(x, y)$ is the data record of a randomly selected user, $\nabla \ell'(\beta_t; x, y)$ is the gradient of $\ell'$ at $\beta_t$, and $\gamma_t$ is called the learning rate at the $t$-th iteration. The learning rate $\gamma_t$ is commonly set by a function (called the learning schedule) of the iteration number $t$; a popular learning schedule is $\gamma_t = O(1/\sqrt{t})$.

In the non-private setting, SGD terminates when the difference between $\beta_{t+1}$ and $\beta_t$ is sufficiently small. Under $\epsilon$-LDP, however, $\nabla \ell'$ is not directly available to the aggregator, and needs to be collected in a private manner. Towards this end, existing studies \[14, 21\] have suggested that the aggregator asks the selected user in each iteration to submit a noisy version of $\nabla \ell'$, by using the Laplace mechanism or Duchi et al.’s solution (i.e., Algorithm \[3\]). Our baseline approach is based on this idea, and improves these existing methods by perturbing $\nabla \ell'$ using Algorithm \[4\]. In particular, in each iteration, we involve a group $G$ of users, and ask each of them to submit a noisy version of the gradient; after that, we update the parameter vector $\beta_t$ with the mean of the noisy gradients, i.e.,

$$\beta_{t+1} = \beta_t - \gamma_t \cdot \frac{1}{|G|} \sum_{i \in G} \nabla \ell_{i}^{*},$$

where $\nabla \ell_{i}^{*}$ is the noisy gradient submitted by the $i$-th user in group $G$. This helps because the amount of noise in the average gradient is $O\left(\sqrt{d \log d} / \epsilon \sqrt{|G|}\right)$, which could be acceptable if $|G| = \Omega\left(d (\log d) / \epsilon^2 \right)$.

Note that in the non-private case, the aggregator often allows each user to participate in multiple iterations (say $m$ iterations) to improve the accuracy of the model. But it does not work in the local differential privacy setting. To explain this, suppose that the $i$-th $(i \in [1, m])$ gradient returned by the user satisfies $\epsilon_i$-differential privacy. By the composition property of differential privacy \[28\], if we enforce $\epsilon$-differential privacy for the user’s data, we should have $\sum_{i=1}^{m} \epsilon_i \leq \epsilon$. Consider that we set $\epsilon_i = \epsilon/m$. Then, the amount of noise in each gradient becomes $O\left(\frac{m \sqrt{d \log d}}{\epsilon}\right)$; accordingly, the group size becomes $|G| = \Omega\left(m^2 d \log d / \epsilon^2 \right)$, which is $m^2$ times larger compared to the case where each user only participates in at most one iteration. It then follows that the total number of iterations in the algorithm is inverse proportional to $1/m$; i.e., setting $m > 1$ only degrades the performance of the algorithm.

### VI. Experiments

We have implemented the proposed methods and evaluated them using two public datasets extracted from the Integrated Public Use Microdata Series \[1\]. BR and MX, which contains census records from Brazil and Mexico, respectively. BR contains 4M tuples and 16 attributes, among which 6 are numerical (e.g., age) and 10 are categorical (e.g., gender); MX has 4M records and 19 attributes, among which 5 are numerical and 14 are categorical. Both datasets contain a numerical attribute “total income”, which we use as the dependent attribute in linear regression, logistic regression, and SVM (explained further in Section \[VI-B\]). We normalize the domain of each numerical attribute into $[-1, 1]$. In all experiments, we report average results over 100 runs.

#### A. Results on Mean Value / Frequency Estimation

In the first set of experiments, we consider the task of collecting a noisy, multidimensional tuple from each user, in order to estimate the mean of each numerical attribute and the frequency of each categorical value. Since no existing solution can directly support this task, we take the following best-effort approach combining state-of-the-art solutions through the composition property of differential privacy \[28\]. Specifically, let $t$ be a tuple with $d_n$ numeric attributes and $d_c$ categorical attributes. Given total privacy budget $\epsilon$, we allocate $d_n \epsilon/d$ budget to the numeric attributes, and $d_c \epsilon/d$ to the categorical ones, respectively. Then, for the numeric attributes, we estimate the mean value for each of them using either (i) Duchi et al.’s solution (i.e., Algorithm \[3\]), which directly handles multiple numeric attributes, or (ii) the Laplace mechanism, applied to each numeric attribute individually using $\epsilon/d$ budget. Regarding categorical attributes, since no previous solution addresses the multidimensional case, we apply the optimized local hashing (OLH) protocol of Wang et al. \[36\], the state of the art for frequency estimation on a single categorical attribute, to each attribute independently with $\epsilon/d$ budget. Clearly, by the composition property of differential privacy \[28\], the above approach satisfies $\epsilon$-LDP.

We evaluate both the above best-effort approach using existing methods, and the proposed solution in Section \[IV\] on the two real datasets BR and MX. For each method, we measure the mean square error (MSE) in the estimated mean values (for numeric attributes) and value frequencies (for categorical attributes). Figure \[4\] plots the MSE results as a function of the total privacy budget $\epsilon$. Overall, the proposed solution consistently and significantly outperforms the best-effort approach combining existing methods. One major reason is that the estimation error of the proposed solution is asymptotically optimal, which scales sublinearly to the data dimensionality $d$; in contrast, the best-effort combination of existing approaches...
Fig. 4: Result accuracy for mean estimation (on numeric attributes) and frequency estimation (on categorical attributes).

Fig. 5: Result accuracy on synthetic datasets with 16 dimensions, each of which follows a Gaussian distribution $N(\mu, 1/16)$ truncated to $[-1, 1]$.

Fig. 6: Logistic Regression.

Fig. 7: Support Vector Machines (SVM).

Fig. 8: Linear Regression.

Involves privacy budget splitting, which is sub-optimal. For instance, on the categorical attributes, applying OLH [36] on each attribute individually leads to $O\left(\frac{\mu^2 \log \frac{3}{\epsilon}}{\epsilon \sqrt{n}}\right)$ error (where $n$ is the number of users, and $k$ is the number of distinct values in each attribute), which grows linearly with data dimensionality $d$. This also explains the consistent performance gap between Duchi et al.’s solution [14] and the Laplace mechanism on numeric attributes.

Meanwhile, on numeric attributes, the proposed solutions PM and HM outperform Duchi et al.’s solution in all settings. This is because (i) although all three methods are asymptotically optimal, Duchi et al.’s solution incurs a larger constant than the proposed algorithms and (ii) Duchi et al.’s solution cannot handle categorical attributes, and, thus, needs to be combined with OLH through privacy budget splitting, which is sub-optimal. To eliminate the effect of (ii), we run an additional set of experiments with only numeric attributes on a synthetic dataset. Specifically, the synthetic data contains 16 numeric attributes (i.e., same number of attributes in BR), where each attribute value is generated from a Gaussian distribution with mean value $\mu$ and standard deviation $1/4$, but discarding any value that fall out of $[-1, 1]$. Figure 5 shows the results with varying privacy budget $\epsilon$, and 4 different values for $\mu$. In all settings, PM and HM outperform Duchi et al.’s solution, and the performance gap slightly expands with increasing $\epsilon$, which agrees with our analysis in Section IV. Finally, comparing PM and HM, the difference in their performance is small, and the relative performance of the two can be different in different settings. Note that the main advantage of HM over PM is that on a single numeric attribute, HM is never worse than Duchi et al., whereas PM does not have this guarantee.
B. Results on Empirical Risk Minimization

In the second set of experiments, we evaluate the accuracy performance of the proposed methods for linear regression, logistic regression, and SVM classification on BR and MX. For both datasets, we use the numeric attribute “total income” as the dependent variable, and all other attributes as independent variables. Following common practice, we transform each categorical attribute \( A_j \) with \( k \) values into \( k - 1 \) binary attributes with a domain \( \{-1, 1\} \), such that (i) the \( l \)-th (\( l < k \)) value in \( A_j \) is represented by 1 on the \( l \)-th binary attribute and -1 on each of the remaining \( k - 2 \) attributes, and (ii) the \( k \)-th value in \( A_j \) is represented by -1 on all binary attributes. After this transformation, the dimensionality of BR (resp. MX) becomes 90 (resp. 94). For logistic regression and SVM, we also covert “total income” into a binary attribute by mapping the values larger than the mean value to 1, and 0 otherwise.

Since each user sends gradients to the aggregator, which are all numeric, the experiment involves the 4 competitors in Section VI-A for numeric data: PM, HM, Duchi et al. [14], and the Laplace mechanism applied to each attribute independently with equally split privacy budget \( \epsilon \). Similar to the results in Section VI-A, the Laplace mechanism leads to significantly higher than the other three solutions, due to the fact that its error rate is sub-optimal. The proposed algorithms PM and HM consistently outperform Duchi et al.’s solution with clear margins, since (i) the former two have smaller constant as analyzed in Section IV, and (ii) the gradient of each user often consists of elements whose absolute values are small, for which PM and HM are particularly effective, as we mention in Section III-B. Further, in some settings such as SVM with \( \epsilon \geq 2 \) on BR, the accuracy of PM and HM approaches that of the non-private method. Comparing the results with those in Section VI-A, we observe that the misclassification rates for logistic regression and SVM classification do not drop as quickly with increasing privacy budget \( \epsilon \) as in the case of MSE for mean values and frequency estimates. This is due to the inherent stochastic nature of SGD: that accuracy in gradients does not have a direct effect on the accuracy of the model. For the same reason, there is no clear trend for the performance gap between PM/HM and Duchi et al.’s solution.

Figure 8 demonstrates the mean squared error (MSE) of the linear regression model generated by each method with varying \( \epsilon \). We omit the MSE results for the Laplace mechanism, since they are far higher than the other three methods. The proposed solutions PM and HM once again consistently outperform Duchi et al.’s solution. Overall, our experimental results demonstrate the effectiveness of PM and HM for empirical risk minimization under local differential privacy, and their consistent performance advantage over existing approaches.

VII. Related Work

Differential privacy [16] is a strong privacy standard that provides semantic, information-theoretic guarantees on individuals’ privacy, which has attracted much attention from various fields, including data management [8], [10], [26], machine learning [4], theory [5], [13], [24], and systems [6]. Earlier models of differential privacy [16], [17], [28] rely on a trusted data curator, who collects and manages the exact private information of individuals, and releases statistics derived from the data under differential privacy requirements. Recently, much attention has been shifted to the local differential privacy (LDP) model (e.g., [13], [24]), which eliminates the data curator and the collection of exact private information.

LDP can be connected to the classical randomized response technique in surveys [39]. Duchi et al. [13] propose the minimax framework for LDP based on information theory, prove upper and lower error bounds of LDP-compliant methods, and analyze the trade-off between privacy and accuracy. Erlingsson et al. [18] propose the RAPPOR framework, which is based on the randomized response mechanism for publishing a value for binary attributes under LDP. They use this mechanism with a Bloom filter, which intuitively adds another level of protection and increases the difficulty for the adversary to infer private information. A follow-up paper [19] extends RAPPOR to more complex statistics such as joint-distributions and association testing, as well as categorical attributes that contain a large number of potential values, such as a user’s home page. Kairouz et al. [23] show that a version of randomized response is universally optimal in low privacy regime for frequency estimation on a single binary attribute. They further study how to handle a categorical attribute with an arbitrary number of possible values [22]. Wang et al. [36] investigate the same problem, and propose a different method: they transform \( k \) possible values into a noisy vector with \( k \) elements, and send the latter to curator. Bassily and Smith [5] propose an asymptotically optimal solution for building succinct histograms over a large categorical domain under LDP. This solution can also be used to publish spatial data [8]. Note that all of the above methods focus on a single attribute, and, thus, are orthogonal to our work on multidimensional data. Ren et al. [33] investigate the problem of publishing multiple attributes, and employ the idea of \( k \)-sized vector, similar to [36]. This approach, however, incurs rather high communication costs between the aggregator and the users, since it involves the transmission of multiple \( k \)-sized vectors.

Various data analytics and machine learning problems have been studied under LDP, such as probability distribution estimation [2], [15], [22], [29], [31], [40], heavy hitter discovery [4], [7], [32], [37], frequent new term discovery [35], frequency estimation [5], [36], frequent itemset mining [38], marginal release [10], clustering [30], and hypothesis testing [20].
Finally, a recent work [3] introduces a hybrid model that involves both centralized and local differential privacy. Bittau et al. [6] evaluate real-world implementations of LDP. Also, LDP has been considered in several applications including the collection of indoor positioning data [25], inference control on mobile sensing [27], and the publication of crowdsourced data [33].

VIII. CONCLUSION

This work systematically investigates the problem of collecting and analyzing users’ personal data under $\epsilon$-local differential privacy, in which the aggregator only collects randomized data from the users, and computes statistics based on such data. The proposed solution is able to collect data records that contain multiple numerical and categorical attributes, and compute accurate statistics from simple ones such as mean and frequency to complex machine learning models such as linear regression, logistic regression and SVM classification. Our solution achieves both optimal asymptotic error bound and high accuracy in practice. Extensive experiments demonstrate its effectiveness on real data. In the next step, we plan to apply the proposed solution to more complex data analysis tasks such as deep neural networks. Further, we intend to adapt our methods to $(\epsilon, \delta)$-local differential privacy, which may promise higher accuracy in the analysis results under a relaxed privacy model.

REFERENCES