This full paper appears in the Proceedings of the IEEE 35th Annual International Conference on Data Engineering (ICDE), held in April 2019.

Collecting and Analyzing Multidimensional Data with Local Differential Privacy

Ning Wang[1], Xiaokui Xiao[2], Yin Yang[3], Jun Zhao[4], Siu Cheung Hui[4], Hyejin Shin[5], Junbum Shin[5], Ge Yu[6]

[1]*School of Information Science and Engineering, Ocean University of China*
[2]*School of Computing, National University of Singapore, Singapore*
[3]*Division of Information and Computing Techologies, College of Science and Engineering, Hamad Bin Khalifa University, Qatar*
[4]*School of Computer Science and Engineering, Nanyang Technological University, Singapore*
[5]*Samsung Research, Samsung Electronics, Korea*
[6]*School of Computer Science and Engineering, Northeastern University, China*

[1]wangning8687@ouc.edu.cn, [2]xkxiao@nus.edu.sg, [3]yyang@hbku.edu.qa, [4]{junzhao, asschui}@ntu.edu.sg,
[5]{hyejin1.shin, junbum.shin}@samsung.com, [6]yuge@mail.neu.edu.cn

***Abstract*—Local differential privacy (LDP) is a recently proposed 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 systems such as Google Chrome [18], Apple iOS and macOS [36], and Microsoft Windows Insiders [12], is a rigorous 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 data 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, e.g., 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[1] and Netflix[2] 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 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. 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 complicated and involves a large constant. Finally, to our knowledge, there is no existing solution that

[1]https://en.wikipedia.org/wiki/AOL_search_data_leak
[2]https://www.wired.com/2009/12/netflix-privacy-lawsuit/

TABLE I: Main theoretical results comparing the proposed mechanisms PM and HM, as well as Duchi et al.'s solution [14]. The terms $MaxVar_{\mathrm{PM}}$, $MaxVar_{\mathrm{HM}}$, and $MaxVar_{\mathrm{Du}}$ denote the worst-case noise variance of these three methods, respectively, for perturbing a d-dimensional numeric tuple under ϵ-local differential privacy (elaborated in Section II). In addition, $\epsilon^{\#} = \ln\left(\frac{7+4\sqrt{7}+2\sqrt{20+14\sqrt{7}}}{9}\right) \approx 1.29$ and $\epsilon^{*} = \ln\left(\frac{-5+2\sqrt[3]{6353-405\sqrt{241}}+2\sqrt[3]{6353+405\sqrt{241}}}{27}\right) \approx 0.61$.

Setting		Result
$d > 1$	$\epsilon > 0$	$MaxVar_{\mathrm{HM}} < MaxVar_{\mathrm{PM}} < MaxVar_{\mathrm{Du}}$
$d = 1$	$\epsilon > \epsilon^{\#}$	$MaxVar_{\mathrm{HM}} < MaxVar_{\mathrm{PM}} < MaxVar_{\mathrm{Du}}$
	$\epsilon = \epsilon^{\#}$	$MaxVar_{\mathrm{HM}} < MaxVar_{\mathrm{PM}} = MaxVar_{\mathrm{Du}}$
	$\epsilon^{*} < \epsilon < \epsilon^{\#}$	$MaxVar_{\mathrm{HM}} < MaxVar_{\mathrm{Du}} < MaxVar_{\mathrm{PM}}$
	$0 < \epsilon \leq \epsilon^{*}$	$MaxVar_{\mathrm{HM}} = MaxVar_{\mathrm{Du}} < MaxVar_{\mathrm{PM}}$

can perturb multidimensional data containing both numeric and categorical data with optimal worst-case error.

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 [14] 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 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.

II. Preliminaries

In the problem setting, an *aggregator* collects data from a set of *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 *ϵ-local differential privacy* (*ϵ-LDP*) [18], defined as follows:

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

$$\Pr\left[f(t) = t^*\right] \leq \exp(\epsilon) \cdot \Pr\left[f(t') = t^*\right]. \tag{1}$$

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 ϵ), 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 ϵ-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 *Var*$[\cdot]$ to denote a random variable's expected value and variance, respectively.

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 ϵ-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 and its variants. 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 $t_i^* = t_i + \mathit{Lap}\left(\frac{2}{\epsilon}\right)$, where $\mathit{Lap}(\lambda)$ denotes a random variable that follows a Laplace distribution of scale λ, with the following probability density function: $\mathit{pdf}(x) = \frac{1}{2\lambda}\exp\left(-\frac{|x|}{\lambda}\right)$.

Clearly, this estimate t_i^* is unbiased, since the injected Laplace noise $\mathit{Lap}\left(\frac{2}{\epsilon}\right)$ 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 $O\left(\frac{1}{\epsilon\sqrt{n}}\right)$.

Soria-Comas and Domingo-Ferrer [35] propose a more sophisticated variant of Laplace mechanism, hereafter referred to as *SCDF*, that obtains improved result accuracy for multi-dimensional data. Later, Geng et al. [21] propose *Staircase mechanism*, which achieves optimal performance for unbounded input values (e.g., from a domain of $(-\infty, +\infty)$). Specifically, for a single attribute value t_i, both methods inject random noise n_i drawn from the following piece-wise constant probability distribution:

$$\mathit{pdf}(n_i = x) = \begin{cases} \frac{a(m)}{e^{j\epsilon}}, & \text{if } x \in \left[-m-2(j+1), -m-2j\right], j \in \mathbb{N}, \\ a(m), & \text{if } x \in [-m, m], \\ \frac{a(m)}{e^{j\epsilon}}, & \text{if } x \in \left[m+2j, m+2(j+1)\right], j \in \mathbb{N}. \end{cases} \tag{2}$$

In SCDF, $m = 2 \cdot \frac{1-\exp(-\epsilon)-\epsilon\exp(-\epsilon)}{\epsilon-\epsilon\exp(-\epsilon)}$ and $a(m) = \frac{\epsilon}{4}$; in Staircase mechanism, $m = \frac{2}{1+e^{\epsilon/2}}$ and $a(m) = \frac{1-e^{-\epsilon}}{2m+4e^{-\epsilon}-2me^{-\epsilon}}$. Note that the optimality result in [21] does not apply to the case with bounded inputs. We experimentally compare the proposed solutions with both SCDF and Staircase in Section VI.

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 algorithm returns a perturbed tuple t_i^* that equals either $\frac{e^\epsilon+1}{e^\epsilon-1}$ or $-\frac{e^\epsilon+1}{e^\epsilon-1}$, with the following probabilities:

$$\Pr\left[t_i^* = x \mid t_i\right] = \begin{cases} \frac{e^\epsilon-1}{2e^\epsilon+2}\cdot t_i + \frac{1}{2}, & \text{if } x = \frac{e^\epsilon+1}{e^\epsilon-1}, \\ -\frac{e^\epsilon-1}{2e^\epsilon+2}\cdot t_i + \frac{1}{2}, & \text{if } x = -\frac{e^\epsilon+1}{e^\epsilon-1}. \end{cases} \tag{3}$$

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:

$$\begin{aligned} \mathit{Var}[t_i^*] &= \mathbb{E}\left[(t_i^*)^2\right] - (\mathbb{E}[t_i^*])^2 \\ &= \left(\tfrac{e^\epsilon+1}{e^\epsilon-1}\right)^2 \cdot \tfrac{t_i\cdot(e^\epsilon-1)+e^\epsilon+1}{2e^\epsilon+2} + \left(-\tfrac{e^\epsilon+1}{e^\epsilon-1}\right)^2 \cdot \tfrac{-t_i\cdot(e^\epsilon-1)+e^\epsilon+1}{2e^\epsilon+2} - {t_i}^2 \\ &= \left(\tfrac{e^\epsilon+1}{e^\epsilon-1}\right)^2 - {t_i}^2. \end{aligned} \tag{4}$$

Therefore, the worst-case variance of t_i^* equals $\left(\frac{e^\epsilon+1}{e^\epsilon-1}\right)^2$, 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.

Algorithm 1: Duchi et al.'s Solution [14] for One-Dimensional Numeric Data.

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

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 $\quad t_i^* = \frac{e^\epsilon+1}{e^\epsilon-1}$;
4 **else**
5 $\quad t_i^* = -\frac{e^\epsilon+1}{e^\epsilon-1}$;
6 **return** t_i^*

Deficiencies of existing solutions. Fig. 1 illustrates the worst-case variance of the noisy values returned by the Laplace mechanism and Duchi et al.'s solution, when ϵ varies. Duchi et al.'s solution offers considerably smaller variance than the Laplace mechanism when $\epsilon \le 2$, but is significantly outperformed by the latter when ϵ is large. To explain, recall that Duchi et al.'s solution returns either $t_i^* = \frac{e^\epsilon+1}{e^\epsilon-1}$ or $t_i^* = -\frac{e^\epsilon+1}{e^\epsilon-1}$, 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 $\frac{e^\epsilon+1}{e^\epsilon-1} > 1$, due to which t_i^*'s variance is always larger than 1 when $t_i = 0$, regardless of how large the privacy budget ϵ is. In contrast, the Laplace mechanism incurs a noise variance of $8/\epsilon^2$, which decreases quadratically with the increase of ϵ, due to which it is preferable when ϵ is large. However, when ϵ 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 $\left[-\frac{e^\epsilon+1}{e^\epsilon-1}, \frac{e^\epsilon+1}{e^\epsilon-1}\right]$.

This full paper appears in the Proceedings of the IEEE 35th Annual International Conference on Data Engineering (ICDE), held in April 2019.

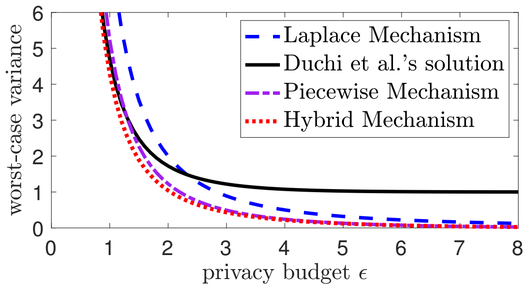


Fig. 1: Different mechanisms' worst-case noise variances for one-dimensional numeric data versus the privacy budget ϵ. Our Piecewise Mechanism and Hybrid Mechanism will be discussed in Sections III-B and III-C, respectively.

SCDF and Staircase mechanism suffer from the same issue as the Laplace mechanism, as demonstrated in our experiments in Section VI.

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 *wide* spectrum of ϵ? 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:

$$pdf(t_i^* = x \mid t_i) = \begin{cases} p, & \text{if } x \in [\ell(t_i),\, r(t_i)], \\ \frac{p}{\exp(\epsilon)}, & \text{if } x \in [-C,\, \ell(t_i)) \cup (r(t_i),\, C]. \end{cases} \tag{5}$$

where

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

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

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

Let $pdf(t_i^*)$ be short for $pdf(t_i^* = x \mid t_i)$. 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: Piecewise Mechanism for One-Dimensional Numeric Data.

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

Algorithm 2 shows the pseudo-code of PM, assuming the input domain is $[-1, 1]$. In general, when the input domain is $t_i \in [-r, r], r > 0$, the user (i) computes $t_i' = t_i/r$, (ii) perturbs t_i' using PM, since $t_i' \in [-1, 1]$, and (iii) submits $r \cdot t_i^*$ to the server, where t_i^* denotes the noisy value output by Algorithm 2. It can be verified that $r \cdot t_i^*$ is an unbiased estimator of t_i. The above method requires that the user knows r, which is a common assumption in the literature, e.g., in Duchi et al.'s work [14].

The following lemmas establish the theoretical guarantees of Algorithm 2.

Lemma 1. *Algorithm 2 satisfies ϵ-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*

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

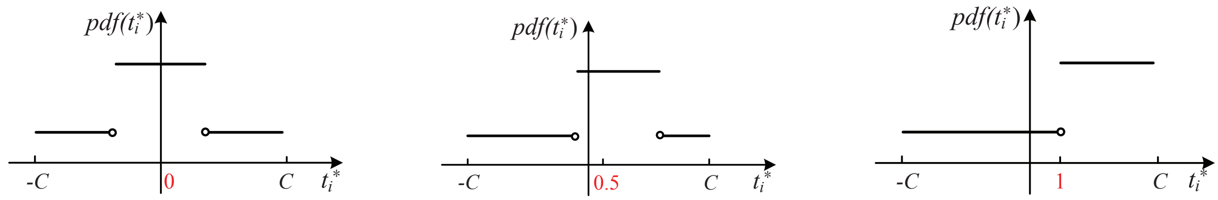
The proof appears in the full version [2].

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 ϵ. Observe that PM's worst-case variance is considerably smaller than that of Duchi et al.'s solution when $\epsilon \geq 1.29$, and is only slightly larger than the latter when $\epsilon < 1.29$, where 1.29 is x-coordinate of the point that the Duchi et al.' solution curve intersects that of PM in Fig. 1. Furthermore, it can be proved that PM's worst-case variance is strictly smaller than Laplace mechanism's, regardless of the value of ϵ. This makes PM be 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



(a) $t_i = 0$ (b) $t_i = 0.5$ (c) $t_i = 1$

Fig. 2: The noisy output t_i^*'s probability density function *pdf*(t_i^*) in the Piecewise Mechanism.

class of applications.) In contrast, Duchi et al.'s solution incurs a noise variance that *increases* with the decrease of $|t_i|$, as shown in Equation 4.

Now consider the estimator $\frac{1}{n}\sum_i^n t_i^*$ 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 $\frac{1}{n}\sum_i^n t_i^*$.

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,*

$$\big|Z - X\big| = O\left(\frac{\sqrt{\log(1/\beta)}}{\epsilon\sqrt{n}}\right).$$

We omit the proof of Lemma 2 as it is a special case of Lemma 5 to be presented in Section IV-B.

Remark. PM bears some similarities to SCDF [35] and Staircase mechanism [21] described in Section III-A, in the sense that the added noise in PM also follows a piecewise constant distribution, as in SCDF and Staircase. On the other hand, there are two crucial differences between PM and SCDF/Staircase. First, SCDF and Staircase mechanism assume an unbounded input, and produce an unbounded output (i.e., with range $(-\infty, +\infty)$) accordingly. In contrast, PM has both bounded input (with domain $[-1, 1]$) and output (with range $[-C, C]$). Second, the noise distribution of SCDF/Staircase consists of an infinite number of "pieces" that are data *independent*, whereas the output distribution of the piecewise mechanism consists of up to three "pieces" whose lengths and positions depend on the input data.

C. Hybrid Mechanism

As discussed in Section III-B, the worst-case result accuracy of PM dominates that of the Laplace mechanism, and yet it can still be (slightly) worse than Duchi et al's solution, since the noise variance incurred by the former (resp. latter) decreases (resp. increases) with the decrease of $|t_i|$. Can we construct a method that that preserves the advantages of PM and is at the same time always no worse than Duchi et al's solution? The answer turns out to be positive: that we can combine PM and Duchi et al's solution into a new *Hybrid Mechanism* (*HM*). Further, the combination used in HM is non-trivial; as a result, the noise variance of HM is often smaller than both PM and Duchi et al's solution, as shown in Fig. 1 on Page 4.

In particular, given an input value t_i, HM flips a coin whose head probability equals a constant α; if the coin shows a head (resp. tail), then we invoke PM (resp. Duchi et al.'s solution) to perturb t_i. Given t_i and ϵ, the noise variance incurred by HM is

$$\sigma_H^2(t_i,\epsilon) = \alpha\cdot\sigma_P^2(t_i,\epsilon) + (1-\alpha)\cdot\sigma_D^2(t_i,\epsilon),$$

where $\sigma_P^2(t_i,\epsilon)$ and $\sigma_D^2(t_i,\epsilon)$ denote the noise variance incurred by PM and Duchi et al.'s solution, respectively, when given t_i and ϵ as input. We have the following lemma.

Lemma 3. *Let ϵ^* be defined as:*

$$\epsilon^* = \ln\left(\frac{-5+2\sqrt[3]{6353-405\sqrt{241}}+2\sqrt[3]{6353+405\sqrt{241}}}{27}\right) \approx 0.61. \tag{6}$$

The term $\max_{t_i\in[-1,1]}\sigma_H^2(t_i,\epsilon)$ is minimized when

$$\alpha = \begin{cases} 1-e^{-\epsilon/2}, & \textit{for } \epsilon > \epsilon^*, \\ 0, & \textit{for } \epsilon \le \epsilon^*. \end{cases} \tag{7}$$

The proof appears in the full version [2].

By Lemma 3, when α satisfies Equation 7, the worst-case noise variance of HM is:

$$\max_{t_i\in[-1,1]}\sigma_H^2(t_i,\epsilon) = \begin{cases} \frac{e^{\epsilon/2}+3}{3e^{\epsilon/2}(e^{\epsilon/2}-1)} + \frac{(e^{\epsilon}+1)^2}{e^{\epsilon/2}(e^{\epsilon}-1)^2}, & \text{for } \epsilon > \epsilon^*, \\ \left(\frac{e^{\epsilon}+1}{e^{\epsilon}-1}\right)^2, & \text{for } \epsilon \le \epsilon^*. \end{cases} \tag{8}$$

Based on Equation 8, Lemma 1, and Equation 4, 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 I of Section I, where ϵ^* follows from Equation 6 and $\epsilon^{\#}$ is derived by solving ϵ 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 α satisfies Equation 7. 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,

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

The red line in Fig. 1 on Page 4 shows the worst-case noise variance incurred by HM, which is consistently no higher than

Algorithm 3: Duchi et al.'s Solution [14] for Multidimensional Numeric Data.

input : tuple $t_i \in [-1,1]^d$ and privacy parameter ϵ.
output: tuple $t_i^* \in \{-B, B\}^d$.

Generate a random tuple $v \in \{-1,1\}^d$ by sampling each $v[A_j]$ independently from the following distribution:

$$\Pr[v[A_j] = x] = \begin{cases} \frac{1}{2} + \frac{1}{2} t_i[A_j], & \text{if } x = 1 \\ \frac{1}{2} - \frac{1}{2} t_i[A_j], & \text{if } x = -1 \end{cases};$$

Let T^+ (resp. T^-) be the set of all tuples $t^* \in \{-B, B\}^d$ such that $t^* \cdot v \geq 0$ (resp. $t^* \cdot v \leq 0$);
Sample a Bernoulli variable u that equals 1 with $e^\epsilon/(e^\epsilon + 1)$ probability;
if $u = 1$ **then**
  return a tuple uniformly at random from T^+;
else
  return a tuple uniformly at random from T^-;

Algorithm 4: Our Method for Multiple Numeric Attributes.

input : tuple $t_i \in [-1,1]^d$ and privacy parameter ϵ.
output: tuple $t_i^* \in [-C \cdot d,\ C \cdot d]^d$.

Let $t_i^* = \langle 0, 0, \ldots, 0 \rangle$;
Let $k = \max\left\{1, \min\left\{d,\ \left\lfloor \frac{\epsilon}{2.5} \right\rfloor\right\}\right\}$;
Sample k values uniformly without replacement from $\{1, 2, \ldots, d\}$;
for *each sampled value j* **do**
  Feed $t_i[A_j]$ and $\frac{\epsilon}{k}$ as input to PM or HM, and obtain a noisy value $x_{i,j}$;
  $t_i^*[A_j] = \frac{d}{k} x_{i,j}$;
return t_i^*

those of all other three methods (HM reduces to Duchi et al.'s solution for $\epsilon \leq \epsilon^*$). 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 ϵ/d. Then, by the composition theorem [17], the collection of all attributes satisfies ϵ-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 ϵ/d. Then, by Lemma 2, the amount of noise in the estimated mean of each attribute is $O\left(\frac{d\sqrt{\log d}}{\epsilon\sqrt{n}}\right)$, 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 ϵ, and outputs a perturbed vector $t_i^* \in \{-B, B\}^d$, where B is a constant decided by d and ϵ. 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.

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:

$$C_d = \begin{cases} \frac{2^{d-1}}{\binom{d-1}{(d-1)/2}}, & \text{if } d \text{ is odd,} \\ \frac{2^{d-1} + \frac{1}{2}\binom{d}{d/2}}{\binom{d-1}{d/2}}, & \text{otherwise.} \end{cases} \tag{9}$$

Then, B is calculated by:

$$B = \frac{\exp(\epsilon) + 1}{\exp(\epsilon) - 1} \cdot C_d. \tag{10}$$

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

$$\mathbb{E}\left[\max_{j \in [1,d]} \left| \frac{1}{n}\sum_{i=1}^{n} t_i^*[A_j] - \frac{1}{n}\sum_{i=1}^{n} t_i[A_j] \right| \right] = O\left(\frac{\sqrt{d \log d}}{\epsilon\sqrt{n}}\right), \tag{11}$$

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

$$k = \max\left\{1, \min\left\{d,\ \left\lfloor \frac{\epsilon}{2.5} \right\rfloor\right\}\right\}. \tag{12}$$

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 $\frac{d}{k} \cdot x$, where x is generated by PM or HM given $t_i[A_j]$ and $\frac{\epsilon}{k}$ 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 ϵ/d to ϵ/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, which is shown in Equation 12. We derive the setting of k by minimizing the worst-case noise variance of Algorithm 4 when it utilizes PM (resp. HM)[3].

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

The proof appears in the full version [2].

By Lemma 4, the aggregator can use $\frac{1}{n}\sum_{i=1}^{n} t^*[A_j]$ 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 numeric data (see Equation 11), which has been proved 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 $Z[A_j] = \frac{1}{n}\sum_{i=1}^{n} t_i^*[A_j]$ and $X[A_j] = \frac{1}{n}\sum_{i=1}^{n} t_i[A_j]$. With at least $1-\beta$ probability,*

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

The proof appears in the full version [2].

Lemma 6 discusses the noise variances induced by Duchi et al.'s solution, PM and HM, respectively.

Lemma 6. *For a d-dimensional numeric tuple t_i which is perturbed as t_i^* under ϵ-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*

$$Var_D\left[t_i^*[A_j]\right] = \left(\tfrac{e^\epsilon+1}{e^\epsilon-1}\right)^2 {C_d}^2 - \left(t_i[A_j]\right)^2, \tag{13}$$

where C_d is defined by Equation 9. Meanwhile, the variance of $t_i^*[A_j]$ induced by PM is

$$Var_P\left[t_i^*[A_j]\right] = \tfrac{d(e^{\epsilon/(2k)}+3)}{3k(e^{\epsilon/(2k)}-1)^2} + \left[\tfrac{d\cdot e^{\epsilon/(2k)}}{k(e^{\epsilon/(2k)}-1)} - 1\right]\left(t_i[A_j]\right)^2; \tag{14}$$

and the variance of $t_i^*[A_j]$ induced by HM is

$$Var_H\left[t_i^*[A_j]\right] = \begin{cases} \frac{d}{k}\left[\frac{e^{\epsilon/(2k)}+3}{3e^{\epsilon/(2k)}(e^{\epsilon/(2k)}-1)} + \frac{(e^{\epsilon/k}+1)^2}{e^{\epsilon/(2k)}(e^{\epsilon/k}-1)^2}\right] + \left(\frac{d}{k}-1\right)\left(t_i[A_j]\right)^2, & \text{for } \epsilon/k > \epsilon^*, \\ \frac{d}{k}\left(\frac{e^{\epsilon/k}+1}{e^{\epsilon/k}-1}\right)^2 + \left(\frac{d}{k}-1\right)\left(t_i[A_j]\right)^2, & \text{for } \epsilon/k \le \epsilon^*, \end{cases} \tag{15}$$

where ϵ^* is defined by Equation 6.

From Equations 13, 14, and 15, we can prove the following:

[3]We discuss how to obtain the value of k in the full version [2].

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 $\epsilon > 0$,*

$$\max_{t_i[A_j]\in[-1,1]} Var_H\left[t_i^*[A_j]\right] < \max_{t_i[A_j]\in[-1,1]} Var_P\left[t_i^*[A_j]\right] < \max_{t_i[A_j]\in[-1,1]} Var_D\left[t_i^*[A_j]\right]. \tag{16}$$

To illustrate Corollary 2, Fig. 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 ϵ. Observe that for $d = 5, 10, 20, 40$, the wort-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.

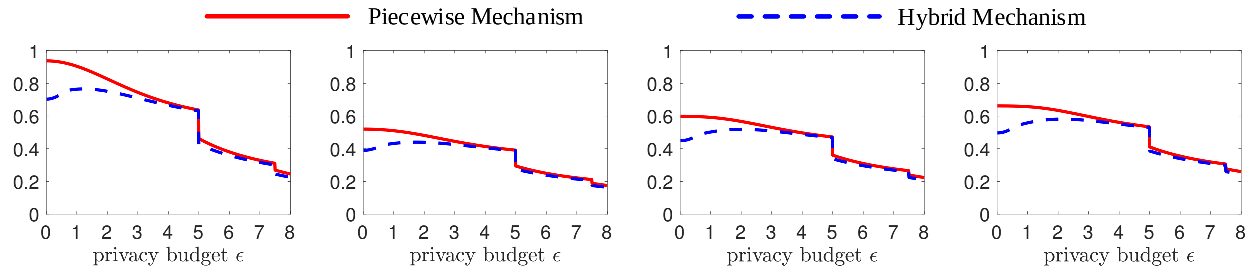
C. Handling Categorical Attributes

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 II 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], [38]) 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 ϵ and a one-dimensional tuple t_i with a categorical attribute A, and outputs a perturbed tuple t_i^* while ensuring ϵ-LDP. In addition, we assume there is a function $g(x, y)$ that $g(x, y) = 1$ if $x = y$; and 0, otherwise. Then for any value $v \in A$, $\frac{1}{n}\sum_{i\in S} g(t_i^*, v)$ is an estimator of the frequency of v over all users, where S is the set of $\{1, 2, \ldots n\}$. 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 12;
2) For each sampled j, if A_j is a numerical attribute, then submit a noisy version of $t[A_j]$ computed as in Lines 3–5 of Algorithm 4; otherwise (i.e., A_j is a categorical attribute), submit $f(t[A_j], \epsilon/k)$, where f can be any existing solution for perturbing a single categorical attribute under ϵ-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 $\frac{d}{kn}\sum_{v^*\in V^*} g(v^*, v)$, 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 unary



(a) $d = 5$ (b) $d = 10$ (c) $d = 20$ (d) $d = 40$

Fig. 3: The worst-case variance of PM (resp. HM) as a fraction of the worst-case variance of Duchi et al.'s solution, for various privacy budget ϵ and data dimensionality d.

encoding (OUE) protocol of Wang et al. [38] 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 ϵ-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 $\langle x_i, y_i \rangle$, where $x_i \in [-1,1]^d$ and $y_i \in [-1,1]$ (for linear regression) or $y_i \in \{-1,1\}$ (for logistic regression and SVM classification). Let $\ell(\cdot)$ be a *loss function* that maps a d-dimensional *parameter vector* β into a real number, and is parameterized by x_i and y_i. We aim to identify a parameter vector β^* such that

$$\beta^* = \arg\min_{\beta} \left[\frac{1}{n} \left(\sum_{i=1}^{n} \ell(\beta; x_i, y_i) \right) + \frac{\lambda}{2} \|\beta\|_2^2 \right],$$

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

For convenience, we define

$$\ell'(\beta; x_i, y_i) = \ell(\beta; x_i, y_i) + \frac{\lambda}{2} \|\beta\|_2^2.$$

The proposed approach solves β^* using SGD, which starts from an initial parameter vector β_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 $\langle x, y \rangle$ is the data record of a randomly selected user, $\nabla \ell'(\beta_t; x, y)$ is the gradient of ℓ' at β_t, and γ_t is called the *learning rate* at the t-th iteration. The learning rate γ_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 β_{t+1} and β_t is sufficiently small. Under ϵ-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], [22] 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 using Algorithm 4. Here, if any entry of $\nabla \ell_i$ is greater than 1 (resp. smaller than -1), then the user should clip it to 1 (resp. -1) before perturbation, where $\nabla \ell_i$ is the gradient generated by the i-th user in group G. That is a common technique referred to as "gradient clipping" in the deep learning literature. After that, we update the parameter vector β_t with the mean of the noisy gradients, i.e.,

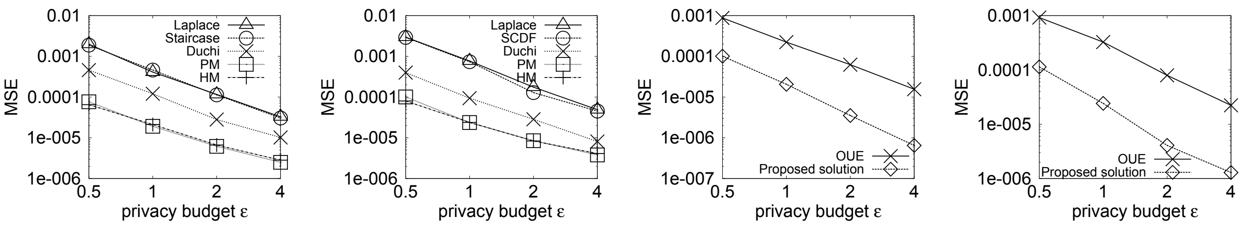
$$\beta_{t+1} = \beta_t - \gamma_t \cdot \tfrac{1}{|G|} \textstyle\sum_{i=1}^{|G|} \nabla \ell_i^*,$$

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 $O\left(\frac{\sqrt{d \log d}}{\epsilon \sqrt{|G|}}\right)$, 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 ϵ_i-differential privacy. By the composition property of differential privacy [29], if we enforce ϵ-differential privacy for the user's data, we should have $\sum_{i=1}^{m} \epsilon_i \le \epsilon$. 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.

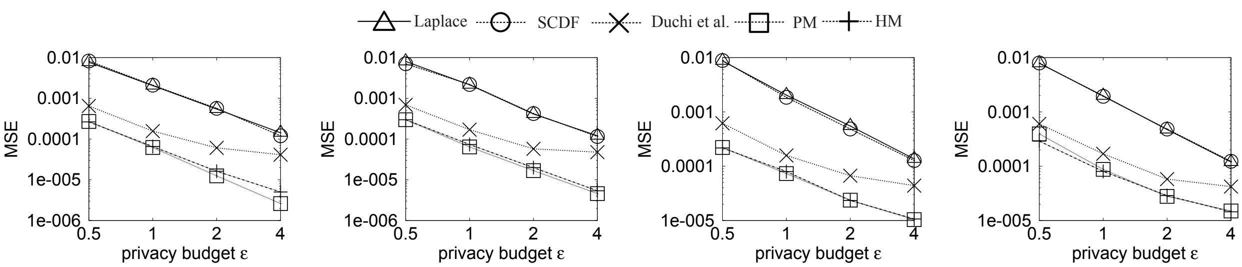
VI. Experiments

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



(a) BR-Numeric (b) MX-Numeric (c) BR-Categorical (d) MX-Categorical

Fig. 4: Result accuracy for mean estimation (on numeric attributes) and frequency estimation (on categorical attributes).



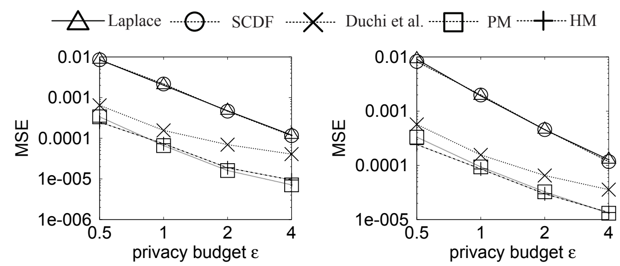
(a) $\mu = 0$ (b) $\mu = 1/3$ (c) $\mu = 2/3$ (d) $\mu = 1$

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

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

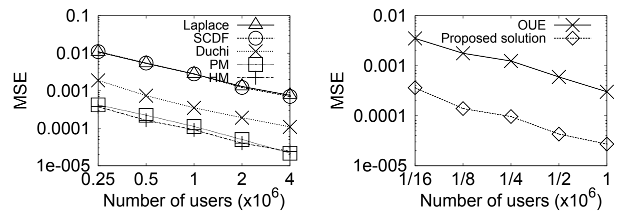
In the first set of experiments, we consider the task of collecting a noisy, multidimensional tuple from each user, in order to estimate the mean of each numerical attribute and the frequency of each categorical value. Since no existing solution can directly support this task, we take the following best-effort approach combining state-of-the-art solutions through the composition property of differential privacy [29]. Specifically, let t be a tuple with d_n numeric attributes and d_c categorical attributes. Given total privacy budget ϵ, 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, (ii) the Laplace mechanism or (iii) SCDF [35], which is applied to each numeric attribute individually using ϵ/d budget. The Staircase mechanism leads to similar performance as SCDF, and we omit its results for brevity. Regarding categorical attributes, since no previous solution addresses the multidimensional case, we apply the optimized unary encoding (OUE) protocol of Wang et al. [38], the state of the art for frequency estimation on a single categorical attribute, to each attribute independently with ϵ/d budget. Clearly, by the composition property of differential privacy [29], the above approach satisfies ϵ-LDP.



(a) Uniform distribution (b) Power law distribution

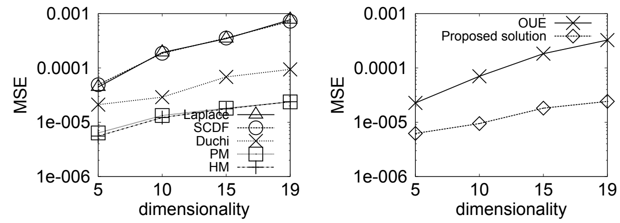
Fig. 6: Result accuracy vs. privacy budget on different data distributions.

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). Fig. 4 plots the MSE results as a function of the total privacy budget ϵ. 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 involves privacy budget splitting, which is sub-optimal. For instance, on the categorical attributes, applying OUE [38] on each attribute individually leads to $O\left(\frac{d}{\epsilon\sqrt{n}}\right)$ error (where n is the number of users), 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 (SCDF mechanism) on numeric attributes.



(a) Numeric (b) Categorical

Fig. 7: Result accuracy vs. number of users.



(a) Numeric (b) Categorical

Fig. 8: Result accuracy vs. dimensionality.

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 OUE 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 several synthetic datasets. Specifically, the first 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 u and standard deviation of $1/4$, but discarding any value that fall out of $[-1, 1]$. Fig. 5 shows the results with varying privacy budget ϵ, and 4 different values for u. In all settings, PM and HM outperform Duchi et al.'s solution, and the performance gap slightly expands with increasing ϵ, 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.

We repeat the experiments on two additional synthetic datasets with the same properties as the first synthetic one, except that their attribute values are drawn from different distributions. One follows the uniform distribution where each attribute value is sampled from $[-1, 1]$ uniformly; the other one follows the power law distribution where each attribute value x is sampled from $[-1, 1]$ with probability proportional to $c \cdot (x + 2)^{-10}$. Fig. 6 presents the results, which lead to similar conclusions as the results on real and Gaussian-distributed data.

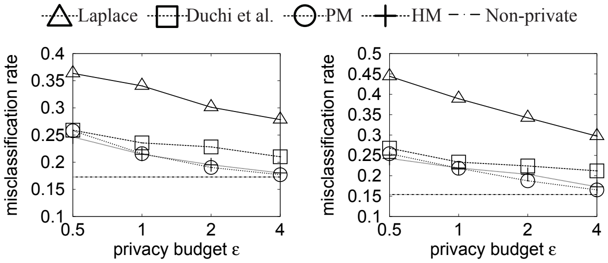
Lastly, Figs. 7 and 8 show the result accuracy in terms of MSE with varying the number of users n and dimensionality d on the MX dataset. Observe that more users and lower dimensionality both lead to more accurate results, which agrees with the theoretical analysis in Lemma 5. Meanwhile, in all settings the proposed solutions consistently outperform their competitors by clear margins. In the next subsection, we omit the results for SCDF, which are comparable to that of the Laplace mechanism.

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 $\{0, 1\}$, such that (i) the l-th ($l < k$) value in A_j is represented by 1 on the l-th binary attribute and 0 on each of the remaining $k-2$ attributes, and (ii) the k-th value in A_j is represented by 0 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.

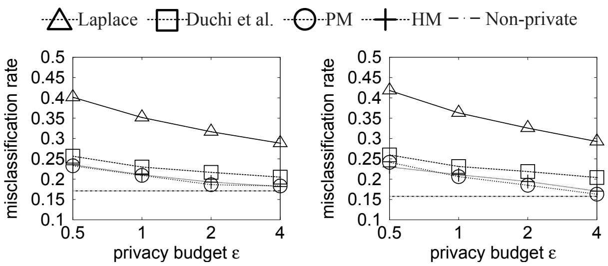
Since each user sends gradients to the aggregator, which are all numeric, the experiment involves the 4 competitors in Section VI-A for numeric data: PM, HM, Duchi et al. [14], and the Laplace mechanism applied to each attribute independently with equally split privacy budget (i.e., ϵ/d for each attribute). Additionally, we also include the result in the non-private setting. For all methods, we set the regularization factor $\lambda = 10^{-4}$. On each dataset, we use 10-fold cross validation 5 times to assess the performance of each method.

Fig. 9 and Fig. 10 show the misclassification rate of each method for logistic regression and SVM classification, respectively, with varying values of the privacy budget ϵ. 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 ϵ 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



(a) BR (b) MX

Fig. 9: Logistic Regression.



(a) BR (b) MX

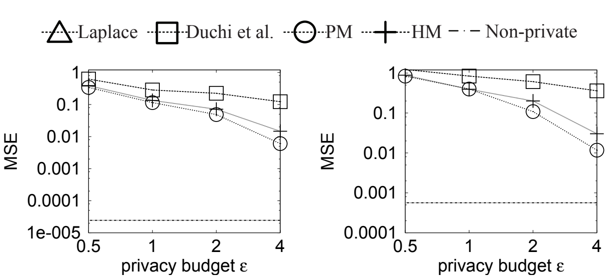
Fig. 10: Support Vector Machines (SVM).

no clear trend for the performance gap between PM/HM and Duchi et al.'s solution.

Fig. 11 demonstrates the mean squared error (MSE) of the linear regression model generated by each method with varying ϵ. 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], [27], machine learning [4], theory [5], [13], [25], and systems [6]. Earlier models of differential privacy [16], [17], [29] 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], [25]), which eliminates the data curator and the collection of exact private information.



(a) BR (b) MX

Fig. 11: Linear Regression.

LDP can be connected to the classical randomized response technique in surveys [41]. 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. Wang et al. [38] 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. Note that all of the above methods focus on a single categorical attribute, and, thus, are orthogonal to our work on multidimensional data including numeric attributes. Ren et al. [34] investigate the problem of publishing multiple attributes, and employ the idea of k-sized vector, similar to [38]. This approach, however, incurs rather high communication costs between the aggregator and the users, since it involves the transmission of multiple k-sized vectors. 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. Besides, Kairouz et al. [24] propose the extremal mechanisms, which are a family of LDP mechanisms for data with discrete inputs, i.e., each input domain $\mathcal{X}$ contains a finite number of possible values. These mechanisms have an output distribution pdf with a key property: for any input $x \in \mathcal{X}$ and any output y, $Pr[y \mid x]$ has only two possible values that differ by a factor of $\exp(\epsilon)$. Kairouz et al. show that for any given utility measure, there exists an extremal mechanism with optimal utility under this measure, using a linear program with $2^{|\mathcal{X}|}$ variables. It is unclear how to apply extremal mechanisms to continuous input domains with an infinite number of possible values, which is the focus on this paper.

Various data analytics and machine learning problems have been studied under LDP, such as probability distribution estimation [15], [23], [30], [32], [42], heavy hitter discovery [4], [7], [33], [39], frequent new term discovery [37], frequency estimation [5], [38], frequent itemset mining [40], marginal release [10], clustering [31], 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 [26], inference control on mobile sensing [28], and the publication of crowd-sourced data [34].

VIII. Conclusion

This work systematically investigates the problem of collecting and analyzing users' personal data under ϵ-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. In the next step, we plan to apply the proposed solution to more complex data analysis tasks such as deep neural networks.

Acknowledgment

This research was supported by National Natural Science Foundation of China (61672475, 61433008), by Samsung via a GRO grant, by the National Research Foundation, Prime Minister's Office, Singapore under its Strategic Capability Research Centres Funding Initiative and by Nanyang Technological University Startup Grant (M4082311.020). This publication was made possible by NPRP grant 10-0208-170408 from the Qatar National Research Fund (a member of Qatar Foundation). The findings herein reflect the work, and are solely the responsibility, of the authors. We thank T. Nguyen for his contributions to an earlier version of this work.

References

[1] Integrated public use microdata series, 2017. https://www.ipums.org.
[2] Collecting and analyzing multidimensional data with local differential privacy (technical report), 2018. https://sites.google.com/site/icde2019tr/tr.
[3] B. Avent, A. Korolova, D. Zeber, T. Hovden, and B. Livshits. BLENDER: Enabling local search with a hybrid differential privacy model. In *USENIX Security Symposium*, pages 747–764, 2017.
[4] R. Bassily, K. Nissim, U. Stemmer, and A. Thakurta. Practical locally private heavy hitters. In *NIPS*, pages 2285–2293, 2017.
[5] R. Bassily and A. Smith. Local, private, efficient protocols for succinct histograms. In *ACM STOC*, pages 127–135, 2015.
[6] A. Bittau, U. Erlingsson, P. Maniatis, I. Mironov, A. Raghunathan, D. Lie, M. Rudominer, U. Kode, J. Tinnes, and B. Seefeld. PROCHLO: Strong privacy for analytics in the crowd. In *ACM SOSP*, pages 441–459, 2017.
[7] M. Bun, J. Nelson, and U. Stemmer. Heavy hitters and the structure of local privacy. In *ACM PODS*, pages 435–447, 2018.
[8] R. Chen, H. Li, A. K. Qin, S. P. Kasiviswanathan, and H. Jin. Private spatial data aggregation in the local setting. In *IEEE ICDE*, pages 289–300, 2016.
[9] G. Cormode, S. Jha, T. Kulkarni, N. Li, D. Srivastava, and T. Wang. Privacy at scale: Local differential privacy in practice. In *ACM SIGMOD Tutorials*, 2018.
[10] G. Cormode, T. Kulkarni, and D. Srivastava. Marginal release under local differential privacy. In *ACM SIGMOD*, 2018.
[11] C. Cortes and V. Vapnik. Support-vector networks. *Machine Learning*, 20(3):273–297, 1995.
[12] B. Ding, J. Kulkarni, and S. Yekhanin. Collecting telemetry data privately. In *NIPS*, pages 3574–3583, 2017.
[13] J. C. Duchi, M. I. Jordan, and M. J. Wainwright. Local privacy and statistical minimax rates. In *IEEE FOCS*, pages 429–438, 2013.
[14] J. C. Duchi, M. I. Jordan, and M. J. Wainwright. Minimax optimal procedures for locally private estimation. *Journal of the American Statistical Association*, 113(521):182–201, 2018.
[15] J. C. Duchi, M. J. Wainwright, and M. I. Jordan. Local privacy and minimax bounds: Sharp rates for probability estimation. In *NIPS*, pages 1529–1537, 2013.
[16] C. Dwork, F. McSherry, K. Nissim, and A. Smith. Calibrating noise to sensitivity in private data analysis. In *TCC*, pages 265–284, 2006.
[17] C. Dwork and A. Roth. The algorithmic foundations of differential privacy. *Foundations and Trends in Theoretical Computer Science*, 9(3-4):211–407, 2014.
[18] Ú. Erlingsson, V. Pihur, and A. Korolova. RAPPOR: Randomized aggregatable privacy-preserving ordinal response. In *ACM CCS*, pages 1054–1067, 2014.
[19] G. Fanti, V. Pihur, and Ú. Erlingsson. Building a RAPPOR with the unknown: Privacy-preserving learning of associations and data dictionaries. *PoPETs*, 2016(3):41–61, 2016.
[20] M. Gaboardi and R. Rogers. Local private hypothesis testing: Chi-square tests. *arXiv preprint arXiv:1709.07155*, 2017.
[21] Q. Geng, P. Kairouz, S. Oh, and P. Viswanath. The staircase mechanism in differential privacy. *J. Sel. Topics Signal Processing*, 9(7):1176–1184, 2015.
[22] J. Hamm, A. C. Champion, G. Chen, M. Belkin, and D. Xuan. Crowd-ML: A privacy-preserving learning framework for a crowd of smart devices. In *IEEE ICDCS*, pages 11–20, 2015.
[23] P. Kairouz, K. Bonawitz, and D. Ramage. Discrete distribution estimation under local privacy. In *ICML*, 2016.
[24] P. Kairouz, S. Oh, and P. Viswanath. Extremal mechanisms for local differential privacy. In *NIPS*, pages 2879–2887, 2014.
[25] S. P. Kasiviswanathan, H. K. Lee, K. Nissim, S. Raskhodnikova, and A. Smith. What can we learn privately? In *IEEE FOCS*, pages 531–540, 2008.
[26] J. W. Kim, D.-H. Kim, and B. Jang. Application of local differential privacy to collection of indoor positioning data. *IEEE Access*, 6:4276–4286, 2018.
[27] S. Krishnan, J. Wang, M. J. Franklin, K. Goldberg, and T. Kraska. PrivateClean: Data cleaning and differential privacy. In *ACM SIGMOD*, pages 937–951, 2016.
[28] C. Liu, S. Chakraborty, and P. Mittal. DEEProtect: Enabling inference-based access control on mobile sensing applications. *arXiv preprint arXiv:1702.06159*, 2017.
[29] F. McSherry and K. Talwar. Mechanism design via differential privacy. In *IEEE FOCS*, pages 94–103, 2007.
[30] T. Murakami, H. Hino, and J. Sakuma. Toward distribution estimation under local differential privacy with small samples. *PoPETs*, 2018(3):84–104, 2018.
[31] K. Nissim and U. Stemmer. Clustering algorithms for the centralized and local models. In *Algorithmic Learning Theory (ALT)*, pages 619–653, Apr 2018.
[32] A. Pastore and M. Gastpar. Locally differentially-private distribution estimation. In *IEEE International Symposium on Information Theory (ISIT)*, pages 2694–2698, 2016.
[33] Z. Qin, Y. Yang, T. Yu, I. Khalil, X. Xiao, and K. Ren. Heavy hitter estimation over set-valued data with local differential privacy. In *ACM CCS*, pages 192–203, 2016.
[34] X. Ren, C. M. Yu, W. Yu, S. Yang, X. Yang, J. A. McCann, and P. S. Yu. LoPub: High-dimensional crowdsourced data publication with local differential privacy. *IEEE Transactions on Information Forensics and Security*, 13(9):2151–2166, Sept 2018.
[35] J. Soria-Comas and J. Domingo-Ferrer. Optimal data-independent noise for differential privacy. *Inf. Sci.*, 250:200–214, 2013.
[36] J. Tang, A. Korolova, X. Bai, X. Wang, and X. Wang. Privacy loss in Apple's implementation of differential privacy on macOS 10.12. *arXiv preprint arXiv:1709.02753*, 2017.
[37] N. Wang, X. Xiao, Y. Yang, T. D. Hoang, H. Shin, J. Shin, and G. Yu. PrivTrie: Effective frequent term discovery under local differential privacy. In *IEEE ICDE*, 2018.
[38] T. Wang, J. Blocki, N. Li, and S. Jha. Locally differentially private protocols for frequency estimation. In *USENIX Security*, 2017.
[39] T. Wang, N. Li, and S. Jha. Locally differentially private heavy hitter identification. *arXiv preprint arXiv:1708.06674*, 2017.
[40] T. Wang, N. Li, and S. Jha. Locally differentially private frequent itemset mining. In *IEEE Symposium on Security and Privacy (S&P)*, 2018.
[41] S. L. Warner. Randomized response: A survey technique for eliminating evasive answer bias. *Journal of the American Statistical Association*, 60(309):63–69, 1965.
[42] M. Ye and A. Barg. Optimal schemes for discrete distribution estimation under local differential privacy. In *IEEE International Symposium on Information Theory (ISIT)*, pages 759–763, 2017.