Estimating Numerical Distributions under Local Differential Privacy

Zitao Li[1], Tianhao Wang[1], Milan Lopuhaä-Zwakenberg[2],
Ninghui Li[1], Boris Škorić[2]

[1]Purdue University, [2]Eindhoven University of Technology
{li2490,wang2842,ninghui}@purdue.edu,{m.a.lopuhaa,b.skoric}@tue.nl

ABSTRACT

When collecting information, local differential privacy (LDP) relieves the concern of privacy leakage from users' perspective, as user's private information is randomized before sent to the aggregator. We study the problem of recovering the distribution over a numerical domain while satisfying LDP. While one can discretize a numerical domain and then apply the protocols developed for categorical domains, we show that taking advantage of the numerical nature of the domain results in better trade-off of privacy and utility. We introduce a new reporting mechanism, called the square wave (SW) mechanism, which exploits the numerical nature in reporting. We also develop an Expectation Maximization with Smoothing (EMS) algorithm, which is applied to aggregated histograms from the SW mechanism to estimate the original distributions. Extensive experiments demonstrate that our proposed approach, SW with EMS, consistently outperforms other methods in a variety of utility metrics.

ACM Reference Format: Zitao Li, Tianhao Wang, Milan Lopuhaä-Zwakenberg, and Ninghui Li, Boris Škorić. 2020. Estimating Numerical Distributions under Local Differential Privacy. In *Proceedings of the 2020 ACM SIGMOD International Conference on Management of Data (SIGMOD'20), June 14–19, 2020, Portland, OR, USA.* ACM, NY, NY, USA, 15 pages. https://doi.org/10.1145/3318464.3389700

SIGMOD '20, June 14–19, 2020, Portland, OR, USA

ACM ISBN 978-1-4503-6735-6/20/06…$15.00
https://doi.org/10.1145/3318464.3389700

1 INTRODUCTION

Differential privacy [11] has been accepted as the *de facto* standard for data privacy. Recently, techniques for satisfying differential privacy (DP) in the local setting, which we call LDP, have been studied and deployed. In the local setting for DP, there are many *users* and one *aggregator*. Each user sends randomized information to the aggregator, who attempts to infer the data distribution based on users' reports. LDP techniques enable the gathering of statistics while preserving privacy of every user, without relying on trust in a single trusted third party. LDP techniques have been deployed by companies like Apple [32], Google [14], and Microsoft [9].

Most existing work on LDP focuses on the situations collecting categorical attributes. Existing research [1, 5, 14, 35, 41] has developed frequency oracle (FO) protocols for categorical domains, where the aggregator can estimate the frequency of any chosen value in the specified domain (fraction of users with that private value). We call these Categorical Frequency Oracle (CFO) protocols.

Many attributes are ordinal or numerical in nature, e.g., income, age, the amount of time viewing a certain page, the amount of communications, the number of times performing a certain actions, etc. A numerical domain consists of values that have a meaningful total order. One natural approach for dealing with ordinal and numerical attributes under LDP is to first apply binning and then use CFO protocols. That is, one treats all values in a range as one categorical value when reporting. This approach faces the challenge of finding the optimal number of bins, which depends on both the privacy parameter and the data distribution. One improvement over this approach is to apply Hierarchical Histogram-based approaches [17, 27, 40], which uses multiple granularities at the same time, and exploit the natural consistency relationships between estimations at different granularities. Recently, Kulkarni et al. [21] studied the accuracy of answering range queries using this approach.

We note that the stronger privacy guarantee offered by LDP (as compared with DP) comes with the cost of significantly higher noises. As a result, many estimated frequencies

will be negative. Existing approaches (such as [21]) do not correct this, and are sub-optimal. We propose to apply Alternating Direction Method of Multipliers (ADMM) optimization [7] to improve Hierarchical Histograms, utilizing the constraints that all estimations are non-negative and sum up to 1. Experiments show that the improved version of hierarchy histogram, which we call HH-ADMM, has significantly better utility.

The above methods still use CFO protocols in a blackbox fashion, and existing CFO protocols ignore any semantic relationship between different values. An intriguing research question is whether one can design frequency oracle protocols that directly utilize the ordered nature of the domain and produce better estimations. In this paper, we answer this affirmatively. We propose an approach that combines what we call a Square Wave reporting mechanism with post-processing using Expectation Maximization and Smoothing.

The key intuition under the Square Wave mechanism is that given input v, one should report a value close to v with higher probability than a value farther away from v. More specifically, assuming the input domain of numerical values is $\mathscr{D} = [0, 1]$, the output domain of Square Wave mechanism is $\tilde{\mathscr{D}} = [-b, 1 + b]$, where b is a parameter depending on the privacy parameter ϵ. A user with value $v \in \mathscr{D}$ reports a value $\tilde{v}$ randomly drawn from a distribution with probability density function $\mathbf{M}_v$. For any $\tilde{v} \in [v - b, v + b]$, probability density is $\mathbf{M}_v(\tilde{v}) = p$, and any $\tilde{v} \in [-b, 1 + b] \setminus [v - b, v + b]$, probability density is $\mathbf{M}_v(\tilde{v}) = q$, where $\frac{p}{q} = e^\epsilon$. We define and study different wave shapes of General Wave mechanism other than the above Square Wave, and conclude that Square Wave has the best utility. We also study how to determine the key parameter b, the width of the wave. We propose to choose b to maximize the upper bound of mutual information between the input and the output variable, and can compute b when given the privacy parameter ϵ. Experiments demonstrate the effectiveness of this approach.

Conceptually, the aggregator, after observing the reported values, without any prior knowledge of the input distribution, should perform Maximum Likelihood Estimation (MLE) to infer the input distribution, which can be carried out by the Expectation Maximization (EM) algorithm. Through experiments, we have observed that the result of applying EM is highly sensitive to the parameter controlling terminating condition. This is because the observed distribution is a combination of the true distribution and the effect of random noise. When EM terminates too early, the result does not fit the true distribution well. When EM terminates too late, the result fits both the true distribution and the effect of noises. It is unclear how one can set the parameter so that one fits the distribution, but not the noise, across different datasets and privacy parameters.

To deal with this challenge, we propose to use smoothing together with the EM algorithm. In each iteration, after the E step and the M step, we add an S (smoothing) step, which averages each estimation with its nearest neighbours, by binomial coefficients. The Expectation Maximization with Smoothing approach was developed in the context of positron emission tomography and image reconstruction [24, 30], and was shown to be equivalent to adding a regularization term penalizing the spiky estimation [24]. Intuitively, EMS uses the prior knowledge that the observation is affected by noise and prefer a smoother distribution to a jagged one. In the experiment, we observe that EMS is stable under different settings, and requires no parameter tuning.

To compare different algorithms for reconstructing distributions of numerical attributes, we first use two metrics measuring the distance of reconstructed cumulative distribution from the true one, namely the Wasserstein distance and Kolmogorov–Smirnov distance (KS distance). In addition, we also consider accuracy for answering range queries, and accuracy of estimations of different statistics from the reconstructed distributions such as mean, variance and quantiles.

The contributions of this paper are as follows. (1) We define the problem of reconstructing distributions of numerical attributes under LDP (with non-negativity and sum-up-to-1 constraints) and propose multiple metrics for comparing competing algorithms. (2) We introduce HH-ADMM, which improves upon existing hierarchy histogram based methods. (3) We introduce the method of combining Square Wave (SW) reporting with Expectation Maximization and Smoothing (EMS), and showed that Square Wave is preferable to other wave shapes, and introduce techniques to choose the bandwidth parameter b using mutual information. (4) We conduct extensive experimental evaluations, comparing the proposed methods with state-of-the-art methods (e.g., [21]). Results demonstrate that SW with EMS and HH-ADAM significantly out-perform existing methods. In addition, SW with EMS generally performs the best under a wide range of metrics, and HH-ADMM performs better than SW-EMS on a very spiking distribution under some of the metrics.

Roadmap. In Section 2, we review the LDP definition and existing LDP protocols. In Section 3, we discuss metrics for measuring the quality of the reconstructed distribution. We describe CFO with binning and HH-ADMM in Section 4. SW reporting and EMS reconstruction are introduced in Section 5. We show our experimental results in Section 6. We give an overview of the related work in Section 7, and conclude in Section 8.

2 BACKGROUND

Assume there are *n users* and one *aggregator*. Each user possesses a value $v \in \mathscr{D}$, and the aggregator wants to learn

Symbol	Description
v	Private input
$\tilde{v}$	Randomized output
$\mathscr{D}$	Domain of private input
$\tilde{\mathscr{D}}$	Domain of the randomized output
$\mathbf{x}$	True private input frequencies
$\hat{\mathbf{x}}$	Estimate of private input frequencies (normalized)
$\tilde{\mathbf{v}}$	Randomized output frequencies (normalized)
$\mathbf{P}$	Cumulative distribution function (CDF)
$\mathbf{M}_v$	Probability density function given input v

Table 1: Notations.

the distribution of values from all users. To protect privacy, each user randomizes the input value v using an algorithm $\Psi(\cdot) : \mathscr{D} \to \tilde{\mathscr{D}}$, where $\tilde{\mathscr{D}}$ is the set of all possible outputs, and sends $\tilde{v} = \Psi(v)$ to the aggregator.

Definition 1 (ϵ-Local Differential Privacy). *An algorithm $\Psi(\cdot) : \mathscr{D} \to \tilde{\mathscr{D}}$ satisfies ϵ-local differential privacy (ϵ-LDP), where $\epsilon \geq 0$, if and only if for any input $v_1, v_2 \in \mathscr{D}$, we have*

$$\forall T \subseteq \tilde{\mathscr{D}} : \ \Pr[\Psi(v_1) \in T] \leq e^{\epsilon} \Pr[\Psi(v_2) \in T],$$

where $\tilde{\mathscr{D}}$ denotes the set of all possible outputs of Ψ.

Since a user never reveals v to the aggregator and reports only $\tilde{v} = \Psi(v)$, the user's privacy is still protected even if the aggregator is malicious.

Notational Conventions. Throughout the paper, we use bold letters to denote vectors. For example, $\mathbf{v} = \langle v_1, \ldots, v_n \rangle$ is all users' values, and $\mathbf{x} = \langle x_1, \ldots, x_d \rangle$ is frequencies of all values (i.e., $x_i = |\{j \mid v_j = i\}|/n$). If the notation is associated with a tilde (e.g., $\tilde{\mathbf{v}}$), it is the value after LDP perturbation; and a hat (e.g., $\hat{\mathbf{x}}$) denotes the value computed by the aggregator. Capital bold letters denote matrices and functions that take more than one input. Table 1 gives some of the frequently used symbols.

2.1 Categorical Frequency Oracles

A *frequency oracle (*FO*)* protocol enables the estimation of the frequency of any value $v \in \mathscr{D}$ under LDP. Existing protocols are designed for situations where $\mathscr{D}$ is a categorical domain. We call them *categorical frequency oracle* (CFO) protocols in this paper. The following are two commonly used CFO protocols.

Generalized Randomized Response (GRR**).** This CFO protocol generalizes the *randomized response* technique [39], and uses $\tilde{\mathscr{D}} = \mathscr{D}$. It uses as input perturbation function $\mathrm{GRR}(\cdot)$, where $\mathrm{GRR}(v)$ outputs the true value v with probability $p = \frac{e^{\epsilon}}{e^{\epsilon}+d-1}$, and any value $v' \neq v$ with probability $q = \frac{1-p}{d-1} = \frac{1}{e^{\epsilon}+d-1}$, where $d = |\mathscr{D}|$ is the domain size. To estimate the frequency of $v \in \mathscr{D}$ (i.e., the ratio of the users who have v as private value to the total number of users), one counts how many times v is reported, and denote the count as $C(v)$, and then computes

$$\tilde{x}_v = \frac{(C(v)/n) - q}{p - q},$$

where n is the total number of users. In [37], it is shown that this is an unbiased estimate of the true count, and the variance for this estimate is

$$\mathrm{Var}[\tilde{x}_v] = \frac{d - 2 + e^{\epsilon}}{(e^{\epsilon} - 1)^2 \cdot n}. \tag{1}$$

The variance given in (1) is linear to d; thus when the domain size d increases, the accuracy of this protocol is low.

Optimized Local Hashing (OLH**) [37].** This protocol deals with a large domain size $d = |\mathscr{D}|$ by first using a hash function to map an input value into a smaller domain of size g (typically $g \ll |\mathscr{D}|$), and then applying randomized response to the hashed value (which leads to $p = \frac{e^{\epsilon}}{e^{\epsilon}+g-1}$). In this protocol, both the hashing step and the randomization step result in information loss. The choice of the parameter g is a tradeoff between losing information during the hashing step and losing information during the randomization step. In [37], it is found that the optimal choice of g that leads to minimal variance is $(e^{\epsilon} + 1)$.

In OLH, one reports $\langle H, \mathrm{GRR}(H(v)) \rangle$, where H is randomly chosen from a family of hash functions that hash each value in $\mathscr{D}$ to $\{1 \ldots g\}$, and $\mathrm{GRR}(\cdot)$ is the perturbation function for Generalized Randomized Response, while operating on the domain $\{1 \ldots g\}$. Let $\langle H^j, y^j \rangle$ be the report from the j'th user. For each value $v \in \mathscr{D}$, to compute its frequency, one first computes $C(v) = |\{j \mid H^j(v) = y^j\}|$, and then transforms $C(v)$ to its unbiased estimate

$$\tilde{x}_v = \frac{(C(v)/n) - (1/g)}{p - 1/g}.$$

The approximate variance of this estimate is

$$\mathrm{Var}[\tilde{x}_v] = \frac{4e^{\epsilon}}{(e^{\epsilon} - 1)^2 \cdot n}.$$

Compared with (1), the factor $d - 2 + e^{\epsilon}$ is replaced by $4e^{\epsilon}$. This suggests that for smaller $|\mathscr{D}|$ (such that $|\mathscr{D}| - 2 < 3e^{\epsilon}$), GRR is better; but for large $|\mathscr{D}|$, OLH is better and has a variance that does not depend on $|\mathscr{D}|$.

2.2 Handling Numerical Attributes

Two methods have been proposed for mean estimation under LDP for numerical attributes. Note that using these methods one can estimate the mean, and not the distribution.

Stochastic Rounding (SR**) [10].** The main idea of Stochastic Rounding (SR) is that, no matter what is the input value v,

each user reports one of two extreme values, with probabilities depending on v. Here we give an equivalent description of the protocol. Following [10], we assume that the input domain is $[-1, 1]$. Given a value $v \in [-1, 1]$, let $p = \frac{e^\epsilon}{e^\epsilon+1}$ and $q = 1 - p = \frac{1}{e^\epsilon+1}$, the SR method outputs a random variable v', which takes the value -1 with probability $q + \frac{(p-q)(1-v)}{2}$ and value 1 with probability $q + \frac{(p-q)(1+v)}{2}$. Since

$$\begin{aligned}\mathbb{E}[v'] &= (-1)\left(q + \frac{(p-q)(1-v)}{2}\right) + q + \frac{(p-q)(1+v)}{2} \\ &= (p-q)v\ ,\end{aligned}$$

let $\tilde{v} = \frac{v'}{p-q}$, we have $\mathbb{E}[\tilde{v}] = v$; thus the mean of $\tilde{v}$ provides an unbiased estimate of the mean for the distribution.

Piecewise Mechanism (PM) [33]. In the Piecewise Mechanism, the input domain is $[-1, 1]$, and the output domain is $[-s, s]$, where $s = \frac{e^{\epsilon/2}+1}{e^{\epsilon/2}-1}$. For each $v \in [-1, 1]$, there is an associated range $[\ell(v), r(v)]$ where $-s \le \ell(v) < r(v) \le s$, such that with input v, a value in the range $[\ell(v), r(v)]$ will be reported with higher probability than a value outside the range. More precisely, we have $\ell(v) = \frac{e^{\epsilon/2} \cdot v - 1}{e^{\epsilon/2}-1}$ and $r(v) = \frac{e^{\epsilon/2} \cdot v + 1}{e^{\epsilon/2}-1}$. The width of the range is $r(v) - \ell(v) = \frac{2}{e^{\epsilon/2}-1}$, and the center is $\frac{\ell(v)+r(v)}{2} = \frac{e^{\epsilon/2}}{e^{\epsilon/2}-1} \cdot v$. Specifically, PM works as follows:

$$\Pr[\mathsf{PM}(v) = \tilde{v}] = \frac{e^{\epsilon/2}}{2} \cdot \frac{e^{\epsilon/2}-1}{e^{\epsilon/2}+1} \text{ if } \tilde{v} \in [\ell(v), r(v)],$$

$$\Pr[\mathsf{PM}(v) = \tilde{v}] = \frac{1}{2e^{\epsilon/2}} \cdot \frac{e^{\epsilon/2}-1}{e^{\epsilon/2}+1} \text{ otherwise.}$$

It is shown that $\tilde{v}$ is unbiased, and has better variance than SR when ϵ is large [33].

3 UTILITY METRICS

When the private values are in a numerical domain, we need utility metrics that are different from those in categorical domains. In particular, the metrics should reflect the ordered nature of the underlying domain.

3.1 Metrics based on Distribution Distance

We want a metric to measure the distance between the recovered density distribution and the true distribution. However, since the distribution is over a metric space, we do not want to use point-wise distance metrics such as the L_1 and L_2 distance or the Kullback–Leibler (KL) divergence. For a simple example, consider the case where $\mathscr{D} = \{1, 2, 3, 4\}$, the true distribution is $\mathbf{x} = [0.7, 0.1, 0.1, 0.1]$. The two estimations $\hat{\mathbf{x}}_1 = [0.1, 0.7, 0.1, 0.1]$ and $\hat{\mathbf{x}}_2 = [0.1, 0.1, 0.1, 0.7]$ have the same L_1, L_2, and KL distance from $\mathbf{x}$, but the distance between $\hat{\mathbf{x}}_1$ and $\mathbf{x}$ should be smaller than the distance between $\hat{\mathbf{x}}_2$ and $\mathbf{x}$ when we consider the numerical nature. To capture this requirement, we propose to use two popular distribution distances as metrics.

Wasserstein Distance (aka. Earth Mover Distance). Wasserstein distance measures the cost of moving the probability mass (or density) from distribution to another distribution. In this paper, we use the one dimensional Wasserstein distance. For discrete domain, define the cumulative function $\mathbf{P} : [0, 1]^d \times \mathscr{D} \mapsto [0, 1]$ that takes a distribution $\mathbf{x}$ and a value v, and output $\mathbf{P}(\mathbf{x}, v) = \sum_{i=1}^{v} x_v$. Let $\mathbf{x}$ and $\hat{\mathbf{x}}$ be two distributions. The one dimensional Wasserstein distance is the L_1 difference between their cumulative distributions:

$$W_1(\mathbf{x}, \hat{\mathbf{x}}) = \sum_{v \in \mathscr{D}} |\mathbf{P}(\mathbf{x}, v) - \mathbf{P}(\hat{\mathbf{x}}, v)|\ .$$

For continuous domain, $\mathbf{x}$ is the probability density function with support on $[0, 1]$, $\mathbf{P}(\mathbf{x}, v) = \int_{t=0}^{v} x(t)dt$. The one dimensional Wasserstein distance is

$$W_1(\mathbf{x}, \hat{\mathbf{x}}) = \int_{v \in \mathscr{D}} |\mathbf{P}(\mathbf{x}, v) - \mathbf{P}(\hat{\mathbf{x}}, v)|\ dv\ .$$

Kolmogorov-Smirnov (KS) Distance. KS distance is the maximum absolute difference at any point between the cumulative functions of two distributions:

$$d_{KS}(\mathbf{x}, \hat{\mathbf{x}}) = \sup_{v \in \mathscr{D}} |\mathbf{P}(\mathbf{x}, v) - \mathbf{P}(\hat{\mathbf{x}}, v)|\ .$$

Both of Wasserstein distance and KS distance can be considered as measures for the errors of answering prefix range queries on numerical domains with constraints that the estimate must be non-negative and sum up to 1. Wasserstein distance is the error of sum of all prefix queries; and the KS distance is the maximum error of prefix queries.

3.2 Semantic and Statistical Quantities

Range queries have been used as the main utility metrics for research in this area [17, 21, 36, 38]. Also, we consider the basic statistics from the estimated data distributions and check whether they are accurate.

Range Query. Define the range query function $\mathbf{R}(\mathbf{x}, i, \alpha) = \mathbf{P}(\mathbf{x}, i+\alpha) - \mathbf{P}(\mathbf{x}, i)$, where α specifies the range size. Given the true distribution $\mathbf{x}$ and the estimated distribution $\hat{\mathbf{x}}$, range queries reflect the quality of estimate with randomly sampling i and calculating the following:

$$|\mathbf{R}(\mathbf{x}, i, \alpha) - \mathbf{R}(\hat{\mathbf{x}}, i, \alpha)|\ .$$

Mean. We denote μ as the mean of the true distribution, and $\hat{\mu}$ as the estimated mean. To measure mean accuracy, we use the absolute value of the difference between these two, i.e. $|\mu - \hat{\mu}|$.

Variance. We use σ^2 to denote the variance of the true distribution, and $\hat{\sigma}^2$ for the variance from the reconstructed

distribution. To measure variance accuracy, we use the absolute value of the difference between these two, i.e. $|\sigma^2 - \hat{\sigma}^2|$.

Quantiles. Quantiles are cut points dividing the range of a probability distribution into intervals with equal probabilities. Formally, $\mathbf{Q}(\mathbf{x}, \beta) = \arg\max_v \{\mathbf{P}(\mathbf{x}, v) \leq \beta\}$. In the experiment, define $B = \{10\%, 20\%, \ldots, 90\%\}$, we measure the following:

$$\frac{1}{|B|} \sum_{\beta \in B} |\mathbf{Q}(\mathbf{x}, \beta) - \mathbf{Q}(\hat{\mathbf{x}}, \beta)| \ .$$

4 USING CFO PROTOCOLS FOR NUMERICAL DOMAINS

In this section, we present two approaches that use CFO protocols to reconstruct distributions over an discrete numerical domain $\mathscr{D} = \{1, 2 \cdots, d\}$. Continuous numerical domains can be buckized into discrete ones.

4.1 CFO with Binning

Given a numerical domain, one can make it discrete using binning, and then have each user report which bin the private value is in using a CFO protocol. For a given domain size and privacy parameter ϵ, one chooses either OLH or GRR, based on which one gives lower estimation variance. After obtaining density estimations for all the bins, one computes a density distribution for the domain by assuming uniform distribution within each bin. However, some estimated values may be negative, which does not lead to valid cumulative distribution functions on the domain. In [38], it is shown that a post-processing method called Norm-Sub can be applied to improve estimation. Norm-sub converts negative estimates to 0 and subtracts the same amount to all the positive estimates so that they sum up to 1. If some positive estimates become negative after the subtraction, the process is repeated. This results in an estimation such that each estimation is non-negative and all estimations sum up to 1. It can thus be interpreted as a probability distribution.

Challenge of Choosing Bin Size. When using binning, there are two sources of errors: noise and bias due to grouping values together. More bins lead to greater error due to noises. Fewer bins lead to greater error due to biases. Choosing the bin size is a trading-off of the above two sources of errors, and the effect of each choice depends both on the privacy parameter ϵ, and on property of the distribution. For example, when a distribution is smooth, one would prefer using less bins, as the bias error is small, and when a distribution is spiky, using more bins would perform better. In our experiments, we observe that even if we could choose the optimal bin size empirically for each dataset and ϵ value (which is infeasible to do in practice due to privacy), the result would still be worse than the method to be proposed in Section 5. We thus chose not to develop ways to choose bin size based on ϵ, and just report results of this method under several different bin sizes.

4.2 Hierarchy-based Methods

Hierarchy-based methods, including Hierarchy Histogram (HH) in [17, 27] and Haar in [40], were first proposed in the centralized setting of DP. In [21], Kulkarni et al. studied the HH method and the Haar in the context of LDP. In order to adapt Haar method to the local setting, they used Hadamard random response (HRR) as the frequency oracle. HRR is similar to Local Hashing method introduced in the Section 2.1, but fixing $g = 2$ and using a Hadamard matrix as the family of hash functions. To make it clear in the context, we call the LDP version of Haar as HaarHRR.

HH in LDP. Given a positive integer β and a discrete, ordered domain with size $d = |\mathscr{D}|$, one can construct a β-ary tree with d leaves corresponding to values in $\mathscr{D}$. There are $(h + 1)$ layers in the tree, where $h = \log_\beta d$ (for simplicity, we assume that $\log_\beta d$ is an integer). The $(h + 1)$-th layer is the root. A user with value v chooses a layer $\ell \in \{1, \ldots, h\}$ uniformly at random, and then reports ℓ as well as the perturbed value of v's ancestor node at layer ℓ. For each node in the tree, the aggregator can obtain an estimate of its frequency. Assuming that the distribution differences among the h groups are negligible, for each parent-child relation, one expects that the sum of child estimations equals the that of the parent. Constrained inference techniques [17] are applied to ensure this property.

HaarHRR. Similar to HH, one can use a binary tree to estimate distribution with Discrete Haar Transform [21]. Specifically, each leaf represents the frequency of a value. Define the height of a leaf node as 0; and the height of an inner nodes a is denotes as $h(a)$. Each inner node now represents the Haar coefficient $c_a = \frac{C_l^{(a)} - C_r^{(a)}}{2^{h(a)/2}}$, where $C_l^{(a)}$ (or $C_r^{(a)}$) is the sum of all leaves of left (or right) subtree of node a.

In the LDP setting, for a user with value v, the Haar coefficients on each layer has exactly one element equal to -1 or 1, while others are all zeros. Similar to HH, each user chooses a layer $\ell \in \{1, \ldots, h\}$ uniformly at random, then apply Hadamard randomized response (HRR) on layer ℓ which depends on Hadamard matrix $\phi \in \{-1, 1\}^{2^{h-\ell} \times 2^{h-\ell}}$. With HRR reports from users, the aggregator can calculate unbiased estimates for the Haar coefficients on layer ℓ. Due the limit of space, more details can be found in [21].

Difference from the Centralized Setting. When using hierarchy-based method, there are two ways to ensure the privacy constraint. One is to divide the privacy budget, where one builds a single tree for all values. Since each value affects the counts at every level, one splits the privacy budget among

the levels. The other is to divide the population among the layers, where each value contributes to the estimation of a single layer, and one can use the whole privacy budget for each count. When dividing the population, the absolute level of noise is less than the case of dividing privacy budget; however, the total count also decreases, magnifying the impact of noise. In addition, dividing the population introduces sampling errors, as users are divided into different groups, which may have different distribution from the global one.

In the centralized setting, because the amount of added noise is low, it is better to divide the privacy budget, as one avoids sampling errors. In [27], it was found that in the centralized setting, the optimal branching factor for HH is around 16. And this results in better performance than using the Haar method, which can be applied only to a binary hierarchy. In the LDP setting, because the amount of noise is much larger, sampling errors can be mostly ignored, and it is better to divide the population instead of privacy budget. As a result, the optimal branching factor for HH is around 5, making it similar to the Haar method. This was theoretically proved and empirically demonstrated in [21, 36].

4.3 HH-ADMM

We note that there are other ways to improve hierarchy-based mechanism in the LDP setting. First, the larger noise in the LDP setting results in negative estimates. We can exploit the prior knowledge that the true counts are non-negative to improve the negative estimates. Second, the total true count is known, as LDP protects privacy of reported values and not the fact that one is reporting. These are not exploited in [21]. We propose to use the Alternating Direction Method of Multipliers (ADMM) algorithm [7] to post-process the hierarchy estimation. The usage of ADMM was proposed in [22] for the centralized setting. Our method applies this to LDP, and has two additional differences from [22]. First, we use L_2 norm in the objective function because the noise by CFO is well approximated by Gaussian noise, and minimizing L_2 norm achieves MLE. In the centralized setting, Laplace noise is used, and L_1 norm is minimized in [22]. Second, we pose an additional constraint that the estimates sum up to n, which is known in LDP setting. In the setting considered in [22], n is unknown.

The HH-ADMM Algorithm. Given a constant vector $\tilde{\mathbf{x}}$, ADMM is an efficient algorithm that aims to find $\hat{\mathbf{x}}$ that satisfies the following optimization problem:

$$\begin{aligned} \text{minimize} \quad & \frac{1}{2}\|\hat{\mathbf{x}} - \tilde{\mathbf{x}}\|_2^2 \\ \text{subject to} \quad & \mathbf{A}\hat{\mathbf{x}} = 0, \quad \hat{\mathbf{x}} \succcurlyeq 0, \quad \hat{\mathbf{x}}_0 = 1 \end{aligned} \tag{2}$$

In the hierarchy histogram case of LDP, $\tilde{\mathbf{x}}$ represents the concatenation of estimates from all the layers, where $\tilde{\mathbf{x}}_0$ is the root. $\hat{\mathbf{x}}$ is the post-processed estimates. The hierarchical constraints state that the estimate of each internal node should be equal to the sum of estimates of its children nodes. This can be represented by an equation $\mathbf{A}\hat{\mathbf{x}} = 0$, where $\mathbf{A}$ has one row for each internal node and one column for each node, and a_{ij} is defined as:

$$a_{ij} = \begin{cases} 1, & \text{if } i = j \\ -1, & \text{node } j \text{ is a child of node } i \\ 0, & \text{otherwise} \end{cases}$$

The optimization problem (2) improves the estimation by enforcing the non-negativity ($\hat{\mathbf{x}} \succcurlyeq 0$) and sum-up-to-1 ($\hat{\mathbf{x}}_0 = 1$) compared with [21]. Because of the limit of space, we refer the readers who want to know the detail of derivation to [22] for more information.

5 SQUARE WAVE AND EXPECTATION MAXIMIZATION WITH SMOOTHING

The methods we presented in Section 4 use CFO protocols as black-boxes and do not fully exploit the ordered nature of the domains. We propose a new approach that uses a Square Wave reporting mechanism with post-processing conducted using Expectation Maximization with Smoothing (EMS).

5.1 General Wave Reporting

We first study a family of randomized reporting mechanisms that we call General Wave mechanisms. The intuition behind this approach is to try to increase the probability that a noisy reported value carries meaningful information about the input. This is also the implicit goal driving the development of CFO protocols beyond GRR. In GRR, one reports a value in $\mathscr{D}$. Intuitively, if the reported value is the true value, then the report is a "useful signal", as it conveys the extract correct information about the true input. If the reported value is not the true value, the report is in some sense noise that needs to be removed. The probability that a useful signal is generated is $p = \frac{e^\epsilon}{e^\epsilon + d - 1}$, where $d = |\mathscr{D}|$ is the size of the domain. When d is large, p is small, and GRR performs poorly. The essence of OLH and other CFO protocols is that one reports a randomly selected set of values, where one's true value has a higher probability of being selected than other values. In some sense, each "useful signal" is less sharp, since it is a set of values, but there is a much higher probability that a useful signal is transmitted.

Exploiting the ordinal nature of the domain, we note that a report that is different from but close to the true value v also carries useful information about the distribution. Therefore, given input v, we can report values closer to v with a higher probability than values that are farther away from v.

Without loss of generality, we assume that $\mathscr{D} = [0, 1]$ consists of floating point numbers between 0 and 1. The

random reporting mechanism can be defined by a family of probability density functions (PDF) over the output domain, with one PDF for each input value. We denote the output probability density function for v as $\mathbf{M}_v(\tilde{v}) = \Pr[\Psi(v) = \tilde{v}]$.

Following the above intuition, we want $\mathbf{M}_v(\tilde{v})$ to satisfy the property that $\mathbf{M}_v(\tilde{v}) = q$ when $|\tilde{v} - v| > b$, and $q \leq \mathbf{M}_v(\tilde{v}) \leq e^\epsilon q$ when $|\tilde{v} - v| \leq b$, where b is a parameter to be chosen. To ensure that for values close to the two ends, the range of near-by values is the same, we enlarge the output domain $\tilde{\mathscr{D}} = [-b, 1 + b]$. We formalize the idea as the following general wave mechanism.

Definition 2 (General Wave Mechanism (GW)). *With input domain $\mathscr{D} = [0, 1]$ and output domain $\tilde{\mathscr{D}} = [-b, 1 + b]$, a randomization mechanism $\Psi : \mathscr{D} \to \tilde{\mathscr{D}}$ is an instance of general wave mechanism if for all $v \in \mathscr{D}$, there is a wave function $W : \mathbb{R} \to [q, e^\epsilon q]$ with constants $q > 0$ and $\epsilon > 0$, such that the output probability density function $\mathbf{M}_v(\tilde{v}) = W(\tilde{v} - v)$:*

(1) $W(z) = q$ for $|z| > b$;
(2) $\int_{-b}^{b} W(z)\, dz = 1 - q$.

Theorem 1. GW *satisfies ϵ-LDP.*

Proof. For any two possible input value $v_1, v_2 \in \mathscr{D}$ and any set of possible output $T \subseteq \tilde{\mathscr{D}}$ of GW, we have $\frac{\Pr[\mathrm{GW}(v_1) \in T]}{\Pr[\mathrm{GW}(v_2) \in T]} = \frac{\int_{\tilde{v} \in T} \mathbf{M}_{v_1}(\tilde{v}) d\tilde{v}}{\int_{\tilde{v} \in T} \mathbf{M}_{v_2}(\tilde{v}) d\tilde{v}}$. By definition for all $v_1, v_2 \in \mathscr{D}$ and $T \subset \tilde{\mathscr{D}}$ we have $\frac{\Pr[\mathrm{GW}(v_1) \in T]}{\Pr[\mathrm{GW}(v_2) \in T]} \leq \frac{\int_{\tilde{v} \in T} e^\epsilon q \; d\tilde{v}}{\int_{\tilde{v} \in T} q \; d\tilde{v}} = e^\epsilon$. □

5.2 The Square Wave mechanism

GW can have different wave shapes. An intriguing question is what shape should be used. Following the same intuition in [1], given different values $v \neq v'$, if $\mathbf{M}_v$ and $\mathbf{M}_{v'}$ are identical, then there is no way to distinguish those different input values. Therefore, the hope is that the farther apart $\mathbf{M}_v$ and $\mathbf{M}_{v'}$ are, the easier it is to tell them apart. We use the difference between two output distributions, Wasserstein (a.k.a., earth-mover) distance as the utility metric. Based on this, we find the Square Wave mechanism, where supports for $[v - b, v + b]$ are the same, is optimal. We also empirically compare GW of other shapes with Square Wave mechanism in Section 6.4. The experimental results support our intuition.

Specification of Square Wave Reporting. The Square Wave mechanism SW is defined as:

$$\forall v \in \mathscr{D}, \tilde{v} \in \tilde{\mathscr{D}},\ \mathbf{M}_v(\tilde{v}) = \begin{cases} p, & \text{if } |v - \tilde{v}| \leq b\ , \\ q, & \text{otherwise}\ . \end{cases} \tag{3}$$

By maximizing the difference between p and q while satisfying the total probability adds up to 1, the values p, q can be derived as:

$$p = \frac{e^\epsilon}{2be^\epsilon + 1}\ , \quad q = \frac{1}{2be^\epsilon + 1}\ .$$

For each input v, the probability mass distribution for the perturbed output looks like a square wave, with the high plateau region centered around v. We thus call it the Square Wave (SW) reporting mechanism.

Theorem 2. *For any fixed b and ϵ, the* SW *is the* GW *that maximizes the Wasserstein distance between any two output distributions of two different inputs.*

Theorem 2 can be proved by using the following Lemma 1 and Lemma 2.

Lemma 1. *Given $v_1, v_2 \in \mathscr{D}$ as inputs to general wave mechanism, where $v_2 > v_1$ and let $\Delta = v_2 - v_1 > 0$, the Wasserstein distance between the output distributions of general wave mechanism is $\Delta(1 - (2b + 1)q)$.*

Proof. Given two different input values v_1 and v_2 which satisfy $v_2 - v_1 = \Delta > 0$, let $\mathbf{M}_{v_1}$ and $\mathbf{M}_{v_2}$ are the corresponding output distributions. Define a function $\mathrm{DIFF}(z)$ as the following:

$$\mathrm{DIFF}(z) = \begin{cases} 0\ , & \text{if } z \leq -b \\ 1 - (2b + 1)q\ , & \text{if } z \geq b \\ \int_{-b}^{z} (W(z') - q)\, dz'\ , & \text{otherwise.} \end{cases}$$

The cumulative function of SW can be written as

$$\mathbf{P}(\mathbf{M}_v, \tilde{v}) = (b + \tilde{v})q + \mathrm{DIFF}(\tilde{v} - v)$$

Therefore,

$$\begin{aligned} \int_{-b}^{1+b} \mathbf{P}(\mathbf{M}_v, \tilde{v}) d\tilde{v} =& \frac{q}{2}(1 + 2b)^2 + \int_{-b}^{b} \mathrm{DIFF}(z) dz \\ &+ (1 - (2b + 1)q)(1 - v)\ . \end{aligned}$$

Following the definition of Wasserstein distance of one dimensional data with ℓ_1 norm in Section 3, and as $\mathbf{P}(\mathbf{M}_{v_1}, \tilde{v}) \geq \mathbf{P}(\mathbf{M}_{v_2}, \tilde{v})$ for all $\tilde{v}$, it follows that

$$\begin{aligned} W_1(\mathbf{M}_{v_1}, \mathbf{M}_{v_2}) &= \int_{\tilde{\mathscr{D}}} |\mathbf{P}(\mathbf{M}_{v_1}, \tilde{v}) - \mathbf{P}(\mathbf{M}_{v_2}, \tilde{v})| d\tilde{v} \\ &= \int_{-b}^{1+b} \left(\mathbf{P}(\mathbf{M}_{v_1}, \tilde{v}) - \mathbf{P}(\mathbf{M}_{v_2}, \tilde{v})\right) d\tilde{v} \\ &= (1 - (2b + 1)q)\Delta\ . \end{aligned}$$

□

Lemma 1 shows that we need to minimize q if we want to maximize the Wasserstein distance between any two output distributions. Thus, we have the following lemma.

Lemma 2. *For any fixed b and ϵ, the minimum q for general wave mechanism is $q = \frac{1}{2be^\epsilon + 1}$, which can be achieved if and only if the mechanism is* SW.

Proof. By criteria of the definition of GW, we have

$$1 = q + \int_{-b}^{b} W(z)dz \le 1 + (2b)e^{\epsilon}q$$
$$\Rightarrow q \ge \frac{1}{2be^{\epsilon} + 1}$$

We have equality iff $\mathbf{M}_v(\tilde{v}) = e^{\epsilon}q$ for all $\tilde{v} \in [v - b, v + b]$, which turns out to be SW. □

Comparison with PM Mechanism. Square Wave (SW) reporting is similar to the Piecewise Mechanism (PM) for mean estimation [33] (see Section 2.2). PM directly sums up the randomized reports to estimate the mean of distribution, while the outputs of SW are used to reconstruct the whole distribution (the reconstruction method will be described in Subsection 5.5). Driven by the different focus, the reporting mechanisms are also different. PM has to be unbiased for mean estimation, so the input values are not always at the center of high probability region. For example, given input $v = -1$, the high probability range in PM is $[-\frac{e^{\epsilon/2}+1}{e^{\epsilon/2}-1}, -1]$.

Communication cost. With SW, each report consists of a single floating point number. The communication cost is thus a small constant for each user, similar to protocols such as GRR and OLH.

5.3 Choosing *b*

An important parameter to choose for the Square Wave reporting mechanism is b. In Square Wave reporting, a value that is within b of true input is reported with a probability that is e^{ϵ} times the probability that a "far" value is reported. The optimal choice of b depends on the privacy parameter ϵ. For a larger ϵ, a smaller b is preferred. When ϵ goes to infinity, a value of $b \to 0$ leads to total recovery of input distribution, and any $b > 0$ leads to information loss. Intuitively, the optimal choice of b also depends on the input distribution. For a distribution with probability density concentrated at one point, one would prefer smaller b. For a distribution with more or less evenly distributed probability density, one would prefer a larger b. However, since we do not know the distribution of the private values, we want to choose a b value independent of the distribution, but can perform reasonably well over different distributions.

In this paper, we choose b to maximize the upper bound of mutual information between the input and output of the Square Wave reporting. We also empirically study the effect of varying b (see Section 6.4). The experimental results show that choosing b by this method results in optimal or close to optimal choices of b.

Let V and $\tilde{V}$ be the input and output random variables representing the input and output of SW, respectively. The mutual information between V and $\tilde{V}$ can be represented by the difference between differential entropy and conditional differential entropy of V and $\tilde{V}$:

$$I(V, \tilde{V}) = h(V) - h(V|\tilde{V}) = h(\tilde{V}) - h(\tilde{V}|V) .$$

The quantity $I(V, \tilde{V})$ depends on the input distribution, which we want to avoid. Therefore, we consider an upper bound of $I(V, \tilde{V})$, which is achieved when $\tilde{V}$ is uniformly distributed on $\tilde{\mathcal{D}}$. Let U be the random variable that is uniformly distributed in $\tilde{\mathcal{D}}$. Because $h(\tilde{V}) \le h(U)$, we have:

$$I(V, \tilde{V}) \le h(U) - h(\tilde{V}|V). \tag{4}$$

In (4), the first term of RHS is

$$h(U) = \log(2b + 1).$$

The second term of RHS only depends on SW:

$$\begin{aligned} h(\tilde{V}|V) &= -\int_{v} \Pr[V = v](2bp \log p + q \log q) \\ &= -(2bp \log p + q \log q) \\ &= -\frac{2b\epsilon e^{\epsilon}}{2be^{\epsilon} + 1} + \log(2be^{\epsilon} + 1) . \end{aligned}$$

So the mutual information is determined by a function of b,

$$\log\left(\frac{2b + 1}{2be^{\epsilon} + 1}\right) + \frac{2b\epsilon e^{\epsilon}}{2be^{\epsilon} + 1} .$$

By making its derivative to 0, we get

$$b = \frac{\epsilon e^{\epsilon} - e^{\epsilon} + 1}{2e^{\epsilon}(e^{\epsilon} - 1 - \epsilon)} .$$

Note that b is a non-increasing function with ϵ. When ϵ goes to ∞, b goes to 0. When ϵ goes to 0, b goes to $1/2$, which leads to an output domain that doubles the size of the input domain, and for each input value, half of the output domain are considered "close" to the input value.

5.4 Bucketizing

The aggregator receives perturbed reports from users and needs to reconstruct the distribution on $\mathcal{D}$. Our approach performs this reconstruction on a discretized domain, i.e., histograms over the domain. The bucketization step can be performed either before or after applying the randomization step. We discuss the two approaches below. In experiments, we use the "randomize before bucketize" approach.

"Randomize before bucketize" (R-B). Here each user possesses a floating point number in $\tilde{\mathcal{D}} = [0, 1]$, applies the Square Wave mechanism in Section 5.2, and sends the result to the aggregator. The aggregator receives values in $\tilde{\mathcal{D}} = [-b, 1 + b]$, discretizes the reported values into $\tilde{d}$ buckets in $\tilde{\mathcal{D}}$, and constructs a histogram with $\tilde{d}$ bins. Using the method in Section 5.5, the aggregator can reconstruct an estimated input histogram of d bins. In experiments, we set $\tilde{d} = d$ for simplicity.

We compare the results of choosing different $\tilde{d}$ in Section 6.4, and found that the results are similar so long as $\tilde{d}$ does not deviate far from $\sqrt{N}$.

"Bucketize before randomize" (B-R) or discrete input domain. Alternatively, a user can perform the discretization step first, and then perform randomization. The SW mechanism can be naturally applied in a discrete domain as well. Assume input domain size is $d = |\mathscr{D}|$, discrete SW mechanism has output domain size $\tilde{d} = |\tilde{\mathscr{D}}| = d + 2b$, and randomizes input values as the following:

$$\forall v \in \mathscr{D}, \tilde{v} \in \tilde{\mathscr{D}},\ \Pr[\mathrm{SW}(v) = \tilde{v}] = \begin{cases} p, & \text{if } |v - \tilde{v}| \le b \\ q, & \text{otherwise,} \end{cases}$$

where $p = \frac{e^\epsilon}{(2b+1)e^\epsilon + d - 1}$ and $q = \frac{1}{(2b+1)e^\epsilon + d - 1}$. In this case, one can set $b = \left\lfloor \frac{\epsilon e^\epsilon - e^\epsilon + 1}{2e^\epsilon(e^\epsilon - 1 - \epsilon)} d \right\rfloor$.

The above discrete SW mechanism can also be applied when the input domain is already discrete (e.g., age). We conducted experiments comparing doing R-B versus B-R, and found that they are very similar. Detailed results are omitted due to space limitation.

5.5 Estimating Distribution from Reports

The aggregator receives perturbed values and faces an estimation problem. Note that post-processing of the output of a mechanism that satisfies differential privacy (the perturbed values from users) does not affect its privacy guarantee [12].

Without relying on any prior knowledge of the actual distribution, the natural approach is to conduct Maximum Likelihood Estimation (MLE). We use a $\tilde{d} \times d$ matrix $\mathbf{M}$ to characterize the randomization process. More specifically, the matrix $\mathbf{M} \in [0,1]^{\tilde{d} \times d}$ denotes the transformation probabilities, where $\mathbf{M}_{j,i}$ represents the probability of output value falling in bucket $\tilde{B}_j$, $j \in [\tilde{d}]$, given input in bucket B_i, $i \in [d]$, (assuming the input data fall uniformly at random within bucket B_i). Each column of $\mathbf{M}$ sums up to 1.

Expectation-Maximization (EM) Algorithm. Given the probability matrix $\mathbf{M}$ as defined above, we can use an Expectation-Maximization (EM) algorithm to reconstruct the distribution. The aggregator receives n randomized values from users, which are denoted as $\tilde{\mathbf{v}} = \{\tilde{v}_1, \ldots, \tilde{v}_n\}$, and finds $\hat{\mathbf{x}}$ that maximizes the log-likelihood $L(\hat{\mathbf{x}}) = \ln \Pr[\tilde{\mathbf{v}}|\hat{\mathbf{x}}]$.

Let n_j be the number of values in $\tilde{B}_j$ is reported. The EM algorithm for post-processing the square wave reporting is shown in Algorithm 1. Note that there are existing works that use EM to post-process results of CFO (e.g., [16, 28]), but our proposed EM algorithm takes aggregated results and is thus more efficient. Because of limitation of space, we omit the derivation of EM algorithm.

Theorem 3. *The EM algorithm converges to the maximum-likelihood (ML) estimator of the true frequencies* $\mathbf{x}$.

Algorithm 1 Post-processing EM algorithm

Input: $\mathbf{M}, \tilde{\mathbf{v}}$
Output: $\hat{\mathbf{x}}$
while not converge **do**
E-step: $\forall i \in \{1, \ldots, d\}$,

$$P_i = \hat{\mathbf{x}}_i \sum_{j \in [\tilde{d}]} n_j \frac{\Pr\left[\tilde{v} \in \tilde{B}_j | v \in B_i, \hat{\mathbf{x}}\right]}{\Pr\left[\tilde{v} \in \tilde{B}_j | \hat{\mathbf{x}}\right]} = \hat{\mathbf{x}}_i \sum_{j \in [\tilde{d}]} n_j \frac{\mathbf{M}_{j,i}}{\sum_{k=1}^{d} \mathbf{M}_{j,k} \hat{\mathbf{x}}_k}$$

M-step: $\forall i \in \{1, \ldots, d\}$,

$$\hat{\mathbf{x}}_i = \frac{P_i}{\sum_{k'=1}^{d} P_{k'}}$$

end while
Return $\hat{\mathbf{x}}$

Proof. To prove EM algorithm converges to the maximum likelihood estimator, it is enough to show the log-likelihood function is concave [6]. In the context of our problem

$$L(\mathbf{x}) = \ln \Pr[\tilde{\mathbf{v}}|\mathbf{x}] = \ln \prod_{k=1}^{n} \Pr[\tilde{v}_k|\mathbf{x}] = \sum_{k=1}^{n} \ln \left(\sum_{i=1}^{d} \mathbf{x}_i \Pr[\tilde{v}_k | v \in B_i] \right),$$

where $\Pr[\tilde{v}_k | v \in B_i]$ are constants determined by SW method. Thus, $L(\mathbf{x})$ is a concave function. □

Stopping Criteria. Through experiments, we have observed that the result of applying EM is highly sensitive to the parameter controlling terminating condition. If EM terminates too early, the reconstructed distribution is still far from the true one. If EM terminates too late, while the reconstructed distribution does fit the observation better (higher likelihood), it is also getting farther away from the true distribution to fit the noise. One of the most common stopping criteria for EM algorithm is checking whether the relative improvement of log-likelihood is small [16]. Namely, when $\left|L(\hat{\mathbf{x}}^{(t+1)}) - L(\hat{\mathbf{x}}^{(t)})\right| < \tau$ for some small positive number τ, EM algorithm stops. The choice of τ depends on many factors, including the smoothness of distribution and the amount of noise added by the square wave distribution. Empirically, we find that if we set τ proportional to e^ϵ, EM algorithm generally performs better than the one using a fixed τ. However, on some datasets that have a smoother distribution, the recovered result still over-fits the noise. Several of our

attempts at finding a stopping condition that make EM perform well consistently did not succeed. This motivates us to apply smoothing in EM.

EMS Algorithm. Using prior knowledge in estimation can make results less sensitive to the noise and more accurate than MLE solution. By the nature of numerical domain, adjacent numerical values' frequencies should not vary dramatically. With this observation, we can add a smoothing step after the M-step in the EM algorithm, boosting the accuracy with prior knowledge. We call the EM algorithm with smoothing steps as EMS algorithm. The idea of adding smoothing step into EM algorithm dates back to 1990s [24, 30] in the context of positron emission tomography and image reconstruction. The authors showed that a simple local smoothing method, the weight average with binomial coefficients of a bin value and the values of its nearest neighbours, could improve the estimation dramatically. We adopt this smoothing method. That is, after the M-step, the smoothing step will average each estimate with its adjacent ones with binomial coefficients $(1, 2, 1)$:

$$\hat{\mathbf{x}}_i = \frac{1}{2}\hat{\mathbf{x}}_i + \frac{1}{4}\left(\hat{\mathbf{x}}_{i-1} + \hat{\mathbf{x}}_{i+1}\right).$$

It was proved that adding the smoothing step is equivalent to adding a regularization term penalizing the spiky estimation [24], which can be viewed as applying Bayesian inference with a prior that prefers smoother distribution to jagged ones [26]. In more recent work, the idea of EMS is also applied to spatial data [15] and biophysics data [18].

6 EXPERIMENTS

6.1 Experimental Setup

Datasets. We use the following datasets to conduct our experiments. One of them is synthetic, and the other three are real world datasets. All of them consist of numerical values. For CFO based methods, we discretize the values to the same granularity as the output of SW with EMS/EM method. Also, in order to compare with HH and HH-ADMM, which have optimal branching factor close to 4 [21], we choose the granularity (number of buckets in histogram) to be power of 4.

Synthetic Beta(5, 2) dataset. Originally, the distribution is in the continuous domain $[0, 1]$. One hundred thousand samples are generated. In experiments, we reconstruct the histogram with 256 buckets for all methods.

Taxi dataset's attribute pick-up time. Taxi pickup time dataset comes from 2018 January New York Taxi data [31]. Originally, the dataset contains the pickup time in a day (in seconds). We map the values into $[0, 1]$. There are $2,189,968$ samples in the dataset. In experiments, all estimated histograms have 1024 buckets.

Methods \ Metrics	Wasserstein and KS distance	Range Query	Mean & Variance	Quantile
SW with EMS/EM (this paper)	✓	✓	✓	✓
HH-ADMM (this paper)	✓	✓	✓	✓
CFO binning	✓	✓	✓	✓
HH [21] and HaarHRR [21]		✓		
PM [33] and SR [10]			✓	

Table 2: Methods and evaluated metrics.

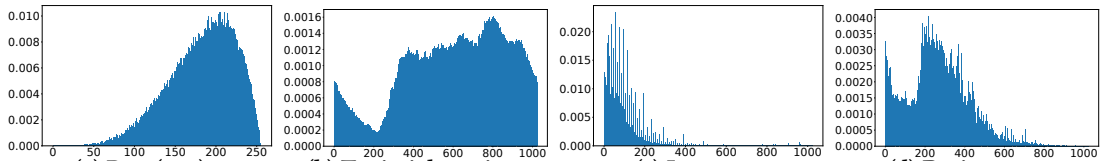
Income dataset. We use the income information of the 2017 American Community Survey [29]. The data range is $[0, 1563000)$. We extract the values that are smaller than 524288 (i.e., 2^{19}) and map them into $[0, 1]$. There are $2,308,374$ samples after pre-processing. We choose to set the estimated histograms with 1024 buckets.

Retirement dataset. The San Francisco employee retirement plans data [25] contains integer values from $-28,700$ to $101,000$. We extract values that are non-negative and smaller than $60,000$, and map them into $[0, 1]$. There are $178,012$ samples after post-processing. In experiments, we reconstruct the histogram with 1024 buckets for all methods.

The income dataset is spiky because many people tend to report with precision up to hundreds or thousands (e.g., people are more likely to report \$3000 instead of more precise value like \$3050 or \$2980.)

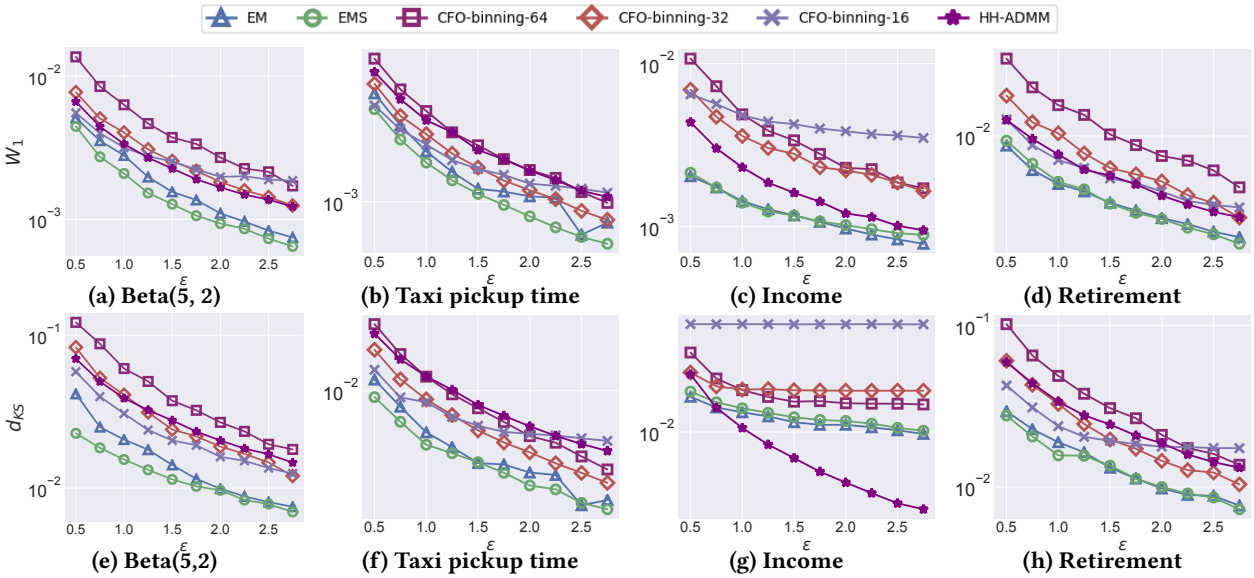
Competitors. In the experiments, we consider several existing methods, including methods that obtain mean (PM, SR) and Hierarchy-based Methods (HH, HaarHRR). We also consider CFO with binning methods, our proposed method HH-ADMM, and SW with EMS/EM. To the more specific, we summarize the methods and metrics evaluated in Table 2.

- Piecewise Mechanism (PM) and Stochastic Rounding (SR) (See Section 2.2) are only evaluated for mean and variance. They were designed for mean, and we adapted them to also estimate variance.
- For CFO with binning, we partition $\mathscr{D}$ into c consecutive, non-overlapping chunks. We consider $c = 16, 32, 64$, which are the best performing c values.
- For HH, HaarHRR and HH-ADMM, similar to [21], we use a branching factor of 4. HH and HaarHRR are only evaluated for range queries as they produce estimation results with negative values, which are not valid probability distributions. Other metrics are defined for probability distributions.
- For SW with EM and EMS as post-processing, we set $\tau = 10^{-3}e^{\epsilon}$ for EM and $\tau = 10^{-3}$ for EMS.



(a) Beta(5, 2) (b) Taxi pickup time (c) Income (d) Retirement

Figure 1: Normalized frequencies of datasets for experiments.



(a) Beta(5, 2) (b) Taxi pickup time (c) Income (d) Retirement

(e) Beta(5,2) (f) Taxi pickup time (g) Income (h) Retirement

Figure 2: Results of distribution distances (first row: Wasserstein distance, second row: KS distance), varying ϵ.

As a brief overview of the experiment results, SW with EMS performs best with different privacy budgets and different metrics. HH-ADMM performs best on the income dataset under some of the metrics. We also experimentally demonstrate the better utility of SW over other wave shapes in GW and the near-optimal choice of b for SW.

Evaluation methodology. The algorithms are implemented using Python 3.6 and Numpy 1.15; the experiments are conducted on a server with Intel Xeon 4108 and 128GB memory. For each dataset and each method, we repeat the experiment 100 times and take the mean.

6.2 Distribution Distance

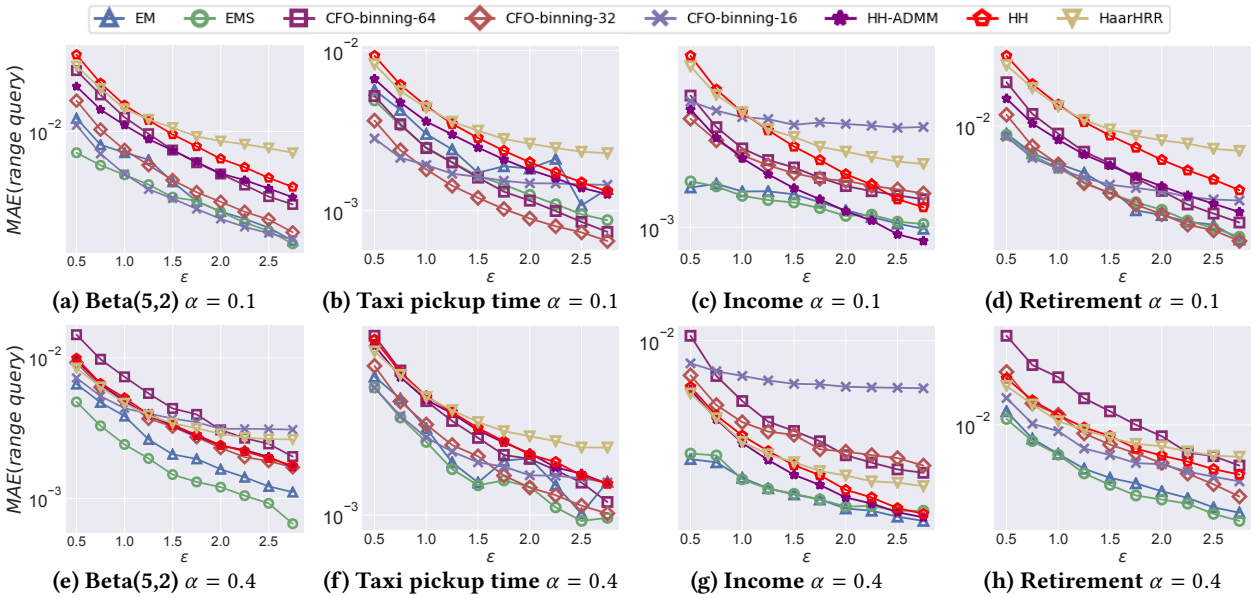
We first evaluate metrics that capture the quality of the recovered distributions. Note that HH and haarHRR are not included (but HH-ADMM is) because HH or haarHRR does not result in valid distributions.

Wasserstein Distance. Figure 2(a)-2(d) shows the Wasserstein distance W_1 of reconstructed distribution and the true distribution. In most cases, SW with EMS performs best, followed by EM and HH-ADMM. For the CFO-binning methods, when ϵ is small, larger binning sizes (i.e., fewer number of bins) tend to give better performance. The lines for larger binning sizes flatten as ϵ increases, showing that the errors are dominated by biases due to binning. When ϵ becomes larger, CFO-binning with smaller bin sizes (i.e., more bins) becomes better. We observe that even if we could choose the optimal bin size empirically for each dataset and ϵ value, the result would still be worse than SW with EMS.

KS Distance. Figure 2(e)-2(h) show the K-S distance. For Beta, taxi pickup time and retirement datasets, SW with EMS generally performs the best, followed by EM. For the income dataset, HH-ADMM performs better than EM and EMS under this metric, especially under larger ϵ values. This is because the income dataset is more spiky, due to the fact that people tend to report income using round numbers. HH-ADMM is better at preserving some of the spikes in the distribution, whereas SW with EM or EMS will smooth the spikes. Since KS distance measures maximum difference at one point in CDF, HH-ADMM results in lower errors under KS distance, even though it produces higher error under Wasserstein Distance. For similar reason, CFO with larger bin size also perform poorly on the income dataset under KS distance.

6.3 Semantic and Statistical Quantities

We compare the results of different methods using the range query and statistic quantities including mean, variance, quantiles. For mean and variance, we also consider the SR and PM, which were designed for mean estimations. All results are measured by Mean Absolute Error (MAE).



(a) Beta(5,2) $\alpha = 0.1$ (b) Taxi pickup time $\alpha = 0.1$ (c) Income $\alpha = 0.1$ (d) Retirement $\alpha = 0.1$

(e) Beta(5,2) $\alpha = 0.4$ (f) Taxi pickup time $\alpha = 0.4$ (g) Income $\alpha = 0.4$ (h) Retirement $\alpha = 0.4$

Figure 3: MAE of random range query with range $\alpha = 0.1$ (first row) and $\alpha = 0.4$ (second row).

Range Query. The queries are randomly generated, but with fixed range sizes. Denote the left and right of the range as i and $i + \alpha$, we randomly generate $i \in [0, 1 - \alpha]$ with $\alpha = 0.1$ and 0.4. The results in Figure 3 shows that SW with EMS outperforms HH and HaarHRR [21]. In fact, it is the best in most cases, except when $\alpha = 0.1$ in the taxi pickup time dataset and in low privacy region of income dataset. However, SW with EMS has performance similar to CFO-binning-64 when $\alpha = 0.1$ and still outperforms all the hierarchy-based approaches in taxi pickup time dataset. For the income dataset, EM and EMS performs well in high privacy range (i.e., $\epsilon \leq 2$), while HH-ADMM performs best in low privacy range, followed by EM and EMS.

Mean Estimation. Results for mean estimation are showed in Figure 4(a)-4(d). SR performs better than PM when ϵ is small, but worse when ϵ is larger. This is consistent with the analysis in [33]. Note that SR and PM devote all privacy budget to estimate mean. While SW with EMS can estimate the full distribution, it performs comparable to the best of SR and PM for estimating the mean. We also see that HH-ADMM has better performances than all other CFO-binning methods, but is still inferior to SW with EMS.

Variance Estimation. Although SR and PM are proposed for mean estimation, they can be modified to support variance estimation as well. Specifically, we randomly sample 50% of users to estimate mean first. The estimated mean is then broadcast to the remaining users. Then each user compares his secret value and the received estimated mean, and reports the squared difference (i.e., $(v_i - \tilde{\mu})^2$) to the server, who averages them to obtain variance.

The experimental results are showed in Figure 4(e)- 4(h). As we can see, the error of SR and PM is larger than EM or EMS in most cases. One reason is that only half of the users are used for variance estimation (the other half is necessary for mean estimation). The relative performance of other methods are similar to previous experiments.

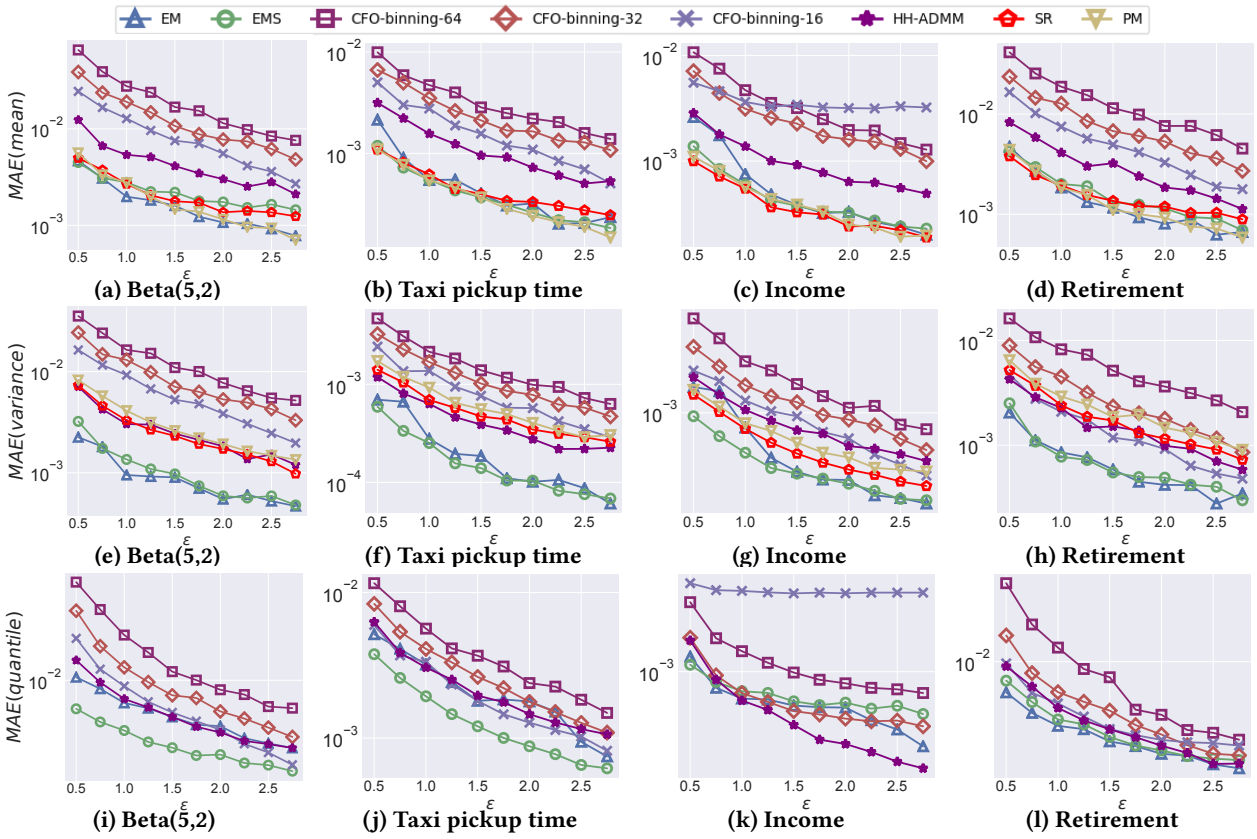
Quantile Estimation. Experimental results are shown in Figure 4(i)-4(l). Ignoring the spiky income dataset for now, our proposed SW with EMS performs best. Moreover, we observe that SW with EM sometimes performs better but is not stable, because it is sensitive to parameters. HH-ADMM performs worse than SW, but close to the best of CFO with binning. For CFO with binning, because of the trade-off between estimation noise and the bias within the bins, larger bin sizes typically perform better in smaller ϵ ranges, while the smaller bin sizes narrows the gap as ϵ increases.

For the spiky income dataset, even for $\epsilon = 0.5$, larger bin sizes give worse utility (1 to 2 orders of magnitude) than other mechanisms. This also demonstrates that the optimal bin size is data-dependent. HH-ADMM successfully captures the spikiness of the dataset and thus performs the best.

6.4 Wave Shapes and Parameters

Here we compare the different shapes of General Wave (GW) with SW, and different parameters of SW.

Different shapes of wave in GW. In Section 5.2, we analytically show that SW is preferred because it maximizes the Wasserstein distance between output distributions. We empirically compare SW with other wave forms. We consider 5 other GW mechanism with different shape, including 4 trapezoid shapes and one triangle shape. The upper side to



(a) Beta(5,2) (b) Taxi pickup time (c) Income (d) Retirement

(e) Beta(5,2) (f) Taxi pickup time (g) Income (h) Retirement

(i) Beta(5,2) (j) Taxi pickup time (k) Income (l) Retirement

Figure 4: MAE for estimating mean (first row), variance (second row), and quantiles (third row).

bottom side length ratio of trapezoid wave are 0.2, 0.4, 0.6 and 0.8. The experimental results in Figure 5 show when $\epsilon = 1$, SW gives the best estimated distributions in terms of Wasserstein distance, no matter how we change b. As the ratio decreases, the recovery accuracy also degrades in general. The results support our intuition in Section 5.2.

SW **with different** b**.** In Section 5.3, we propose to use $b_{\text{SW}} = \frac{\epsilon e^{\epsilon} - e^{\epsilon} + 1}{2e^{\epsilon}(e^{\epsilon} - 1 - \epsilon)}$. Figure 6 reports experimental results with different b. Our choice of b_{SW}, which is indicated as the vertical dotted line, is among the ones that provide best utility. We have also evaluated b on other metrics; the results give similar conclusion, and are omitted because of space limitation.

Bucketization granularity. To see what is the optimal bucketization granularity on different datasets, we choose 4 different numbers of buckets (256, 512, 1024 and 2048) then compare the Wasserstein distance between the estimated distributions and the true distributions. For simplicity, we use same number of buckets for both $\tilde{\mathscr{D}}$ and $\mathscr{D}$. The experimental results in Figure 7 show different datasets have different optimal bucketization granularity. For Beta(5,2), we have best result when the number of buckets is 256. For the other 3 datasets, dividing $\mathscr{D}$ into 1024 buckets can give us best performance in most cases.

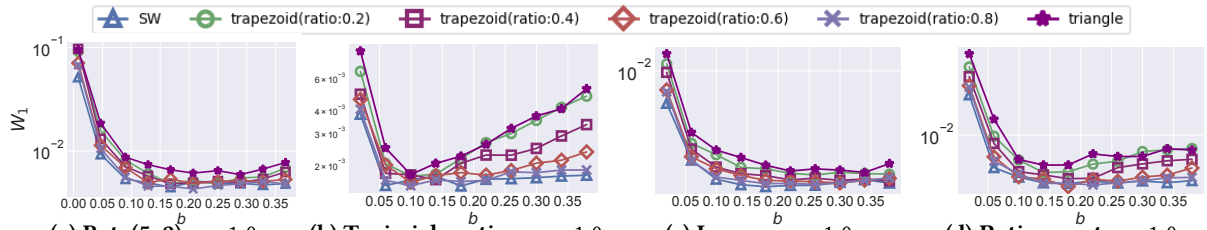
7 RELATED WORK

Differential privacy has been the *de facto* notion for protecting privacy. In the local setting, we have seen real world deployments: Google Chrome extension [14], spelling prediction of Apple [32] and telemetry collection by Microsoft [9].

Categorical Frequency Oracle. One basic mechanism in LDP is to estimate frequencies of values. There have been several mechanisms [1, 4, 5, 9, 14, 35] proposed for this task. Among them, [35] introduces OLH, which achieves low estimation errors and low communication costs. Our paper develop new frequency oracles for numerical attributes.

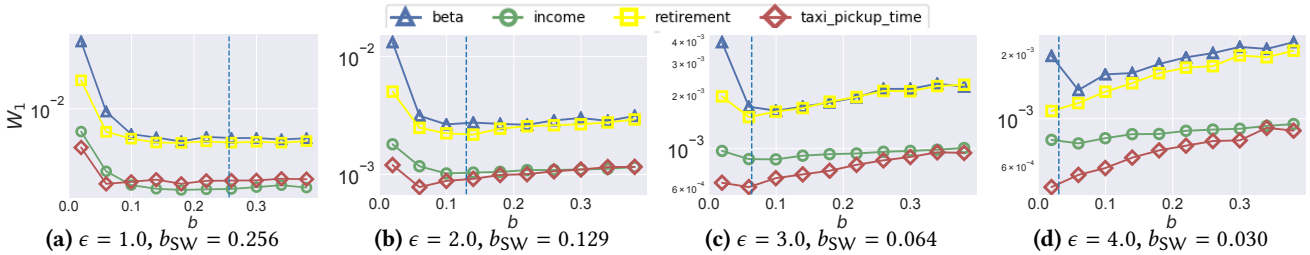
Handling Ordinal/Numerical Data. When the data is ordinal, the straightforward approach is to bucketize the data and apply categorical frequency oracles. [34] considers distribution estimation, but with a strictly weaker privacy definition. There are also mechanisms that can handle numerical setting, but focusing on the specific task of mean estimation, i.e. SR [9, 10] and PM [33]. These two approaches have been discussed in Section 2 and compared in the experiments.

Post-processing. Given the result of a privacy-preserving algorithm, one can utilize the structural information to post-process it so that the utility can be improved. In the setting of centralized DP, Hay et al. [17] propose an efficient hierarchical method to minimize L_2 difference between the original



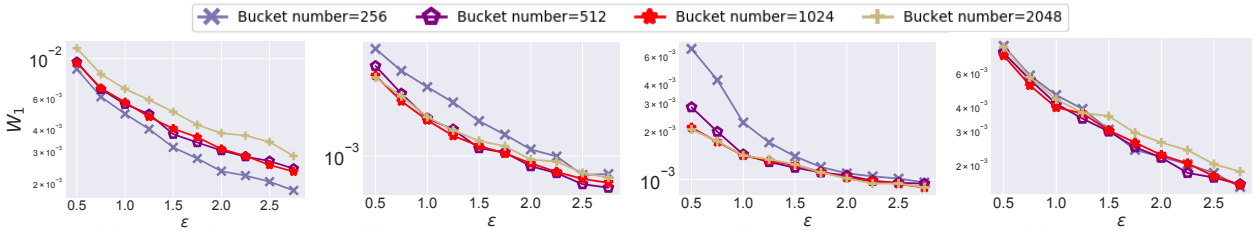
(a) Beta(5, 2) $\epsilon = 1.0$ (b) Taxi pickup time $\epsilon = 1.0$ (c) Income $\epsilon = 1.0$ (d) Retirement $\epsilon = 1.0$

Figure 5: Comparison of different shapes of wave in GW. Ratios are the upper/lower length ratios for trapezoids.



(a) $\epsilon = 1.0$, $b_{SW} = 0.256$ (b) $\epsilon = 2.0$, $b_{SW} = 0.129$ (c) $\epsilon = 3.0$, $b_{SW} = 0.064$ (d) $\epsilon = 4.0$, $b_{SW} = 0.030$

Figure 6: Wasserstein distances between the true data and the estimation produced by EMS algorithm with fixed ϵ values and varying b from 0.01 to 0.38. Dotted vertical lines means the used b_{SW} in Section 5.3.



(a) Beta(5,2) (b) Taxi Pickup time (c) Income (d) Retirement

Figure 7: Wasserstein distance between estimated and true distribution with different bucketization granularity.

result and the processed result. Besides that, the authors of [22] also consider the non-negativity constraint and propose to use ADMM to obtain result that achieves maximal likelihood. As ADMM is not efficient for high dimensional case, a gradient descent based algorithm is proposed [23].

In the LDP setting, [33] and [21] also consider the hierarchy structure and apply the technique of [17]. We propose to use ADMM instead of [17], which improves utility.

Without using the hierarchical constraint (only consider CFO), Jia et al. [19] propose to utilize external information about the dataset (e.g., assume it follows a power-law distribution), and Wang et al. [38] consider the constraints that the distribution is non-negative and sum up to 1. Bassily [3] and Kairouz et al. [20] study the post-processing for some CFO with MLE. Compared with those existing methods, our work is also a post-processing method but is applied to a new Square Wave reporting method and requires different techniques (such as EMS algorithm).

Shuffling. Recently, shuffle-DP [2, 8, 13] is introduced as an intermediate framework between centralized DP and LDP. By assuming there is a trusted third party who shuffles the reports of a ϵ-LDP protocol before sending them to the aggregator, it is proved in [2] that the output of those shuffled reports will satisfy (ϵ', δ)-DP, for some $\epsilon' = O((1 \wedge \epsilon)e^{\epsilon}\sqrt{\log(1/\delta)/n})$. Our SW mechanism is fully compatible with shuffling, and its privacy amplification effects can be analyzed by the same tools introduced in [2].

8 CONCLUSIONS

We have studied the problem of reconstructing the distribution of a numerical attribute under LDP. We introduce HH-ADMM as an improvement to existing hierarchy-based methods. Most importantly, we propose the method of combining Square Wave reporting with Expectation Maximization and Smoothing. We show that Square Wave mechanism has the best utility among general wave mechanisms, and introduce techniques to choose the bandwidth parameter b by maximizing an upper bound of mutual information. Extensive experimental evaluations demonstrate that SW with EMS generally performs the best under a wide range of metrics. We expect these protocols and findings to help improving the deployment of LDP protocols to collect and analyse numerical information.

ACKNOWLEDGEMENT

This project is supported by NSF grant 1640374, NWO grant 628.001.026, and NSF grant 1931443. We thank the anonymous reviewers for their helpful suggestions.

REFERENCES

[1] J. Acharya, Z. Sun, and H. Zhang. Hadamard response: Estimating distributions privately, efficiently, and with little communication. *arXiv preprint arXiv:1802.04705*, 2018.

[2] B. Balle, J. Bell, A. Gascón, and K. Nissim. The privacy blanket of the shuffle model. In *Annual International Cryptology Conference*, pages 638–667. Springer, 2019.

[3] R. Bassily. Linear queries estimation with local differential privacy. In *AISTATS*, 2019.

[4] R. Bassily, K. Nissim, U. Stemmer, and A. G. Thakurta. Practical locally private heavy hitters. In *NIPS*, 2017.

[5] R. Bassily and A. D. Smith. Local, private, efficient protocols for succinct histograms. In *STOC*, 2015.

[6] J. A. Bilmes et al. A gentle tutorial of the em algorithm and its application to parameter estimation for gaussian mixture and hidden markov models. *International Computer Science Institute*, 4(510):126, 1998.

[7] S. Boyd, N. Parikh, E. Chu, B. Peleato, J. Eckstein, et al. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends® in Machine learning*, 3(1):1–122, 2011.

[8] A. Cheu, A. D. Smith, J. Ullman, D. Zeber, and M. Zhilyaev. Distributed differential privacy via shuffling. In *EUROCRYPT*, pages 375–403, 2019.

[9] B. Ding, J. Kulkarni, and S. Yekhanin. Collecting telemetry data privately. In *NIPS*, 2017.

[10] 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.

[11] C. Dwork, F. McSherry, K. Nissim, and A. Smith. Calibrating noise to sensitivity in private data analysis. In *TCC*, 2006.

[12] C. Dwork and A. Roth. The algorithmic foundations of differential privacy. *Foundations and Trends in Theoretical Computer Science*, 9(3-4), 2014.

[13] Ú. Erlingsson, V. Feldman, I. Mironov, A. Raghunathan, K. Talwar, and A. Thakurta. Amplification by shuffling: From local to central differential privacy via anonymity. *arXiv preprint arXiv:1811.12469*, 2018.

[14] Ú. Erlingsson, V. Pihur, and A. Korolova. RAPPOR: randomized aggregatable privacy-preserving ordinal response. In *CCS*, 2014.

[15] C.-P. S. Fan, J. Stafford, and P. E. Brown. Local-em and the ems algorithm. *Journal of Computational and Graphical Statistics*, 20(3):750–766, 2011.

[16] G. C. Fanti, V. Pihur, and Ú. Erlingsson. Building a RAPPOR with the unknown: Privacy-preserving learning of associations and data dictionaries. *PoPETs*, 2016(3), 2016.

[17] M. Hay, V. Rastogi, G. Miklau, and D. Suciu. Boosting the accuracy of differentially private histograms through consistency. *PVLDB*, 3(1), 2010.

[18] Q. J. Huys and L. Paninski. Smoothing of, and parameter estimation from, noisy biophysical recordings. *PLoS computational biology*, 5(5):e1000379, 2009.

[19] J. Jia and N. Z. Gong. Calibrate: Frequency estimation and heavy hitter identification with local differential privacy via incorporating prior knowledge. *INFOCOM*, 2019.

[20] P. Kairouz, K. Bonawitz, and D. Ramage. Discrete distribution estimation under local privacy. In *ICML*, 2016.

[21] T. Kulkarni, G. Cormode, and D. Srivastava. Answering range queries under local differential privacy. *PVLDB*, 2019.

[22] J. Lee, Y. Wang, and D. Kifer. Maximum likelihood postprocessing for differential privacy under consistency constraints. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 635–644. ACM, 2015.

[23] R. McKenna, D. Sheldon, and G. Miklau. Graphical-model based estimation and inference for differential privacy. *ICML*, 2019.

[24] D. Nychka. Some properties of adding a smoothing step to the em algorithm. *Statistics & probability letters*, 9(2):187–193, 1990.

[25] S. F. C. Office. Sf employee compensation. https://www.kaggle.com/san-francisco/sf-employee-compensation#employee-compensation.csv.

[26] D. Ormoneit and V. Tresp. Averaging, maximum penalized likelihood and bayesian estimation for improving gaussian mixture probability density estimates. *IEEE Transactions on Neural Networks*, 9(4):639–650, 1998.

[27] W. H. Qardaji, W. Yang, and N. Li. Understanding hierarchical methods for differentially private histograms. *PVLDB*, 6(14), 2013.

[28] X. Ren, C. 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 Trans. Information Forensics and Security*, 13(9), 2018.

[29] S. Ruggles, S. Flood, R. Goeken, J. Grover, E. Meyer, J. Pacas, and M. Sobek. Integrated public use microdata series: Version 9.0 [database], 2019.

[30] B. Silverman, M. Jones, J. Wilson, and D. Nychka. A smoothed em approach to indirect estimation problems, with particular reference to stereology and emission tomography. *Journal of the Royal Statistical Society: Series B (Methodological)*, 52(2):271–303, 1990.

[31] N. Taxi and L. Commission. Tlc trip record data. https://www1.nyc.gov/site/tlc/about/tlc-trip-record-data.page, 2018.

[32] A. D. P. Team. Learning with privacy at scale, 2017.

[33] 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 *ICDE*, 2018.

[34] S. Wang, Y. Nie, P. Wang, H. Xu, W. Yang, and L. Huang. Local private ordinal data distribution estimation. In *IEEE INFOCOM 2017-IEEE Conference on Computer Communications*, pages 1–9. IEEE, 2017.

[35] T. Wang, J. Blocki, N. Li, and S. Jha. Locally differentially private protocols for frequency estimation. In *USENIX Security*, 2017.

[36] T. Wang, B. Ding, J. Zhou, C. Hong, Z. Huang, N. Li, and S. Jha. Answering multi-dimensional analytical queries under local differential privacy. In *Proceedings of the 2019 International Conference on Management of Data*, pages 159–176. ACM, 2019.

[37] T. Wang, N. Li, and S. Jha. Locally differentially private frequent itemset mining. In *SP*, 2018.

[38] T. Wang, Z. Li, N. Li, M. Lopuhaä-Zwakenberg, and B. Skoric. Locally differentially private frequency estimation with consistency. In *NDSS*, 2020.

[39] S. L. Warner. Randomized response: A survey technique for eliminating evasive answer bias. *Journal of the American Statistical Association*, 60(309), 1965.

[40] X. Xiao, G. Wang, and J. Gehrke. Differential privacy via wavelet transforms. In *ICDE*, 2010.

[41] M. Ye and A. Barg. Optimal schemes for discrete distribution estimation under locally differential privacy. *IEEE Transactions on Information Theory*, 2018.