Linear and Range Counting under Metric-based Local Differential Privacy

Zhuolun Xiang*
UIUC
xiangzl@illinois.edu

Bolin Ding
Alibaba Group
bolin.ding@alibaba-inc.com

Xi He
University of Waterloo
xihe@uwaterloo.ca

Jingren Zhou
Alibaba Group
jingren.zhou@alibaba-inc.com

***Abstract*—Local differential privacy (LDP) enables private data sharing and analytics without the need for a trusted data collector. Error-optimal primitives (for, *e.g.*, estimating means and item frequencies) under LDP have been well studied. For analytical tasks such as range queries, however, the best known error bound is dependent on the domain size of private data, which is potentially prohibitive. This deficiency is inherent as LDP protects the same level of indistinguishability between any pair of private data values for each data downer.**

In this paper, we utilize an extension of ϵ-LDP called Metric-LDP or E-LDP, where a metric E defines heterogeneous privacy guarantees for different pairs of private data values and thus provides a more flexible knob than ϵ does to relax LDP and tune utility-privacy trade-offs. We show that, under such privacy relaxations, for analytical workloads such as linear counting, multi-dimensional range counting queries, and quantile queries, we can achieve significant gains in utility. In particular, for range queries under E-LDP where the metric E is the L^1-distance function scaled by ϵ, we design mechanisms with errors independent on the domain sizes; instead, their errors depend on the metric E, which specifies in what granularity the private data is protected. We believe that the primitives we design for E-LDP will be useful in developing mechanisms for other analytical tasks, and encourage the adoption of LDP in practice.

Full version of this paper at: https://arxiv.org/abs/1909.11778

I. Introduction

After more than a decade of research and development, differential privacy (DP) [1] has become the *de facto* standard for privacy protection, and is being used or actively explored by major companies in various data applications and services, *e.g.*, Apple [2], Google [3], Uber [4], Microsoft [5], and Alibaba [6]. This privacy guarantee allows releasing aggregate information of the population while protecting individual's data. The degree of protection is characterized by a parameter ϵ, which is used to tune a trade-off between the level of privacy protection and the error of data analytics.

Two models of DP have been studied: *centralized differential privacy (CDP)* and *local differential privacy (LDP)*. In CDP, a *trusted* centralized data curator receives data from data owners and ensures a differentially private data release to mistrustful data analysts. In LDP, there is no trusted data curator; each data owner perturbs her data locally and sends the noisy output (LDP report) to the curator.

Recently, LDP has received a significant amount of attention in the real-world deployments of DP [3], [5], as it prevents single-point failures for data breaches and relieves the burden on the data curator to keep data secure. For primitives such as frequency estimation, a sufficient number of data owners and their LDP reports (*e.g.*, refer to lower bounds in [7], [8]) are required to achieve high utility. In more useful tasks such as range queries, more error has to be introduced with additional terms that depend on the domain size and the dimensionality. Improving the utility for queries on datasets with large domain sizes and dimensionalities, where the additional error terms are prohibitive, has been the research focus of LDP algorithms [9]–[12], to encourage the adoption of LDP.

In many applications, LDP is too strict and not flexible, as not all pairs of values require the same level of protection. For instance, when website visits are collected, the website type, *e.g.*, shopping or video website, is less sensitive than the particular website, YouTube or Hulu, or video being visited; when a person's age is collected, whether s/he is an adult or a kid is less sensitive than the exact year or month of birth. Such relaxations have been formalized as Blowfish [13] and $d_{\mathcal{X}}$-privacy [14] in CDP, and geo-indistinguishability [15] and Metric-LDP [16] in LDP. In fact, we can show that these notations are equivalent in terms of how privacy is relaxed.

In this paper, we utilize Metric-LDP [16], which has a metric function defining different levels of privacy requirements for different pairs of values. We study how to make the best of such privacy relaxations to optimize utility-privacy trade-offs and to achieve provably significant utility gains for analytical tasks. We first consider the tasks of linear counting and range counting queries under Metric-LDP. For multi-dimensional range counting queries and a concrete class of metric, we introduce a novel mechanism whose error is *independent on the domain sizes of dimensions*. It achieves significantly better utility than the best known ϵ-LDP algorithms [9] and [6], whose error is prohibitive when the domain sizes and dimensionality are non-trivial under ϵ-LDP. Our algorithms can be applied as primitives in other tasks such as quantile queries for provable utility gains. There were no known algorithms utilizing such relaxations to gain utility for multi-dimensional range counting in the local model. For the equivalent relaxation in CDP, the best-known utility gain [17] (under Blowfish [13]) is much less significant than ours (relatively).

A. Preliminaries

Let $\mathcal{X}$ denote the domain of private values. Suppose there are n *data owners*, each holding a private value $x \in \mathcal{X}$. A *data collector* wants to collect these private values from data owners to conduct analytical tasks. In the *local model of differential privacy* (LDP), a data owner does not trust the data collector; she encodes her private value x locally with a randomized algorithm $\mathcal{A}$, and sends the *LDP report* $\mathcal{A}(x)$ to the data collector. LDP formalizes a type of plausible deniability: given any output $\mathcal{A}(x)$, the likelihoods to generate $\mathcal{A}(x)$ with $\mathcal{A}$ from x and from any other value are approximately the same.

Definition 1 (Local Differential Privacy [18], [19]): A randomized algorithm $\mathcal{A} : \mathcal{X} \to \mathcal{Y}$ is ϵ-locally differentially private (or ϵ-LDP), if for any pair of private values $x, x' \in \mathcal{X}$, and any subset of output $S \subseteq \mathcal{Y}$, we have that $\Pr[\mathcal{A}(x) \in S] \leq e^{\epsilon} \cdot \Pr[\mathcal{A}(x') \in S]$.

Local differential privacy on metric spaces. ϵ-LDP guarantees the same level of protection for all pairs of private values. However, such homogeneous privacy definition may be too strong for many applications. We adopt an extension of LDP called *Metric-LDP* [16], which uses a metric function to customize heterogeneous (different

[*] Work done at Alibaba Group.

levels of) privacy guarantees for different pairs of private values and to tune utility-privacy trade-offs in analytical tasks.

Definition 2 (Metric-based Local Differential Privacy [15], [16]): Let $E : \mathcal{X} \times \mathcal{X} \to \mathbb{R}_{\geq 0}$ define a metric function on the input domain. A randomized algorithm $\mathcal{A} : \mathcal{X} \to \mathcal{Y}$ satisfies Metric-LDP or E-LDP if for any pair of values $x, x' \in \mathcal{X}$ and any subset of output $S \subset \mathcal{Y}$, we have that $\Pr[\mathcal{A}(x) \in S] \leq e^{E(x,x')} \cdot \Pr[\mathcal{A}(x') \in S]$.

Here, smaller $E(x, x')$ implies that it is more sensitive to the data owners whether the private value is x or x'. Similar to DP and LDP, E-LDP also has the sequential composability.

Relationship to other relaxations. Metric-LDP is a generic form of Blowfish [13] and $d_{\mathcal{X}}$-privacy [14] adapted to the local model. In particular, Blowfish introduces the concept of *policy graph*, where each vertex corresponds to a data value and the distance between two vertices measures how strong the protection between the two corresponding values is (the smaller the stronger). Indeed, distance on graphs is a metric. One attempt to further generalize the relaxation is to consider an arbitrary function $E : \mathcal{X} \times \mathcal{X} \to \mathbb{R}_{\geq 0}$, instead of restricting E to the class of metric functions. However, as shown in our full version [20], it is sufficient to focus on metrics on $\mathcal{X}$ (if E is not a metric, an E-LDP algorithm $\mathcal{A}$ also satisfies E'-LDP, where E' defines a metric and E'-LDP is *stronger* than E-LDP [20]).

B. Problem Statement and Our Main Results

Each of the n data owners holds a private value $x_i \in \mathcal{X}$, and let $X = \{x_i\}_{i \in [n]}$ be the whole private dataset. An analytical task $\mathbf{q}(X)$ is to be conducted on X by the data collector.

We focus on single-round LDP mechanisms in this work. With E-LDP reports $\hat{X} = \{\mathcal{A}(x_i)\}_{i \in [n]}$ collected from data owners, the data collector wants to estimate the answer to $\mathbf{q}(X)$ as $\hat{\mathbf{q}}(\hat{X})$ from $\hat{X}$. The privacy is guaranteed on each LDP report $\mathcal{A}(x_i)$, and thus, we do not need to worry about privacy in designing the estimator $\hat{\mathbf{q}}$ as it can be regarded as "post-processing" of LDP reports. $\mathcal{A}$ and $\hat{\mathbf{q}}$ often need to be co-designed, as a *mechanism*, for an analytical task.

Previous work [15], [16] define utility loss as the hardness of reconstructing the real data distribution from LDP reports, represented as the expected difference between the statistical properties based on LDP reports and those based on the real data. However, for a concrete analytical task, there is no guarantee on estimation errors for the algorithms in [15], [16]. An important contribution of our paper is that, for several tasks, we propose mechanisms that achieve provable end-to-end utility (error bounds) under Metric-LDP.

Linear counting (Section III). Let's consider a domain $\mathcal{X} = [m] = \{1, \ldots, m\}$. An *indicator* $\mathbb{1}_{\mathsf{P}}$ is defined to be 1 if the predicate P is true, or 0 if otherwise. The *frequency vector* on the dataset X is $\mathbf{c} = [c_x]^{\intercal}_{x \in [m]}$, where $c_x = \sum_{i=1}^{n} \mathbb{1}_{x_i = x}$ represents the number owners holding a private value x. A *linear counting task* $\mathbf{q}(X)$ is specified by a $q \times m$ *workload matrix* $\mathbf{W}$ with q rows, and asks for $\mathbf{W} \cdot \mathbf{c}$. In particular, each row of $\mathbf{W}$ is a *linear counting query* asking for a linear combination of frequencies. We use the *total expected squared error*, $\mathrm{E}[\|\hat{\mathbf{q}}(\hat{X}) - \mathbf{W} \cdot \mathbf{c}\|^2]$, to measure the utility of an estimation (the expectation is taken over the randomness of n instances of $\mathcal{A}$).

As warm-up, for this class of counting queries, we introduce a mechanism to minimize the above error based on a generic matrix formulation, which is a reminiscence of the class of matrix mechanisms, [21]–[23] and [17], under CDP. But here, we need to carefully model the flexibility introduced by Metric-LDP to optimize the utility, allowing noises of heterogeneous magnitudes to be added at each dimension of the data. This mechanism can be applied for answering one-dim range counting queries with provable error bounds.

Multi-dimensional range counting (Section IV). Let's consider a D-dim domain $\mathcal{X} = [m]^D$, and each data owner i has a private value $x_i \in \mathcal{X} = [m]^D$. A *D-dim range query* is specified by an interval $R = [l_1, r_1] \times \ldots \times [l_D, r_D]$, asking for $\sum_{i=1}^{n} \mathbb{1}_{x_i \in R}$. We want to bound the *expected squared error* for any given range query.

A metric E_{L^1} on $\mathcal{X} = [m]^D$ is defined based on the L^1-distance: $E_{\mathsf{L}^1}(x, y) = \epsilon \|x - y\|_1 = \epsilon \sum_{i=1}^{D} |x[i] - y[i]|$. For any given multi-dimensional range query on $[m]^D$, we introduce an E_{L^1}-LDP mechanism with expected squared error bounded by $\mathrm{O}(n(\frac{2}{\epsilon^2})^D)$, which completely removes the dependency on the domain size m in error. Our algorithm can be extended for weighted range queries.

In comparison, the best known ϵ-LDP algorithms [6], [9] for multi-dimensional range queries have error $\mathrm{O}(\frac{n \log^{2D} m}{\epsilon^2})$. E_{L^1} is equivalent to the policy graph under Blowfish adopted by Haney *et al.* [17] for answering range queries in the centralized setting. The techniques in [17] can be extended to the local model, leading to the best previously known error bound $\mathrm{O}(\frac{nD(\log m)^{2(D-1)}}{\epsilon^2})$ under E_{L^1}-LDP.

Our algorithm replaces the term $\log^{2D} m$ (in previous works) with $1/\epsilon^{2D}$ in the error bound. As ϵ is usually chosen to be a constant no smaller than 1 for reasonable utility in data analytics *under LDP*, especially in the real-world deployments, *e.g.*, $\epsilon \geq 1$ in [5] by Microsoft and $\epsilon \geq 4$ in [2] by Apple, we have $1/\epsilon \ll \log m$ and thus obtain a significant utility boost from the privacy relaxation.

Quantile queries (Section V). We consider quantile queries in a one-dim domain $\mathcal{X} = [m]$. We defer the formal definitions of quantile queries and their errors to Section V, where we will apply our algorithm for range queries as a primitive to answer quantile queries with provable accuracy gain under E_{L^1}-LDP.

All missing proofs are in the full version of this paper [20].

II. Related Work

Generalized privacy notations. An orthogonal line of work under CDP generalizes the quantification of privacy loss, *i.e.*, the divergence between the output distributions of an algorithm on neighboring datasets. Examples are KL- [24], Renyi- [25] differential privacy, and capacity bounded differential privacy [26]. These generalizations aim to achieve tighter privacy composition properties of DP.

A more relevant line of work considers semantic privacy frameworks which (i) clarify assumptions on the adversary and (ii) redefine sensitive information to be kept secret, such as Pufferfish privacy [27], [28] and membership privacy [29]. Specifying a weaker version of adversary under a semantic framework [29], [30] or weaker protection on the sensitive information [13], [17] allows the design of algorithms with better utility than the standard differentially private algorithms. In particular, Blowfish is an instance under such frameworks and provides improved utilities for several tasks including k-means clustering and estimating cumulative histograms [13]. Readers can refer to [31] for a survey on other variants under the centralized setting.

Primitives under LDP. We give a brief summary on the analytical primitives under LDP (without relaxation). Mean/median estimation under ϵ-LDP has been well studied [5], [8], [18] with a matching upper and lower bound. Frequency estimation under LDP is also studied extensively in, *e.g.*, [3], [5], [19], [32]–[35]. They use techniques like hashing (*e.g.*, [34]) and Hadamard transform (*e.g.*, [33], [35]) for good utility. For locally differentially private range queries, the work of [6], [9], [12] present the state-of-the-art.

III. Warmup: Linear Counting Queries

We first consider the task of answering linear counting queries, defined in Section I-B: how to collect each private value in X under E-LDP, and estimate $\mathbf{W} \cdot \mathbf{c}$ for a given workload matrix $\mathbf{W}$.

A. A Generic Matrix Formulation under Metric-LDP

In our mechanism introduced below, the *matrix mechanism* under CDP [21]–[23] is adapted to the local setting, and more importantly, extended to make the best of the flexibility in Metric-LDP.

E-LDP encoding algorithm $\mathcal{A}_{\mathbf{A},\mathbf{B},\mathbf{s}}(x)$. Data owners use the same $p \times m$ *strategy matrix* $\mathbf{A} = [\mathbf{a}_1 \ldots \mathbf{a}_p]^{\intercal}$ to encode their data. Every row of the workload matrix $\mathbf{W}$ can be reconstructed using a linear combination of rows of $\mathbf{A}$, i.e., $\mathbf{W} = \mathbf{B}\mathbf{A}$ for some $q \times p$ matrix $\mathbf{B}$. A properly chosen $\mathbf{A}$ can reduce the noise to be injected and enable the reconstruction of $\mathbf{W}$. Each data owner first encode her value x as a length-m binary vector $\mathbf{h}_x = [0, ..., 0, 1, 0, ..., 0]^\intercal$ where only the x-th position is 1. We use $\mathrm{Lap}(s)$ to represent a random sample drawn from Laplace distribution with parameter s. Each data owner draws p independent random samples $\mathrm{Lap}(\mathbf{s}) = [\mathrm{Lap}(s_1), \ldots, \mathrm{Lap}(s_p)]^\intercal$, with parameters $\mathbf{s} = [s_1, \ldots, s_p]^\intercal$, and reports:

$$\mathcal{A}_{\mathbf{A},\mathbf{B},\mathbf{s}}(x) = \mathbf{A}\cdot\mathbf{h}_x + [\mathrm{Lap}(s_1), \ldots, \mathrm{Lap}(s_p)]^\intercal = \mathbf{A}\cdot\mathbf{h}_x + \mathrm{Lap}(\mathbf{s}).$$

Proposition 1: $\mathcal{A}_{\mathbf{A},\mathbf{B},\mathbf{s}}$ is E-LDP, if for any pair of $x, x' \in [m]$,

$$\begin{bmatrix}\frac{1}{s_1} & \frac{1}{s_2} & \cdots & \frac{1}{s_p}\end{bmatrix} |\mathbf{A}(\mathbf{h}_x - \mathbf{h}_{x'})| \le E(x, x'),$$

where $|\mathbf{A}(\mathbf{h}_x - \mathbf{h}_{x'})| = \begin{bmatrix}|\mathbf{a}_1^\intercal(\mathbf{h}_x - \mathbf{h}_{x'})| & \cdots & |\mathbf{a}_p^\intercal(\mathbf{h}_x - \mathbf{h}_{x'})|\end{bmatrix}^\intercal$, namely, $|\mathbf{A}(\mathbf{h}_x - \mathbf{h}_{x'})|$ is the vector obtained by taking the absolute values of entries in vector $\mathbf{A}(\mathbf{h}_x - \mathbf{h}_{x'})$.

Answering linear counting workload. After collecting $\hat{X} = \{\mathbf{r}_i = \mathcal{A}_{\mathbf{A},\mathbf{B},\mathbf{s}}(x_i)\}_{i\in[n]}$ from n data owners, the data collector estimates the linear counting queries $\mathbf{W}\cdot\mathbf{c}$ as $\mathbf{B}\cdot\sum_{i=1}^{n}\mathbf{r}_i$. We can show that it is unbiased and its error depends on the choice of $\mathbf{B}$.

Proposition 2: The estimation $\hat{\mathbf{q}}(\hat{X}) = \mathbf{B}\cdot\sum_{i=1}^{n}\mathbf{r}_i$ is an unbiased estimation of $\mathbf{W}\cdot\mathbf{c}$. The total expected squared error of $\hat{\mathbf{q}}(\hat{X})$ is

$$\mathrm{E}[\|\hat{\mathbf{q}}(\hat{X}) - \mathbf{W}\cdot\mathbf{c}\|^2] = 2n\cdot\mathrm{Trace}[\mathbf{B}^\intercal\mathbf{B}\cdot\mathrm{diag}(s_1^2, \ldots, s_p^2)]$$

where $\mathrm{diag}(s_1^2, \ldots, s_p^2)$ is a $p\times p$ diagonal matrix with diagonal elements $s_1^2, \ldots, s_p^2$.

An optimization problem. Given a workload $\mathbf{W}$ and a metric E, we can choose $\mathbf{A}$, $\mathbf{B}$, and $\mathbf{s} = [s_1, \ldots, s_p]^\intercal$ properly to gain utility, *i.e.*, minimizing the total expected squared error: formally,

$$\begin{aligned}&\min_{\mathbf{A},\mathbf{B},\mathbf{s}} 2n\cdot\mathrm{Trace}[\mathbf{B}^\intercal\mathbf{B}\cdot\mathrm{diag}(s_1^2, \ldots, s_p^2)] \qquad (1)\\ &\text{s.t. } \begin{bmatrix}\frac{1}{s_1} & \frac{1}{s_2} & \cdots & \frac{1}{s_p}\end{bmatrix} |\mathbf{A}(\mathbf{h}_x - \mathbf{h}_{x'})| \le E(x, x'), \forall x, x' \in [m]\\ &\quad \mathbf{B}\mathbf{A} = \mathbf{W}, \qquad s_k \ge 0,\ \forall k \in [p]\end{aligned}$$

It is hard to solve (1) efficiently, unless $\mathbf{A}$ is fixed (then it becomes convex but a bad choice of $\mathbf{A}$ may lead to a suboptimal solution).

B. One-dimensional Range Queries

We consider one-dim range queries now. A range query is specified by an interval $R = [l, r] \subseteq [m]$, and asks for $\sum_{i=1}^{n}\mathbb{1}_{x_i\in R}$. Mechanisms are developed to handle range queries under ϵ-LDP [6], [9]. It is natural to consider a metric $E_{\mathsf{L}^1}(x, x') = \epsilon|x - x'|$ for $x, x' \in [m]$, which means values that are closer are more sensitive to each other. We can achieve better utility under E_{L^1}-LDP by solving the optimization problem (1) for this special case.

Let $\mathbf{W}_m$ be the workload matrix for all possible one-dimensional range queries on $[m]$. We consider a strategy matrix $\mathbf{A} = \mathbf{L}_m$ (an $m\times m$ $\{0, 1\}$-matrix with bottom-left triangular area filled with 1), which intuitively means that each user creates an LDP report for estimating every prefix sum of the frequencies (a range query can be answered as the difference between two prefixes).

$$\mathbf{W}_3 = \begin{bmatrix}1 & 1 & 1 & 0 & 0 & 0\\ 0 & 1 & 1 & 1 & 1 & 0\\ 0 & 0 & 1 & 0 & 1 & 1\end{bmatrix}^\intercal \quad \mathbf{L}_3 = \begin{bmatrix}1 & 0 & 0\\ 1 & 1 & 0\\ 1 & 1 & 1\end{bmatrix} \qquad (2)$$

Above are examples of $\mathbf{W}_3$ and $\mathbf{L}_3$. With $\mathbf{A} = \mathbf{L}_m$, we can rewrite the optimization problem (1) as:

$$\min_{\mathbf{s}} \quad 2n\sum_{x=1}^{m} m s_x^2 \qquad \text{s.t.} \sum_{i=l}^{r-1}\frac{1}{s_i} \le \epsilon(r - l),\ \forall 1 \le l < r \le m. \\ s_x \ge 0,\ \forall x \in [m]$$

We can easily derive the optimal solution to the above problem as $s_k = \frac{1}{\epsilon}$ for $k \in [m-1]$ and $s_m = 0$, and thus, the total expected squared error is $\frac{2nm(m-1)}{\epsilon^2}$ for $m(m-1)/2$ range queries. For each range query, the squared error is $\mathrm{O}(\frac{n}{\epsilon^2})$. This already gives an $\mathrm{O}(\log^2 m)$ improvement (indeed, under a relaxed privacy notation, E_{L^1}-LDP) on the utility in comparison to the mechanisms in [9] and [6] (which have expected squared error $\mathrm{O}(\frac{n\log^2 m}{\epsilon^2})$, under ϵ-LDP).

IV. Multi-dimensional Range Counting Queries

We now consider the task of answering range counting queries in a D-dim domain, defined in Section I-B: a range query is specified by a range $R = [l_1, r_1]\times\ldots\times[l_D, r_D] \subseteq [m]^D$, asking for $c(R) = \sum_{i=1}^{n}\mathbb{1}_{x_i\in R}$. We provide E_{L^1}-LDP when collecting each x_i.

Our results can be extended for E_{L^p}, due to the relation: $\|x\|_p \le \|x\|_1 \le D^{1-\frac{1}{p}}\|x\|_p$ for any $p \ge 1$. Therefore, any algorithm that is E_{L^1}-LDP with parameter ϵ is also E_{L^p}-LDP with parameter $D^{1-\frac{1}{p}}\epsilon$.

Notations. For $x \in [m]^D$ or a vector $\mathbf{v}$, we use $x[i]$ or $\mathbf{v}[i]$ to denote the coordinate on the ith dimension, respectively. We assign an *index* ($\mathsf{ind} : [m]^D \to [m^D]$) to each value in the D-dim domain $[m]^D$, which numbers all the values in $[m]^D$ from 1 to m^D: $\mathsf{ind}(x) = 1 + \sum_{d=1}^{D} m^{d-1}(x[d] - 1)$. If it is clear from the context, we will refer to x as both a value in $[m]^D$ and its index $\mathsf{ind}(x)$, interchangeably.

Comparison to existing approaches. Existing methods for answering range queries under ϵ-LDP are either based on hierarchical histograms [6], [9] or discrete Haar transform [9]; both schemes, however, rely on $(\log m)^{\mathrm{O}(D)}$ independent frequency estimations per query, and each frequency estimation as a black box is inherently hard with error as least $\Omega(\frac{n}{\epsilon^2})$ [36], even under E_{L^1}-LDP (consider a domain with two possible values). And thus, the $(\log m)^{\mathrm{O}(D)}$ term is inevitable for existing methods even under relaxation. In the centralized Blowfish, under a policy equivalent to E_{L^1}, Haney *et al.* [17] made some improvement but failed to completely remove the $(\log m)^{\mathrm{O}(D)}$ term. Their approach has expected squared error $\mathrm{O}(\frac{D(\log m)^{3(D-1)}}{\epsilon^2})$ under E_{L^1}-CDP, which is only better than the Privelet mechanism [37] under ϵ-CDP by a $\Theta(\log^3 m)$ factor. Haney *et al.* [17]'s method can be extended to the local model E_{L^1}-LDP, with error $\mathrm{O}(\frac{nD(\log m)^{2(D-1)}}{\epsilon^2})$, reducing the expected squared error in the methods [6], [9] under ϵ-LDP only by a factor of $\Theta(\log^2 m)$.

Our goal here is to remove the prohibitive term $(\log m)^{\mathrm{O}(D)}$ from error bounds under E_{L^1}-LDP. Our mechanism in Section IV-A has error bounded by $\mathrm{O}(n(\frac{2}{\epsilon^2})^D)$ when ϵ is small. Our error bounds are completely independent on the domain size m. As ϵ is usually chosen to be a constant ≥ 1 in real-world deployments, *e.g.*, [2] and [5], we have $1/\epsilon \ll \log m$, and thus replacing $\log^{2D} m$ with $1/\epsilon^{2D}$ improves the utility significantly. Our method can be considered as a special type of transformation similar to discrete Haar transform, but with a nice property that during the summation of frequency estimations of single values, most noise from perturbation will be canceled out.

A. Multi-dimensional Range Query under E_{L^1}-LDP

E_{L^1}-LDP encoding algorithm $\mathcal{A}(x)$. Let x denote the D-dim private value held by a data owner. She first encodes each dimension d of x, $x[d] \in [m]$, into a length-m vector $\mathbf{b}_d$:

$$\mathbf{b}_d = [\underbrace{-1, -1, \ldots, -1}_{x[d]-1}, \underbrace{1, 1, \ldots, 1}_{m-x[d]+1}],$$

where the first up to the $(x[d]-1)$-th position are -1's and the rest are 1's. She will then perturb the vector $\mathbf{b}_d$ into $\mathbf{r}_d$ with standard random-flipping operation on each position $k \in [m]$:

$$\mathbf{r}_d[k] = \begin{cases}\mathbf{b}_d[k] & \text{with prob. } \frac{e^\epsilon}{e^\epsilon+1}\\ -\mathbf{b}_d[k] & \text{with prob. } \frac{1}{e^\epsilon+1}\end{cases}.$$

Each data owner reports $\mathcal{A}(x) = \mathbf{R} = [\mathbf{r}_1, \mathbf{r}_2, ..., \mathbf{r}_D]^\intercal$ (a $D\times m$ matrix), to the data collector. It is easy to verify its privacy guarantee.

Proposition 3: $\mathcal{A}(x)$ is E_{L^1}-LDP.

Range query estimation. After collecting data owners' reports $\mathbf{R}_1$, $\dots$, $\mathbf{R}_n$, where $\mathbf{R}_i = \mathcal{A}(x_i)$, the data collector first obtains a length-m^D vector $\mathbf{o} = [o_1, \dots, o_{m^D}]^\intercal$, called *observations*:

$$o_x = \sum_{i=1}^{n} \prod_{d=1}^{D} \mathbf{R}_i[d, x[d]], \quad \text{for each } x \in [m]^D, \tag{3}$$

where $\mathbf{R}[a,b]$ denotes the value in row a and column b of matrix $\mathbf{R}$. Recall that the index $\mathsf{ind} : [m]^D \to [m^D]$ numbers values $x \in [m]^D$ as $\mathsf{ind}(x) = 1 + \sum_{d=1}^{D} m^{d-1}(x[d]-1)$. When referring to indexes of entries in a vector, we will use x and $\mathsf{ind}(x)$, interchangeably. Thus, by o_x, we mean the $\mathsf{ind}(x)$-th position $o_{\mathsf{ind}(x)}$ in the vector $\mathbf{o}$.

For example, if $n = 2$, $D = 2$ and $m = 3$, with $\mathbf{R}_1 = \begin{bmatrix} 1 & -1 & 1 \\ -1 & -1 & -1 \end{bmatrix}$ and $\mathbf{R}_2 = \begin{bmatrix} 1 & 1 & -1 \\ 1 & -1 & -1 \end{bmatrix}$, for $x = (1,1)$, $o_x = \mathbf{R}_1[1, x[1]] \cdot \mathbf{R}_1[2, x[2]] + \mathbf{R}_2[1, x[1]] \cdot \mathbf{R}_2[2, x[2]] = 1 \cdot (-1) + 1 \cdot 1$.

We will use $\mathbf{o}$ to estimate the frequencies of all values in $[m]^D$. Let $\mathbf{c} = [c_1, ..., c_{m^D}]^\intercal$ be the vector representing true frequencies of all values $x \in [m]^D$ among the n data owners ($c_x = \sum_{i=1}^{n} \mathbb{1}_{x_i = x}$). As proved in Theorem 1 of the full version [20], there exists a relation

$$\mathrm{E}[\mathbf{o}] = (\frac{e^\epsilon - 1}{e^\epsilon + 1})^D \mathbf{B}_{m,D} \cdot \mathbf{c}, \tag{4}$$

where $\mathbf{B}_{m,D}$ is an $m^D \times m^D$ matrix that can be partitioned into $m \times m$ submatrices $\mathbf{B}_{m,D-1}$, satisfying the following recursive relation,

$$\mathbf{B}_{m,d} = \begin{bmatrix} \mathbf{B}_{m,d-1} & -\mathbf{B}_{m,d-1} & \cdots & -\mathbf{B}_{m,d-1} \\ \vdots & \ddots & \ddots & \vdots \\ \vdots & & \ddots & -\mathbf{B}_{m,d-1} \\ \mathbf{B}_{m,d-1} & \cdots & \cdots & \mathbf{B}_{m,d-1} \end{bmatrix}, \tag{5}$$

for $2 \le d \le D$. That is, after partition, the submatrices in the bottom-left triangle are all $\mathbf{B}_{m,d-1}$ and rest of the submatrices are all $-\mathbf{B}_{m,d-1}$. For the base case when $D = 1$,

$$\mathbf{B}_{m,1} = \begin{bmatrix} 1 & -1 & \cdots & -1 \\ \vdots & \ddots & \ddots & \vdots \\ \vdots & & \ddots & -1 \\ 1 & \cdots & \cdots & 1 \end{bmatrix}.$$

- *Estimating single-value frequencies.* The estimated frequency vector $\hat{\mathbf{c}} = [\hat{c}_1, ..., \hat{c}_{m^D}]^\intercal$ can be thus computed from (4) as follows:

$$\hat{\mathbf{c}} = (\frac{e^\epsilon + 1}{e^\epsilon - 1})^D \mathbf{B}_{m,D}^{-1} \cdot \mathbf{o}. \tag{6}$$

For any value $x \in [m]^D$, $\hat{c}_x$ is the estimated frequency of x.

- *Estimating answers to range queries.* For a D-dim range query $R = [l_1, r_1] \times \cdots \times [l_D, r_D]$, the data collector can estimate its answer by directly summing up the estimated frequencies of all $x \in R$:

$$\hat{c}(R) = \sum_{x \in R} \hat{c}_x = (\frac{e^\epsilon + 1}{e^\epsilon - 1})^D \sum_{x \in R} \mathbf{e}_x \mathbf{B}_{m,D}^{-1} \cdot \mathbf{o}, \tag{7}$$

where $\mathbf{e}_x$ is a 0-1 row vector with only the $\mathsf{ind}(x)$-th entry as 1, and $\mathbf{e}_x \mathbf{B}_{m,D}^{-1}$ gives the $\mathsf{ind}(x)$-th row in $\mathbf{B}_{m,D}^{-1}$.

- *Computing inverse* $\mathbf{B}_{m,D}^{-1}$. The rest question is thus how to compute the matrix inverse $\mathbf{B}_{m,D}^{-1}$. It turns out that we can efficiently compute it in a recursive way. $\mathbf{B}_{m,D}^{-1}$ can be partitioned into $m \times m$ submatrices $\mathbf{B}_{m,D-1}^{-1}$, defined by the following recursive relation for $2 \le d \le D$:

$$\mathbf{B}_{m,d}^{-1} = \frac{1}{2} \begin{bmatrix} \mathbf{B}_{m,d-1}^{-1} & 0 & \cdots & 0 & \mathbf{B}_{m,d-1}^{-1} \\ -\mathbf{B}_{m,d-1}^{-1} & \mathbf{B}_{m,d-1}^{-1} & \ddots & \vdots & 0 \\ 0 & -\mathbf{B}_{m,d-1}^{-1} & \ddots & 0 & \vdots \\ \vdots & \ddots & \ddots & \mathbf{B}_{m,d-1}^{-1} & 0 \\ 0 & \cdots & 0 & -\mathbf{B}_{m,d-1}^{-1} & \mathbf{B}_{m,d-1}^{-1} \end{bmatrix}. \tag{8}$$

Recursively, $\mathbf{B}_{m,d-1}^{-1}$ is a $m^{d-1} \times m^{d-1}$ matrix. In the base case, $\mathbf{B}_{m,1}^{-1}$ is the $m \times m$ matrix:

$$\mathbf{B}_{m,1}^{-1} = \frac{1}{2} \begin{bmatrix} 1 & 0 & \cdots & 0 & 1 \\ -1 & 1 & \ddots & \vdots & 0 \\ 0 & -1 & \ddots & 0 & \vdots \\ \vdots & \ddots & \ddots & 1 & 0 \\ 0 & \cdots & 0 & -1 & 1 \end{bmatrix}.$$

It can be shown that $\mathbf{B}_{m,d}^{-1}$ is indeed the inverse of $\mathbf{B}_{m,d}$. Please refer to Lemma 2 in the full version [20] for more details.

B. Analysis of Algorithm

We focus on accuracy analysis here. Computational and space complexity is also analyzed in Section 4.2 of the full version [20].

First, we can show that the estimations are unbiased.

Theorem 1 (Unbiasedness): The estimates for the frequency of any single value and the answer to any range query R (Equations (6) and (7), respectively) are unbiased, *i.e.*, $\mathrm{E}[\hat{\mathbf{c}}] = \mathbf{c}$ and $\mathrm{E}[\hat{c}(R)] = c(R)$.

According to Equation (7), our mechanism estimates range query by by summing up all estimations of single values' frequencies in the range R. Surprisingly, the range query's estimation error has the same upper bound as the single-value frequency's estimation error, instead of $\mathrm{O}(m^D)$ times larger as one may expect naturally.

Theorem 2 (Single-value frequency): For any value $x \in [m]^D$, the expected squared error of estimation $\hat{c}_x$ is $\mathrm{E}[\|\hat{c}_x - c_x\|^2] =$

$$\mathrm{Var}[\hat{c}_x] = \mathrm{O}\bigg((\frac{e^\epsilon + 1}{e^\epsilon - 1})^{2D} 2^{-D} (1 - (\frac{e^\epsilon - 1}{e^\epsilon + 1})^{2D}) n \bigg).$$

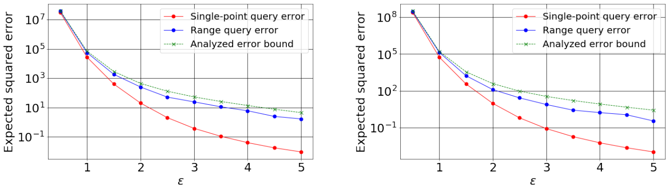
Theorem 3 (Range query): For any range query $R = [l_1, r_1] \times [l_2, r_2] \times \cdots \times [l_D, r_D]$, the expected squared error of the estimated answer $\hat{c}(R)$ is $\mathrm{E}[\|\hat{c}(R) - c(R)\|^2] =$

$$\mathrm{Var}[\hat{c}(R)] = \mathrm{O}\bigg((\frac{e^\epsilon + 1}{e^\epsilon - 1})^{2D} 2^{-D} (1 - (\frac{e^\epsilon - 1}{e^\epsilon + 1})^{2D}) n \bigg).$$

When ϵ is small, we have $\mathrm{Var}[\hat{c}_x]$, $\mathrm{Var}[\hat{c}(R)] \approx \mathrm{O}((\frac{2}{\epsilon^2})^D n)$. Complete proofs of Theorems 1-3 are in the full version [20].

Here, let's give some intuitive explanation on why the expected squared errors for both range query and single value have the same upper bound. This is from the nice property of the bias correction matrix $\mathbf{B}_{m,D}^{-1}$. When calculating the estimation for the range query in (7), the expected squared error of the estimation is affected by the non-zero terms in $\sum_{x \in R} \mathbf{e}_x \mathbf{B}_{m,D}^{-1}$, which is a summation of multiple rows of $\mathbf{B}_{m,D}^{-1}$ with each row corresponding to one point in R. Fortunately, instead of exploding the number of non-zero terms in the summation by $\mathrm{O}(|R|) = \mathrm{O}(m^D)$, most of the terms are canceled out, leaving the number of remaining non-zero terms to be equal to that in a single-value frequency query. Therefore, the expected squared error is not amplified from single values to range queries.

Simulation results. We perform a simulation to evaluate the empirical error of our mechanism and verify our theoretical analysis (Theorems 2 and 3). We use synthetic data generated as follows.



(a) $D = 5$, $n = 1000$, $m = 10$

(b) $D = 6$, $n = 1000$, $m = 10$

Fig. 1: Squared error in estimating single-value frequencies and multi-dimensional range queries

For any D-dimensional private value, each of its dimension follows Zipf distribution with parameter 1.1. We implement our mechanism in Section IV-A and measure the average squared error of frequency queries over all single values. We also randomly generate 100 range queries and measure the average squared error of all these range queries. The mechanism (both encoding and estimation) is repeated three times. The analyzed error bound is the one in Theorem 2 or 3 with the constant set to be 1 in the big-oh upper bound – it is equal to the analytical upper bound shown in the proofs in the full version [20]. As we can observe from Figure 1, both the empirical squared errors of single-value frequencies and range queries are below our analyzed error bound, verifying the effectiveness of our mechanism.

Handling continuous domains. In general, the input may be vectors from a real domain $[0, \Sigma]^D$. To apply the mechanism introduced in this section, a mapping from $[0, \Sigma]^D$ to $[m]^D$ is needed (*e.g.*, partitioning each dimension $[0, \Sigma]$ evenly into sub-intervals and mapping each of them to a value in $[m]$). At first glance, it is appealing to choose a larger m for such a discretization process, since the *truncation error* (due to the rounding from $[0, \Sigma]$ to $[m]$) can be smaller as a more accurate range in $[m]$ can be used for the range query in $[0, \Sigma]$, while the error bounds in Theorems 2 and 3 are independent on m. However, a larger m means the "real" distance between i and $i+1$ in $[m]$ is smaller in the original domain $[0, \Sigma]$, and thus a smaller ϵ is needed to guarantee the same level of privacy protection, resulting in a larger *estimation error* according to Theorems 2 and 3. Therefore, it is possible to choose an optimal value of m to minimize the total error of the two types introduced above. The optimal selection of m may depend on the distribution of the input data, which is hard to be quantified, and is private, too. We leave it as an intriguing open question for future work.

C. Extensions to More Complex Range Queries

We briefly discuss how our method can be extended to the case where each dimension has a different size, and weighted range query. More details can be found in Section 4.3 of the full version [20].

When dimension sizes are different. When each dimension has a different size, *i.e.*, the private values are in domain $[m_1] \times \ldots \times [m_D]$, our mechanism in Section IV-A also applies after a few changes: (1) Each data owner reports a *length-m_d vector* instead of a length-m vector for each dimension d, constructed in the same way as in Section IV-A. (2) The data collector estimates the frequencies of all single values using a $(\prod_{j=1}^{D} m_j) \times (\prod_{j=1}^{D} m_j)$ matrix $\mathbf{C}_D^{-1}$ (defined in [20]) instead of $\mathbf{B}_D^{-1}$. Any range query R is also answered by $\hat{c}(R) = \sum_{x \in R} \hat{c}_x$. The correctness proof and accuracy analysis are similar to the case with identical domain sizes, and the error bounds of the estimations are identical to the bounds in Theorems 2-3.

Weighted range queries. Each data owner i may holds a weight $w_i \in W$. A *weighted range query* asks $c_{\mathbf{w}}(R) = \sum_{i=1}^{n} w_i \mathbb{1}_{x_i \in R}$. W.l.o.g., we consider weights from the domain $W = [0, \Delta]$. *If the weights are non-private information*, we can partition all data owners into groups g_w by their weights w, and construct an estimator of unweighted multi-dimensional range queries $\hat{c}_{g_w}(\cdot)$ for each group g_w. To answer a weighted range query, we sum up the weighted answers from all groups, $\hat{c}_{\mathbf{w}}(R) = \sum_{w \in W} w \cdot \hat{c}_{g_w}(R)$. *If the weights are private information*, we can treat the weight as an extra private dimension for each data owner, and then use unweighted $(D+1)$-dimensional range query oracle to answer weighted D-dimensional range queries. In both cases, the error bounds have an additional term Δ^2 but are still independent on the domain size m.

V. Application: Quantile Queries

Consider quantile queries in a one-dim domain $\mathcal{X} = [m]$. The *percentile* of a value x in $X = \{x_i\}_{i \in [n]}$ is $\sigma(x) = \frac{1}{n} \sum_{i=1}^{n} \mathbb{1}_{x_i \leq x}$, which calculates the fraction of values that are no larger than x in X. The interval $I(x) = (\sigma(x-1), \sigma(x)] \subseteq [0, 1]$ is said to be the *percentile interval* of x. The *p-quantile* of X is defined to be the value x^*, such that $\sigma(x^* - 1) < p \leq \sigma(x^*)$, *i.e.*, p is in x^*'s percentile interval. Let $\hat{x}^*$ be an estimated p-quantile. The estimation goal is to make sure that the percentile interval $I(\hat{x}^*)$ is close to p. We define the error of the estimation $\hat{x}^*$ to be $\mathrm{Err}[\hat{x}^*] = \inf_{\hat{p} \in I(\hat{x}^*)} |\hat{p} - p|$. We want to bound error $\mathrm{Err}[\hat{x}^*]$ with high probability. Note that our error definition is essentially the ϵ-approximate p-quantile in literature, *e.g.*, [38] (with inf considered here as X is a multiset).

Answering quantile queries under E_{L^1}-LDP. Our mechanism follows the approach proposed in Section 4.7 of [9], which uses one-dimensional range query mechanism as a primitive and perform binary search to estimate the p-quantile. *Our main contribution here is to provide formal analysis on the utility of the mechanism, and compare the mechanisms under ϵ-LDP and E_{L^1}-LDP.*

For data owners, private values are encoded using the algorithm (its 1-dim case) in Section IV-A to guarantee E_{L^1}-LDP. For the data collector, let $\hat{c}([l, r])$ be the answer estimated using the mechanism introduced in Section IV-A for a one-dim range query $[l, r]$. We can then estimate the percentile of value x as $\hat{\sigma}(x) = \hat{c}([1, x])/n$. Our mechanism answers a p-quantile query as follows.

1) Construct an oracle (Section IV-A) for answering 1-dim range queries on data owners' reports.
2) Perform binary search on the domain $[m]$ until a value $\hat{x}^*$ s.t. $\hat{\sigma}(\hat{x}^* - 1) < p \leq \hat{\sigma}(\hat{x}^*)$ is found, with $\hat{\sigma}$ defined above.
3) Output $\hat{x}^*$ as the estimation for the quantile query.

Accuracy analysis. With high probability, the error is bounded.

Theorem 4: With probability at least $1 - \delta$, our quantile query mechanism guarantees that for an estimated p-quantile $\hat{x}^*$

$$\mathrm{Err}[\hat{x}^*] \leq \frac{2(e^\epsilon + 1)}{e^\epsilon - 1} \cdot \sqrt{\frac{2}{n} \log \frac{2 \log m}{\delta}}.$$

The proof of the above theorem can be found in Section 4.4 of the full version [20]. From Theorem 4, the estimation error of our mechanism is bounded by $\mathrm{O}(\frac{1}{\epsilon\sqrt{n}} \sqrt{\log \log m})$ with high probability. The state-of-the-art ϵ-LDP mechanisms [6], [9] for one-dim range queries can be plugged in step 1 (as suggested in [9]), leading to error $\mathrm{O}(\frac{1}{\epsilon\sqrt{n}} \log m \sqrt{\log \log m})$ using our analysis, which is $\mathrm{O}(\log m)$ times larger compared to our mechanism under E_{L^1}-LDP.

VI. Conclusion

This paper investigates local differential privacy on metric spaces (or E-LDP), which is a relaxation of ϵ-LDP to customize the levels of indistinguishability among different pairs of values using a metric function E. In this work, we design a generic E-LDP mechanism (generalizing matrix mechanisms in CDP) to trade-off privacy for utility of linear counting queries. For multi-dimensional range queries, we introduce a novel E-LDP algorithm under L^1-metric with an error which is independent on the size m of each dimension. This technique can also help reduce the error of ϵ-LDP algorithms for quantile queries by a factor of $\log m$ under E-LDP. Our techniques apply to L^p-LDP as well. As future work, we would apply techniques in this paper as primitives for other analytical tasks; we would also expect that the transform matrices developed in this paper can be used to improve algorithms in the centralized setting under similar relaxations (*e.g.*, in Blowfish privacy).

References

[1] C. Dwork, "Differential privacy," in *Proceedings of the 33rd International Conference on Automata, Languages and Programming (ICALP)*, 2006, pp. 1–12.

[2] A. D. P. Team, "Learning with privacy at scale," *Apple Machine Learning J.*, 2017.

[3] Ú. Erlingsson, V. Pihur, and A. Korolova, "Rappor: Randomized aggregatable privacy-preserving ordinal response," in *Proceedings of the 2014 ACM Conference on Computer and Communications Security (CCS)*, 2014, pp. 1054–1067.

[4] N. M. Johnson, J. P. Near, and D. Song, "Towards practical differential privacy for SQL queries," *PVLDB*, vol. 11, no. 5, pp. 526–539, 2018.

[5] B. Ding, J. Kulkarni, and S. Yekhanin, "Collecting telemetry data privately," in *Advances in Neural Information Processing Systems (NIPS)*, 2017, pp. 3571–3580.

[6] 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 (SIGMOD)*, 2019, pp. 159–176.

[7] T.-H. H. Chan, E. Shi, and D. Song, "Optimal lower bound for differentially private multi-party aggregation," in *Proceedings of the 20th Annual European Conference on Algorithms (ESA)*, 2012, pp. 277–288.

[8] J. C. Duchi, M. I. Jordan, and M. J. Wainwright, "Minimax optimal procedures for locally private estimation," *Journal of the American Statistical Association*, vol. 113, no. 521, pp. 182–201, 2018.

[9] G. Cormode, T. Kulkarni, and D. Srivastava, "Answering range queries under local differential privacy," *PVLDB*, vol. 12, no. 10, pp. 1126–1138, 2019.

[10] 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*, vol. 13, no. 9, pp. 2151–2166, 2018.

[11] T. Wang, Z. Li, N. Li, M. Lopuhaä-Zwakenberg, and B. Skoric, "Consistent and accurate frequency oracles under local differential privacy," *CoRR*, vol. abs/1905.08320, 2019.

[12] M. Xu, T. Wang, B. Ding, J. Zhou, C. Hong, and Z. Huang, "Dpsaas: Multi-dimensional data sharing and analytics as services under local differential privacy," *PVLDB*, vol. 12, no. 12, pp. 1862–1865, 2019.

[13] X. He, A. Machanavajjhala, and B. Ding, "Blowfish privacy: Tuning privacy-utility trade-offs using policies," in *Proceedings of the 2014 International Conference on Management of Data (SIGMOD)*, 2014, pp. 1447–1458.

[14] K. Chatzikokolakis, M. E. Andrés, N. E. Bordenabe, and C. Palamidessi, "Broadening the scope of differential privacy using metrics," in *Proceedings of the 13th International Symposium on Privacy Enhancing Technologies (PETS)*, 2013, pp. 82–102.

[15] M. E. Andrés, N. E. Bordenabe, K. Chatzikokolakis, and C. Palamidessi, "Geo-indistinguishability: Differential privacy for location-based systems," in *Proceedings of the 2013 ACM Conference on Computer and Communications Security (CCS)*, 2013, pp. 901–914.

[16] M. Alvim, K. Chatzikokolakis, C. Palamidessi, and A. Pazii, "Local differential privacy on metric spaces: optimizing the trade-off with utility," in *Proceedings of the 31st IEEE Computer Security Foundations Symposium (CSF)*, 2018, pp. 262–267.

[17] S. Haney, A. Machanavajjhala, and B. Ding, "Design of policy-aware differentially private algorithms," *PVLDB*, vol. 9, no. 4, pp. 264–275, 2015.

[18] J. C. Duchi, M. I. Jordan, and M. J. Wainwright, "Local privacy and statistical minimax rates," in *Proceedings of the 54th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, 2013, pp. 429–438.

[19] J. C. Duchi, M. J. Wainwright, and M. I. Jordan, "Local privacy and minimax bounds: Sharp rates for probability estimation," in *Advances in Neural Information Processing Systems (NIPS)*, 2013, pp. 1529–1537.

[20] Z. Xiang, B. Ding, X. He, and J. Zhou, "Linear and range counting under metric-based local differential privacy," 2019. [Online]. Available: https://arxiv.org/abs/1909.11778

[21] C. Li, M. Hay, V. Rastogi, G. Miklau, and A. McGregor, "Optimizing linear counting queries under differential privacy," in *Proceedings of the 29th ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems (PODS)*, 2010, pp. 123–134.

[22] C. Li and G. Miklau, "An adaptive mechanism for accurate query answering under differential privacy," *PVLDB*, vol. 5, no. 6, pp. 514–525, 2012.

[23] ——, "Optimal error of query sets under the differentially-private matrix mechanism," in *Proceedings of the 16th International Conference on Database Theory (ICDT)*, 2013, pp. 272–283.

[24] Y. Wang, J. Lei, and S. E. Fienberg, "On-average kl-privacy and its equivalence to generalization for max-entropy mechanisms," in *Proceedings of the 2016 International Conference on Privacy in Statistical Databases (PSD)*, 2016, pp. 121–134.

[25] I. Mironov, "Rényi differential privacy," in *Proceedings of the 30th IEEE Computer Security Foundations Symposium (CSF)*, 2017, pp. 263–275.

[26] K. Chaudhuri, J. Imola, and A. Machanavajjhala, "Capacity bounded differential privacy," *CoRR*, vol. abs/1907.02159, 2019.

[27] D. Kifer and A. Machanavajjhala, "A rigorous and customizable framework for privacy," in *Proceedings of the 31st ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems (PODS)*, 2012, pp. 77–88.

[28] ——, "Pufferfish: A framework for mathematical privacy definitions," *ACM Transactions on Database Systems*, vol. 39, no. 1, pp. 3:1–3:36, 2014.

[29] N. Li, W. Qardaji, D. Su, Y. Wu, and W. Yang, "Membership privacy: a unifying framework for privacy definitions," in *Proceedings of the 2013 ACM Conference on Computer and Communications Security (CCS)*, 2013, pp. 889–900.

[30] F. Tramèr, Z. Huang, J. Hubaux, and E. Ayday, "Differential privacy with bounded priors: Reconciling utility and privacy in genome-wide association studies," in *Proceedings of the 2015 ACM Conference on Computer and Communications Security (CCS)*, 2015, pp. 1286–1297.

[31] D. Desfontaines and B. Pejó, "Sok: Differential privacies," *CoRR*, vol. abs/1906.01337, 2019.

[32] R. Bassily and A. D. Smith, "Local, private, efficient protocols for succinct histograms," in *Proceedings of the 47th ACM Symposium on Theory of Computing (STOC)*, 2015, pp. 127–135.

[33] R. Bassily, K. Nissim, U. Stemmer, and A. G. Thakurta, "Practical locally private heavy hitters," in *Advances in Neural Information Processing Systems (NIPS)*, 2017, pp. 2285–2293.

[34] T. Wang, J. Blocki, N. Li, and S. Jha, "Locally differentially private protocols for frequency estimation," in *Proceedings of the 26th USENIX Security Symposium*, 2017, pp. 729–745.

[35] J. Acharya, Z. Sun, and H. Zhang, "Hadamard response: Estimating distributions privately, efficiently, and with little communication," in *Proceedings of the 22nd International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2019, pp. 1120–1129.

[36] M. Bun, J. Nelson, and U. Stemmer, "Heavy hitters and the structure of local privacy," in *Proceedings of the 37th ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems (PODS)*, 2018, pp. 435–447.

[37] X. Xiao, G. Wang, and J. Gehrke, "Differential privacy via wavelet transforms," in *Proceedings of the 26th International Conference on Data Engineering (ICDE)*, 2010, pp. 225–236.

[38] G. S. Manku, S. Rajagopalan, and B. G. Lindsay, "Approximate medians and other quantiles in one pass and with limited memory," *ACM SIGMOD Record*, vol. 27, no. 2, pp. 426–435, 1998.