Locally Differentially Private and Consistent Frequency Estimation of Longitudinal Data

**Antonio A. Marreiras Neto, Eduardo R. Duarte Neto,
José S. Costa Filho, Javam C. Machado**

[1] Laboratório de Sistemas e Bancos de Dados (LSBD)
Departamento de Computação / UFC – Fortaleza – CE – Brazil

{anotnio.marreiras, eduardo.rodrigues,
serafim.costa, javam.machado}@lsbd.ufc.br

Abstract. *Local Differential Privacy (LDP) was developed as a Differential Privacy (DP) model that protects user data from the collector. However, tasks such as frequency estimation over time are challenging to apply LDP guarantees to, as privacy and utility goals are subjected to increasing privacy budget consumption. Utility can be enhanced through post-processing techniques, but it's important to be aware that they may introduce unintended bias. In this paper, we analyze the performance of a range of longitudinal LDP protocols coupled with various post-processing techniques, of which we determined Norm Sub and PowerNS to be the best-performing and warned against the use of Norm Mul.*

1. Introduction

Differential Privacy (DP) has come to be accepted as the *de facto* standard for data privacy. Nonetheless, as the originally proposed model of DP, central DP relies on a trusted curator [Dwork et al., 2006], which is not a reliable assumption for real-life scenarios; research in the field has recently pivoted towards pursuing more restrictive models in a local setting to bypass the need for a trusted curator in the central model [Erlingsson et al., 2014]. Said pursuits have resulted in the growing popularity of Local Differential Privacy (LDP), which aims to guarantee privacy in a local setting. In this setting, user data passes through an automatized sanitization process immediately after sampling [Team, 2017; Ding et al., 2017; Johnson et al., 2018]. Thus, once the data reaches the server, it has already been processed in a way that guarantees the user's anonymity, even in the scenario of a data leak or a malicious curator. However, as the LDP model requires noise to be added to each new user data sample, compliant protocols may add excessive amounts of noise, resulting in data that diverges significantly from the raw counterpart, and subsequent analysis may have inaccurate results. Said concerns become even more challenging when dealing with longitudinal data, as with each new query over time, the mechanism consumes the privacy budget so that anonymity can still be guaranteed, and the added noise leads to loss of data utility [Wang et al., 2021; Ren et al., 2022]. Alternatively, a larger budget can be provided at the cost of user privacy [Dwork et al., 2006].

There have been efforts to provide more flexible alternatives to achieve some form of LDP in a streaming scenario, including longitudinal LDP protocols [Erlingsson et al., 2014; Arcolezi et al., 2022a,b]. In previous research, L-LDP protocols have been applied to frequency counting, which has been used in localization and census scenarios. One difficulty in said application has been how to guarantee the consistency of the protocols' outputs, as it often requires some form of post-processing [Wang et al., 2019] that will introduce different biases in the data, resulting in varying levels of utility.

Main contribution. This paper systematically analyzes the state-of-the-art post-processing techniques applied to the leading Local Differential Privacy (LDP) protocols for longitudinal data. Our research stands out by thoroughly and meticulously demonstrating which post-processing techniques are best suited, depending on the data characteristics, protocol configurations, and protocols. This comprehensive analysis provides a clear overview of current practices and valuable guidelines for selecting optimized techniques, significantly advancing the efficient application of LDP in continuous and dynamic data environments. We implemented and evaluated six longitudinal LDP protocols in combination with ten different post-processing techniques for each of the four real datasets. We show that the post-processing methods that ensured greater utility are those that guarantee non-negative results and a sum equal to 1. However, even for these methods, when analyzing different datasets with distinct domains, we observed that the effectiveness varied significantly, as they can introduce unwarranted bias, affecting their utility.

Paper structure: The subsequent sections of this article are divided and presented in the following order: In Section 2, we present the required theoretical background for understanding the problem of interest to this paper. In Section 3, we describe our problem of interest in greater detail. In Section 4, we present basic LDP solutions that serve as building blocks for the protocols presented in Section 5, developed for longitudinal data and subjects of this paper's evaluation. In Section 6, we detail the effects of post-processing, and we list techniques to be evaluated in conjunction with the protocols presented in Section 5. In Section 7, we present the datasets and experimental setup used. In Section 8, we first present a preliminary evaluation to determine the most promising LDP protocols and post-processing methods and continue to discuss the results found through an in-depth analysis of the most promising pairings and their utility across increasing budgets. Section 9 briefly summarizes and highlights our most important and promising findings.

2. Theoretical background

2.1. Longitudinal Data

We define longitudinal data as data that evolves over time, captured through repeated sampling at increasing time intervals, which are represented as timestamps in the database. In this context, different individuals send a sample to the server at each timestamp. Therefore, the server aggregates the data where each row corresponds to all the data collected from timestamp t_0 to the current timestamp t_c for a single user.

2.2. Local Differential Privacy

Under LDP, sensitive information v from each user is encoded by a randomized algorithm Ψ, and its output $\Psi(v)$ is sent to the aggregator responsible for collecting all users' reports. Intuitively, LDP guarantees that, no matter what the value of $\Psi(v)$ is, it is approximately equally as likely to be a result of perturbing v as any other v' differing from v. Therefore, if $\Psi(v)$, instead of v, is collected, the users never reveal their private value v. The user's degree of privacy is controlled by the privacy budget ϵ. More formally,

Definition 1. *(Local Differential Privacy [Erlingsson et al., 2014]) An algorithm $\Psi(\cdot)$ satisfies ϵ-local differential privacy (ϵ-LDP), where $\epsilon \geq 0$, if and only if for any pair of inputs (v, v'), and any possible output y of Ψ, we have $Pr[\Psi(v) = y] \leq e^{\epsilon} Pr[\Psi(v') = y]$*

For any pair of distinct inputs, an LDP mechanism has the probability to output the same value limited by e^{ϵ}. In the same fashion as central DP, LDP is robust to post-processing and sequential composition[Dwork et al., 2014].

Post-Processing: post-processing is any function that receives the output of a ϵ-LDP mechanism as input, and regardless of which function it is, the output will remain ϵ-Locally Differentially Private, *i.e.,* if M is a ϵ-LDP mechanism, then $f(M)$ is also ϵ-LDP for any function f [Dwork et al., 2014].

Sequential composition: if M_t is a ϵ_t-LDP mechanism, for $t \in [\tau]$. Then, the sequence of outputs $[M_1(v), ..., M_\tau(v)]$ is $\sum_{t=1}^{\tau} \epsilon_t$-LDP. Moreover, if M is an ϵ-LDP mechanism and v is a finite sequence of k values, then the sequence $[M(v_1), ..., M(v_k)]$ of outputs is $k\epsilon$-LDP [Dwork et al., 2014].

3. Problem

We consider a setting where there are many users and one aggregator. Each user has a sequence s = $[v_1, v_2, ..., v_\tau]$ of values in a domain D, and the aggregator wants to learn the frequency distribution of values among all users for τ timestamps in a way that protects the privacy of individual users. More specifically, the aggregator wants to estimate, for each value $v \in D$, the fraction of users having v in each timestamp t *i.e.,* the number of users having v divided by the population size. We measure utility using the MSE averaged by the number of data collection τ, denoted by MSE_{avg}. Thus, for each time $t \in [1...\tau]$, we compute for each value $v \in D$ the estimated frequency $f(v)_t$ and the real one $\bar{f}(v)_t$ and calculate their differences before averaging by τ. More formally,

$$MSE_{avg} = \frac{1}{\tau} \sum_{t=1}^{\tau} \frac{1}{|D|} \sum_{v \in D} \left(\bar{f}(v)_t - f(v)_t \right)^2 \tag{1}$$

Goal. We want to understand how to leverage different post-processing techniques with LDP longitudinal data collection algorithms in order to improve utility.

4. Frequency Oracle Protocols

A *frequency oracle (FO)* protocol can be used to estimate the frequency of any value $v \in D$ under LDP, where D is the domain. A FO consists of two algorithms. The first one is Ψ, which users use locally to perturb their private data. The second one is Φ, which the aggregator uses to estimate the frequencies regarding the perturbed data received. In the literature, FOs have been employed in many different LDP tasks, including marginal release [Liu et al., 2023], answering range queries [Filho and Machado, 2023], answering queries on geospatial data [Hong et al., 2021] and identifying heavy hitters [Zhu et al., 2024].

Traditional FOs do not account for budget consumption over time when processing longitudinal data. Still, most state-of-the-art protocols designed to tackle longitudinal scenarios are adaptations of traditional ones, usually through two rounds of sanitization, a technique accomplished by sequentially composing two traditional FOs and memoization. Below, we present two traditional FOs, which serve as the basis for the ones with two rounds of sanitization that interest this paper.

4.1. Generalized Randomized Response (GRR)

Randomized Response [Warner, 1965] was introduced for binary responses, but it can easily be generalized to larger domains [Kairouz et al., 2016]. In GRR, users send their true private value $v \in D$ with probability p. Otherwise, with probability $1 - p$, the users send a randomly chosen value $v' \in D$. Formally, the algorithm is

$$\forall_{x \in D}\ Pr\big[\Psi_{GRR_{(\epsilon)}}(v) = x\big] = \begin{cases} p = \frac{e^{\epsilon}}{e^{\epsilon}+|D|-1} & \text{if x = v} \\ q = \frac{1-p}{|D|-1} = \frac{1}{e^{\epsilon}+|D|-1} & \text{if x} \neq \text{v} \end{cases}$$

GRR satisfies ϵ-LDP since $p/q = e^{\epsilon}$. From a population of n users, the aggregator receives a vector $\mathbf{x} = \big\langle x_1, x_2, ...x_n \big\rangle$ of length $|\mathbf{x}| = n$ where $x_i \in D$ is the reported value of the i-th user. Then, it estimates the frequency of $v \in D$, which consists of the ratio of users with private value v among all n users. Considering $C(n)$ as the number of times v appears in vector **x**, the unbiased [Wang et al., 2017] estimator for the frequency of $v \in D$ is $\Phi_{f(\epsilon)}(v) := (C(v)/n - q)/(p - q)$.

4.2. Unary Encoding (UE))

In Unary Encoding, a value $v \in D$ with domain size k is encoded as a length-k binary vector $B = [0,\cdots, 0, 1, 0,\cdots, 0]$ where only the v-th position is 1. The private mechanism returns a perturbed B' as

$$Pr\big[\Psi_{UE_{(\epsilon)}} B'[i] = 1\big] = \begin{cases} p, if B[i] = 1 \\ q, if B[i] = 0 \end{cases}$$

[Wang et al., 2017] show that UE satisfies ϵ-LDP for $\epsilon = ln\big(\frac{p(1-q)}{(1-p)q}\big)$. [Wang et al., 2017] define two UE protocols: Symmetric Unary Encoding (SUE) and Optimized Unary Encoding (OUE). SUE selects $p = \frac{e^{\epsilon/2}}{e^{\epsilon/2}+1}$ and $q = \frac{1}{e^{\epsilon/2}+1}$. It is symmetric since $p + q = 1$. OUE sets $p = 1/2$ and $q = \frac{1}{e^{\epsilon}+1}$. It is considered a better option than SUE.

5. FOs to longitudinal data

Traditional FOs are inadequate for longitudinal data due to increased budget consumption and decreased user privacy. Many modern solutions use *memoization*, where a value is first sanitized, memoized, and then sanitized again with a fraction of the original budget for extra protection. Popularized by RAPPOR and adapted across various protocols[Arcolezi et al., 2022a], the *2-round* memoization approach is the most accepted and will be the focus of our evaluation. However, as proven in Arcolezi et al. [2022b], most influential works, such as RAPPOR, claim to be able to guarantee LDP by making bold assumptions about the data[Erlingsson et al., 2014], which is not always realistic. That is why we will be adhering to a relaxed definition of LDP:

Definition 2. *(Longitudinal Local Differential Privacy [Arcolezi et al., 2022b]) For a longitudinal memoizing mechanism $M : A^{\tau} \rightarrow B^{\tau}$, in which $A = [1..k]$, let M^* denote a mechanism that takes as input a permutation x of A and outputs $M^*(x) := x''$ by shuffling the k entries of x, yielding x', and letting $x''_i := M^*(x'_i)$ for each $i = 1..k$, sequentially. M is said to be ϵ-LDP on the users' values iff $M^*$is ϵ-LDP.*

All FOs in this paper comply with the above L-LDP definition. In L-LDP, the sanitization parameters ϵ_{∞} and ϵ_1, the original budget and the α fraction of it, can be defined as the upper bound and the lower bound for ϵ-LDP, respectively. We have the upper bound guarantee when τ, the number of timestamps, tends to infinity, and we have the lower bound when $\tau = 1$. All the L-LDP FOs presented in this paper use the same unbiased estimator:

$$\Phi_{f_L}(v) := \frac{C(v) - nq_1(p_2 - q_2) - nq_2}{n(p_1 - q_1)(p_2 - q_2)} = \frac{\frac{C(v)/n - q_2}{p_2 - q_2} - q_1}{p_1 - q_1} \tag{2}$$

5.1. L-GRR (*Longitudinal Generalized Randomized Response*)

L-GRR is an adaption of GRR to the longitudinal scenario, adding memoization with 2-round sanitization, using the full and a downsized alpha percentage of the budget for it, which is executed by two instances of the traditional GRR protocol. The perturbation algorithm is the same as GRR for the first round:

$$\forall_{x \in D}\ Pr\big[\Psi_{L-GRR_{(\epsilon_{\infty})}}(v) = x\big] = \begin{cases} p_1 = \frac{e^{\epsilon}}{e^{\epsilon}+|D|-1} & \text{if x = v} \\ q_2 = \frac{1-p}{|D|-1} = \frac{1}{e^{\epsilon}+|D|-1} & \text{if x} \neq \text{v} \end{cases}$$

followed by a second round that outputs a report x':

$$\forall_{x' \in D}\ Pr\big[\Psi_{L-GRR_{(\epsilon_1)}}(x) = x'\big] = \begin{cases} p_2 & \text{if x' = x} \\ q_2 = \frac{1-p_2}{|D|-1} & \text{if x'} \neq \text{x} \end{cases}$$

where $p_2 = \frac{q_1 - e^{\epsilon_1}p_1}{(-p_1e^{\epsilon_1})+|D|q_1e^{\epsilon_1} - q_1e^{\epsilon_1} - p_1(|D|-1)+q_1}$ as $\epsilon_1 = ln(\frac{p_1p_2+q_1q_2}{p_1q_2+q_1p_2})$ for L-GRR.

5.2. RAPPOR and L-SUE (*Longitudinal Symmetric Unary Encoding*)

RAPPOR was a pioneer of the 2-round sanitization approach. The utility-oriented implementation of it is equivalent to the L-SUE protocol. L-SUE follows the same structure as L-GRR but uses SUE for the two rounds and consequentially requires the data to be encoded before being processed by it. The perturbation algorithm for RAPPOR and all other UE-based L-LDP FOs for the first round is

$$Pr\big[\Psi_{UE_{(\epsilon)}}B'[i] = 1] = \begin{cases} p1, if B[i] = 1 \\ q1, if B[i] = 0 \end{cases}$$

for the second round is

$$Pr\big[\Psi_{UE_{(\epsilon)}}B'[i] = 1] = \begin{cases} p2, if B[i] = 1 \\ q2, if B[i] = 0 \end{cases}$$

In L-SUE p_1 and q_1 are the same as p and q for standard SUE presented in Section 4.2, and $p_2 + q_2 = 1$. To ensure privacy for all UE algorithms [Arcolezi et al., 2022a], the following equation must be satisfied:

$$\epsilon_1 = ln(\frac{(p_1p_2 - q_2(p_1 - 1))(p_2q_1 - q_2(q_1 - 1) - 1)}{(p_2q_1 - q_2(q_1 - 1))(p_1p_2 - q_2(p_1 - 1) - 1)}) \tag{3}$$

5.3. L-OUE (*Longitudinal Optimized Unary Encoding*)

Similar to L-SUE but built on top of the OUE protocol. OUE is generally regarded as the preferred state-of-the-art solution for the traditional scenario, but L-OUE is prone to adding excessive noise [Arcolezi et al., 2022a], leading to a significant loss of utility over time. The algorithm follows the same structure, with $p_1 = p_2 = 0.5$, $q_1 = \frac{1}{e^{\epsilon}_{\infty}+1}$, and q_2 may be calculated using the Equation (3), according the definition of OUE.

5.4. L-OSUE (*Longitudinal Optimized-Symmetric Unary Encoding*)

As proven by Arcolezi et al. [2022a], it is valid to chain both UE protocols and still achieve L-LDP. L-OSUE is a hybrid solution that uses OUE for the first round and SUE for the second, thus avoiding the excessive addition of noise over time as it happens with data processed under L-OUE. The L-OSUE protocol in first round has $p_1 = 0.5$, $q_1 = \frac{1}{e^{\epsilon}_{\infty}+1}$, followed by SUE with p_2 and q_2 that satisfy $p_2 + q_2 = 1$, satisfying Equation (3).

5.5. LOLOHA

Proposed in Arcolezi et al. [2022b], LOLOHA builds on the GRR protocol and applies the technique of Local Hashing to shrink the domain size *k* to g, up to $g = 2$, leading to slower budget consumption. LOLOHA can define g as $g = 2$ (BiLOLOHA) for the strongest longitudinal LDP guarantees or compute an optimal g (OLOLOHA) value by

$$g = 1 + \max\left(1, \left\lfloor \frac{1 - a^2 + \sqrt{a^4 - 14a^2 + 12ab(1 - ab) + 12a^3b + 1}}{6(a - b)} \right\rceil\right) \tag{4}$$

As it builds on the GRR protocol, it first uses a random hash function that maps the user value to a domain of size g, and then follows the same perturbation algorithm, but with $|D| = g$ given our reduced domain size. The sanitization step outputs both the report and the hash function seed, so it can be used for counting by the aggregator. When it comes to the estimation step, it first updates the value of q_1 to $q_1 = 1/g$, and then it counts all values which the output of the hash function matches the report given the user seed and uses the same unbiased estimator (2) as other L-LDP FOs.

6. Post-processing and utility

The output of LDP FOs requires post-processing to improve the utility, ideally achieving what Wang et al. [2019] defines as consistency: all values are non-negative, and the sum of all frequencies must be 1. However, Wang et al. [2019] focused only on FOs with one round of memoization (including RAPPOR, as only the simple one-time version of it was considered), did not account for the unique properties of longitudinal data, and discussed the results for only one FO. We aim to expand the analysis of post-processing techniques

Method	Description	Non-neg	Sum to 1
Base pos	Round negative frequencies to 0	Yes	No
Base Cut	Round frequencies below k to 0. We fixed k=4 for most experiments	Yes	No
Norm Std	Normalize by adding δ	No	Yes
Norm Mul	Round negative frequencies to 0, and normalize by multiplying by ϕ	Yes	Yes
Norm Cut	Find θ, round frequencies bellow θ to 0	Yes	≈ 1
Norm Sub	Round negative frequencies to 0,normalize by adding δ	Yes	Yes
MLE Apx	Compute Apx. MLE to recover values with consistency	Yes	Yes
Power	Fit Power-Law dist. and minimize expected squared error	Yes	No
PowerNS	Execute Power, and follow with Norm Sub	Yes	Yes

Table 1. Summary of post-processing methods

to a focus on FOs developed for longitudinal data via the *2-round* memoization approach, how they can benefit from it, and discuss our findings for a greater range of protocols.

The methods' detailed definition and theoretical proof can be found in Wang et al. [2019]. Table 1 presents a summary of the post-processing methods analyzed in this work, indicating those that guarantee no negative values among the frequency distribution and those that produce an output with a sum of all values equal to 1.

7. Experimental analysis

For our experiments, four distinct datasets were used to analyze the performance of the protocols and the effects of post-processing in varying scenarios. We implemented our framework in Python 3.10. All experiments were conducted on a server with Ubuntu 20.04, Intel Core i7-7820X, and 128GB memory.

7.1. Datasets

The datasets used, their specific features, and how pre-processing was done for each are described as follows:

- GeoLife [1]: We used a sample of 100 users from the GeoLife Trajectories dataset, with 10 timestamps each. We pre-processed the locations as labels in a grid divided into cells with 1km² of area each.
- Adult [2]: We selected the attribute of hours per week and interpolated values to achieve 260 timestamps (simulating varying work hours by each user for five years)
- Loan [3]: Randomly sampling a fifth of the lending club dataset and selecting one attribute, we interpolated values to achieve 5 timestamps.
- Bfive[4]: A dataset representing personality test results. We selected an attribute and interpolated its values to achieve 20 timestamps.

As a result of pre-processing, GeoLife has a large domain size, and given a small sample size, its resulting frequency distribution can be considered sparse. Adult and Loan both have skewed frequency distributions, a more realistic scenario to apply LDP [Erlingsson et al., 2014; Wang et al., 2019]. Bfive has a small domain, a large sample size, and a smooth underlying frequency distribution.

[1] https://www.microsoft.com/en-us/download/details.aspx?id=52367

[2] http://archive.ics.uci.edu/ml/datasets/Adult

[3] https://www.kaggle.com/datasets/wordsforthewise/lending-club

[4] https://www.kaggle.com/datasets/tunguz/big-five-personality-test

7.2. Setup for experiments

We selected a privacy budget range starting in 0.5, up to 5 in incremental steps of 0.5, for a total of 10 distinct values for ϵ_∞, with $\alpha = 0.4$ for a lower bound of $\epsilon_1 = 0.4\epsilon_\infty$.

8. Results

This section presents the findings from our comprehensive analysis of post-processing methods applied to the FOs protocols. First, we decided to determine which protocols and methods are the most promising and if post-processing is a high requirement for useful results. As a means to do so, we built tables for each dataset in Table 2 that showcase the MSE between the real frequency distribution and the L-LDP output, with and without post-processing. We fixed $\epsilon_\infty = 2.5$ as the midpoint of our budget range for the sake of legibility, and so we do not differentiate between protocols that perform best under greater and lower budgets for now. Instead, we aim to first identify general trends, propose hypotheses, and determine which are the most promising FOs and post-processing techniques through a preliminary analysis, before proceeding with more in-depth evaluation.

From Table 2a, we find L-OSUE to be the best-performing FO, and *Norm Mul* as the best post-processing method. L-GRR performed poorly without post-processing, as it is unsuited to large domain sizes [Arcolezi et al., 2022a]. However, its performance gap was greatly diminished via post-processing methods that guarantee both requirements of consistency, mentioned in Section 6.

Results found in Table 2b point to L-GRR being by far the best performing FO when coupled with *Norm Mul*. Other protocols and methods present a very similar performance. Thus, we can classify the best result in this table as an outlier, which may point to unexpected behavior resulting from the interaction between the dataset features and post-processing bias.

Table 2c, despite its corresponding dataset having similar features to that of 2b, does not replicate its results. OLOLOHA was the best among FOs, and again *Norm Mul* has a lead over other methods, performing best coupled with BiLOLOHA, unexpectedly even better than with OLOLOHA. All these findings reinforce an unexpected behavior hypothesis. Lastly, Table 2d presents L-OSUE as the consistently best performing FO and *Norm Mul* with a close lead over *Norm Sub*.

In summary, it is possible to infer that the best-performing post-processing methods will output results in which all values are non-negative and sum to 1. Thus, our evaluation will not focus on the *Base Pos*, *Base Cut*, *Power*, and *Norm* methods since these methods do not meet the aforementioned guarantees, as observed in Table 1. *Base Cut* performed poorly when processing results for the bfive dataset, likely due to a positive bias when processing a dataset with a smooth distribution and small domain.

As for the FOs, L-GRR was generally the worst performing; however, it presented the best performance for Table 2b, so we will analyze it further. The performance among UE-based protocols was close at times, but the best performance was L-OSUE, as showcased by the highlighted results in Tables 2a and 2d. Therefore, we plan to discuss its results in greater detail. In most cases, OLOLOHA presented a matching or better performance than BiLOLOHA, as expected, given that it is the optimal setting of the LOLOHA

Table 2. MSE

Method	L-GRR	RAPPOR	L-OUE	L-OSUE	OLOLOHA	BiLOLOHA
None	$9.1 \cdot 10^4$	0.0265	0.0294	**0.0248**	0.0277	0.0316
Base Pos	$8.51 \cdot 10^4$	0.013	0.0155	**0.0128**	0.0141	0.0159
Base Cut	$8.51 \cdot 10^4$	0.0016	0.0028	**0.0015**	0.0018	0.0028
Norm Std	$9.1 \cdot 10^4$	0.0264	0.0293	**0.0248**	0.0277	0.0316
Norm Mul	2.5e-05	2.2e-05	2.2e-05	**2.19e-05**	2.2e-05	2.2e-05
Norm Cut	4.93e-04	2.52e-04	2.57e-04	**2.46e-04**	2.52e-04	2.55e-04
Norm Sub	1.84e-04	6.73e-05	7.31e-05	**6.68e-05**	6.75e-05	6.83e-05
MLE Apx	**4.87e-05**	9.85e-05	6.17e-05	5.98e-05	0.0001	0.0001
Power	0.999	0.999	0.999	0.999	0.999	0.999
PowerNS	2.17e-05	2.17e-05	2.17e-05	2.17e-05	2.17e-05	2.17e-05

(a) geolife dataset

Method	L-GRR	RAPPOR	L-OUE	L-OSUE	OLOLOHA	BiLOLOHA
None	0.0138	0.0035	0.0035	**0.0034**	0.0035	0.0035
Base Pos	0.0082	0.0034	0.0034	0.0034	0.0034	0.0034
Base Cut	**0.0034**	0.0036	0.0036	0.0036	0.0036	0.0036
Norm Std	0.0138	0.0035	0.0035	**0.0034**	0.0035	0.0035
Norm Mul	**7.27e-04**	0.0026	0.0026	0.0027	0.0026	0.0026
Norm Cut	0.0047	0.0034	0.0034	0.0034	0.0034	0.0034
Norm Sub	**0.0030**	0.0034	0.0034	0.0034	0.0034	0.0033
MLE Apx	**0.0031**	0.0099	0.0144	0.0144	0.0072	0.0072
Power	0.968	0.968	0.968	0.968	0.968	0.968
PowerNS	1.59e-04	1.59e-04	1.59e-04	1.59e-04	1.59e-04	1.59e-04

(b) adult dataset

Method	L-GRR	RAPPOR	L-OUE	L-OSUE	OLOLOHA	BiLOLOHA
None	0.0037	2.35e-04	2.38e-04	2.35e-04	**2.36e-04**	2.38e-04
Base Pos	0.002	2.31e-04	2.33e-04	**2.31e-04**	2.32e-04	2.33e-04
Base Cut	0.0015	0.0012	0.0012	0.0012	0.0012	0.0012
Norm Std	0.0037	2.35e-04	2.38e-04	2.38e-04	**2.36e-04**	2.38e-04
Norm Mul	9.64e-04	1.65e-04	1.65e-04	1.66e-04	1.63e-04	**1.62e-04**
Norm Cut	9.15e-04	2.26e-04	2.27e-04	2.26e-04	**2.25e-04**	2.27e-04
Norm Sub	6.82e-04	2.29e-04	2.3e-04	2.29e-04	2.29e-04	2.3e-04
MLE Apx	8.65e-04	0.0015	0.0022	0.0022	9.93e-04	**6.4e-04**
Power	0.981	0.981	0.981	0.981	0.981	0.981
PowerNS	0.0018	0.0018	0.0018	0.0018	0.0018	0.0018

(c) loan dataset

	LGRR	RAPPOR	L-OUE	L-OSUE	OLOLOHA	BILOLOHA
None	2.43e-05	1.83e-05	2.38e-05	**1.81e-05**	1.97e-05	2.18e-05
Base post	2.17e-05	1.65e-05	2.12e-05	**1.58e-05**	1.69e-05	1.86e-05
Base cut	0.032	0.032	0.032	0.032	0.032	0.032
Norm std	2.43e-05	1.62e-05	2.05e-05	**1.54e-05**	1.72e-05	1.9e-05
Norm mul	2.13e-05	1.54e-05	1.9e-05	**1.42e-05**	1.56e-05	1.68e-05
Norm cut	2.14e-05	1.61e-05	1.98e-05	**1.51e-05**	1.66e-05	1.84e-05
Norm sub	2.12e-05	1.54e-05	1.94e-05	**1.43e-05**	1.53e-05	1.69e-05
MLE Apx	6.88e-04	8.33e-04	0.001	0.001	7.59e-04	**3.31e-04**
Power	0.747	0.747	0.747	0.747	0.747	0.747
PowerNS	0.012	0.012	0.012	0.012	0.012	0.012

(d) bfive dataset

protocol [Arcolezi et al., 2022b]. There were instances of BiLOLOHA taking a lead in Tables 2c and 2d, but these were either very close or as a result of post-processing by *MLE Apx*, which at no point achieved the best results. Given these results, and with it being the best performing protocol for Table 2c, we decided to select OLOLOHA for more in-depth analysis.

Now, we measure the MSE across budgets. We can see that most protocols and post-processing techniques have the expected result of lowering MSE as the budget increases, except for *Norm Mul*, which shows unexpected and unstable behavior in experiments using the Adult and Loan datasets. Analysis of the datasets shows that the frequency distribution of the domain for both datasets is very skewed (to values above 20 for Adult, and lower than 60 for Loan), which results in many negative frequencies in the aggregator's output when the budget is small.

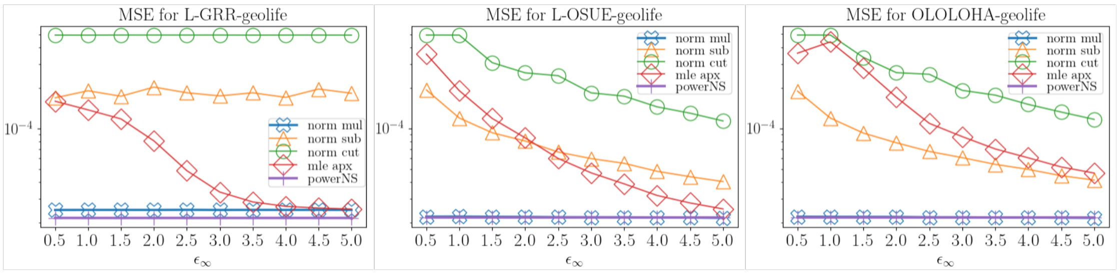

Figure 1. Geolife dataset: MSE

In GeoLife, due to a comparatively small sample for a large domain resulting in a sparse distribution, L-LDP FOs will output many negative values among frequency estimates without post-processing. From Figure 1, we can see that *Norm Mul* has the best performance, making it an outlier in comparison to results from other datasets. However, in Figures 2 and 3, we can see unexpected behavior: increasing MSE for larger budgets. This is the reverse of what is expected of L-LDP and presented by most other methods, and thus can only be caused by the bias of *Norm Mul*.

As a technique, *Norm Mul* has negative bias for high frequencies items and positive bias for low frequencies items [Wang et al., 2019], meaning that the high frequencies present in the raw FO output are lowered, and the low frequencies are increased. This leads us to the conclusion that the exceptional performance of *Norm Mul* is the result of its positive bias nullifying utility loss caused by a large number of negative values among frequency estimates. The unexpected behavior in Figures 2 and 3 can also be explained by positive bias: for smaller budgets, the skewed frequency distributions of the Adult and Loan datasets, resulting in a raw FO output with a great number of negative values, but unlike for a sparse dataset like our Geolife sample, as the budget increases the number of negative estimates decreases drastically, and *Norm Mul* positive bias for low frequencies no longer improves utility; instead it contributes to greater loss.

Budget consumption tends to increase over time in the real world, and data changes can make the dataset no longer sparse. It is likely that even in scenarios favored by *Norm Mul*, such as the results for our GeoLife sample, it may output worse and present unstable behavior as time goes on, making it inadequate for use in longitudinal data. This is concerning as *Norm Mul* follows a simple algorithm: remove negative frequencies and divide the result by the sum of all values. Therefore, developers might intuitively converge on it as a technique.

MLE Apx performed best only for the larger budgets in Figure 1. One explanation

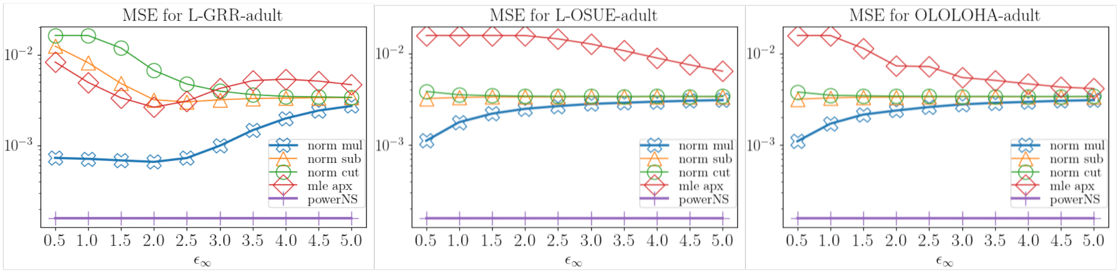

Figure 2. adult dataset: MSE

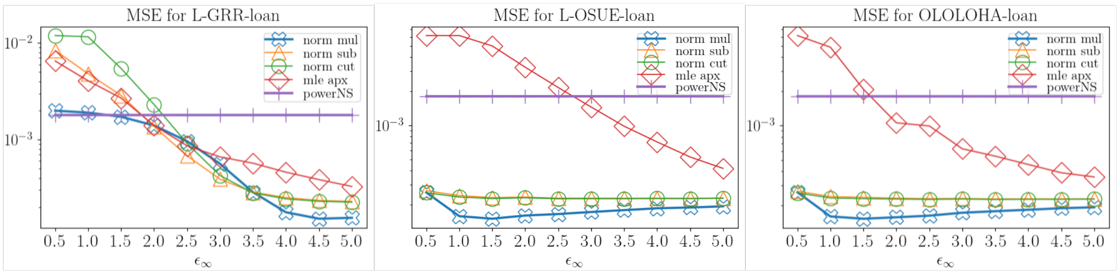

Figure 3. loan dataset: MSE

is that due to the two rounds of sanitization: the additional noise makes it harder to find a *MLE* (Maximum Likelihood Estimation) that is as close to the real distribution as in the case of a single round. It may also contribute to the unexpected behavior when coupled with L-GRR observed in Figure 2: as it follows a similar trend of increasing error for larger budgets, a case of positive bias for low frequencies interacting with negative values may be the cause, which explains the good performance observed in Figure 1 as well, even if the effect of said bias is not as strong as in the case of *Norm Mul*. L-GRR makes it easier to perceive said bias effect due to its sensibility to the domain size [Arcolezi et al., 2022a] resulting in a noisier output, as evident given its consistently lower utility of unprocessed results in comparison to other FOs as shown in Table 2. In previous works, *MLE Apx* did not show a positive bias for low frequencies, however, the additional sanitization round may be impacting the method's output. Given these findings and subpar overall performance, it is not recommended.

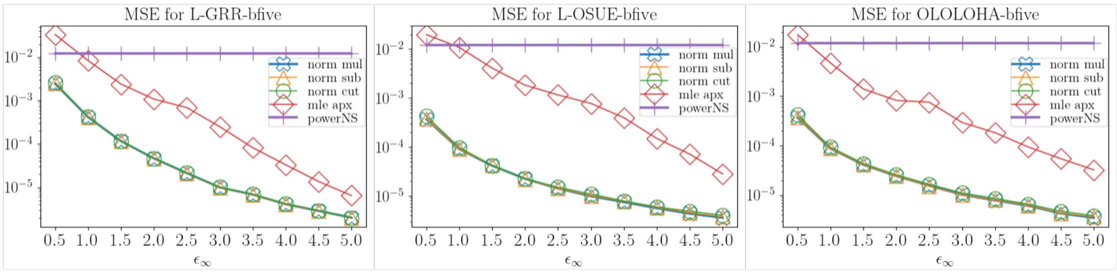

Figure 4. bfive dataset: MSE

Norm Sub and *Norm Cut* presented the most consistent results, most very close and generally among the best. However, *Norm Sub* outperformed *Norm Cut* by a large margin in Figure 1. Also, as *Norm Cut* does not always guarantee that its output will sum to 1, we regard *Norm Sub* as the best approach. We note that *Norm Sub* has the same biases as *Norm Mul*, however, their distribution is more even [Wang et al., 2019], and thus have a lesser impact on its output.

PowerNS resulted in a constant value across FOs and budgets, likely due to comparable FO performance, and (2) approximate variance not being heavily affected by the privacy budget [Arcolezi et al., 2022b]. It outperformed all other methods aside from *Norm Mul* in Figure 1, and all in Figure 2, but performed worst in Figures 3 and 4. It performed best when the MSE for the unprocessed output was larger, as in Table 2.

Among the FOs L-GRR presented the most inconsistent behavior, as the MSE stayed near constant for decreasing budgets in Figure 1 most likely the result of its sensitivity to a large domain size. L-OSUE and OLOLOHA presented comparable results, however, L-OSUE did present the most stable behavior across our experiments. This can partially be explained as a result of OLOLOHA being built on the GRR protocol with local hashing as a means to shrink the domain size, thus not being as sensitive to it as L-GRR but still more than L-OSUE. Still, there are benefits to OLOLOHA over L-OSUE such as lower budget consumption Arcolezi et al. [2022b], so only L-GRR is not recommended.

In summary, we do not recommend *Norm Mul* as its bias affects the FOs output unexpectedly when dealing with longitudinal data. *MLE Apx* would need a revised version to process data sanitized through two rounds. *Norm Cut* has no clear advantages over *Norm Sub*. We conclude that L-GRR is not ideal due to its sensitivity to large domain sizes, highlighting L-OSUE and OLOLOHA as the best-performing L-LDP FOs. Lastly, we recommend *Norm Sub* as a reliable solution for most use cases, and *PowerNS* performs best when the raw FO output has low utility.

9. Conclusion

In this paper, we conducted an exhaustive evaluation of the behavior and performance of nine post-processing techniques applied to six frequency oracle protocols focused on longitudinal data through experimentation in four real datasets. We found L-OSUE and OLOLOHA as the best FOs for longitudinal data. Our results also showed that only a few post-processing methods produced consistent and good results, namely *Norm Cut*, *Norm Sub*, and *PowerNS*. We also discovered that, among them, only the latter two have practical use cases. We warn against using *Norm Mul*, which can lead to unexpected behavior.

For future work, we aim to investigate more FOs and post-processing methods applied to specific tasks with longitudinal data, such as identifying heavy hitters, answering range queries, building graph models, and itemset mining.

ACKNOWLEDGMENTS

This work was funded by Lenovo Brasil as part of its R&D investment under Brazil's Informatics Law.

References

H. H. Arcolezi, J.-F. Couchot, B. Al Bouna, and X. Xiao. Improving the utility of locally differentially private protocols for longitudinal and multidimensional frequency estimates. *Digital Communications and Networks*, 2022a.

H. H. Arcolezi, C. Pinzón, C. Palamidessi, and S. Gambs. Frequency estimation of evolving data under local differential privacy. *arXiv preprint arXiv:2210.00262*, 2022b.

B. Ding, J. Kulkarni, and S. Yekhanin. Collecting telemetry data privately. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, page 3574–3583, Red Hook, NY, USA, 2017. Curran Associates Inc. ISBN 9781510860964.

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

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

Ú. Erlingsson, V. Pihur, and A. Korolova. Rappor: Randomized aggregatable privacy-preserving ordinal response. In *Proceedings of the 2014 ACM SIGSAC conference on computer and communications security*, pages 1054–1067, 2014.

J. S. C. Filho and J. C. Machado. Felip: A local differentially private approach to frequency estimation on multidimensional datasets. In *International Conference on Extending Database Technology*, 2023.

D. Hong, W. Jung, and K. Shim. Collecting geospatial data with local differential privacy for personalized services. In *2021 IEEE 37th International Conference on Data Engineering (ICDE)*, pages 2237–2242, 2021.

N. Johnson, J. P. Near, and D. Song. Towards practical differential privacy for sql queries. 11(5), 2018. ISSN 2150-8097.

P. Kairouz, K. Bonawitz, and D. Ramage. Discrete distribution estimation under local privacy. In *International Conference on Machine Learning*, pages 2436–2444. PMLR, 2016.

G. Liu, P. Tang, C. Hu, C. Jin, and S. Guo. Multi-dimensional data publishing with local differential privacy. In *Proceedings 26th International Conference on Extending Database Technology, EDBT 2023, Ioannina, Greece, March 28-31, 2023*, pages 183–194. OpenProceedings.org, 2023.

X. Ren, L. Shi, W. Yu, S. Yang, C. Zhao, and Z. Xu. Ldp-ids: Local differential privacy for infinite data streams. In *Proceedings of the 2022 International Conference on Management of Data*, SIGMOD '22, page 1064–1077, New York, NY, USA, 2022. Association for Computing Machinery. ISBN 9781450392495.

A. D. P. Team. Learning with privacy at scale. 2017. URL `https://machinelearning.apple.com/research/learning-with-privacy-at-scale`.

T. Wang, J. Blocki, N. Li, and S. Jha. Locally differentially private protocols for frequency estimation. In *26th USENIX Security Symposium (USENIX Security 17)*, pages 729–745, 2017.

T. Wang, M. Lopuhaä-Zwakenberg, Z. Li, B. Skoric, and N. Li. Locally differentially private frequency estimation with consistency. *arXiv preprint arXiv:1905.08320*, 2019.

T. Wang, J. Q. Chen, Z. Zhang, D. Su, Y. Cheng, Z. Li, N. Li, and S. Jha. Continuous release of data streams under both centralized and local differential privacy. In *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security*, CCS '21, New York, NY, USA, 2021. Association for Computing Machinery. ISBN 9781450384544.

S. L. Warner. Randomized response: A survey technique for eliminating evasive answer bias. *Journal of the American Statistical Association*, 60(309):63–69, 1965.

Y. Zhu, Y. Cao, Q. Xue, Q. Wu, and Y. Zhang. Heavy hitter identification over large-domain set-valued data with local differential privacy. *IEEE Transactions on Information Forensics and Security*, 19:414–426, 2024. doi: 10.1109/TIFS.2023.3324726.