Multi-Freq-LDPy: Multiple Frequency Estimation Under Local Differential Privacy in Python

Héber H. Arcolezi[1][0000−0001−8059−7094], Jean-François Couchot[2][0000−0001−6437−5598], Sébastien Gambs[3], Catuscia Palamidessi[1][0000−0003−4597−7002], and Majid Zolfaghari[1,4][0000−0002−9932−2944]

[1] Inria and École Polytechnique (IPP), Palaiseau, France
{heber.hwang-arcolezi, catuscia.palamidessi, majid.zolfaghari}@inria.fr
[2] Femto-ST Institute, Univ. Bourg. Franche-Comté, UBFC, CNRS, Belfort, France
jean-francois.couchot@univ-fcomte.fr
[3] Université du Québec à Montréal, UQAM, Montreal, Canada
gambs.sebastien@uqam.ca
[4] Sharif University of Technology, Tehran, Iran

Abstract. This paper introduces the `multi-freq-ldpy` Python package for multiple frequency estimation under Local Differential Privacy (LDP) guarantees. LDP is a gold standard for achieving local privacy with several real-world implementations by big tech companies such as Google, Apple, and Microsoft. The primary application of LDP is frequency (or histogram) estimation, in which the aggregator estimates the number of times each value has been reported. The presented package provides an easy-to-use and fast implementation of state-of-the-art solutions and LDP protocols for frequency estimation of: multiple attributes (*i.e.*, multidimensional data), multiple collections (*i.e.*, longitudinal data), and both multiple attributes/collections. Multi-Freq-LDPy is built on the well-established *Numpy* package – a *de facto* standard for scientific computing in Python – and the *Numba* package for fast execution. These features are illustrated in this demo paper with different tutorial-like case studies. This package is open-source and publicly available under an MIT license via GitHub (https://github.com/hharcolezi/multi-freq-ldpy) and can be installed via PyPi.

Keywords: Local Differential Privacy · Frequency Estimation · Multidimensional Data · Longitudinal Data · Open Source.

1 Introduction

Differential privacy (DP) [4] is a formal privacy that allows to quantify the privacy-utility trade-off originally designed for the centralized setting. In contrast, the local DP (LDP) variant satisfies DP at the user-side. One fundamental task in LDP is frequency (or histogram) estimation in which the data collector (*a.k.a.* the aggregator) decodes all the sanitized data of the users and can then estimate the number of times each value has been reported. The single frequency estimation task has received considerable attention in the literature (*e.g.*, [8,6,5]).

In particular, a state-of-the-art Python package for single frequency estimation under LDP guarantees has been proposed in [3], which covers a wide range of LDP protocols (*a.k.a.* frequency oracles) reviewed in [8].

More recently, in [1] we have investigated the frequency estimation task of multiple attributes and proposed a solution named Random Sampling Plus Fake Data (RS+FD) that outperforms the state-of-the-art solution (divide users into groups to report a single attribute) commonly reported in the literature (*e.g.*, adopted in [7]). In addition, our work in [2] optimized state-of-the-art frequency oracle protocols [5,8] for longitudinal studies (*i.e.*, multiple frequency estimation over time), which are based on the *memoization* framework from [5].

In this demo paper, we introduce Multi-Freq-LDPy[1], which is an open-source Python package providing an easy-to-use and fast implementation of state-of-the-art solutions and LDP protocols for the task of private multiple frequency estimation [1,2]. By "multiple", we mean either multidimensional data (*i.e.*, multiple attributes), longitudinal data (*i.e.*, multiple collections throughout time), or both multiple attributes/collections. The package can be installed with PyPI through "`pip install multi-freq-ldpy`", which is under an MIT license. The *multi-freq-ldpy* package is based on the standard *numpy* and *numba* libraries, as the goal is to enable an easy-to-use and fast execution toolkit. The source code, documentation, several tutorials as well as an introductory video are available at the GitHub page (https://github.com/hharcolezi/multi-freq-ldpy).

2 Presentation and Use Case Demo of multi-freq-ldpy

In terms of LDP protocols for single-frequency estimation, it currently features the best-performing[2] frequency oracle protocols presented in [8]. For the frequency estimation task of multiple attributes, three solutions are implemented from [1]. For single longitudinal frequency estimation, *multi-freq-ldpy* features all the longitudinal LDP protocols developed in [2]. Finally, for longitudinal frequency estimation of multiple attributes, the package features two multidimensional solutions from [1] featuring the longitudinal protocols from [2]. For each solution and/or protocol, there is always a *client* and an *aggregator* function, to simulate the data collection pipeline between users and the server.

For example, the following use case demonstrates how easy it is to perform single longitudinal frequency estimation with the `L_SUE` protocol [2] (*a.k.a.* Basic RAPPOR [5]) using *multi-freq-ldpy*. In this specific example, there is a single attribute $A = \{a_1, ..., a_k\}$ with domain size $k = |A|$, n users, and the privacy guarantees ϵ_{perm} (upper bound for infinity reports, *a.k.a.* ϵ_∞ in [5]) and ϵ_1 (lower bound for a single report[3]). The complete code to execute this task is illustrated in Figure 1 with the resulting estimated frequency for a given set of parameters and a randomly generated dataset.

[1] https://pypi.org/project/multi-freq-ldpy/

[2] A more complete Python package for single frequency estimation can be found in (https://pypi.org/project/pure-ldp/) [3].

[3] Naturally, one should set $\epsilon_1 < \epsilon_{perm}$.

```python
# Numpy library
import numpy as np

# Multi-Freq-LDPy functions for L-SUE protocol (a.k.a. Basic RAPPOR)
from multi_freq_ldpy.long_freq_est.L_SUE import L_SUE_Client, L_SUE_Aggregator

# Parameters for simulation
epsilon_perm = 2 # longitudinal privacy guarantee, i.e., upper bound (infinity reports)
epsilon_1 = 0.5 * epsilon_perm # single report privacy guarantee, i.e., lower bound
n = int(1e6) # number of users
k = 5 # attribute's domain size

# Simulation dataset where every user has a number between [0,k) with n users
data = np.random.randint(k, size=n)

# Simulation of client-side
l_sue_reports = [L_SUE_Client(input_data, k, epsilon_perm, epsilon_1) for input_data in data]

# Simulation of server-side aggregation
l_sue_est_freq = L_SUE_Aggregator(l_sue_reports, epsilon_perm, epsilon_1)
```

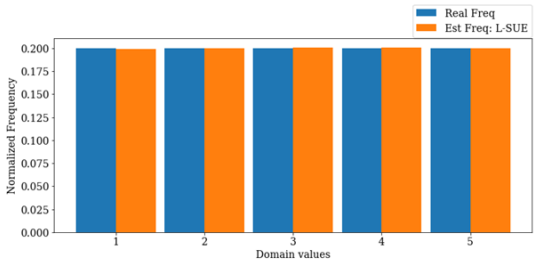

Fig. 1. Code snippet for performing single longitudinal frequency estimation with the `L_SUE` protocol and the resulting estimated frequencies for a given set of parameters.

```python
# Numpy library
import numpy as np

# Multi-Freq-LDPy functions for RS+FD solution with GRR protocol
from multi_freq_ldpy.mdim_freq_est.RSpFD_solution import RSpFD_GRR_Client, RSpFD_GRR_Aggregator

# Parameters
epsilon = 1 # privacy guarantee
n = int(1e6) # number of users
k = 5 # attribute's domain size
d = 3 # number of attributes
lst_k = [k for _ in range(d)] # list of attribute's domain size (uniformly set as k)

# Simulation dataset where every user has tuple of d values between [0-k) with n users
data = np.random.randint(k, size=(n, d))

# Simulation of client-side
rspfd_reports = [RSpFD_GRR_Client(input_tuple, lst_k, d, epsilon) for input_tuple in data]

# Simulation of server-side aggregation
rspfd_est_freq = RSpFD_GRR_Aggregator(rspfd_reports, lst_k, d, epsilon)
```

Fig. 2. Code snippet for performing frequency estimation of multiple attributes with the `RSpFD_GRR` [1] protocol.

After the import functions, we essentially need two lines of codes to simulate the data collection pipeline through applying the `L_SUE_Client` and `L_SUE_Aggregator` functions. In another example, we demonstrate how to perform frequency estimation of multiple attributes with the RS+FD solution from [1] and the GRR protocol [6] using *multi-freq-ldpy*. In this setting, a pro-

file is composed of d attributes $\mathcal{A} = \{A_1, \ldots, A_d\}$ in which each attribute A_j has a discrete domain of size $k_j = |A_j|$, for $j \in [1, d]$, n users, and the privacy parameter ϵ. The complete code to execute this task is illustrated in Figure 2.

3 Conclusion

In this demonstration paper, we have showcased the first open-source Python package named `multi-freq-ldpy` for private multiple frequency estimation under LDP guarantees. This package features separate and combined multidimensional and longitudinal data collections, *i.e.*, the frequency estimation of multiple attributes, of a single attribute throughout time, and of multiple attributes throughout time. In addition to the multiple frequency estimation task, collecting multidimensional tabular data with LDP guarantees also allows the training of machine learning classifiers/regressors on the differentially private data.

Acknowledgements

This work was partially supported by the ERC project Hypatia with grant agreement N$^{\underline{o}}$ 835294 and by the EIPHI-BFC Graduate School (contract "ANR-17-EURE-0002"). Sébastien Gambs is supported by the Canada Research Chair program as well as a Discovery Grant from NSERC.

References

1. Arcolezi, H.H., Couchot, J.F., Al Bouna, B., Xiao, X.: Random sampling plus fake data: Multidimensional frequency estimates with local differential privacy. In: Proceedings of the 30th ACM International Conference on Information & Knowledge Management. pp. 47–57 (2021). https://doi.org/10.1145/3459637.3482467
2. Arcolezi, H.H., Couchot, J.F., Bouna, B.A., Xiao, X.: Improving the utility of locally differentially private protocols for longitudinal and multidimensional frequency estimates. arXiv preprint arXiv:2111.04636 (2021)
3. Cormode, G., Maddock, S., Maple, C.: Frequency estimation under local differential privacy. Proceedings of the VLDB Endowment **14**(11), 2046–2058 (Jul 2021). https://doi.org/10.14778/3476249.3476261
4. Dwork, C., McSherry, F., Nissim, K., Smith, A.: Calibrating noise to sensitivity in private data analysis. In: Theory of Cryptography, pp. 265–284. Springer Berlin Heidelberg (2006). https://doi.org/10.1007/11681878_14
5. Erlingsson, U., Pihur, V., Korolova, A.: RAPPOR: Randomized aggregatable privacy-preserving ordinal response. In: Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security. pp. 1054–1067. ACM, New York, NY, USA (2014). https://doi.org/10.1145/2660267.2660348
6. Kairouz, P., Bonawitz, K., Ramage, D.: Discrete distribution estimation under local privacy. In: Int. Conf. on Machine Learning. pp. 2436–2444. PMLR (2016)
7. Nguyên, T.T., et al.: Collecting and analyzing data from smart device users with local differential privacy. arXiv preprint arXiv:1606.05053 (2016)
8. Wang, T., Blocki, J., Li, N., Jha, S.: Locally differentially private protocols for frequency estimation. In: 26th USENIX Security Symposium (USENIX Security 17). pp. 729–745. USENIX Association, Vancouver, BC (Aug 2017)