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

Héber Hwang Arcolezi, Sébastien Gambs, Jean-François Couchot, Catuscia Palamidessi, Majid Zolfaghari

▶ To cite this version:

Héber Hwang Arcolezi, Sébastien Gambs, Jean-François Couchot, Catuscia Palamidessi, Majid Zolfaghari. Multi-Freq-LDPy: Multiple Frequency Estimation Under Local Differential Privacy in Python. ESORICS 2022 - European Symposium on Research in Computer Security, Sep 2022, Copenhague, Denmark. pp.770 - 775, 10.1007/978-3-031-17143-7_40 . hal-03816212

HAL Id: hal-03816212
https://inria.hal.science/hal-03816212

Submitted on 15 Oct 2022

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Distributed under a Creative Commons Attribution 4.0 International License

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

Héber H. Arcolezi[1], Jean-François Couchot[2], Sébastien Gambs[3],
Catuscia Palamidessi[1], and Majid Zolfaghari[1,4]

[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: single attribute (*i.e.*, the building blocks), 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 described and illustrated in this paper with four worked examples. 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 (`https://pypi.org/project/multi-freq-ldpy/`).

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

1 Introduction

Differential privacy (DP) [7] 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) [6, 12] variant satisfies DP at the user-side, which is formalized as:

* Authors are listed by order of contribution. Version of Record (ESORICS 2022): `https://doi.org/10.1007/978-3-031-17143-7_40`.

Definition 1 (ϵ-Local Differential Privacy). *A randomized algorithm $\mathcal{M}$ satisfies ϵ-local-differential-privacy (ϵ-LDP), where $\epsilon > 0$, if for any pair of input values $v_1, v_2 \in Domain(\mathcal{M})$ and any possible output y of $\mathcal{M}$:*

$$\frac{\Pr[\mathcal{M}(v_1) = y]}{\Pr[\mathcal{M}(v_2) = y]} \leq e^{\epsilon}.$$

The privacy budget ϵ controls the privacy-utility trade-off for which lower values of ϵ result in tighter privacy protection. 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.*, [3,11,19,20,22]) as it is a building block for more complex tasks dealing with temporal and/or multidimensional aspects.

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 adopted in the literature [14,18]. In addition, our work in [2] optimized state-of-the-art LDP protocols [9,11,20] for longitudinal studies (*i.e.*, multiple frequency estimation over time), which are based on the *memoization* framework from [9].

In this paper, we introduce `multi-freq-ldpy`[1], which is the first 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. By "multiple", we mean either multidimensional data (*i.e.*, multiple attributes) [1, 14, 18], longitudinal data (*i.e.*, multiple collections throughout time) [2, 5, 9], or both multiple attributes/collections [2]. The package can be installed with PyPI using the pip command.

```
$ pip install multi-freq-ldpy
```

The `multi-freq-ldpy` package is mainly based on the standard *numpy* [17] and *numba* [13] libraries, as the goal is to enable an easy-to-use and fast execution toolkit. The source code, documentation, several (Jupyter notebook) tutorials as well as an introductory video are available at the GitHub page (`https://github.com/hharcolezi/multi-freq-ldpy`). Released under the MIT open source license, `multi-freq-ldpy` is free to use and modify, and user contributions are encouraged to help enhance the library's functionality and capabilities.

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

`Multi-freq-ldpy` is a function-based package that simulates the LDP data collection pipeline of users and the server. Thus, for each solution and/or protocol, there is always a *client* and an *aggregator* function. This section briefly presents the tasks that `multi-freq-ldpy` covers and presents four use-case of the library.

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

2.1 Main Modules (Tasks Covered)

The first task `multi-freq-ldpy` covers is **single-frequency estimation** under the `pure_frequency_oracles` module, which is a building block for the other tasks. The package currently features six[2] state-of-the-art LDP protocols, namely: Generalized Randomized Response (GRR) [11], Binary Local Hashing (BLH) [3, 20], Optimal Local Hashing (OLH) [20], Subset Selection (SS) [19, 22], Symmetric Unary Encoding[3] (SUE) [20], and Optimal Unary Encoding (OUE) [20].

Secondly, for **multidimensional frequency estimation** (*i.e.*, multiple attributes), three solutions are implemented from [1] with all aforementioned LDP protocols. These solutions, under the `mdim_freq_est` module, are: SPL) a naïve solution that splits the privacy budget ϵ over the total number of attributes; SMP) a state-of-the-art solution that randomly samples a single attribute and report it with ϵ-LDP [2, 14, 18, 20], and RS+FD) a state-of-the-art solution that randomly samples a single attribute to report with an amplified ($\epsilon' > \epsilon$)-LDP as it also generates one uniformly random fake data for each non-sampled attribute.

Thirdly, for **single longitudinal frequency estimation**, `multi-freq-ldpy` features Microsoft's *d*BitFlipPM [5] protocol and all the longitudinal LDP protocols developed in [2] based on the Google's RAPPOR [9] memoization solution (*i.e.*, two rounds of sanitization). These protocols, following the `long_freq_est` module, are: Longitudinal GRR (L-GRR) that chains GRR in both rounds and four Longitudinal Unary Encoding (L-UE) protocols that chains SUE and/or OUE in both rounds of sanitization (*i.e.*, L-SUE, L-SOUE, L-OUE, and L-OSUE). Indeed, L-SUE refers to the utility-oriented version of RAPPOR that chains SUE twice (*a.k.a.* basic RAPPOR [9]).

Finally, for **longitudinal multidimensional frequency estimation**, the package features both SPL and SMP multidimensional solutions with all the longitudinal protocols from [2] and Microsoft's *d*BitFlipPM [5], under the `long_mdim_freq_est` module.

2.2 Worked Example: Single Frequency Estimation

For example, the following use case demonstrates how easy it is to perform single frequency estimation with the `GRR` [11] protocol. In this example, there is a single attribute $A = \{a_1, ..., a_k\}$ with domain size $k = |A|$, n users, and the privacy guarantee ϵ. The complete code to execute this task is illustrated in Listing 1.1 with the resulting estimated frequency for a given set of parameters and a randomly generated dataset. One can note that after the import functions, we essentially need two lines of codes to simulate the LDP data collection pipeline through applying the `GRR_Client` and `GRR_Aggregator` functions.

[2] A more complete Python package for *single* frequency estimation can be found in (`https://pypi.org/project/pure-ldp/`) [4].

[3] Originally known as basic one-time RAPPOR [9].

```
# Multi-Freq-LDPy functions for GRR protocol
from multi_freq_ldpy.pure_frequency_oracles.GRR import GRR_Client,
    GRR_Aggregator

# NumPy library
import numpy as np

# Parameters for simulation
eps = 1 # privacy guarantee
n = int(1e6) # number of users
k = 5 # attribute’s domain size

# Simulation dataset following Uniform distribution
dataset = np.random.randint(k, size=n)

# Simulation of data collection
reports = [GRR_Client(user_data, k, eps) for user_data in dataset]

# Simulation of server-side aggregation
est_freq = GRR_Aggregator(reports, k, eps)
>>> array([0.199, 0.201, 0.199, 0.202, 0.199])
```

Listing 1.1: Code snippet for performing single frequency estimation with the `GRR` [11] protocol.

2.3 Worked Example: Longitudinal Frequency Estimation

In this second example, we demonstrate how to perform single longitudinal frequency estimation with the `L-SUE` protocol [2] (*i.e.*, RAPPOR [9]) 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 [9]) and ϵ_1 (lower bound for the first report[4]). The complete code to execute this task is illustrated in Listing 1.2 with the resulting estimated frequency for a given set of parameters and a randomly generated dataset.

2.4 Worked Example: Multidimensional Frequency Estimation

In another example, we demonstrate how to perform frequency estimation of multiple attributes with the RS+FD [1] solution and the GRR protocol [11] using `multi-freq-ldpy`. In this setting, there are n users, the privacy parameter ϵ, and each user's profile is a tuple 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 $j \in [1, d]$. The complete code to execute this task is illustrated in Listing 1.3 with the resulting estimated frequencies for a given set of parameters and a randomly generated dataset.

2.5 Worked Example: Longitudinal Multidimensional Frequency Estimation

This last example demonstrates how to perform frequency estimation of multiple attributes (*i.e.*, multidimensional data) throughout time (*i.e.*, longitudinal data)

[4] Naturally, $0 < \epsilon_1 \ll \epsilon_{perm}$ because higher values of ϵ_1 are undesirable [2, 9].

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

# NumPy library
import numpy as np

# Parameters for simulation
eps_perm = 2 # longitudinal privacy
eps_1 = 0.5 * eps_perm # first report privacy
n = int(1e6) # number of users
k = 5 # attribute's domain size

# Simulation dataset following Uniform distribution
dataset = np.random.randint(k, size=n)

# Simulation of data collection
reports = [L_SUE_Client(user_data, k, eps_perm, eps_1) for user_data in
    dataset]

# Simulation of server-side aggregation
est_freq = L_SUE_Aggregator(reports, eps_perm, eps_1)
>>> array([0.199, 0.201, 0.2, 0.198, 0.202])
```

Listing 1.2: Code snippet for performing single longitudinal frequency estimation with the `L-SUE` [2] (*i.e.*, RAPPOR [9]) protocol.

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

# NumPy library
import numpy as np

# Parameters
eps = 1 # privacy guarantee
n = int(1e6) # number of users
k = 4 # attribute's domain size
d = 3 # number of attributes
lst_k = [k for _ in range(d)] # attributes' domain size

# Simulation dataset following Uniform distribution
dataset = np.random.randint(k, size=(n, d))

# Simulation of data collection
reports = [RSpFD_GRR_Client(user_tuple, lst_k, d, eps) for user_tuple in
    dataset]

# Simulation of server-side aggregation
est_freq = RSpFD_GRR_Aggregator(reports, lst_k, d, eps)
>>> array([0.255, 0.246, 0.248, 0.251], [0.252, 0.247, 0.249, 0.252], [0.252,
     0.255, 0.244, 0.249])
```

Listing 1.3: Code snippet for performing multidimensional frequency estimation with the `RS+FD[GRR]` [1] protocol.

```python
# Multi-Freq-LDPy functions for SMP solution and L-GRR protocol
from multi_freq_ldpy.long_mdim_freq_est.L_SMP_Solution import
    SMP_L_GRR_Client, SMP_L_GRR_Aggregator

# NumPy library
import numpy as np

# Parameters for simulation
eps_perm = 2 # longitudinal privacy
eps_1 = 0.5 * eps_perm # first report privacy
n = int(1e6) # number of users
k = 4 # attribute's domain size
d = 3 # number of attributes
lst_k = [k for _ in range(d)] # attributes' domain size

# Simulation dataset following Uniform distribution
dataset = np.random.randint(k, size=(n, d))

# Simulation of data collection
reports = [SMP_L_GRR_Client(user_tuple, lst_k, d, eps_perm, eps_1) for
    user_tuple in dataset]

# Simulation of server-side aggregation
est_freq = SMP_L_GRR_Aggregator(reports, lst_k, d, eps_perm, eps_1)
>>> array([0.253, 0.255, 0.242, 0.250], [0.248, 0.253, 0.25, 0.249], [0.25,
    0.245, 0.25, 0.255])
```

Listing 1.4: Code snippet for performing longitudinal multidimensional frequency estimation with the SMP solution and `L-GRR` [2] protocol.

with the SMP solution and the `L-GRR` protocol [2] using `multi-freq-ldpy`. In this setting, there are n users, the privacy parameters ϵ_{perm} and ϵ_1, and each user's profile is a tuple 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 $j \in [1, d]$. The complete code to execute this task is illustrated in Listing 1.4 with the resulting estimated frequencies for a given set of parameters and a randomly generated dataset.

3 Conclusion

In this paper, we have showcased the first open-source Python package named `multi-freq-ldpy` for private multiple frequency estimation under LDP guarantees. More specifically, we presented the modules of Version 0.2.4 of the library, but also its easy-to-use essence, requiring two lines of code to simulate the LDP data collection pipeline. In addition to the standard single frequency estimation task, `multi-freq-ldpy` 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. As an open source project, we welcome and encourage code contributions from the community to help grow and improve the library in all of its forms. For future work, we aim to implement the LDP protocols of [8, 10, 15, 21] for longitudinal data and those of [16] for multidimensional data.

Acknowledgements

This work was supported by the European Research Council (ERC) project HYPATIA under the European Union's Horizon 2020 research and innovation programme. Grant agreement n. 835294. The work of Jean-François Couchot was supported 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. Digital Communications and Networks (2022). https://doi.org/10.1016/j.dcan.2022.07.003
3. Bassily, R., Smith, A.: Local, private, efficient protocols for succinct histograms. In: Proceedings of the forty-seventh annual ACM symposium on Theory of Computing. ACM (2015). https://doi.org/10.1145/2746539.2746632
4. Cormode, G., Maddock, S., Maple, C.: Frequency estimation under local differential privacy. Proceedings of the VLDB Endowment **14**(11), 2046–2058 (2021). https://doi.org/10.14778/3476249.3476261
5. Ding, B., Kulkarni, J., Yekhanin, S.: Collecting telemetry data privately. In: Guyon, I., Luxburg, U.V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., Garnett, R. (eds.) Advances in Neural Information Processing Systems 30, pp. 3571–3580. Curran Associates, Inc. (2017)
6. Duchi, J.C., Jordan, M.I., Wainwright, M.J.: Local privacy and statistical minimax rates. In: 2013 IEEE 54th Annual Symposium on Foundations of Computer Science. IEEE (2013). https://doi.org/10.1109/focs.2013.53
7. 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
8. Erlingsson, Ú., Feldman, V., Mironov, I., Raghunathan, A., Talwar, K., Thakurta, A.: Amplification by shuffling: From local to central differential privacy via anonymity. In: Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms. pp. 2468–2479. SIAM (2019)
9. 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
10. Joseph, M., Roth, A., Ullman, J., Waggoner, B.: Local differential privacy for evolving data. In: Bengio, S., Wallach, H., Larochelle, H., Grauman, K., Cesa-Bianchi, N., Garnett, R. (eds.) Advances in Neural Information Processing Systems. vol. 31. Curran Associates, Inc. (2018)
11. Kairouz, P., Bonawitz, K., Ramage, D.: Discrete distribution estimation under local privacy. In: Int. Conf. on Machine Learning. pp. 2436–2444. PMLR (2016)

12. Kasiviswanathan, S.P., Lee, H.K., Nissim, K., Raskhodnikova, S., Smith, A.: What can we learn privately? In: 2008 49th Annual IEEE Symposium on Foundations of Computer Science. IEEE (2008). https://doi.org/10.1109/focs.2008.27
13. Lam, S.K., Pitrou, A., Seibert, S.: Numba: A llvm-based python jit compiler. In: Proceedings of the Second Workshop on the LLVM Compiler Infrastructure in HPC. LLVM '15, Association for Computing Machinery, New York, NY, USA (2015). https://doi.org/10.1145/2833157.2833162
14. Nguyên, T.T., Xiao, X., Yang, Y., Hui, S.C., Shin, H., Shin, J.: Collecting and analyzing data from smart device users with local differential privacy. arXiv preprint arXiv:1606.05053 (2016)
15. Ohrimenko, O., Wirth, A., Wu, H.: Randomize the future: Asymptotically optimal locally private frequency estimation protocol for longitudinal data. In: Proceedings of the 41st ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems. p. 237–249. PODS '22, Association for Computing Machinery, New York, NY, USA (2022). https://doi.org/10.1145/3517804.3526226
16. Varma, G., Chauhan, R., Singh, D.: Sarve: synthetic data and local differential privacy for private frequency estimation. Cybersecurity **5**(26) (2022). https://doi.org/10.1186/s42400-022-00129-6
17. van der Walt, S., Colbert, S.C., Varoquaux, G.: The numpy array: A structure for efficient numerical computation. Computing in Science & Engineering **13**(2), 22–30 (2011). https://doi.org/10.1109/MCSE.2011.37
18. Wang, N., Xiao, X., Yang, Y., Zhao, J., Hui, S.C., Shin, H., Shin, J., Yu, G.: Collecting and analyzing multidimensional data with local differential privacy. In: 2019 IEEE 35th International Conference on Data Engineering (ICDE). IEEE (2019). https://doi.org/10.1109/icde.2019.00063
19. Wang, S., Huang, L., Wang, P., Nie, Y., Xu, H., Yang, W., Li, X.Y., Qiao, C.: Mutual information optimally local private discrete distribution estimation. arXiv preprint arXiv:1607.08025 (2016)
20. 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 (2017)
21. Xue, Q., Ye, Q., Hu, H., Zhu, Y., Wang, J.: DDRM: A continual frequency estimation mechanism with local differential privacy. IEEE Transactions on Knowledge and Data Engineering pp. 1–1 (2022). https://doi.org/10.1109/TKDE.2022.3177721
22. Ye, M., Barg, A.: Optimal schemes for discrete distribution estimation under locally differential privacy. IEEE Transactions on Information Theory **64**(8), 5662–5676 (2018). https://doi.org/10.1109/TIT.2018.2809790