Internship: Private Sparse Histograms and Implementation in Open Source Library

Team and Contact

- Inria team PreMeDICaL, based in Montpellier
- Supervisors: Aurélien Bellet (aurelien.bellet@inria.fr), Christian Janos Lebeda (christian-janos.lebeda@inria.fr), Tudor Cebere (tudor.cebere@inria.fr)

Keywords

Differential Privacy, Sparse Histogram, Open Source, Rust

Context

Differential Privacy (DP) [Dwork et al., 2006] is a rigorous mathematical framework that provides privacy guarantees for individuals in a dataset. It ensures that the outcome of any analysis is not significantly affected by any individual's data, thereby protecting personal information from being inferred. The privacy properties of any differential private mechanism follows from a mathematical proof. However, in academic papers the privacy guarantees are often shown only for the pseudocode of the proposed mechanism. This can be problematic in practice, because implementation details can negatively affect (or even break) the privacy guarantees (see [Mironov, 2012, Haney et al., 2022, Casacuberta et al., 2022] for some examples). For this reason, it is best practice to rely on library implementations when working with differential privacy.

This internship provides an opportunity to contribute to the popular OpenDP framework (`https://opendp.org`). The OpenDP core library (`https://github.com/opendp/opendp`) provides implementations of several differentially private mechanisms. The library also contains tools for combining basic mechanisms to enable more complicated data analysis. The mechanisms are implemented in Rust and can be accessed through Python and R bindings. We focus on the Rust implementation.

The OpenDP library contains an implementation of the ALP mechanism [Aumüller et al., 2022] by Christian Janos Lebeda, one of the supervisors for this internship. The ALP mechanism is used to release a sparse histogram under differential privacy. However, the support for ALP can be improved in several ways. One of the primary goals of the internship is to make a production-ready implementation of another version of the algorithm that performs better in some regimes ([Aumüller et al., 2022, Algorithm 10]). Furthermore, improving documentation and adding experiments with both synthetic and real data would help to guide users to use the right technique and parameters for their task.

Objectives

The goal of this internship is to improve the support of privately releasing sparse histograms with the OpenDP library. The intern will spend the first part of the project getting familiar with the ALP mechanism [Aumüller et al., 2022] and the relevant resources for contributing to the OpenDP library. The internship will focus on (a subset of) the following questions and tasks:

1. **Adapting integer based ALP mechanism for OpenDP**: The variant of the mechanism that is implemented in the OpenDP library was designed for real-valued data. However, the setting from the paper differs from the actual implementation, where data is discrete. The variant designed for this setting [Aumüller et al., 2022, Algorithm 10] performs better when the privacy parameter is not too low. Additionally, it is more intuitive to use, because it avoids a scaling step that is used by the original implementation for technical reasons. Implementing this variant would be a nice additonal to the library. In addition to the source code, a privacy proof in the style of the OpenDP framework should be provided.

2. **Experiments with multiple algorithms and parameters**: The error guarantees of the ALP mechanism focus on a single entry in a worst-case setting. This is not very intuitive for most users. It would be helpful to create experiments and simple examples similar to `https://docs.opendp.org/en/stable/getting-started/examples/histograms.html`. The experiments should compare multiple different techniques for releasing sparse histograms. The ALP mechanism requires us to set multiple parameters. The default values are chosen based on some simple heuristics. Are those the best parameters? Perhaps the best parameters depends on the type of data and use case.

3. **Is compact encoding practical?**: The ALP mechanism relies on a simple encoding of integers. Both the space and the decoding time relies on the encoding/decoding scheme. A recent result shows that a more complicated encoding scheme can significantly reduce the decoding time [Lolck and Pagh, 2024]. The technique has the same asymptotic error as the ALP mechanism in terms of the privacy parameter. However, it is not clear how practical this technique is. If the constants for the error is small, it would be a great improvement for the ALP mechanism. On the other hand, if the error increases by a large constant, the reduced evaluation time will not be worth the increased error.

4. **Reducing bias/supporting negative values**: The current implementation can only output a value in the range $[0, \beta]$ for some value $\beta \in \mathbb{N}$. Therefore the implementation cannot be used if the data can contain negative values. It also means that the output is biased for zero-valued entries in the histogram. The mechanism can be extended to output a value in $[-\beta, \beta]$. This task would require an updated privacy proof. This would allow for support of negative values and reduce bias.

Skills Required

- Background in Algorithms and Computer Science.
- Familiarity in Rust programming (or similar languages if you are willing to learn Rust).
- Familiarity with standard Probability Theory.
- Familiarity with Differential Privacy is a plus.

References

Martin Aumüller, Christian Janos Lebeda, and Rasmus Pagh. Representing sparse vectors with differential privacy, low error, optimal space, and fast access. *Journal of Privacy and Confidentiality*, 2022.

Sílvia Casacuberta, Michael Shoemate, Salil P. Vadhan, and Connor Wagaman. Widespread underestimation of sensitivity in differentially private libraries and how to fix it. In *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security, CCS 2022, Los Angeles, CA, USA, November 7-11, 2022*, pages 471–484. ACM, 2022. doi: 10.1145/3548606.3560708. URL https://doi.org/10.1145/3548606.3560708.

Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. In Shai Halevi and Tal Rabin, editors, *Theory of Cryptography*, pages 265–284, Berlin, Heidelberg, 2006. Springer Berlin Heidelberg. ISBN 978-3-540-32732-5.

Samuel Haney, Damien Desfontaines, Luke Hartman, Ruchit Shrestha, and Michael Hay. Precision-based attacks and interval refining: how to break, then fix, differential privacy on finite computers. *CoRR*, abs/2207.13793, 2022. doi: 10.48550/arXiv.2207.13793. URL https://doi.org/10.48550/arXiv.2207.13793.

David Rasmussen Lolck and Rasmus Pagh. Shannon meets gray: Noise-robust, low-sensitivity codes with applications in differential privacy. In David P. Woodruff, editor, *Proceedings of the 2024 ACM-SIAM Symposium on Discrete Algorithms, SODA 2024, Alexandria, VA, USA, January 7-10, 2024*, pages 1050–1066. SIAM, 2024. doi: 10.1137/1.9781611977912.40. URL https://doi.org/10.1137/1.9781611977912.40.

Ilya Mironov. On significance of the least significant bits for differential privacy. In *Proceedings of the 2012 ACM Conference on Computer and Communications Security*, CCS '12, 2012.