Differentially Private Sparse Vectors with Low Error, Optimal Space, and Fast Access

Martin Aumüller†, Christian Janos Lebeda*†, and Rasmus Pagh*‡

†IT University of Copenhagen, *BARC, ‡University of Copenhagen

IT UNIVERSITY OF COPENHAGEN

Problem formulation

Given $x \in \mathbf{R}^d$ with sparsity $\|x\|_0 \le k$ and $\|x\|_\infty \le u$, apply a mechanism $\mathscr{A}$ s.t. $\mathscr{A}(x)$ is ε-differentially private with respect to pairs of vectors $x, x' \in \mathbf{R}^d$ with $\|x - x'\|_1 \le 1$

<u>Goals</u>:

- $\mathscr{A}(x)$ stored in small-space data structure, space depending on k, d, u, and $n = \|x\|_1$
- Ability to quickly query for a value x_i
- Per-query error similar to $\mathrm{Lap}(1/\varepsilon)$

Known trade-offs*

Reference	Space in bits	Access time	Per-query error (expected, worst x)
DMNS06	$O(d \log u)$	$O(1)$	$O(1/\varepsilon)$
KKMN09**	$O(k \log (d+u))$	$O(1)$	$O(\log(1/\delta)/\varepsilon)$
CPST12	$O(k \log (d+u))$	$O(1)$	$O(\log(d)/\varepsilon)$
BV18	$\tilde{O}(n/\varepsilon \log d)$	$\tilde{O}(n/\varepsilon)$	$O(1/\varepsilon)$

* Results are explicit or follow directly from the references

** Approximate differential privacy

Main result

We introduce the *Approximate Laplace Projection* (ALP) private sparse vector representation, with these properties:

Space in bits	Access time	Per-query error
$O(k \log (d+u))$	$O(\log d)$	$O(1) \cdot \mathrm{Lap}(1/\varepsilon)$

Expected value of $\mathrm{Lap}(1/\varepsilon)$ is $O(1/\varepsilon)$

Techniques

Initially scale to values $y_i = \varepsilon x_i$

<u>Difficult case</u>: Small values, $|y_i| = O(\log d)$

<u>Idea for the case $k = 1$</u>:

- Randomly round y_i to an integer y_i'
- Use *unary* $O(\log d)$-bit representation of y_i'
- Flip each bit with probability $1/3$
- Maximum-likelihood estimator $\hat{y}_i'$ for y_i'
- Estimate for x_i is $\hat{y}_i'/\varepsilon$

<u>Extending to $k > 1$:</u>

Use *hashing* to randomly choose where to place each bit in unary representation of y_i'

Approximate DP

Previously investigated by, e.g., [KKMN09], [BNS16], [BV18], [LKSS18], [CGSS20]. In this setting, it was known how to improve the per-query error bound to $O(\log(1/\delta)/\varepsilon)$.

Our mechanism has the following properties:

Space in bits	Access time	Per-query error
$\tilde{O}(k \log (d+u))$	$O(\log (1/\delta))$	$O(1) \cdot \mathrm{Lap}(1/\varepsilon)$

Open problem

Is it possible to improve access time while not increasing space and error?

If so, is it possible to achieve these properties?

Space in bits	Access time	Per-query error
$O(k \log (d+u))$	$O(1)$	$O(1) \cdot \mathrm{Lap}(1/\varepsilon)$

References

[BNS16] Bun, Nissim & Stemmer. Simultaneous Private Learning of Multiple Concepts. ITCS 2016.

[BV18] Balcer & Vadhan. Differential Privacy on Finite Computers. ITCS 2018.

[CGSS20] Cohen, Geri, Sarlos & Stemmer. Differentially Private Weighted Sampling. AISTATS 2021.

[CPST12] Cormode, Procopiuc, Srivastava & Tran. Differentially Private Summaries for Sparse Data. ICDT 2012.

[DMNS06] Dwork, McSherry, Nissim & Smith. Calibrating Noise to Sensitivity in Private Data Analysis. TCC 2006.

[KKMN09] Korolova, Kenthapadi, Mishra & Ntoulas. Releasing Search Queries and Clicks Privately. WWW 2009.

[LKSS18] Li, Karwa, Slavkovic & Steorts. A Privacy Preserving Algorithm to Release Sparse High-dimensional Histograms. J. Priv. Conf. 2018.

Research supported by the VILLUM foundation grant number 16582.

mauu@itu.dk chle@itu.dk pagh@di.ku.dk