Snapping Mechanism and Problems of Finite Precision

Christian Covington

September 30, 2019

Harvard University Privacy Tools Project

Overview

Problem Statement

IEEE 754 Floating Point

Issues with implementing the Laplace Mechanism

Inadequate Fixes

Snapping Mechanism

Implementation Considerations

Problem Statement

What is Differential Privacy and how do we achieve it?

Let $M : \mathcal{X}^n \to \mathcal{R}$ be a randomized algorithm, D and D' be neighboring datasets (differing in one row), and $S \subseteq \mathcal{R}$. Then M satisfies $(\epsilon, 0)$ differential privacy if

$$\mathbb{P}(M(D) \in S) \leq \exp(\epsilon) \cdot \mathbb{P}(M(D') \in S) \text{ [DMNS06]}$$

One way to construct such a randomized algorithm is to add noise to the function we want to compute. We will focus on the Laplace Mechanism, which satisfies $(\epsilon, 0)$ differential privacy:

$$M_{Lap}(\mathcal{D}, f, \epsilon) = f(\mathcal{D}) + Lap\left(\frac{\Delta f}{\epsilon}\right) \text{ [DMNS06]}$$

where $f : \mathcal{D} \to \mathbb{R}$.

For $(\epsilon, 0)$-DP, it is necessary (but not sufficient) that $\operatorname{supp}(M_{Lap}(D, f, \epsilon)) = \operatorname{supp}(M_{Lap}(D', f, \epsilon))$.

Moving from Theory to Practice

Let N be a stand-in for any type of noise we might want to add to produce a randomized algorithm. When $\text{supp}(N) = \mathbb{R}$, the supports of mechanism outputs on neighboring datasets are equivalent. This is not necessarily true when $\text{supp}(N) \neq \mathbb{R}$.[1]

Any software implementation of DP algorithms with necessarily have only finite precision, so $\text{supp}(N) \neq \mathbb{R}$. In the interest of concreteness, we will consider the IEEE-754 double-precision (binary64) floating point format.

[1] E.g. let $f(D) = 0, f(D) = \frac{1}{2}$, and $\text{supp}(N) = \mathbb{Z}$.

IEEE 754 Floating Point

IEEE 754 Floating Point

The IEEE 754 standard (referred to as *double* or *binary64*) floating point number has 3 components:
sign: 1 bit
significand/mantissa: 53 bits (only 52 are explicitly stored)
exponent: 11 bits

Let S be the sign bit, $m_1 \dots m_{52}$ be the bits of the mantissa, and $e_1 \dots e_{11}$ be the bits of the exponent. Then a double is represented as

$$(-1)^S (1.m_1 \dots m_{52})_2 \times 2^{(e_1 \dots e_{11})_2 - 1023}$$

Note that doubles ($\mathbb{D}$) are not uniformly distributed over their range, so arithmetic precision is not constant across $\mathbb{D}$.

Issues with implementing the Laplace Mechanism

Generating the Laplace: Overview

The most common method of generating Laplace noise is to use inverse transform sampling. Let Y be the random variable representing our Laplace noise with scale parameter λ. Then,

$$Y \leftarrow F^{-1}(U) = -\lambda ln(1 - U)$$

where F^{-1} is the inverse cdf of the Laplace and $U \sim Unif(0, 1)$.

Sampling from Uniform

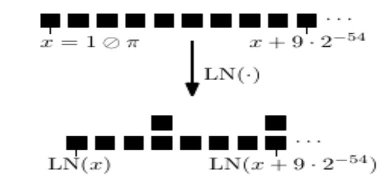
Sampling from $\mathbb{D} \cap (0, 1)$ is not particularly well-defined or consistent across implementations. Typically, the output of a uniform random sample is confined to a small subset of possible elements of $\mathbb{D}$. [Mir12]

Reference and Library	Uniform from $[0, 1)$
Knuth [Knu97]	multiples of 2^{-53}
"Numerical Recipes" [PTVF07]	multiples of 2^{-64}
C#	multiples of $1/(2^{31} - 1)$
SSJ (Java) [L'E]	multiples of 2^{-32} or 2^{-53}
Python	multiples of 2^{-53}
OCaml	multiples of 2^{-90}

Figure 1: Uniform random number generation [Mir12]

Natural Logarithm

When implemented on uniform random numbers as normally generated, the natural log produces some values repeatedly and skips over others entirely. [Mir12]



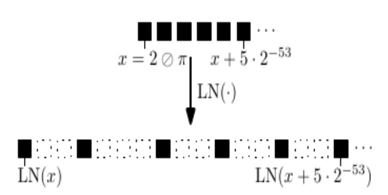


Figure 2: Artefacts of natural logarithm on $\mathbb{D}$ [Mir12]

Attack

Imagine we want to release a private version of the output of a function f with $\Delta f = 1$ and $\epsilon = \frac{1}{3}$. Let $f(D) = 0, f(D') = 1$.

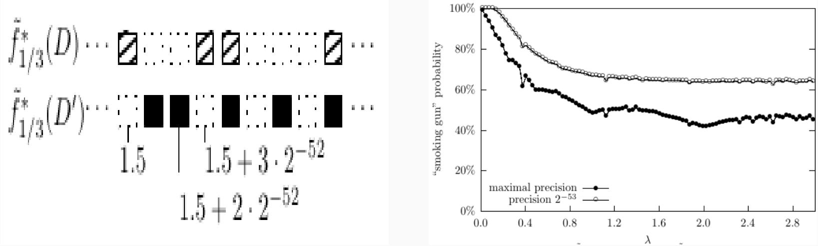


Figure 3: Attack on Laplace Mechanism [Mir12]

Inadequate Fixes

Rounding Noise

Gaps happen at very fine precision, so can we round the noise to be less precise? Consider rounding noise to the nearest integer multiple of 2^{-32}. Then, if $|f(D) - f(D')| < 2^{-32}$, then the supports of the mechanism outputs under the two data sets are completely disjoint.

Smoothing Noise

Can we smooth the noise and ensure that all possible doubles are in the support of the mechanism? Imagine $f(D) = 0, f(D') = 1$ and we are adding $Lap(1)$ noise, with y a sample from $Lap(1)$. Consider the case when our mechanism output $x \in (0, \frac{1}{2})$.

If the private data set is D, then $x = f(D) \oplus y$ will have unit of least precision (ulp) $< 2^{-53}$ because $y \in (0, \frac{1}{2})$. If the private data set is D', then $x = f(D') \oplus y$ will have ulp $= 2^{-53}$ because $y \in (-1, -\frac{1}{2})$.

So, conditional on $x \in (0, \frac{1}{2})$, the support of $f(D') + y$ is a proper subset of the support of $f(D) + y$.

Snapping Mechanism

Generating Uniform Random Numbers

Our goal is to sample from $\mathbb{D} \cap (0, 1)$ while maintaining the properties of $\mathbb{R}$ as closely as possible.

IEEE 754 floating point numbers are of the form

$$(-1)^S(1.m_1 \ldots m_{52})_2 \times 2^{-E}$$

Let $S = 0$, $E \sim Geom(0.5)$, and $m_1, \ldots, m_{52} \sim Bern(0.5)$. This means that every $d \in \mathbb{D} \cap (0, 1)$ has a chance of being represented, and are represented proportional to their unit of least precision.

Mechanism Statement

The Snapping Mechanism [Mir12] is defined as follows:

$$\tilde{f}(D) \triangleq clamp_B\left(\lfloor clamp_B(f(D)) \oplus S \otimes \lambda \otimes LN(U^*)\rceil_\Lambda\right)$$

where $clamp_B$ restricts output to the range $[-B, B]$, $S \otimes \lambda \otimes LN(U^*)$ is Laplace noise generated with our improved random number generator, and $\lfloor\cdot\rceil_\Lambda$ rounds to the nearest Λ, where Λ is the smallest power of two at least as large as λ.

The mechanism guarantees $\left(\frac{1+12B\eta+2\eta\lambda}{\lambda}, 0\right)$-DP, where η is machine epsilon.

Implementation Considerations

Why the lack of implementation?

Not actually sure, but probably some combination of:

- No utility/error bounds in the paper
- Not immediately clear how to properly implement suggested uniform random number generation
- Technical differences from other mechanisms
    - Privacy guarantee is a function of ϵ
    - Non-private function estimate is an input to the mechanism
- Generally seen as low-order concern

[DMNS06] 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.

[Mir12] 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, pages 650–661, New York, NY, USA, 2012. ACM.