Harvard Privacy Tools Project

Geometric Mechanism Notes

Christian Covington

January 29, 2020

1 Overview

This document is a write-up of extra notes regarding implementations of the Geometric mechanism in yarrow.

2 Simple Geometric Mechanism

2.1 Background

The *Simple Geometric Mechanism* is an implementation of the Geometric mechanism proposed in [GRS12]. For a counting query f with true value $f(d)$ and parameter value $\alpha \in (0,1)$, the $\alpha-$geometric mechanism outputs $f(d) + N$, where $\operatorname{supp} N = \mathbb{Z}$ and

$$\Pr(N = n) = \frac{1-\alpha}{1+\alpha}\alpha^{|n|}. \tag{2.1}$$

This mechanism respects pure differential privacy, with a privacy loss parameter of α. To accommodate privacy parameters outside of $(0,1)$, we can choose a privacy parameter $\epsilon > 0$ and let $\alpha = e^{-\epsilon}$.

2.2 Approximate Implementation

Below is pseudocode that is pretty close to our implementation of the mechanism (more on the finer points later). This directly matches the α-Geometric mechanism from [GRS12].

Algorithm 1 (Almost) Simple Geometric Mechanism $M_{SG}(f(D), \epsilon)$

```
1: Let $f(D)$ be the count query we wish to privatize, $\epsilon$ be our privacy parameter and
   $\alpha = e^{-\epsilon}$.
2: $u \leftarrow \text{Unif}(0,1)$
3: if $u < \frac{1-\alpha}{1+\alpha}$ then
4:     return $f(D)$
5: else
6:     $s \leftarrow$ uniformly random draw from $\{-1, 1\}$
7:     $g \leftarrow \text{Geom}(1-\alpha)$ where $g \in \{1, 2, \ldots\}$
8:     return $f(D) + s \cdot g$
9: end if
```

2.2.1 Proof that Algorithm 1 is equivalent to Equation (2.1)

First, note from Equation (2.1) that the mechanism returns $f(d)$ when $N = 0$, which happens with probability $\frac{1-\alpha}{1+\alpha}$. This is reflected in line 3 of Algorithm 1.

Now, we have handled the case where $N = 0$, so let's manipulate Equation (2.1) a bit more. For arbitrary $n \in \mathbb{Z} \setminus \{0\}$:

$$\begin{aligned}
\Pr(N = n | N \neq 0) &= \left(\frac{1}{1 - \Pr(N=0)}\right) \cdot \left(\frac{1-\alpha}{1+\alpha}\alpha^{|n|}\right) \\
&= \left(\frac{1}{1 - \frac{1-\alpha}{1+\alpha}}\right) \cdot \left(\frac{1-\alpha}{1+\alpha}\alpha^{|n|}\right) \\
&= \left(\frac{1+\alpha}{2\alpha}\right) \cdot \left(\frac{1-\alpha}{1+\alpha}\alpha^{|n|}\right) \\
&= \frac{1-\alpha}{2\alpha}\alpha^{|n|}.
\end{aligned}$$

We know that the noise induced by the geometric mechanism should be symmetric, so let's now consider only $n \in \mathbb{Z}^+$, with the knowledge that $\Pr(N = n) = \Pr(N = -n)$. This allows us to remove the factor of 2 from the denominator and remove the absolute value around n:

$$\begin{aligned}
\Pr(N = n | N \neq 0) &= \frac{1-\alpha}{\alpha}\alpha^{n} \\
&= \alpha^{n-1} \cdot (1-\alpha) \\
&= (1-p)^{n-1} p \text{ [for } p = (1-\alpha)].
\end{aligned}$$

Notice that this last statement is exactly the PDF of a $\text{Geom}(p)$ defined on $\{1, 2, \ldots\}$ where $p = 1 - \alpha$. Thus, the combination of lines 6 and 7 from Algorithm 1 is sufficient to generate the modified distribution from Equation (2.1) where we condition on $N \neq 0$.

2.2.2 Actual Implementation

To this point, we have assumed that the noise generated from our mechanism has support $\mathbb{Z}$ (we will call this the untruncated mechanism). This presents two major problems.

First, recall that we need to sample from a Geometric distribution within our mechanism. We do not want to use inverse transform sampling to do this, as doing so requires manipulation of floating-point numbers that can lead to privacy violations.[1] Therefore, we induce a Geometric distribution by randomly sampling bits until we see a 1. If the distribution from which we are sampling has support $\mathbb{Z}$, we could hypothetically sample an arbitrarily large number of flips and not see a 1. We would like some kind of guarantee on the number of samples we need.

Perhaps more importantly, the untruncated mechanism will sometimes yield nonsensical answers. For a data set with known sample size n, the only reasonable answers to a counting query are $\mathcal{S} = \{0, 1, 2, \ldots, n\}$.

If the untruncated mechanism were to return a value outside of $\mathcal{S}$, then we can clip the value so that it is back in the set. This does not violate our privacy guarantee, as it is considered data-independent post-processing, and can only help us in terms of absolute error. We will refer to this as the *Truncated Geometric Mechanism*, as introduced in [GRS12]. We also consider what truncating the eventual mechanism output means for how we need to sample from the Geometric distribution.

Let's return to Algorithm 1, but include truncation so that it actually reflects the algorithm in yarrow.

Algorithm 2 Simple Geometric Mechanism $M_{SG}(f(D), \epsilon,$ count_min, count_max, EFC)

```
 1: Let $f(D)$ be the count query we wish to privatize, $\epsilon$ be our privacy parameter,
    count_min be the minimum possible count (likely 0), count_max be the maximum possible
    count (probably $n$), and EFC (enforce_constant_time) be a boolean for whether or not
    we want to enforce our geometric sampling to always take the same number of steps.
 2: Let $\alpha = e^{-\epsilon}$.
 3: $u \leftarrow \text{Unif}(0, 1)$
 4: if $u < \frac{1-\alpha}{1+\alpha}$ then
 5:     return $f(D)$
 6: else
 7:     $s \leftarrow$ uniformly random draw from $\{-1, 1\}$
 8:     $g \leftarrow \text{Geom}_{Trunc}(1 - \alpha)$ where $g \in \{1, 2, \ldots, \text{count\_max} - \text{count\_min}\}$
 9:     return $\max\Big(\text{count\_min}, \min\big(f(D) + s \cdot g, \text{count\_max}\big)\Big)$
10: end if
```

You can see in line 9 of Algorithm 2 where we clip the final output to the set

$$\mathcal{S} = \{\text{count\_min}, \text{count\_min} + 1, \ldots, \text{count\_max}\},$$

again keeping in mind that, in general, count_min and count_max will be 0 and n, respectively.

Our final step is to define Geom_{Trunc}. We know that our raw count $f(D)$ must be between count_min and count_max and that our mechanism will eventually return a result between those same bounds. Therefore, the absolute maximum noise we could ever add is $r = \text{count\_max} - \text{count\_min}$; any more would always put us outside of the set $\mathcal{S}$ and eventually

[1] See [Mir12] and [Ilv19] for examples. [BV17] is the only place I have seen the known problems with floating-point numbers extended to their effects on inverse transform sampling.

be clipped back into the set. Therefore, we can sample at most r bits when generating the draw from the Geometric distribution. If we have sampled r bits and not seen a 1, then we can set $g = r$ without affecting our eventual result.

References

[BV17] Victor Balcer and Salil Vadhan. Differential privacy on finite computers. *arXiv preprint arXiv:1709.05396*, 2017.

[GRS12] Arpita Ghosh, Tim Roughgarden, and Mukund Sundararajan. Universally utility-maximizing privacy mechanisms. *SIAM Journal on Computing*, 41(6):1673–1693, 2012.

[Ilv19] Christina Ilvento. Implementing the exponential mechanism with base-2 differential privacy, 2019.

[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*, pages 650–661, 2012.