

# Space-Efficient Private Estimation of Quantiles

Massimo Cafaro<sup>1,\*</sup>, Angelo Coluccia<sup>1,†</sup>, Italo Epicoco<sup>1,†</sup> and Marco Pulimeno<sup>1,†</sup>

<sup>1</sup>Department of Engineering for Innovation, University of Salento, Lecce, 73100 Italy

## Abstract

Online estimation of robust statistics, namely quantiles, is of great interest in several applications where high-rate data streams must be processed as quickly as possible and discarded, being their storage usually unfeasible. Fast and accurate estimation is challenging when considering the additional constraint of differential privacy, which leads to the well-known privacy-utility trade-off. Recent approaches further require the use of a minimal amount of space (even a single memory variable), so as to reduce the complexity. In this paper we present three differentially-private streaming algorithms for frugal estimation of a quantile, based on different modifications of the Frugal-1U algorithm: DP-FRUGAL-1U-L, DP-FRUGAL-1U-G, and DP-FRUGAL-1U- $\rho$ . We specifically provide a theoretical analysis and experimental results.

## Keywords

Quantiles, Streaming, Differential Privacy, Frugal Algorithms

## 1. Introduction

Differential privacy (DP) is an important topic among various research communities, including computer science, communications, data and signal processing [1, 2, 3]. Processing of Big Data streams can indeed include a large amount of personal and sensitive information, which need to be protected. At the same time, utility must be preserved, so that the processing can extract useful information for the application at hand. Finding the best algorithm in this trade-off is still an open problem, which is further complicated by the compelling requirement to reduce the computational complexity and memory requirements.

Various DP algorithms for the estimation of mean values or other ensemble statistics under different settings have been proposed, e.g., [4, 5], see also the survey [6]. Among them, a recent trend is the adoption of computational approaches that require only a few memory variables or just a single one, termed “frugal”. In this paper, we are particularly concerned with the online estimation of quantiles under DP. The streaming setting adds additional constraints, since stream items may arrive at a very high rate and must be processed as quickly as possible and discarded [7]. More in details, our aim is to obtain DP quantile estimation algorithms able to cope with data heterogeneity, including the presence of outliers due to heavy-tails or random effects with heterogeneous variance, for which robust tools not requiring knowledge of the data distribution are needed [8, and references therein]. Such data are found in many contexts, including finance [9], Internet [10], database query optimizers, data splitting for parallel computation in database management systems, etc.

Even though recent work has provided DP algorithms for mean values [4, 5], to the best of our knowledge no DP algorithm is available in the literature for quantile estimation via frugal computation. We base our work on the FRUGAL-1U algorithm [11], discussed in Section 3, and present preliminary results. Overall, we provide the following original contributions, without assuming knowledge of the data distribution:

- we analyze the FRUGAL-1U algorithm and prove that its global sensitivity is bounded and equal to 2;

ITADATA2025: The 4<sup>th</sup> Italian Conference on Big Data and Data Science, September 9–11, 2025, Turin, Italy

\*Corresponding author.

<sup>†</sup>These authors contributed equally.

✉ massimo.cafaro@unisalento.it (M. Cafaro); angelo.coluccia@unisalento.it (A. Coluccia); italo.epicoco@unisalento.it (I. Epicoco); marco.pulimeno@unisalento.it (M. Pulimeno)

ORCID 0000-0003-1118-7109 (M. Cafaro); 0000-0001-7118-9734 (A. Coluccia); 0000-0002-6408-1335 (I. Epicoco); 0000-0002-4201-1504 (M. Pulimeno)



© 2025 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

next, we design three DP versions of the algorithm based respectively on the Laplace mechanism, the Gaussian mechanism and on  $\rho$  zero-concentrated DP;

- we validate the theoretical results through simulations, considering different families of statistical distributions, including heavy-tailed ones.

The rest of this paper is organized as follows. Section 2 provides the necessary definitions and notation used throughout the manuscript. Section 3 introduces the FRUGAL-1U algorithm whilst Section 4 presents our analysis and three corresponding DP algorithms. We present the experimental results in section 5 and draw our conclusions in Section 6.

## 2. Preliminary Definitions and Notation

In this Section, we briefly recall the definitions and notations that shall be used throughout this paper. We begin by giving a formal definition of quantiles.

**Definition 1.** (Lower and upper  $q$ -quantile) Given a set  $A$  of size  $n$  over  $\mathbb{R}$ , let  $R(x)$  be the rank of the element  $x$ , i.e., the number of elements in  $A$  smaller than or equal to  $x$ . Then, the lower (respectively upper)  $q$ -quantile item  $x_q \in A$  is the item  $x$  whose rank  $R(x)$  in the sorted set  $A$  is  $\lfloor 1 + q(n - 1) \rfloor$  (respectively  $\lceil 1 + q(n - 1) \rceil$ ) for  $0 \leq q \leq 1$ .

The accuracy related to the estimation of a quantile can be defined either as rank or relative accuracy. In this paper, we deal with algorithms that provide rank accuracy, which is defined as follows.

**Definition 2.** (Rank accuracy) For all items  $v$  and a given tolerance  $\epsilon$ , return an estimated rank  $\tilde{R}$  such that  $|\tilde{R}(v) - R(v)| \leq \epsilon n$ .

Next, we introduce the main concepts underlying DP. We focus on the so-called central model of DP. Actually, two definitions are possible, as follows.

**Definition 3.** (Unbounded differential privacy, also known as the add-remove model [12] [13]) Two datasets  $x$  and  $x'$  are considered neighbors if  $x'$  can be obtained from  $x$  by adding or removing one row. Under unbounded DP, the sizes of  $x$  and  $x'$  are different (by one row):  $|x \setminus x'| + |x' \setminus x| = 1$ .

**Definition 4.** (Bounded differential privacy, also known as the swap or the update/replace model [12] [14]) Two datasets  $x$  and  $x'$  are considered neighbors if  $x'$  can be obtained from  $x$  by changing one row. Under bounded DP, the sizes of  $x$  and  $x'$  are equal:  $|x \setminus x'| = 1$  and  $|x' \setminus x| = 1$ .

In this paper, we adopt bounded DP. Next, we define  $\epsilon$ -DP.

**Definition 5.** ( $\epsilon$ -differential privacy) A function which satisfies DP is called a mechanism; we say that a mechanism  $\mathcal{F}$  satisfies pure DP if for all neighboring datasets  $x$  and  $x'$  and all possible sets of outputs  $\mathcal{S}$ , it holds that  $\frac{\Pr[\mathcal{F}(x) \in \mathcal{S}]}{\Pr[\mathcal{F}(x') \in \mathcal{S}]} \leq e^\epsilon$ . The  $\epsilon$  parameter in the definition is called the privacy parameter or privacy budget.

The  $\epsilon$  parameter is strictly related to the desired amount of privacy. In practice, there is trade-off going on, since smaller values of this parameter provide higher privacy but at the cost of less utility and vice-versa. In this context, utility refers to the possibility of using the obtained results for further investigations, namely statistical analyses. Therefore, the trade-off may be understood considering that setting  $\epsilon$  to a small value require the mechanism  $\mathcal{F}$  to provide very similar outputs when instantiated on similar inputs (so, higher privacy, obtained by injecting more noise which in turn undermines utility); on the contrary, a large value provides less similarity of the outputs (so, less privacy but increased utility). Besides pure DP, a different notion, called approximate (or, alternatively,  $(\epsilon, \delta)$  DP, is also available.

**Definition 6.** ( $(\epsilon, \delta)$ -differential privacy) A mechanism  $\mathcal{F}$  satisfies  $(\epsilon, \delta)$ -DP if for all neighboring datasets  $x$  and  $x'$  and all possible sets of outputs  $\mathcal{S}$ , it holds that  $\Pr[\mathcal{F}(x) \in \mathcal{S}] \leq e^\epsilon \Pr[\mathcal{F}(x') \in \mathcal{S}] + \delta$ , where the privacy parameter  $0 \leq \delta < 1$  represents a failure probability.

The definition implies that (i) with probability  $1 - \delta$  it holds that  $\frac{\Pr[\mathcal{F}(x) \in \mathcal{S}]}{\Pr[\mathcal{F}(x') \in \mathcal{S}]} \leq e^\epsilon$  and (ii) with probability  $\delta$  no guarantee holds. As a consequence,  $\delta$  is required to be very small.

In order to define a mechanism, we need to introduce the notion of sensitivity. In practice, the sensitivity of a function reflects the amount the function's output will change when its input changes. Formally, given the universe of datasets, denoted by  $\mathcal{D}$ , the sensitivity of a function  $f$ , called *global sensitivity*, is defined as follows.

**Definition 7.** (*Global sensitivity*) Given a function  $f : \mathcal{D} \rightarrow \mathbb{R}$  mapping a dataset in  $\mathcal{D}$  to a real number, the global sensitivity of  $f$  is  $GS(f) = \max_{x, x' : d(x, x') \leq 1} |f(x) - f(x')|$ , where  $d(x, x')$  represents the distance between two datasets  $x, x'$ .

We now define two mechanisms, respectively the Laplace and the Gaussian mechanism. The former must be used with pure DP, the latter with approximate DP.

**Definition 8.** (*Laplace mechanism*) Given a function  $f : \mathcal{D} \rightarrow \mathbb{R}$  mapping a dataset in  $\mathcal{D}$  to a real number,  $\mathcal{F}(x) = f(x) + \text{Lap}\left(\frac{s}{\epsilon}\right)$  satisfies  $\epsilon$ -DP.  $\text{Lap}(S)$  denotes sampling from the Laplace distribution with center 0 and scale  $S$ , whilst  $s$  is the sensitivity of  $f$ .

**Definition 9.** (*Gaussian mechanism*) Given a function  $f : \mathcal{D} \rightarrow \mathbb{R}$  mapping a dataset in  $\mathcal{D}$  to a real number,  $\mathcal{F}(x) = f(x) + \mathcal{N}(\sigma^2)$  satisfies  $(\epsilon, \delta)$ -DP, where  $\sigma^2 = \frac{2s^2 \ln(1.25/\delta)}{\epsilon^2}$  and  $s$  is the sensitivity of  $f$ .  $\mathcal{N}(\sigma^2)$  denotes sampling from the Gaussian (normal) distribution with center 0 and variance  $\sigma^2$ .

The Gaussian mechanism also satisfies a stronger notion of privacy, known as  $\rho$  zero-concentrated differential privacy ( $\rho$ -zCDP); its definition uses a single privacy parameter  $\rho$ , and lies between pure and approximate DP. Moreover,  $\rho$ -zCDP has been shown to be equivalent (i.e., it can be translated) to standard notions of privacy.

**Definition 10.** ( $\rho$ -zCDP) A mechanism  $\mathcal{F}$  satisfies zero-concentrated DP if for all neighboring datasets  $x$  and  $x'$  and all  $\alpha \in (1, \infty)$ , it holds that  $D_\alpha(\mathcal{F}(x) \parallel \mathcal{F}(x')) \leq \rho\alpha$ , where  $D_\alpha(P \parallel Q) = \frac{1}{\alpha-1} \ln E_{x \sim Q} \left( \frac{P(x)}{Q(x)} \right)^\alpha$  is the Rényi divergence.

It can be shown that  $\rho$ -zCDP can be converted to  $(\epsilon, \delta)$ -DP as follows. If the mechanism  $\mathcal{F}$  satisfies  $\rho$ -zCDP, then for  $\delta > 0$  it also satisfies  $(\epsilon, \delta)$ -differential privacy for  $\epsilon = \rho + 2\sqrt{\rho \log(1/\delta)}$ . Moreover the Gaussian mechanism can be adapted to work with  $\rho$ -zCDP as follows.

**Definition 11.** ( $\rho$ -zCDP Gaussian mechanism) Given a function  $f : \mathcal{D} \rightarrow \mathbb{R}$  mapping a dataset in  $\mathcal{D}$  to a real number,  $\mathcal{F}(x) = f(x) + \mathcal{N}(\sigma^2)$  where  $\sigma^2 = \frac{s^2}{2\rho}$  satisfies  $\rho$ -zCDP, where  $s$  is the sensitivity of  $f$ .

We briefly introduce the concept of utility, which quantifies how much a DP result is useful for a subsequent data analysis. Therefore, the analysis to be performed plays a key role here, since DP results affected by a significant error may or may not be useful to the analyst. One way to overcome the dependence from the analysis is the use of the related concept of accuracy, which is the distance between the true value computed without DP and the DP released value. Therefore, accuracy is often used in place of utility, because more accurate results are generally more useful for an analysis. The so-called  $(\alpha, \beta)$ -accuracy framework [15] can be used to measure accuracy. Here,  $\alpha$  represents an upper bound on the absolute error committed, whilst  $\beta$  is the probability to violate this bound.

**Definition 12.** ( $(\alpha, \beta)$ -accuracy) Given a function  $f : \mathcal{D} \rightarrow \mathbb{R}$  mapping a dataset  $x \in \mathcal{D}$  to a real number, and a DP mechanism  $\mathcal{M}_f : \mathcal{D} \rightarrow \mathbb{R}$ ,  $\mathcal{M}_f$  is  $(\alpha, \beta)$ -accurate if  $\Pr \left[ \|f(x) - \mathcal{M}_f(x)\|_\infty \geq \alpha \right] \leq \beta$ .

It can be shown [15], starting from the Cumulative Distribution Function for the Laplace distribution  $\text{Lap}(b)$ , that the Laplace mechanism is  $(\alpha, \beta)$ -accurate with

$$\alpha = \ln \left( \frac{1}{\beta} \right) \cdot \left( \frac{s}{\epsilon} \right). \quad (1)$$

Regarding the Gaussian and the  $\rho$ -zCDP mechanisms, we did not find in the literature a corresponding derivation for the  $\alpha$  value; as an additional contribution, here we derive their analytical form. We start by considering the Cumulative Distribution Function for the normal distribution  $\mathcal{N}(\sigma)$ , which is  $\frac{1}{2} \left[ \operatorname{erfc} \left( -\frac{x}{\sigma\sqrt{2}} \right) \right]$ . The probability  $\Pr[X > x]$  is  $1 - \frac{1}{2} \left[ \operatorname{erfc} \left( -\frac{x}{\sigma\sqrt{2}} \right) \right]$ , so that, substituting  $x = t\sigma$ , we get:

$$\Pr[X > x] = 1 - \frac{1}{2} \operatorname{erfc} \left[ -\frac{t}{\sqrt{2}} \right]. \quad (2)$$

Therefore, we need to solve, taking into account that  $0 < \beta < 1$ , the following equation, with regard to  $t$ :

$$1 - \frac{1}{2} \operatorname{erfc} \left[ -\frac{t}{\sqrt{2}} \right] \leq \beta \quad (3)$$

obtaining

$$t \geq -\sqrt{2} \operatorname{erfc}^{-1}(2(1 - \beta)). \quad (4)$$

It follows that the Gaussian mechanism is  $(\alpha, \beta)$ -accurate with

$$\alpha = (-\sqrt{2} \operatorname{erfc}^{-1}(2(1 - \beta))) \sqrt{\frac{2s^2 \ln \left( \frac{1.25}{\delta} \right)}{\epsilon^2}}. \quad (5)$$

Reasoning as before, we can also derive that the  $\rho$ -zCDP mechanism is  $(\alpha, \beta)$ -accurate with

$$\alpha = (-\sqrt{2} \operatorname{erfc}^{-1}(2(1 - \beta))) \sqrt{\frac{s^2}{2\rho}}. \quad (6)$$

Next, we introduce the FRUGAL-1U algorithm.

### 3. The FRUGAL-1U Algorithm

Among the many algorithms that have been designed for tracking quantiles in a streaming setting, FRUGAL [11] besides being fast and accurate, also restricts by design the amount of memory that can be used. It is well-known that in the streaming setting the main goal is to deliver a high-quality approximation of the result (this may provide either an additive or a multiplicative guarantee) by using the lowest possible amount of space. In practice, there is a tradeoff between the amount of space used by an algorithm and the corresponding accuracy that can be achieved. Surprisingly, FRUGAL only requires one unit of memory to track a quantile. The authors of FRUGAL have also designed a variant of the algorithm that uses two units of memory. In this Section, we introduce the one unit of memory version, which is called FRUGAL-1U. Algorithm 1 provides the pseudo-code for FRUGAL-1U.

The algorithm works as follows. First,  $\tilde{m}$  is initialized to zero (however, note that it can be alternatively initialized to the value of the first stream item, in order to increase the speed of convergence of the estimate towards the value of the true quantile). This variable will be dynamically updated each time a new item  $s_i$  arrives from the input stream  $S$ , and its value represents the estimate of the quantile  $q$  being tracked. The update is quite simple, since it only requires  $\tilde{m}$  to be increased or decreased by one. Specifically, a random number  $0 < \text{rand} < 1$  is generated by using a pseudo-random number generator (the call  $\text{random}(0, 1)$  in the pseudo-code) and if the incoming stream item is greater than the estimate  $\tilde{m}$  and  $\text{rand} > 1 - q$ , then the estimate  $\tilde{m}$  is increased, otherwise if the incoming stream item is smaller than the estimate  $\tilde{m}$  and  $\text{rand} > q$ , then the estimate  $\tilde{m}$  is decreased. Obviously, the algorithm is really fast and can process an incoming item in worst-case  $O(1)$  time. Therefore, a stream of length  $n$  can be processed in worst-case  $O(n)$  time and  $O(1)$  space.

Despite its simplicity, the algorithm provides good accuracy, as shown by the authors. The proof is challenging since the algorithm's analysis is quite involved. The complexity in the worst case is  $O(n)$ , since  $n$  items are processed in worst case  $O(1)$  time.

---

**Algorithm 1** Frugal-1U

---

**Require:** Data stream  $S$ , quantile  $q$ , one unit of memory  $\tilde{m}$

**Ensure:** estimated quantile value  $\tilde{m}$

```
1:  $\tilde{m} = 0$ 
2: for each  $s_i \in S$  do
3:    $rand = \text{random}(0, 1)$ 
4:   if  $s_i > \tilde{m}$  and  $rand > 1 - q$  then
5:      $\tilde{m} = \tilde{m} + 1$ 
6:   else if  $s_i < \tilde{m}$  and  $rand > q$  then
7:      $\tilde{m} = \tilde{m} - 1$ 
8:   end if
9: end for
10: return  $\tilde{m}$ 
```

---

Finally, the algorithm has been designed to deal with an input stream consisting of integer values distributed over the domain  $[N] = \{1, 2, 3, \dots, N\}$ . This is not a limitation though, owing to the fact that one can process a stream of real values as follows: fix a desired precision, say three decimal digits, then each incoming stream item with real value can be converted to an integer by multiplying it by  $10^3$  and then truncating the result by taking the floor. If the maximum number of digits following the decimal point is known in advance, truncation may be avoided altogether: letting  $m$  by the maximum number of digits following the decimal point, it suffices to multiply by  $10^m$ . Obviously, the estimated quantile may be converted back to a real number dividing the result by the fixed precision selected or by  $10^m$ .

## 4. Differentially-Private FRUGAL-1U

In this Section, we analyze the FRUGAL-1U algorithm and design DP variants of it. As shown in Section 3, the algorithm is quite simple. In order to estimate a quantile  $q$ , the current estimate  $\tilde{m}$  is either incremented or decremented by one based on the value of the incoming stream item  $s_i$ . The increments are applied with probability  $q$  and the decrements with probability  $1 - q$ .

Our DP versions of the algorithm are based on the definition of bounded DP (see Definition 4), in which two datasets  $x$  and  $x'$  are considered neighbors if  $x'$  can be obtained from  $x$  by changing one row. Owing to our choice, we need to analyze the impact of changing one incoming stream item with a different one on the quantile estimate  $\tilde{m}$ . The following Lemma is our fundamental result to then obtain DP versions of Frugal-1U.

**Lemma 1.** *Under bounded DP, the global sensitivity of the FRUGAL-1U algorithm is 2.*

*Proof.* Let  $s_i$  be the item to be changed, and  $s_j \neq s_i$  the item replacing  $s_i$ . There are a few symmetric cases to consider. Let  $s_i$  be the  $i$ -th stream item, so that the length of the stream  $S$  is equal to  $i - 1$  before the arrival of  $s_i$  and equal to  $i$  immediately after. Moreover, denote by  $\tilde{m}_{i-1}$  the estimate of the quantile  $q$  before the arrival of  $s_i$  and by  $\tilde{m}_i$  after seeing the item  $s_i$ . Suppose that the arrival of  $s_i$  causes  $\tilde{m}_i$  to increase by one with regard to  $\tilde{m}_{i-1}$ , i.e.,  $\tilde{m}_i = \tilde{m}_{i-1} + 1$ . Substituting  $s_i$  with  $s_j$  therefore can lead to the following cases: either  $\tilde{m}_i = \tilde{m}_{i-1} - 1$  or  $\tilde{m}_i = \tilde{m}_{i-1} + 1$ . Therefore, the estimate is unchanged or it is increased by 2. Similarly, assuming that the arrival of  $s_i$  causes  $\tilde{m}_i$  to decrease by one with regard to  $\tilde{m}_{i-1}$ , i.e.,  $\tilde{m}_i = \tilde{m}_{i-1} - 1$ , then there are, symmetrically, the following cases: either  $\tilde{m}_i = \tilde{m}_{i-1} + 1$  or  $\tilde{m}_i = \tilde{m}_{i-1} - 1$ . Therefore, the estimate is unchanged or is decremented by 2. It follows that the global sensitivity of the algorithm is  $\max_{x, x' : d(x, x') \leq 1} |f(x) - f(x')| = 2$ .  $\square$

Since the global sensitivity is 2, DP-FRUGAL-1U-L, a pure DP (see Definition 5) variant of the algorithm can be obtained by using the Laplace mechanism. We are now in the position to state the following theorem.



**Theorem 1** (DP-Frugal-1U-L). *FRUGAL-1U can be made  $\epsilon$ -DP by adding to the quantile estimate returned by the algorithm noise sampled from a Laplace distribution as follows:  $\tilde{m} = \tilde{m} + \text{Lap}(\frac{2}{\epsilon})$ .*

*Proof.* It follows straight from Lemma 1 and Definition 8.  $\square$

Next, we design DP-FRUGAL-1U-G, a  $(\epsilon, \delta)$ -DP (see Definition 6) version of the algorithm, by using the Gaussian mechanism.

**Theorem 2** (DP-Frugal-1U-LG). *FRUGAL-1U can be made  $(\epsilon, \delta)$ -DP by adding to the quantile estimate returned by the algorithm noise sampled from a Gaussian distribution as follows:  $\mathcal{F}(x) = f(x) + \mathcal{N}(\sigma^2)$  where  $\sigma^2 = \frac{8 \ln(1.25/\delta)}{\epsilon^2}$ .*

*Proof.* It follows straight from Lemma 1 and Definition 9.  $\square$

Finally, we design DP-FRUGAL-1U- $\rho$ , a  $\rho$ -zCDF version of the algorithm.

**Theorem 3** (DP-Frugal-1U- $\rho$ ). *FRUGAL-1U can be made  $\rho$ -zCDF by adding to the quantile estimate returned by the algorithm noise sampled from a Gaussian distribution as follows:  $\mathcal{F}(x) = f(x) + \mathcal{N}(\sigma^2)$  where  $\sigma^2 = \frac{2}{\rho}$ .*

*Proof.* It follows straight from Lemma 1 and Definition 11.  $\square$

Finally, we remark here that, contrary to many DP algorithms that initialize a data structure or a sketch using suitable noise, it is not possible to initialize the quantile estimate of FRUGAL-1U using the noise required by one of the proposed mechanisms. The reason is two-fold. First, the algorithm processes integer items, so that its initial estimate must be an integer as well whilst, in general, the noise is a floating point value. But, even assuming that we initialize the estimate to an integer value related to the noise (perhaps taking its floor or the ceiling), this will not help in any way since, by design, the algorithm adapts dynamically to the observed input items and converges to the estimated quantile. Therefore, the second reason is that the noise added will be silently discarded by the algorithm when converging to the quantile estimate. As a consequence, the noise must be added only after the algorithm termination to the returned estimated quantile.

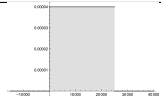
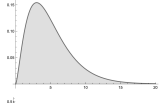
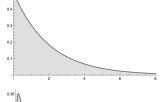
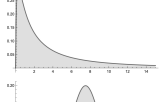
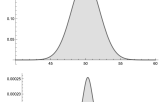
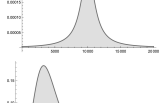
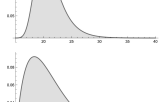
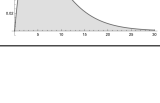
## 5. Experimental Results

In this section we present and discuss the results of the experiments carried out for FRUGAL-1U-L, FRUGAL-1U-G, FRUGAL-1U- $\rho$ . The experiments are limited to our algorithms owing to the fact that, to the best of our knowledge, there are no other frugal algorithms for the estimation of quantiles designed for the central model of differential privacy. The source code has been compiled using the Apple clang compiler v15.0 with the following flags: `-Os -std=c++14` (it is worth recalling that on macOS the flag `-Os` optimizes for size and usually on this platform the resulting executable is faster than the executable obtained by compiling using the `-O3` flag). The tests have been carried out on an Apple MacBook Pro laptop equipped with 64 GB of RAM and a 2.3 GHz 8-core Intel Core i9. The experiments have been repeated ten times for each specific category of test and the results have been averaged; the seeds used to initialize the pseudo-random generators are the same for each experiment and algorithm being tested.

The tests have been performed on eight synthetic datasets, whose properties are summarized in Table 1. The experiments have been executed varying the stream length, the quantile, the privacy budget,  $\epsilon$ ,  $\delta$  and  $\rho$ . Table 2 reports the default settings for the parameters.

We plot the relative error between the true quantile and the DP quantile estimate released by the algorithms under test, by allowing one parameter to vary whilst keeping the values of the others at their defaults. In all of the figures, the distribution used is the normal (later we compare the results obtained when varying the distribution as well).

**Table 1**  
Synthetic datasets.

Dataset	Distribution	Parameters	PDF
D1	Uniform	$[0, 1000]$	
D2	Chi square	$\alpha = 5$	
D3	Exponential	$\alpha = 0.5$	
D4	Lognormal	$\alpha = 1, \beta = 1.5$	
D5	Normal	$\mu = 50, \sigma = 2$	
D6	Cauchy	$\alpha = 10000, \beta = 1250$	
D7	Extreme Value	$\alpha = 20, \beta = 2$	
D8	Gamma	$a = 2, b = 4$	

**Table 2**  
Default settings of the parameters.

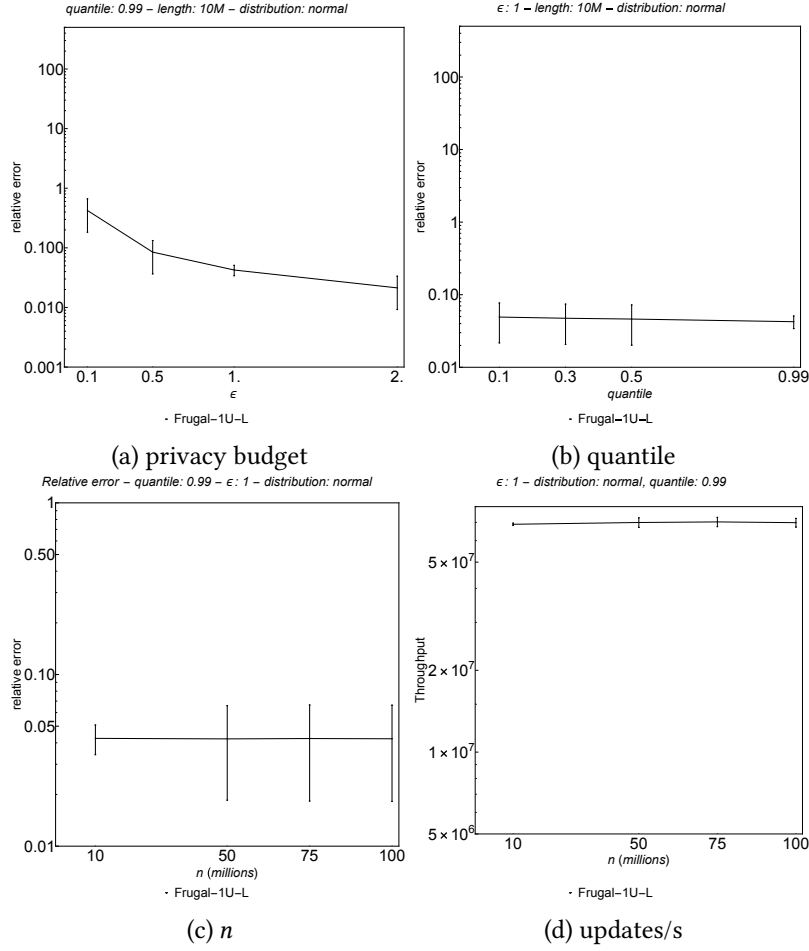
Parameter	Values	Default
quantile	$\{0.1, 0.3, 0.5, 0.99\}$	0.99
stream length	$\{10M, 50M, 75M, 100M\}$	10M
$\epsilon$	$\{0.1, 0.5, 1, 2\}$	1
$\delta$	$\{0.01, 0.04, 0.08, 0.1\}$	0.04
$\rho$	$\{0.1, 0.5, 1, 5\}$	1

The experimental results for FRUGAL-1U-L (using the Laplace mechanism) are shown in Figure 1. As depicted in Figure 1a, the relative error decreases as expected when the privacy budget  $\epsilon$  increases, meaning that the utility (see Section 2) of the released value increases when  $\epsilon$  increases. Therefore, a good tradeoff between privacy and utility is reached for  $0.5 \leq \epsilon \leq 1$ .

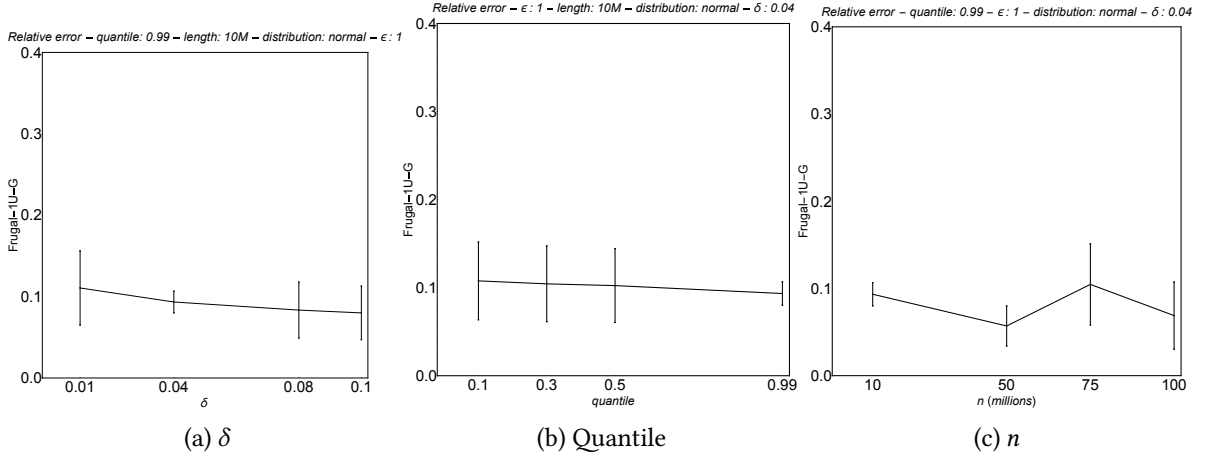
Figure 1b and Figure 1c depict the relative error when varying, respectively, the computed quantile and the stream size. As shown, the two quantities do not affect the security of the released quantile. Finally, Figure 1d depicts the throughput measured in updates/s.

Next, we analyze FRUGAL-1U-G. Increasing  $\delta$ , the probability of failure, provides as expected slightly less privacy, as shown in Figure 2a. By varying the computed quantile, a similar behaviour is observed. In Figure 2b, slightly less privacy is associated to higher quantiles. Finally, the impact of the stream size is depicted in Figure 2c, in which a fluctuating behaviour can be observed, even though the interval of variation is tight.

Regarding FRUGAL-1U- $\rho$ , Figure 3a shows that, as expected, the relative error decreases when the privacy budget  $\rho$  increases, and the utility increases correspondingly. A good privacy-utility tradeoff is reached for  $0.5 \leq \rho \leq 1$ . Figure 3b and 3c, related respectively to the relative error for varying quantile and stream size, present the same behaviour illustrated for the Gaussian mechanism. This is



**Figure 1:** FRUGAL-1U-L. Relative error varying the privacy budget  $\epsilon$ , the quantile  $q$  and the stream size  $n$ . Throughput measured in updates/s.

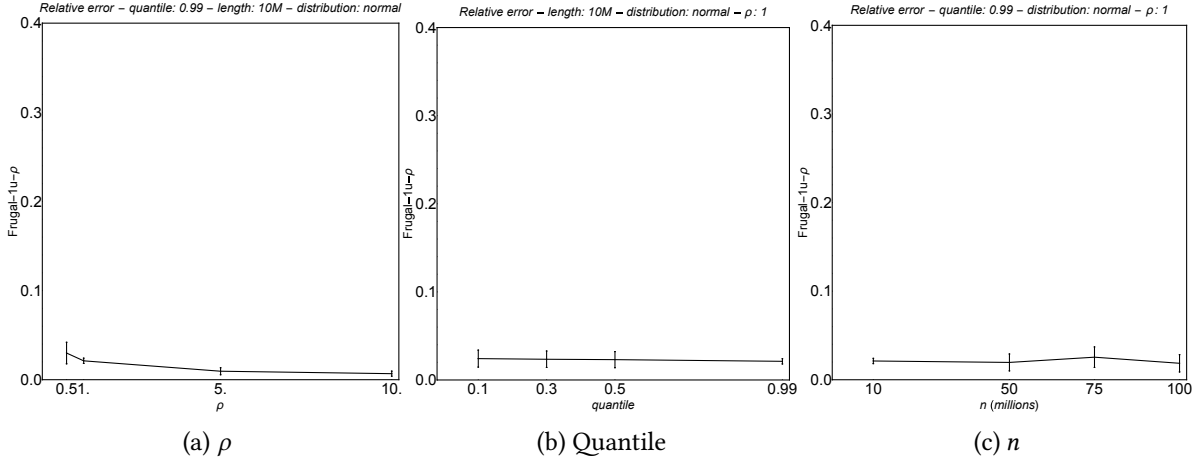


**Figure 2:** FRUGAL-1U-G. Relative error varying the probability  $\delta$ , the quantile  $q$  and the stream size  $n$ .

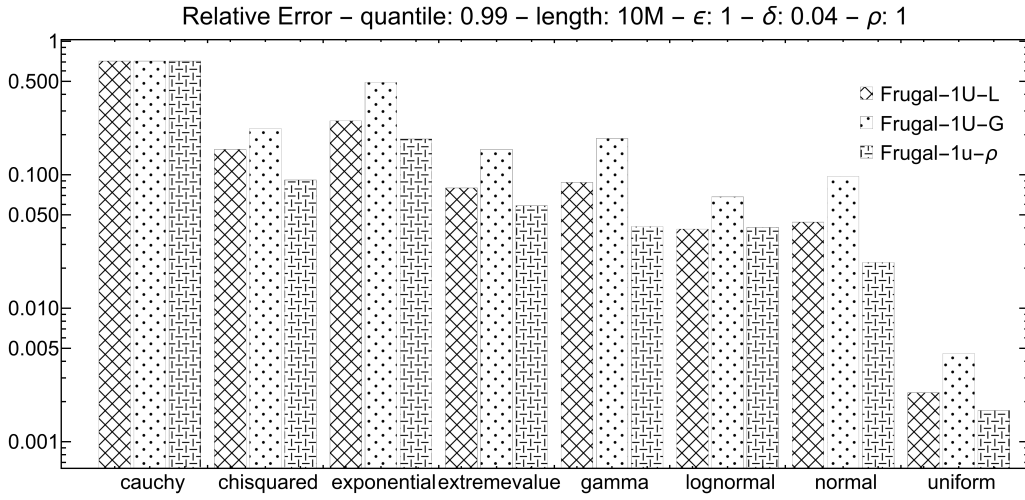
not surprising since this mechanism adds Gaussian noise (though the way noise is derived is of course different).

We now turn our attention to what happens when we vary the distribution, including heavy-tailed instances. Figure 4 provides the results for FRUGAL-1U-L, FRUGAL-1U-G and FRUGAL-1U- $\rho$ . As shown, the relative error between the true quantile and the DP quantile estimate released by one of the algorithms varies with the distribution. However, for our proposed algorithms, as expected (since the global





**Figure 3:** FRUGAL-1U- $\rho$ . Relative error varying the parameter  $\rho$ , the quantile  $q$  and the stream size  $n$ .



**Figure 4:** Relative error varying the distributions.

sensitivity is just 2) the algorithms can be used independently of the actual distribution, with the notable exception related to the Cauchy distribution (which can be considered adversarial for our algorithms based on FRUGAL-1U as discussed in [11]).

Our results show that, having fixed a distribution, the behaviour of our algorithms based on FRUGAL-1U does not depend on the seed used to initialize the pseudo-random number generator used to draw samples from the distribution. In this sense, our algorithms are robust.

Finally, we analyze the  $(\alpha, \beta)$ -accuracy (Definition 12) of FRUGAL-1U-L. Fixing  $\beta = 0.04$ ,  $\epsilon = 1$  and taking into account that the global sensitivity of FRUGAL-1U is  $s = 2$ , by using equation (1) we get  $\alpha = \ln\left(\frac{1}{0.04}\right) \cdot 2 = 6.4$ , so that FRUGAL-1U-L is  $(6.4, 0.04)$ -accurate.

For FRUGAL-1U-G, using Eq. (5) with  $\delta = 0.04$ ,  $\beta = 0.04$  and  $\epsilon = 1$  we get  $\alpha = 9.1$  so that FRUGAL-1U-G is  $(9.1, 0.04)$ -accurate. Finally, FRUGAL-1U- $\rho$  accuracy is determined by using equation (6) with  $\rho = 1$  and  $\beta = 0.04$ , so that  $\alpha = 2.4$  and FRUGAL-1U- $\rho$  is  $(2.4, 0.04)$ -accurate.

## 6. Conclusions

In this paper, we proposed DP algorithms for tracking quantiles in a streaming setting. Our algorithms are DP variants of the well-known FRUGAL-1U algorithm, characterized by the property of requiring just a tiny amount of memory to process a stream whilst guaranteeing surprising accuracy for quantile estimates. In particular, for FRUGAL-1U we gave corresponding  $\epsilon$ -DP,  $(\epsilon, \delta)$ -DP, and  $\rho$ -zCDF algorithms

after proving that the global sensitivity of FRUGAL-1U is equal to 2. Finally, we also showed that the proposed algorithms achieve good accuracy in the experimental results.

## Declaration on Generative AI

The author(s) have not employed any Generative AI tools.

## References

- [1] B. Jiang, J. Li, G. Yue, H. Song, Differential privacy for industrial internet of things: Opportunities, applications, and challenges, *IEEE Internet of Things Journal* 8 (2021) 10430–10451. doi:10.1109/JIOT.2021.3057419.
- [2] A. Machanavajjhala, X. He, M. Hay, Differential privacy in the wild: A tutorial on current practices & open challenges, in: *Proceedings of the 2017 ACM International Conference on Management of Data, SIGMOD '17*, Association for Computing Machinery, New York, NY, USA, 2017, p. 1727–1730. URL: <https://doi.org/10.1145/3035918.3054779>. doi:10.1145/3035918.3054779.
- [3] A. D. Sarwate, K. Chaudhuri, Signal processing and machine learning with differential privacy: Algorithms and challenges for continuous data, *IEEE Signal Processing Magazine* 30 (2013) 86–94. doi:10.1109/MSP.2013.2259911.
- [4] X. Liu, W. Kong, S. Kakade, S. Oh, Robust and differentially private mean estimation, in: M. Ranzato, A. Beygelzimer, Y. Dauphin, P. Liang, J. W. Vaughan (Eds.), *Advances in Neural Information Processing Systems*, volume 34, Curran Associates, Inc., 2021, pp. 3887–3901. URL: [https://proceedings.neurips.cc/paper\\_files/paper/2021/file/1fc5309ccc651bf6b5d22470f67561ea-Paper.pdf](https://proceedings.neurips.cc/paper_files/paper/2021/file/1fc5309ccc651bf6b5d22470f67561ea-Paper.pdf).
- [5] Z. Yang, X. Xu, Y. Gu, A general framework for accurate and private mean estimation, *IEEE Signal Processing Letters* 29 (2022) 2293–2297. doi:10.1109/LSP.2022.3219356.
- [6] T. Zhu, G. Li, W. Zhou, P. S. Yu, Differentially private data publishing and analysis: A survey, *IEEE Transactions on Knowledge and Data Engineering* 29 (2017) 1619–1638. doi:10.1109/TKDE.2017.2697856.
- [7] W. Gao, S. Zhou, Privacy-preserving for dynamic real-time published data streams based on local differential privacy, *IEEE Internet of Things Journal* 11 (2024) 13551–13562. doi:10.1109/JIOT.2023.3337397.
- [8] F. Grassi, A. Coluccia, Distribution-agnostic linear unbiased estimation with saturated weights for heterogeneous data, *IEEE Transactions on Signal Processing* 71 (2023) 2910–2926. doi:10.1109/TSP.2023.3293908.
- [9] U. A. Müller, M. M. Dacorogna, O. V. Pictet, Heavy tails in high-frequency financial data, *A practical guide to heavy tails: Statistical techniques and applications* (1998) 55–78.
- [10] M. Crovella, M. Taqqu, A. Bestavros, Heavy-tailed probability distributions in the world wide web, in: R. Adler, R. Feldmann, M. Taqqu (Eds.), *A Practical Guide to Heavy Tails*, Birkhäuser Boston, Boston, MA, 1998, pp. 3–25.
- [11] Q. Ma, S. Muthukrishnan, M. Sandler, *Frugal Streaming for Estimating Quantiles*, Springer Berlin Heidelberg, Berlin, Heidelberg, 2013, pp. 77–96. URL: [https://doi.org/10.1007/978-3-642-40273-9\\_7](https://doi.org/10.1007/978-3-642-40273-9_7). doi:10.1007/978-3-642-40273-9\_7.
- [12] C. Dwork, F. McSherry, K. Nissim, A. Smith, Calibrating noise to sensitivity in private data analysis, in: S. Halevi, T. Rabin (Eds.), *Theory of Cryptography*, Springer Berlin Heidelberg, Berlin, Heidelberg, 2006, pp. 265–284.
- [13] C. Dwork, A. Roth, The algorithmic foundations of differential privacy, *Foundations and Trends® in Theoretical Computer Science* 9 (2014) 211–407. URL: <http://dx.doi.org/10.1561/04000000042>. doi:10.1561/04000000042.
- [14] S. Vadhan, *The Complexity of Differential Privacy*, Springer International Publishing,

Cham, 2017, pp. 347–450. URL: [https://doi.org/10.1007/978-3-319-57048-8\\_7](https://doi.org/10.1007/978-3-319-57048-8_7). doi:10.1007/978-3-319-57048-8\_7.

- [15] J. P. Near, X. He, Differential privacy for databases, *Foundations and Trends® in Databases* 11 (2021) 109–225. URL: <http://dx.doi.org/10.1561/19000000066>. doi:10.1561/19000000066.