Differentially Private Approximate Quantiles

Haim Kaplan [1 2] **Shachar Schnapp** [3] **Uri Stemmer** [1 2]

Abstract

In this work we study the problem of differentially private (DP) quantiles, in which given dataset X and quantiles $q_1, ..., q_m \in [0, 1]$, we want to output m quantile estimations which are as close as possible to the true quantiles and preserve DP. We describe a simple recursive DP algorithm, which we call Approximate Quantiles (AQ), for this task. We give a worst case upper bound on its error, and show that its error is much lower than of previous implementations on several different datasets. Furthermore, it gets this low error while running time two orders of magnitude faster that the best previous implementation.

1. Introduction

Quantiles are the values that divide a sorted dataset in a certain proportion. They are one of the most basic and important data statistics, with various usages, ranging from computing the median to standardized test scores (GRE, 2021). Given sensitive data, publishing the quantiles can expose information about the individuals that are part of the dataset. For example, suppose that a company wants to publish the median of its users' ages. Doing so means to reveal the date of birth of a certain user, thus compromising the user's privacy. *Differential privacy* (DP) offers a solution to this problem by requiring the output distribution of the computation to be insensitive to the data of any single individual. This leads us to the definition of the DP-quantiles problem:

Definition 1.1 (The DP-Quantiles Problem)**.** Let $a, b \in \mathbb{R}$. Given a dataset $X \subseteq (a, b)$ containing n points from (a, b), and a set of m quantiles $0 < q_1 \leq \cdots \leq q_m < 1$, privately identify quantile estimations $v_1, ..., v_m$ such that for every $j \in [m]$ we have $\Pr_{x \sim U_X}[x \leq v_j] \approx q_j$,[1] where U_X is the uniform distribution over the data X.[2]

On the theory side, the DP-quantiles problem is relatively well-understood, with advanced constructions achieving very small asymptotic error (Beimel et al., 2016; Bun et al., 2015; Kaplan et al., 2020). However, as was recently observed by Gillenwater et al. (2021), due to the complexity of these advanced constructions and their large hidden constants, their practicality is questionable. This led Gillenwater et al. (2021) to design a simple algorithm for the DP-quantiles problem that performs well in practice. In this work we revisit the DP-quantiles problem. We build on the theoretical construction of Bun et al. (2015), and present a new (and simple) practical algorithm that obtains better utility and runtime than the state-of-the-art construction of Gillenwater et al. (2021) (and all other existing implementations). When the number of quantiles is large, the error of AQ algorithm reaches up to 7.14 time lower than the best existing baseline.

1.1. Our Contributions

We provide Approximate Quantiles, an algorithm and implementation for the DP-quantiles problem (Section 3). We prove a worst case bound on the error of the AQ algorithm for arbitrary m quantiles, and a tighter error bound for the case of uniform quantiles $q_i = i/(m+1)$, $i = 1, \ldots, m$ (Section 3.2). We experimentally evaluated the AQ algorithm and conclude that it obtains higher accuracy and faster runtime than the existing state-of-the-art implementations (Section 4). In addition, we adapt our algorithm (and its competitors) to the definition of Concentrated Differential Privacy (zCDP) (Bun & Steinke, 2016). We show that its dominance over other methods is even more significant with this definition of privacy.

1.2. A Technical Overview of Our Construction: Algorithm Approximate Quantiles

Our algorithm operates using a DP algorithm for estimating a *single* quantile. Specifically, we assume that we have a DP algorithm $\mathcal{A} : \mathbb{R}^2 \times \mathbb{R}^n \times (0, 1) \to \mathbb{R}$ that takes a domain $I = (a, b) \in \mathbb{R}^2$ (an interval on the line), a database $X \in \mathbb{R}^n$ (containing n points in I), and a sin-

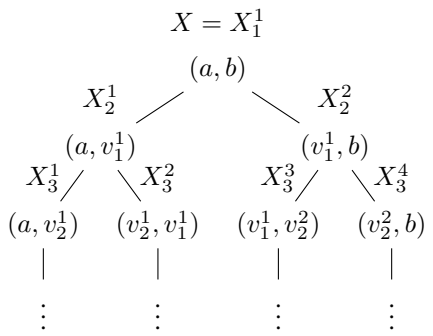
*Equal contribution [1]Tel Aviv University [2]Google Research [3]Ben-Gurion University. Correspondence to: Shachar Schnapp <schnapp@post.bgu.ac.il>.

Proceedings of the 39^{th} International Conference on Machine Learning, Baltimore, Maryland, USA, PMLR 162, 2022.

[1]We will make this precise later.

[2]We assume that there are no duplicate points in X.

Figure 1: The recursion tree of the algorithm. At each node we write the range of the corresponding subproblem and above it the part of the data in this subproblem.



gle quantile $q \in (0, 1)$, and returns a point $v \in I$ such that $\Pr_{x \sim U_X}[x \leq v] \approx q$. Estimating a single quantile is a much easier task, with a very simple algorithm based on the exponential mechanism (McSherry & Talwar, 2007). A naive approach of using $\mathcal{A}$ for solving the DP-quantiles problem would be to run it m times (once for every given quantile). However, due to composition costs of running m DP algorithms on the same data, the error with this approach would scale polynomially with m. As we demonstrate in our experimental results (in Section 4), this leads to a significantly reduced performance. In contrast, as we explain next, in our algorithm the error scales only logarithmically in the number of quantiles m. The AQ algorithm privately estimates (using $\mathcal{A}$) the "middle quantile" $p = q_{\lceil m/2 \rceil}$ and observes an answer v. Then it splits the problem into two sub-problems. The first sub-problem is defined over the dataset X_ℓ which contains the values from X that are smaller than v. Its goal is to privately compute the quantiles $(q_1, .., q_{\lceil m/2 \rceil - 1})/p$ on X_ℓ. The second problem is defined over the dataset X_u which contains the values from X that are greater than v. The goal of the second problem is to privately compute $(q_{\lceil m/2 \rceil + 1} - p, \ldots, q_m - p)/(1-p)$ on X_u. Notice that the recursive sub-problems have smaller ranges. Specifically, at the first level of the recursion tree we compute one quantile $q_1^1 \triangleq q_{\lceil m/2 \rceil}$ on data points from a range (a, b). We denote by v_1^1 the estimate which we receive. At the second level we compute two normalized quantiles $q_2^1 \triangleq q_{\lceil m/4 \rceil}/q_1^1$ and $q_2^2 \triangleq q_{\lceil 3m/4 \rceil}/(1 - q_1^1)$. The quantile q_2^1 is computed on data in the range (a, v_1^1), and we denote its estimate by v_2^1. The second quantile is computed on the data in the range $[v_1^1, b]$, and we denote its estimate by v_2^1, and so on. Figure 1 illustrates this recursion tree. A few remarks are in order.

1. By shrinking the data range from one level to the next, we effectively reduce the error of algorithm $\mathcal{A}$ (because its error depends on the data range). We have found that, in practice, this has a large impact on our accuracy guarantees.
2. Note that every single data point participates only in $\log(m) + 1$ sub-problems (one at each level). This allows our privacy loss (and, hence, our error) to grow only logarithmically in m.

1.3. Related Work

Private (Cumulative Distribution Function) CDF estimation can be applied to estimating all the quantiles (Bun et al., 2015; Kaplan et al., 2020), however the theoretically best known algorithm for private CDF estimation Bun et al. (2015) relies on several reductions, thus limiting its practicality. We present our AQ algorithm, inspired by Bun et al. (2015), taking their CDF estimation algorithm into practice. As opposed to the algorithm proposed by Bun et al. (2015), our algorithm avoids discretization of the domain by solving the DP single quantile problem using the exponential mechanism (Smith, 2011) instead of an interior point algorithm (Bun et al., 2015). Moreover, we split the data according the desired quantiles (rather than uniformly) while avoiding using the laplace mechanism in the process. A common tree-based approach to CDF estimation is included in our empirical error analysis (Section 4). A recent work by Gillenwater et al. (2021) proposed an instance of the exponential mechanism that simultaneously draws m quantiles. The naive implementation of this exponential mechanism for m quantiles is computationally difficult, but Gillenwater et al. (2021) provide a sophisticated $O(mn \log n + m^2 n)$ implementation. The empirical experiments of Gillenwater et al. (2021) show that when the number of quantiles is small, JointExp algorithm performs best. A comparison with this algorithm is included in our experiments (Section 4).

2. Preliminaries

A database X is a set of n points from some data domain $\mathcal{D}$.[3] Differential privacy uses the definition of *neighbors* as follows.

Definition 2.1. Databases X and $X' \in \mathcal{D}^n$ are *neighbors*, denoted $X \sim X'$, if one of them can be obtained from the other by adding or removing a single element.

We use the *add-remove* definition of neighbors, as opposed to the *swap* definition (in which we replace a point instead of deleting or inserting it), although it is important to note that our algorithm Approximate Quantiles (AQ) easily adapts to the swap framework. Differential privacy can be defined using the notion of neighboring databases as follows:

Definition 2.2 (Dwork et al. (2006b))**.** $\mathcal{A}$ randomized algorithm $\mathcal{A}\colon \mathcal{D}^* \to \mathcal{Y}$ is (ε, δ)-differentially private ((ε, δ)-DP) if, for every pair of neighboring databases X, X' and every output subset $Y \subseteq \mathcal{Y}$, $\Pr_{\mathcal{A}}[A(X) \in Y] \leq e^\varepsilon \cdot \Pr_{\mathcal{A}}[A(X') \in Y] + \delta$. When $\delta > 0$, we say $\mathcal{A}$ sat-

[3] The domain in this paper is the interval (a, b).

isfies *approximate* differential privacy. If $\delta = 0$, we say $\mathcal{A}$ satisfies *pure* differential privacy, and shorthand this as ε-differential privacy (ε-DP).

The composition property is a key benefit of differential privacy: an algorithm that relies on differentially private subroutines inherits an overall privacy guarantee by simply adding up the privacy guarantees of its components.

Lemma 2.3 (Dwork et al. (2006a;b)). *Let $\mathcal{A}_1, \ldots, \mathcal{A}_k$ be k algorithms that respectively satisfy $(\varepsilon_1, \delta_1)$-, $\ldots$, $(\varepsilon_k, \delta_k)$-differential privacy. Then running $\mathcal{A}_1, \ldots, \mathcal{A}_k$ satisfies $\left(\sum_{i=1}^k \varepsilon_i, \sum_{i=1}^k \delta_i\right)$-differential privacy.*

The above lemma has come to be known as "basic composition" in the literature of differential privacy (see (Dwork et al., 2010b) for more advanced composition theorems). Given Lemma 2.3, a simplistic approach for the DP-quantiles problem would be to estimate each of the m quantiles separately, using an $\frac{\varepsilon}{m}$-DP algorithm, and then to apply Lemma 2.3 in order to show that the m estimations together satisfy ε-DP. As we will see, our algorithm outperforms this simplistic approach by a large gap. We will also rely on the exponential mechanism, a common building block for differentially private algorithms.

Definition 2.4 (McSherry & Talwar (2007)). Given utility function $u\colon \mathcal{D}^* \times S \to \mathbb{R}$ mapping (database, output) pairs to real-valued scores with L_1 sensitivity

$$\Delta_u \triangleq \max_{X \sim X', s \in S} |u(X, s) - u(X', s)|,$$

the probability that the exponential mechanism M has an output $s \in S$ is:

$$\Pr[M(X) = s] \propto \exp\left(\frac{\varepsilon u(X, s)}{2\Delta_u}\right),$$

where $\propto$ elides the normalization factor.

The exponential mechanism maintains the database's privacy while prioritizing its higher utility outputs.

Lemma 2.5 (McSherry & Talwar (2007)). *The mechanism described in Definition 2.4 is ε-DP and for error parameter $\gamma > 0$ satisfies that:*

$$\Pr_{s \in S}[u(X, s) > \mathrm{Opt} - \gamma] \leq |S| \cdot \exp\left(-\frac{\varepsilon\gamma}{2\Delta}\right),$$

where $\mathrm{Opt} = \max_{s \in S}\{u(X, s)\}$.

In our case in this paper, the solution space S is infinite. Specifically, it is an interval (a, b). Lemma A.1 in Appendix A adapts the exponential mechanism to this setting.

3. Approximate Quantiles Algorithm

This section demonstrates our differentially private quantiles algorithm, Approximate Quantiles. As we mentioned in the introduction, our algorithm uses a subroutine $\mathcal{A}$ for privately estimating a single quantile. We implement algorithm $\mathcal{A}$ using the exponential mechanism (see Appendix A). We remark that, if the dataset is given sorted, then $\mathcal{A}$ runs in linear time, and the time required for all recursive calls at the same level of the recursion tree is $O(n)$. It follows that the overall time complexity of AQ algorithm is $O(n \log m)$.

Algorithm 1 - Approximate Quantiles (AQ)

input Domain $\mathcal{D} = (a, b)$, database $X = (x_1, \ldots, x_n) \in \mathcal{D}^n$, quantiles $Q = (q_1, \ldots, q_m)$, privacy parameter ε.

Let $\mathcal{A} : \mathbb{R}^2 \times \mathbb{R}^n \times (0, 1) \to \mathbb{R}$ be a $\frac{\varepsilon}{\log m + 1}$-DP mechanism for a single quantile.
function $F((a, b), X, Q)$
  if $m = 0$ **then**
    return $null$
  else if $m = 1$ **then**
    return $\{\mathcal{A}((a, b), X, q_1)\}$
  end if
  Let $\hat{m} = \lceil m/2 \rceil$
  Let $p = q_{\hat{m}}$
  Let $v = \mathcal{A}((a, b), X, p)$
  Let $X_\ell, X_u = \{x \in X \mid x < v\}, \{x \in X \mid x > v\}$
  Let $Q_\ell = (q_1, \ldots, q_{\hat{m}-1})/p$
  Let $Q_u = (q_{\hat{m}+1} - p, \ldots, q_m - p)/(1 - p)$
  return $F((a, v), X_\ell, Q_\ell) \cup \{v\} \cup F((v, b), X_u, Q_u)$
end function

We denote by q_i^j the normalized quantile computed by the jth subproblem at the ith level of the algorithm's recursion tree, and by v_i^j the result of this computation. Note that the number of subproblems at level i is 2^{i-1}. (At the last level some of the subproblems may be empty.) We let X_i^j be the data used to compute v_i^j. It was created by splitting the data $X_{i-1}^{\lceil j/2 \rceil}$ into two parts according to $v_{i-1}^{\lceil j/2 \rceil}$. We note that $X_i^{j_1}$ and $X_i^{j_2}$ are disjoint for a fixed level i and $j_1 \neq j_2$. This allows us to avoid splitting our privacy budget between subproblems at the same level of the recursion, and we split it only between different levels, see Section 3.1. We also denote $Q_i \triangleq (q_i^1, \ldots, q_i^{2^{i-1}})$ and $V_i \triangleq (v_i^1, \ldots, v_i^{2^{i-1}})$.

We assume that the data does not contain duplicate points. This can be enforced by adding small perturbations to the points. The answer we return is with respect to the perturbed points. In fact the utility of our algorithm depends on the minimum distance between a pair of points.

3.1. Privacy Analysis

First we prove that AQ algorithm is ε-DP.

Lemma 3.1 (Differential Privacy). *If $\mathcal{A}$ is an $\frac{\varepsilon}{\log m + 1}$-DP mechanism for a single quantile then AQ algorithmis ε-DP.*

Proof. It suffices to show that for each level $1 \leq i \leq \log m + 1$ the output V_i is $\frac{\varepsilon}{\log m+1}$-DP, since the number of levels is $\log m + 1$, from composition (Lemma 2.3) we get that the AQ algorithm satisfies (ε, δ)-DP.

Let X and $X' = X \cup \{x'\}$ be neighboring databases, mark as ${X'}_i^k$ the part of level i that contains x' among ${X'}_i^j$, $1 \leq j \leq 2^{i-1}$ (as explained above, only one ${X'}_i^j$ contains x'). For each $j \neq k$ the data X_i^j equals ${X'}_i^j$ and therefore the probability of the output v_i^j is the same under X or X'. The output v_i^k is obtained by $\mathcal{A}$ which is a $\frac{\varepsilon}{\log m+1}$-DP mechanism, therefore it satisfies $\frac{\varepsilon}{\log m+1}$-DP. □

3.2. Utility Analysis

A q_i-quantile is any point $o_i \in (a, b)$ such that the number points of X which are in (a, o_i) is $\lfloor q_i n \rfloor$. We also define the *gap*, $\text{Gap}_X(d_1, d_2)$, between $d_1, d_2 \in (a, b)$ with respect to X are the number of points in the data X that fall between d_1 and d_2, formally: $\text{Gap}_X(d_1, d_2) = |\{x \in X \mid x \in [\min\{d_1, d_2\}, \max\{d_1, d_2\})\}|$. Using this notion we define the error of the algorithm. Given dataset X, quantiles $Q = (q_1, \ldots, q_m)$, solution $V = (v_1, \ldots, v_m)$ and true quantiles $O = (o_1, \ldots, o_m)$, the *maximum missed points error* is defined as:

$$\text{Err}_X(O, V) = \max_{j\in[m]}\{\text{Gap}(o_j, v_j)\} = \max_{j\in[m]}\{||\{x \in X | x < v_j\}| - \lfloor q_j \cdot n \rfloor|\}.$$

This error was first defined by Smith (2011) and is widely used in the literature on the differentially private quantiles problem. We first analyze the error of our algorithm in the general case, without any constraints on the given quantiles.

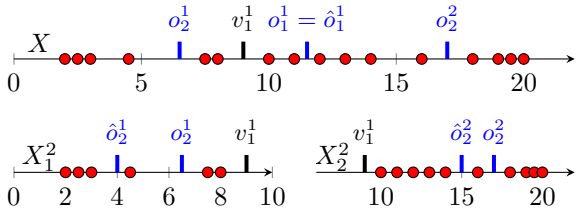
Lemma 3.2 (General Quantiles Utility)**.** *Let $X \in (a, b)^n$ be a database, and let $\mathcal{A}$ be an algorithm that computes an approximation v for a single quantile q of X such that $\Pr[\text{Gap}_X(o, v) > \gamma] \leq \frac{\beta}{m}$. for some constants $\beta, \gamma > 0$, where o is a true q-quantile. We run AQ algorithm using $\mathcal{A}$ on a database X, and quantiles $Q = (q_1, \ldots, q_m)$. Then, with probability $1 - \beta$, we get approximate quantiles $V = (v_1, \ldots, v_m)$ such that $\text{Err}_X(O, V) \leq (\log m + 1)\gamma$.*

Proof. For the computation of m quantiles the AQ algorithm applies $\mathcal{A}$ at most m times (once per each internal node of the recursion tree). Since in each run $\mathcal{A}$ has error at most γ with probability $1 - \beta/m$ it follows by a union bound that:

$$\Pr[\exists i, j \text{ s.t. } \text{Gap}_X(\hat{o}_i^j, v_i^j) > \gamma] \leq \beta, \tag{1}$$

where $\hat{o}_i^j$ is a true q_i^j-quantile with respect to the dataset X_i^j. We also denote by o_i^j a true q_k-quantile with respect to X where q_k is the original fraction in Q that corresponds to q_i^j (see Figure 2).

Figure 2: An error in computing v_1^1 by at most $\gamma = 2$ points from the optimal value o_1^1, might cause an error of at most γ points between the value of o_j^2 compared to $\hat{o}_j^2$. The example demonstrates the computed quantiles $q_1 = \frac{1}{4}$, $q_2 = \frac{1}{2}$ and $q_3 = \frac{3}{4}$. Note that $q_2^1, q_2^2 = \frac{1}{2}$.



At the first level ($i = 1$) of the recursive tree we compute one quantile q_1^1 on the data $X = X_1^1$ and therefore $\text{Gap}_X(\hat{o}_1^1, o_1^1) = 0$. At the second level ($i = 2$), we split the data X according to v_1^1 into X_2^1, X_2^2, since $\text{Gap}_X(v_1^1, o_1^1) \leq \gamma$, the q_2^j-quantile $\hat{o}_2^j$ of the dataset X_2^j satisfies that $\text{Gap}_X(\hat{o}_2^j, o_2^j) \leq \gamma$ (see Figure 2).

By induction, at layer i, for every j we have that $\text{Gap}_X(\hat{o}_i^j, o_i^j) \leq (i-1) \cdot \gamma$. Combining this with Equation (1) results in $\text{Gap}_X(v_i^j, o_i^j) \leq i \cdot \gamma$. At the last level ($i = \log m+1$) we have that $\text{Gap}_X(v_i^j, o_i^j) \leq (\log m+1) \cdot \gamma$. □

Theorem 3.3. *Assume that we implement $\mathcal{A}$ using the exponential mechanism with privacy parameter $\frac{\varepsilon}{\log m+1}$, as described in Appendix A to solve the single quantile problem. Then, given a database $X \in (a, b)^n$ and quantiles $Q = (q_1, \ldots, q_m)$ the AQ algorithm is ε-DP, and with probability $1 - \beta$ output $V = (v_1, \ldots, v_m)$ that satisfies*

$$\text{Err}_X(O, V) \leq O\left(\frac{\log^2 m(\log\psi + \log m + \log\frac{1}{\beta})}{\varepsilon}\right),$$

where $\psi = \frac{b-a}{\min_{k\in[n+1]}\{x_k - x_{k-1}\}}$.[4]

Proof. By Lemma A.1, the exponential mechanism with privacy parameter $\frac{\varepsilon}{\log m+1}$ has an error more than $2(\log m + 1)\frac{\log\psi + \log m - \log\beta}{\varepsilon}$ with probability at most $\frac{\beta}{m}$. Therefore, by Lemma 3.2, the AQ algorithm has an error no larger than $\gamma = 2(\log m+1)^2 \cdot \frac{\log\psi + \log m - \log\beta}{\varepsilon}$ with probability $1-\beta$. Combining this with Lemma 3.1 the theorem follows. □

The uniform quantiles problem is a common instance of the quantiles problem where $q_i = \frac{i}{m+1}$, for $i = 1, \ldots, m$. Lemma 3.4 shows that when the desired quantiles are uniform it is possible to guarantee that $\text{Err}_X(O, V) = O(\gamma)$ with probability $1 - \beta$.

[4] We define $x_0 = a$ and $x_{n+1} = b$. These are not real data points.

Lemma 3.4 (Uniform Quantiles Utility). *Let $X \in (a,b)^n$ be a database, and let $\mathcal{A}$ be an algorithm that computes an approximation v for a single quantile q of X such that*

$$\Pr[\mathrm{Gap}_X(o,v) > \gamma] \leq \frac{\beta}{m}.$$

for some constants $\beta, \gamma > 0$, where o is a true q-quantile. We run AQ algorithm using $\mathcal{A}$ on a database X, and quantiles $Q = (q_1, \ldots, q_m)$ where $q_i = \frac{i}{m+1}$. Then, with probability $1-\beta$, the AQ algorithm returns approximate quantiles $V = (v_1, \ldots, v_m)$ satisfying $\mathrm{Err}_X(O,V) \leq 2\gamma$.

Proof. For simplicity we assume that $m = 2^k - 1$. The proof is similar for other values of m. For this value of m we have that $q_\ell = \ell/2^k$, $\ell = 1, \ldots, 2^k - 1$ and $q_i^j = 1/2$ for all i and j. Furthermore, we have that the number of points in $X \cap (a, o_i^j)$ is $\lfloor \frac{2 \cdot j - 1}{2^i} n \rfloor$, $j = 1, \ldots, 2^{i-1}$. As in Lemma 3.2, Equation (1) holds. So we assume in the rest of the proof that $\mathrm{Gap}_X(v_i^j, \hat{o}_i^j) \leq \gamma$ for all i and j.

For the root ($i = 1$) of the recursion tree we compute one quantile q_1^1 on the data $X = X_1^1$ and therefore $\mathrm{Gap}_X(\hat{o}_1^1, o_1^1) = 0$. At the second level, ($i = 2$), we split the data X according to v_1^1 into X_2^1, and X_2^2. Since $\mathrm{Gap}_X(v_1^1, o_1^1) \leq \gamma$, we have that $\lfloor n/2 \rfloor - \gamma \leq |X_2^j| \leq \lfloor n/2 \rfloor + \gamma$. Recall that $|X \cap (a, o_2^1)| = \lfloor n/4 \rfloor$ and $|X \cap (a, o_2^2)| = \lfloor 3n/4 \rfloor$ and therefore $\mathrm{Gap}_X(\hat{o}_2^j, o_2^j) \leq \gamma/2$ for $j = 1, 2$. Now, since $\mathrm{Gap}_X(v_2^j, \hat{o}_2^j) \leq \gamma$ we get that $\lfloor n/4 \rfloor - 3\gamma/2 \leq |X_3^j| \leq \lfloor n/4 \rfloor + 3\gamma/2$, and therefore $\mathrm{Gap}_X(\hat{o}_3^j, o_3^j) \leq 3\gamma/4$, for $j \in [4]$. Similarly, since $\mathrm{Gap}_X(v_3^j, \hat{o}_3^j) \leq \gamma$, it follows that $\lfloor n/8 \rfloor - 7\gamma/4 \leq |X_4^j| \leq \lfloor n/8 \rfloor + 7\gamma/4$ and therefore $\mathrm{Gap}_X(\hat{o}_4^j, o_4^j) \leq 7\gamma/8$, for $j \in [8]$. We conclude that by induction on the levels we get that,

$$\mathrm{Gap}_X(\hat{o}_i^j, o_i^j) \leq \frac{2^{i-1} - 1}{2^{i-1}} \cdot \gamma \leq \gamma,$$

for all i and j. Combining this upper bound with Equation (1) we get that $\mathrm{Gap}_X(v_j^i, o_j^i) \leq 2\gamma$. □

Theorem 3.5 improves the bound of Theorem 3.3 for uniform quantiles. We omit its proof which is similar to the proof of Theorem 3.3 using Lemma 3.4 instead of Lemma 3.2.

Theorem 3.5. *If we set $\mathcal{A}$ to be the exponential mechanism with privacy parameter $\frac{\varepsilon}{\log m+1}$, as described in Appendix A, then given data $X \in (a,b)^n$ and quantiles $Q = (q_1, \ldots, q_m)$, where $q_i = \frac{i}{m+1}$, the AQ algorithm is ε-DP and with probability $1 - \beta$ output $V = (v_1, \ldots, v_m)$ that satisfies*

$$\mathrm{Err}_X(O,V) = O\left(\frac{\log m(\log \psi + \log m + \log \frac{1}{\beta})}{\varepsilon}\right),$$

where $\psi = \frac{b-a}{\min_{k \in [n+1]}\{x_k - x_{k-1}\}}$.

Quantiles sanitization. Given a database $X = (x_1, .., x_n) \in (a,b)^n$, we can produce a differentially private dataset $\hat{X} = (\hat{x}_1, \ldots, \hat{x}_n) \in (a,b)^n$, such that for each point $\hat{x}_\ell$, the number of points in $\hat{X}$ that are smaller than $\hat{x}_\ell$ is similar to the number of points in X that are smaller than $\hat{x}_\ell$. This is specified precisely in the following Corollary of Theorem 3.3. In particular this corollary implies that for every interval $I \subseteq (a,b)$, $|X \cap I|$ is approximately equal to $|\hat{X} \cap I|$.

Corollary 3.6. *Assume that we implement $\mathcal{A}$ using the exponential mechanism with privacy parameter $\frac{\varepsilon}{\log n+1}$, as described in Appendix A, to solve the single quantile problem. Then, given a database $X \in (a,b)^n$ and n quantiles $Q = (q_1, \ldots, q_n)$, where $q_i = i/n$, the AQ algorithm is ε-DP, and with probability $1 - \beta$ outputs $\hat{X} = (\hat{x}_1, \ldots, \hat{x}_n)$ that satisfies*

$$\mathrm{Err}_X(X, \hat{X}) = O\left(\frac{\log n(\log \psi + \log n + \log \frac{1}{\beta})}{\varepsilon}\right).$$

where $\psi = \frac{b-a}{\min_{k \in [n+1]}\{x_k - x_{k-1}\}}$.

3.3. Zero-Concentrated Differential Privacy

Zero Concentrated Differential Privacy (zCDP) (Bun & Steinke, 2016) offers smoother composition properties than standard (ε, δ)-DP. The general idea is to compare the Rényi divergence of the privacy losses random variables for neighboring databases. We analyse our algorithm also under this definition of privacy. As in Section 3.1, the privacy analysis of our algorithm applies composition of the processing of different levels in the recursion tree. zCDP's composition theorem allows us to run the exponential mechanism at each level with a higher privacy parameter, which results in a tighter error bound for the exponential mechanism. For precise statements see Theorem 3.11 and Theorem 3.12 below. In Section 4.3 we measure empirically the gain from this smoother composition of zCDP.

Definition 3.7 (Zero-Concentrated Differential Privacy (zCDP) Bun & Steinke (2016))**.** An algorithm $M : \mathcal{D}^n \to \mathbb{R}$ is ρ-zCDP if for all neighbouring $X, X' \in \mathcal{D}^n$, and $\alpha \in (1, \infty)$ $\mathrm{RD}_\alpha(M(Z), M(Z')) \leq \rho\alpha$, where RD_α is the α-Rényi [5] divergence between random variables A and B. ($\mathcal{D}$ is the domain of the database elements, in our case it is (a,b).)

The following lemma shows that DP implies zCDP.

Lemma 3.8. *(Bun & Steinke, 2016) if algorithm M satisfies ε-DP, then M satisfies ρ-zCDP with $\rho = \frac{\varepsilon^2}{2}$.*

[5] the Rényi divergence of order $\alpha \in (1, \infty)$ between two probability distributions on P and Q over ω is $\mathrm{RD}_\alpha := \frac{1}{1-\alpha} \log(\int_\omega P(x)^\alpha Q(x)^{1-\alpha}\, dx)$

The above lemma is generic in that it applies to *every* mechanism that satisfies differential privacy. Better bounds are known for specific cases. In particular, we will use the following lemma.

Lemma 3.9. *(Cesar & Rogers, 2021) The exponential mechanism with parameter ε satisfies $\frac{\varepsilon^2}{8}$-zCDP.*

Lemma 3.10. *(Bun & Steinke, 2016) Let $M : \mathcal{D}^n \to \mathcal{Y}$ and $M' : \mathcal{D}^n \to \mathcal{Z}$. Suppose M satisfies ρ-zCDP and M' satisfies ρ'-zCDP. Define $M'' : \mathcal{D}^n \to \mathcal{Z}$ by $M''(X) = M'(X, M(X))$. Then M'' satisfies $(\rho + \rho')$-zCDP.*

Theorem 3.11 (General Quantiles Utility with zCDP)**.** *Suppose we implement $\mathcal{A}$ using the exponential mechanism to solve the single quantile problem, with privacy parameter $\varepsilon = \sqrt{\frac{8\rho}{\log m+1}}$. Then, given data X and quantiles $Q = (q_1, \ldots, q_m)$, the AQ algorithm is ρ-zCDP and with probability $1 - \beta$ output $V = (v_1, \ldots, v_m)$ that satisfies*

$$O\left(\frac{\log^{1.5} m}{\sqrt{\rho}}(\log\psi + \log m + \log\frac{1}{\beta})\right),$$

where $\psi = \frac{b-a}{\min_{k\in[n+1]}\{x_k - x_{k-1}\}}$.

Proof. By Lemma 3.1, the computation at each recursive level $1 \le i \le \log m + 1$ is $\sqrt{\frac{8\rho}{\log m+1}}$-DP, and therefore, by Lemma 3.9, also $(\frac{\rho}{\log m+1})$-zCDP. Since the number of levels is $\log m + 1$, by Lemma 3.10, the AQ algorithm is ρ-zCDP. The error bound follows exactly as in the proof of Theorem 3.3. □

The following theorem is analogous to Theorem 3.5. Its privacy proof is as for Theorem 3.11 and the error analysis is as in the proof of Theorem 3.5.

Theorem 3.12 (Uniform Quantiles Utility with zCDP)**.** *Suppose we implement $\mathcal{A}$ using the exponential mechanism to solve the single quantile problem, with privacy parameter $\sqrt{\frac{8\rho}{\log m+1}}$. Then, given data X and quantiles $Q = (q_1, \ldots, q_m)$, where $q_i = \frac{i}{m+1}$, the AQ algorithm is ρ-zCDP, and with probability $1 - \beta$ output $V = (v_1, \ldots, v_m)$ that satisfies*

$$\mathrm{Err}_X(O, V) = O\left(\sqrt{\frac{\log m}{\rho}}(\log\psi + \log m + \log\frac{1}{\beta})\right),$$

where $\psi = \frac{b-a}{\min_{k\in[n+1]}\{x_k - x_{k-1}\}}$.

4. Experiments

We implemented the AQ algorithm in Python and its code is publicly available on GitHub[6]. We used the exponential mechanism for the DP single quantile algorithm $\mathcal{A}$, with $-\mathrm{Gap}_X(o, v)$ as the utility of a solution v, where o is a true quantile, see Appendix A. We also experimented with the AQ-zCDP algorithm, a version of our algorithm that is private with respect to the definition of zero-concentrated differential privacy (zCDP) (Section 3.3). We compared our algorithms to the three best performing algorithms from Gillenwater et al. (2021) called: (1) JointExp (2) AppindExp and (3) AggTree. We ran the implementations provided by Gillenwater et al. (2021). We describe these baseline algorithms in Section 4.1. We tested the algorithms using two synthetic datasets and four real datasets that are described in Section 4.2. For each dataset we compared the accuracy (Section 4.3) and runtime (Section 4.4) of the competing algorithms.

4.1. Baseline Algorithms

JointExp Gillenwater et al. (2021) solves the DP quantiles problem by an efficient implementation of the exponential mechanism (Definition 2.4) on m-tuples $V = (v_1, \ldots, v_m) \in (a, b)^m$, where the utility of V is defined as follows:

$$u(X, V) = -\sum_{j\in[m+1]} |\mathrm{Gap}_X(o_j, o_{j-1}) - \mathrm{Gap}_X(v_j, v_{j-1})\}|,$$

where we define $v_0 = o_0 = a$ and $v_{m+1} = o_{m+1} = b$. The naive implementation of the exponential mechanism with this utility function is computationally difficult: The number of m tuples V is infinite, and there may even be exponentially many (in m) equivalence classes of such m-tuples. Gillenwater et al. (2021) give an $O(mn \log n + m^2 n)$ time algorithm to sample from the distribution defined by the exponential mechanism. The experiments of Gillenwater et al. (2021) show that when the number of quantiles, m, is small the JointExp algorithm performs best.

AppindExp solves the DP quantiles problem by applying the exponential mechanism as described in Appendix A to find every quantile q_i separately. Since AppindExp applies the exponential mechanism m times, if we use ε/m as the privacy parameter for each application of the exponential mechanism, then by composition we get that AppindExp is ε-DP. The advanced composition theorem Dwork et al. (2010b) shows that if we use $\varepsilon' \approx \varepsilon/\sqrt{m\log(1/\delta)}$ for each application of the exponential mechanism then the overall algorithm would be (ε, δ)-DP. The implementation of Gillenwater et al. (2021) uses a tighter advanced composition theorem specific for nonadaptive applications of the exponential mechanism (Dong et al., 2020), to determine an ε' for each quatile computation such that the overall composition of the m applications is (ε, δ)-DP. We use $n = 1000$ data points in our experiments, so we chose

[6] https://github.com/ShacharSchnapp/DP_AQ

$\delta = 10^{-6}$ in accordance with the common practice that $\delta \ll \frac{1}{n}$.

AggTree Dwork et al. (2010a) and Chan et al. (2011) implement an ε-DP tree-based counting algorithm for CDF estimation. Given a domain (a, b) the algorithm builds a balanced tree T with branching factor b and height h, so T has b^h leaves. The j'th leaf of the tree is associated with sub-domain $[c_{j-1}, c_j]$ where $c_j := a + j(b-a)/b^h$. Given a dataset $D \in (a, b)^n$, the algorithm starts by counting the number of elements from the dataset that fall into each leaf (i.e. are contained in its sub-domain). Each internal node of T is associated with the sum of the counters of its children (which equals to the number of elements in the leaves of its subtree). In particular, the count associated with the root is n. Since each element in the data contributes to at most h nodes (the path from the leaf containing it to the root), it suffice to add $\mathrm{Lap}(h/\varepsilon)$ noise to the value of each node to make the counts of T ϵ-DP. We can approximate any quantile q using this data structure as follows. We find the leftmost leaf z such that the sum of the noisy counts of all leaves to the left of z (including z) is at least $q \cdot n$. In particular, if the counts were not noisy that z would contain a qth quantile. Let $c(z)$ be the noisy count of z and let $c^-(z)$ be the sum of the noisy counts of the leaves to the left of z. Let $p = (qn - c^-(z))/c(z)$. Without noise p would have been the approximate relative quantile of the qth quantile among the elements in z. Let $[c_{k-1}, c_k]$ be the range associated with z. We approxmate the qth quantile using linear interpolation inside $[c_{k-1}, c_k]$. That is we return $(1-p)c_{k-1} + pc_k$. We utilize the AggTree implementation provided by Gillenwater et al. (2021), we used $b = 12$ and $h = 4$ those values came from the exhaustive hyperparameters search of Gillenwater et al. (2021) (see their Appendix E) and the results are given in Section 4.3.

4.2. Datasets

We tested our four algorithms on six datasets. Two data sets are synthetic. One contains independent samples from the uniform distribution $U(-5, 5)$, and the other contains independent samples from the Gaussian $N(0, 5)$. Two real continuous datasets from Goodreads (2019), one contains book ratings and the other contains books' page counts. Last we have two categorial datasets from the adults' census data (Dua & Graff, 2019). One contains working hours per week and the other ages of different persons. Table 1 shows the properties of each dataset, and Figure 3 shows the histograms of 100 equal-width bins for each dataset.

4.3. Empirical Error Analysis

We compare the error of Approximate Quantiles and the baseline algorithms. Our error metric is the average gap of

Figure 3: Histograms of 100 equal-width bins.

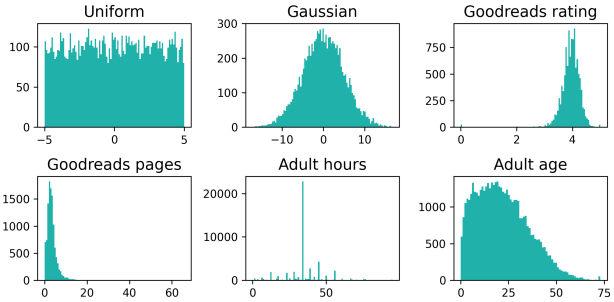


Table 1: Data set properties.

Data set	Size	Data Characteristics
Uniform (synthetic)	10000	Continuous
Gaussian (synthetic)	10000	Continuous
Goodreads rating	11123	Continuous
Goodreads pages	11123	Continuous
Adult hours	48842	96 Categories
Adult age	48842	74 Categories

the approximate quantiles $V = (v_1, \ldots, v_m)$ and and the true ones $O = (o_1, \ldots, o_m)$: $\frac{\sum_{j\in[m]} \mathrm{Gap}_X(o_j, v_j)}{m}$.

DP Error Analysis: We randomly chose 1000 samples from each dataset and checked the error of each algorithm with $m = 1$ to $m = 120$ uniform quantiles and $\mathcal{D} = [-100, 100]$ as a (loose) user-provided domain. We used the privacy parameter $\varepsilon = 1$. This process was repeated 100 times. Figure 4 shows the average of the error across the 100 iterations. Figure 5 zooms in on the error for $m = 1, \ldots, 35$ quantiles. Approximate Quantiles performs better than the baselines almost in all experiments, except for a few small values of m where the performance of JointExp was slightly better. As the number of quantiles increases Approximate Quantiles wins by a larger margin.

zCDP Error Analysis: As in previous experiments we randomly chose 1000 samples from each dataset and checked the error of each algorithm for $m = 1, \ldots, 120$ uniform quantiles and $\mathcal{D} = [-100, 100]$ as a (loose) user-provided domain. All algorithms were ρ-zCDP for $\rho = \frac{1}{8}$. For this we used $\varepsilon' = 1/\sqrt{m}$ in each application of the exponential mechanism by AppindExp, and a Laplace noise of magnitude $\varepsilon' = \sqrt{h}$ in each node of the tree computed by AggTree. In each application of the exponential mechanism by Approximate Quantiles we used $\varepsilon' = \sqrt{1/(\log m + 1)}$ as described in Theorem 3.12. The algorithm **JointExp**

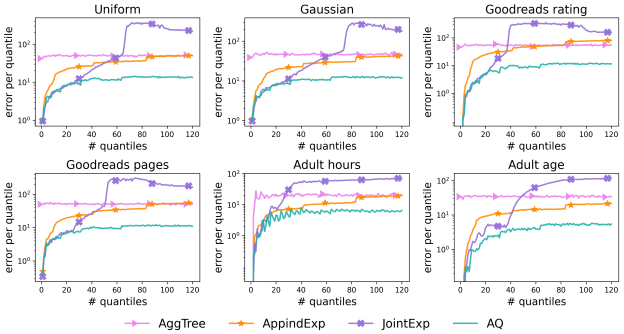


Figure 4: The $x-$axis shows the number of quantiles and the $y-$axis shows the gap per quantile averaged over 100 trials with $\varepsilon = 1$. Note that the graphs are in log scale.

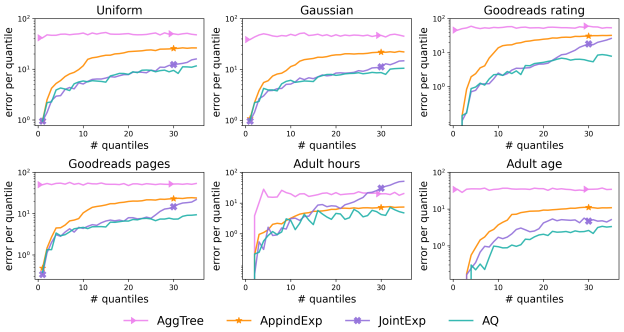


Figure 5: The $x-$axis shows the number of quantiles and the $y-$axis shows the gap per quantile averaged over 100 trials with $\varepsilon = 1$. Note that the graphs are in log scale.

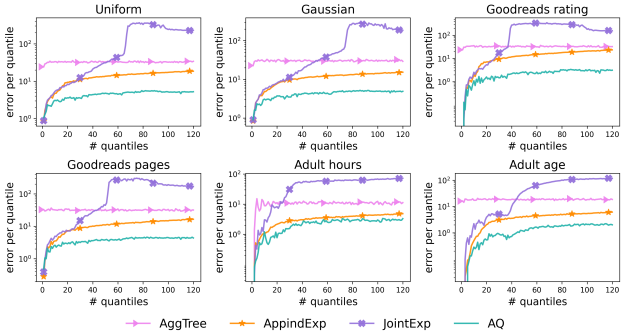


Figure 6: zCDP: The $x-$axis shows the number of quantiles and the $y-$axis shows the gap per quantile averaged over 100 trials with $\rho = 1/8$. Note that the graphs are in log scale.

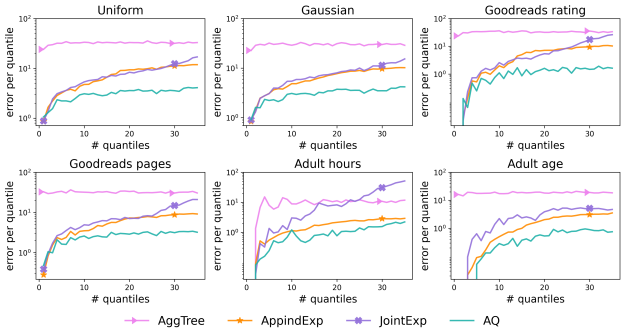


Figure 7: zCDP: The $x-$axis shows the number of quantiles and the $y-$axis shows the gap per quantile averaged over 100 trials with $\rho = 1/8$. Note that the graphs are in log scale.

with $\varepsilon = 1$ is $\frac{1}{8}$-zCDP by Lemma 3.10, so its error is the same as in the previous experiment. Figure 6 shows the average of the error for z-CDP across the 100 iterations. Figure 7 zooms in on the error for $m = 1, \ldots, 35$. Approximate Quantiles performs much better than the baselines even for small number of quantiles. When the number of quantiles is large, the error of AQ algorithm reaches up to 7.14 time lower than the best existing baseline.

4.4. Time Complexity Experiments

Given a sorted dataset, it takes $O(n)$ time to find all quantiles at a single level of the recursion tree of Approximate Quantiles. Therefore the overall time complexity (*i.e.,* without the sort) of the AQ algorithm is $O(n \log m)$, where m is the number of quantiles. In comparison, the baseline algorithms are computationally more expensive: JointExp algorithm runs in $O(mn \log n + m^2 n)$ time, AppindExp algorithm runs in $O(mn)$ time and AggTree algorithm runs in $O(n \log n)$ time. We empirically compared the running time of Approximate Quantiles to the running times of the baseline algorithms. For each dataset we measured the time required to find $m \in [120]$ quantiles in a sub-sample of size 1000 of each dataset, averaged over 100 trials per dataset. Figure 8 shows the average running time across all datasets (Section 4.2), each experiment used one core of an Intel i9-9900K processor. We see that the running time of Approximate Quantiles is about ten times smaller than of AggTree and at least a 100 smaller than of JointExp and AppindExp .

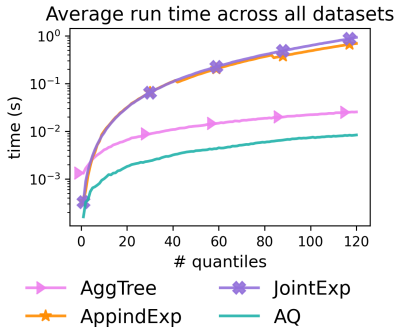


Figure 8: $x-$axis number of quantiles, $y-$axis is the run time averaged across all datasets, each dataset averaged across 100 trials. Note that the graphs are in log scale.

Acknowledgements

Haim Kaplan and Uri Stemmer are partially supported by the Israel Science Foundation (grants 1595/19 and 1871/19) and by Len Blavatnik and the Blavatnik Family foundation. Shachar Schnapp is partially supported by the Cyber Security Research Center at Ben-Gurion University of the Negev.

References

Beimel, A., Nissim, K., and Stemmer, U. Private learning and sanitization: Pure vs. approximate differential privacy. *Theory Comput.*, 12(1):1–61, 2016.

Bun, M. and Steinke, T. Concentrated differential privacy: Simplifications, extensions, and lower bounds. In *Theory of Cryptography Conference*, pp. 635–658. Springer, 2016.

Bun, M., Nissim, K., Stemmer, U., and Vadhan, S. Differentially private release and learning of threshold functions. In *2015 IEEE 56th Annual Symposium on Foundations of Computer Science*, pp. 634–649. IEEE, 2015.

Cesar, M. and Rogers, R. Bounding, concentrating, and truncating: Unifying privacy loss composition for data analytics. In *ALT*, volume 132 of *Proceedings of Machine Learning Research*, pp. 421–457. PMLR, 2021.

Chan, T.-H. H., Shi, E., and Song, D. Private and continual release of statistics. *ACM Transactions on Information and System Security (TISSEC)*, 14(3):1–24, 2011.

Dong, J., Durfee, D., and Rogers, R. Optimal differential privacy composition for exponential mechanisms. In *International Conference on Machine Learning*, pp. 2597–2606. PMLR, 2020.

Dua, D. and Graff, C. UCI machine learning repository, 2019. URL http://archive.ics.uci.edu/ml. Date accessed: 2021-07-01.

Dwork, C. and Roth, A. The algorithmic foundations of differential privacy. *Foundations and Trends in Theoretical Computer Science*, 9(3-4):211–407, 2014.

Dwork, C., Kenthapadi, K., McSherry, F., Mironov, I., and Naor, M. Our data, ourselves: Privacy via distributed noise generation. In *EUROCRYPT*, volume 4004 of *Lecture Notes in Computer Science*, pp. 486–503. Springer, 2006a.

Dwork, C., McSherry, F., Nissim, K., and Smith, A. Calibrating noise to sensitivity in private data analysis. In *Theory of cryptography conference*, pp. 265–284. Springer, 2006b.

Dwork, C., Naor, M., Pitassi, T., and Rothblum, G. N. Differential privacy under continual observation. In *Proceedings of the forty-second ACM symposium on Theory of computing*, pp. 715–724, 2010a.

Dwork, C., Rothblum, G. N., and Vadhan, S. Boosting and differential privacy. In *2010 IEEE 51st Annual Symposium on Foundations of Computer Science*, pp. 51–60, 2010b.

Gillenwater, J., Joseph, M., and Kulesza, A. Differentially private quantiles. In *ICML*, volume 139 of *Proceedings of Machine Learning Research*, pp. 3713–3722. PMLR, 2021.

Goodreads. Goodreads-books dataset, 2019. URL https://www.kaggle.com/jealousleopard/. Date accessed: 2021-07-01.

GRE. Ets. gre guide to the use of scores., 2021. URL https://www.ets.org/s/gre/pdf/gre_guide.pdf. Date accessed: 2021-07-19.

Kaplan, H., Ligett, K., Mansour, Y., Naor, M., and Stemmer, U. Privately learning thresholds: Closing the exponential gap. In *Conference on Learning Theory*, pp. 2263–2285. PMLR, 2020.

McSherry, F. and Talwar, K. Mechanism design via differential privacy. In *48th Annual IEEE Symposium on Foundations of Computer Science (FOCS'07)*, pp. 94–103. IEEE, 2007.

Smith, A. Privacy-preserving statistical estimation with optimal convergence rates. In *Proceedings of the forty-third annual ACM symposium on Theory of computing*, pp. 813–822, 2011.

A. DP Single quantile

In the DP single quantile problem, the input is a single quantile q and a database $X \in (a,b)^n$. The output is a quantile estimate $v \in (a,b)$ such that $\Pr_{x\sim U_X}[x \le v] \approx q$ (in a sense that Lemma A.1 would make precise). We solve this problem using the exponential mechanism of (McSherry & Talwar, 2007) on (a,b) with the utility function:

$$u(X,w) = -\left|\left|\{x \in X | x < w\}\right| - \lfloor q \cdot n \rfloor\right| = -\mathrm{Gap}_X(o,w),$$

where o is a q-quantile and $w \in (a,b)$. This mechanism samples each point $w \in (a,b)$ to be the output with density proportional to $\exp(\varepsilon u(X,w)/2)$. Note that the sensitivity of u is 1, that is $\max\{|u(X,\omega) - u(X',\omega)|\} \le 1$, where the maximum is over neighboring datasets X and X' and points $\omega \in (a,b)$. The largest utility is of a q-quantile and equals to 0. We can sample from this distribution using the technique given by (Smith, 2011) (see their Algorithm 2). The idea is to split the sampling process into two steps:

1. Let $I_k = [x_{k-1}, x_k]$, $k = 1, \ldots, n+1$ where $x_0 = a$, $x_{n+1} = b$, be the set of $n+1$ intervals between data points. We sample an interval from this set of intervals, where the probability of sampling I_k is proportional to

$$\Pr[\mathcal{A}(X) = I_k] \propto \exp\left(\frac{\varepsilon u(X, I_k)}{2}\right) \cdot (x_k - x_{k-1}),$$

   Note that all points in I_k have the same utility which we denote by $u(X, I_k)$.

2. Return a uniform random point from the sampled interval.

Lemma A.1. *Given dataset $X \in (a,b)^n$ and quantile $q \in [0,1]$, the exponential mechanism is ε-DP, and with probability $1-\beta$ outputs v that satisfies*

$$\mathrm{Gap}_X(o,v) \le 2 \cdot \frac{\log\psi - \log\beta}{\varepsilon},$$

where o is a true q-quantile and $\psi = \frac{b-a}{\min_{k\in[n+1]}\{x_k - x_{k-1}\}}$.

Proof. ε-DP follows in a straightforward way by bounding the ratio of the densities of a point w in the destribution defined by X and in the distribution defined by X'; See also (Dwork & Roth, 2014).

Let I_t be an interval such that $u(X, I_t) \le -\gamma$. It follows that the probability of sampling a point from I_t is at most

$$\Pr[\mathcal{A}(X) = I_t] \le \frac{\exp\left(\frac{-\varepsilon\gamma}{2}\right) \cdot (x_t - x_{t-1})}{\sum_{k\in[n+1]} \exp\left(\frac{\varepsilon u(X,I_k)}{2}\right) \cdot (x_k - x_{k-1})}.$$

Using the union bound we get that:

$$\begin{aligned}
\Pr[\mathcal{A}(X) \le -\gamma] &\le \frac{\exp\left(\frac{-\varepsilon\gamma}{2}\right)(b-a)}{\sum_{k\in[n+1]} \exp\left(\frac{\varepsilon u(X,I_k)}{2}\right) \cdot (x_k - x_{k-1})} \\
&\le \frac{\exp\left(\frac{-\varepsilon\gamma}{2}\right)(b-a)}{\exp\left(\frac{\varepsilon u(X,I_o)}{2}\right)(x_o - x_{o-1})} \\
&\le \frac{b-a}{\min_{k\in[n+1]}(x_k - x_{k-1})} \cdot \exp\left(\frac{-\varepsilon\gamma}{2}\right) \\
&= \psi \exp\left(\frac{-\varepsilon\gamma}{2}\right)
\end{aligned}$$

where I_o is the interval containing the q-quantiles so $u(X, I_0) = 0$. It follows that with probability less than β, we sample an interval whose utility is at most $-\gamma$ for

$$\gamma = 2\Delta_u \cdot \frac{\log\psi - \log\beta}{\varepsilon}.$$

Since in the second step we sample the q-quantile v uniformly from the interval selected in the first step, we get that with probability $1 - \beta$ the output v satisfies:

$$\mathrm{Gap}_X(o, v) \leq 2 \cdot \frac{\log \psi - \log \beta}{\varepsilon}.$$

□