Multiplicative Update Methods for Incremental Quantile Estimation

Anis Yazidi and Hugo Hammer
Dept. of Computer Science
Oslo and Akershus University
College of Applied Sciences
Oslo, Norway
Email:anis.yazidi@hioa.no, hugo.hammer@hioa.no

Abstract—**In this paper we focus on the estimation of quantiles when samples arrive sequentially. With time, the amount of data will become large and classical quantile estimators that require storing the whole history of the data (or stream) cannot be deployed. A plausible alternative is to rely on incremental estimators. An incremental estimator utilizes the previously-computed estimates and *only* resorts to the last sample for updating these estimates.**

The state-of-the-art work on obtaining incremental quantile estimators is due to Tierney [1], and is based on the theory of stochastic approximation. The estimator is constructed to efficiently estimate quantiles in a system in which the underlying distribution of the samples does not vary with time. However, from a practical point of view this is rarely the case and variants of the Tierney's estimator have been suggested to cope with dynamic environments, see [2]–[5]. A primary shortcoming of all of these estimators is the requirement to incrementally build local approximations of the distribution function in the neighborhood of the quantiles. This requirement, unfortunately, increases the complexity of these algorithms significantly and additionally renders them vulnerable to numerical issues.

We present two novel lightweight incremental quantile estimators which possess far less complexity than the Tierney [1] estimator and its extensions [2]–[5]. Notably, our algorithms rely only on tuning one single parameter which is a plausible property which we could only find in the discretized quantile estimator Frugal [6]. This makes our algorithms easy tune to perform well. Furthermore, our algorithms are multiplicative which makes them highly suitable to handle quantile estimation in systems in which the underlying distribution varies with time.

The convergence of the two proposed estimators is proven using the theory of stochastic learning. Extensive experimental results show that while our first estimator yields comparable performance to the legacy estimators, our second estimator clearly outperforms the state-of-the-art incremental estimators.

Index Terms—**Quantiles estimation, Time Varying Distributions, Multiplicative Updates.**

I. Introduction

An incremental estimator, by definition, resorts to the last observation(s) in order to update its estimate [7]–[9]. Surprisingly enough, the research on developing incremental quantile estimators is sparse. Probably, one of the outstanding early and unique examples of incremental quantile estimators is due to Tierney, proposed in 1983 [1], and which resorted to the theory of stochastic approximation. Some extensions of the seminal work of Tierney [1] can be found in [2]–[4]. Applications of Tierney's algorithm to network monitoring can be found in [5]. In order to appreciate the qualities of our estimator, we will present the estimator scheme proposed by Tierney [1]. Let $x(n)$ denote a realization of a stochastic variable X at time 'n'. We assume that X is distributed according to the distribution $f_X(x)$. The intention of the exercise is to estimate the q-th quantile, the number Q_q, such that $F_X(Q_q) = q$. Tierney [1] achieved this by maintaining a running estimate $\widehat{Q_q}(n)$ at time 'n'

$$\widehat{Q_q}(n+1) = \widehat{Q_q}(n) + \frac{d_n}{n+1}(q - I(x(n) \leq \widehat{Q_q}(n))) \quad (1)$$

where $d_n = min(\frac{1}{\widehat{f_n(Q_q)}}, d_0 n^a)$. Here $0 < a < 1/2$, $d_o > 0$, and $\widehat{f_n(Q_q)}$ is an estimator of $f(Q_q)$ defined in [1]. The reason for invoking the min operation in the above expression of d_n is the fact that the estimated density must be bounded to prevent the correction factor from "exploding". In other words, f_n is the current estimate of the density of X at the q-th quantile. This is usually done based on maintaining a histogram structure. However, requiring the incremental constructions of local approximations of the distribution function in the neighborhood of the quantiles increases the complexity of the algorithm. Our goal is to present a new family of incremental quantile estimators that does not involve any local approximations of the distribution function.

Another intriguing algorithm is called Frugal [6] that achieves estimation using exactly the same complexity as our two algorithms presented in this paper. Frugal bears similarity to our first algorithm presented here and that we reckoned RUMIQE in the sense that it performs randomized updates. However there are major difference between Frugal and DUMIQE:(1) First, Frugal operates in a discretized space while DUMIQE operates in a continuous-space of values. (2) Second, Frugal has an additive increase-decrease update form in contrast to DUMIQE which has a multiplicative increase-decrease flavor.

We shall first review some of the related work on estimating quantiles from data streams. However, as we will explain later, these related works require some memory restrictions which renders our work to be radically distinct from them. In fact, our approach requires storing only one sample value in order to update the estimate. The most representative work for this type of ”streaming” quantile estimator is due to the seminal work of Munro and Paterson [10]. In [10], Munro and Paterson described a p-pass algorithm for selection using $O(n^{1/(2p)})$ space for any $p \geq 2$. Cormode and Muthukrishnan [11] proposed a more space-efficient data structure, called the Count-Min sketch, which is inspired by Bloom filters, where one estimates the quantiles of a stream as the quantiles of a random sample of the input. The key idea is to maintain a random sample of an appropriate size to estimate the quantile, where the premise is to select a subset of elements whose quantile approximates the true quantile. From this perspective, the latter body of research requires a certain amount of memory that increases as the required accuracy of the estimator increases [12]. Examples of these works are [10], [12]–[15]. Guha and McGregor [15] advocate the use of random-order data models in contrast to adversarial-order models. They show that computing the median requires exponential number of passes in adversarial model while requiring $O(log\ logn)$ in random order model.

In [2], the authors proposed a modification of the stochastic approximation algorithm [1] in order to allow an update similar to the well-known Exponentially Weighted Moving Averages form for updates. This modification is particularly helpful in the case of non-stationary environments in order to cope with non-stationary data. Thus, the quantile estimate is a weighted combination of the new data that has arrived and the previously-computed estimate. Indeed, a ”weighted” update scheme is applied to incrementally build local approximations of the distribution function in the neighborhood of the quantiles.

In many network monitoring applications, quantiles are key indicators for monitoring the performance of the system. For instance, system administrators are interested in monitoring the 95% response time of a web-server so that to hold it under a certain threshold. Quantile tracking is also useful for detecting abnormal events and in intrusion detection systems in general. However, the immense traffic volume of high speed networks impose some computational challenges: little storage and the fact that the computation needs to be ”one pass” on the data. It is worth mentioning that the seminal paper of Robbins and Monro [16] which established the field of research called ”stochastic approximation” [17] have included an incremental quantile estimator as a proof of concept of the vast applications of the theory of stochastic approximation. An extension of the latter quantile estimator which first appeared as example in [16] was further developed in [18] in order to handle the case of ”extreme quantiles”. Moreover, the estimator provided by Tierney [1] falls under the same umbrella of the example given in [16], and thus can be seen as an extension of it.

As Arandjelovic remarks [19], most quantile estimation algorithms are not single-pass algorithms and thus are not applicable for streaming data. On the other hand, the single pass algorithms are concerned with the exact computation of the quantile and thus require a storage space of the order of the size of the data which is clearly an unfeasible condition in the context of big data stream.

Thus, we submit that all work on quantile estimation using more than one pass, or storage of the same order of the size of the observations seen so far is not relevant in the context of this paper.

When it comes to memory efficient methods that require a small storage footprint, histogram based methods form an important class. A representative work in this perspective is due to Schmeiser and Deutsch [20]. In fact, they proposed to use equidistant bins where the boundaries are adjusted online. Arandjelovic et al . [19] use a different idea than equidistant bins by attempting to maintain bins in a manner that maximizes the entropy of the corresponding estimate of the historical data distribution. Thus, the bin boundaries are adjusted in an online manner.

In [21], the authors propose a memory efficient method for simultaneous estimation of several quantiles using interpolation methods and a grid structure where each internal grid point is updated upon receiving an observation. The application of this approach is limited for stationary data. An approximation relies on using linear and parabolic interpolations, while the tails of the distribution are approximated using exponential curves. It is worth mentioning that the latter algorithm is based on the P^2 algorithm [22].

A notable work treating simultaneous estimation of the quantiles using elements from the theory of stochastic approximation is due to Cao et al . [4]. The authors resorted to interpolation by defining some type of distance between the interpolated quantiles so that to ensure no ”crossing” between the monotonic quantile estimates. Nevertheless, the interpolation uses ”the density” estimate as in [1] and in [2], which is an operation that increases the complexity.

In [22], Jain et al. resort to five markers so that to track the quantile, where the markers correspond to different quantiles and the min and max of the observations. Their concept is similar to the notion of histograms, where each marker has two measurements, its height and its position. By definition, each marker has some ideal position, where some adjustments are made so that to keep it in its ideal position by counting number of samples exceeding the marker. In simple terms, for example, if the marker corresponds to the 80% quantile, its ideal position will be around the point corresponding to 80% of the data points below the marker. However, such approach does not handle the case of non-stationary quantile estimation as the position of the markers will be affected by stale data points. Then based on the position of the markers, quantiles are computed by supposing that the curve passing through three adjacent markers is parabolic and using piecewise parabolic prediction function.

Finally, it is worth mentioning that an important research direction that has received little attention in the literature revolves around updating the quantile estimates under the assumption that portions of the data are deleted. Such assumption is realistic in many real life settings where data needs to be deleted due to the occurrence of errors, or because it is merely out-of-date and thus should be replaced. The deletion triggers a re-computation of the quantile [4], which is considered a complex operation. Note that the case of deleted data is more challenging than the case of insertion of new data. In fact, the insertion can be handled easily using either sequential or batch updates, while quantile update upon deletion requires more complex forms of updates.

A. Contributions

We catalogue the contributions of the paper as follows:

- We present two lightweight incremental quantile estimation schemes: the first scheme resorts to randomized updates while the second scheme is based on deterministic updates. Both algorithms are much simpler than the state-of-the-art algorithms [1]–[5] which require locally approximating the distribution function in the neighborhood of the quantile which results in an increased complexity.
- To the best of our knowledge, our randomized and deterministic estimators are the first reported incremental *multiplicative* increase-decrease quantile estimator in the literature as opposed to the legacy additive increase-decrease algorithms [1]–[5]. By virtue of the multiplicative updates, the quantile estimate can be adjusted in a ”geometric” manner yielding fast convergence speed.
- Since the algorithms are multiplicative, in the base versions of our two schemes (randomized and deterministic update), the quantile estimate stays positive (negative) if it initially is positive (negative). In order to cope with the case of changing the sign of the estimate while performing the estimation[1], we extend both estimators using two different approaches:
  - We merge the update operations designed for positive quantiles together with the counterpart operations for updating negative quantile so that to build an estimator that is able to estimate any quantile, and most importantly, a one that is able to change the sign of the quantile estimate in an online-manner.
  - The second approach exploits a different idea in which we reckon with so-called phantom quantiles. The idea is based on the concept inspired by simple observation that $Prob(X \leq a) = Prob(X + b \leq a + b)$ for any real numbers a and b.
- Experimental results show the performance of the schemes and their comparable performance to the state-of-the-art.

[1]The case of changing the sign of the estimate might occur in dynamic environment where the true quantile might drift from positive values to negative values and vice-versa.

B. Paper Organization

In Section I, we both presented the motivation behind our study on incremental quantile estimation, and surveyed the relevant state-of-the-art. In Sections II and III, we give the details of our randomized and deterministic update schemes, respectively. In Section IV we present the two approaches to be able to estimate any quantile (both positive and negative). Finally in Section V we compared the algorithms with state-of-the-art quantile estimators.

II. Randomized Update Based Multiplicative Incremental Quantile Estimator (RUMIQE)

We start by presenting the incremental quantile estimator which is based on randomization. Let X denote a stochastic variable with distribution $f_X(x)$ and further let $x(n)$ be a concrete realization of X at time $'n'$. The intention of the exercise is to estimate the q-th quantile, which is the number Q_q such that $P(X < Q_q) = F_X(Q_q) = q$. We achieve this by maintaining a running estimate $\widehat{Q_q}(n)$ at time $'n'$. We omit the reference to time $'n'$ in $\widehat{Q_q}(n)$ whenever there is no confusion. $\widehat{Q_q}$ is initialized to $\widehat{Q_q}(0)$ such that $\widehat{Q_q}(0) > 0$.

$\widehat{Q_q}(n)$ is updated as per the following simple rule:

$$\widehat{Q_q}(n+1) \leftarrow (1+\lambda)\widehat{Q_q}(n) \quad \text{if } \widehat{Q_q}(n) < x(n) \text{ and } rand() \leq q \tag{2}$$

$$\widehat{Q_q}(n+1) \leftarrow (1-\lambda)\widehat{Q_q}(n) \quad \text{if } \widehat{Q_q}(n) \geq x(n) \text{ and } rand() \leq (1-q) \tag{3}$$

$$\widehat{Q_q}(n+1) \leftarrow \widehat{Q_q}(n) \quad \text{else} \tag{4}$$

where $rand()$ is a random number in $[0, 1]$ and $0 < \lambda < 1$. Note that since the update scheme is multiplicative and $\widehat{Q_q}(0) > 0$, the estimator will stay positive. In Section IV we present modifications of the scheme such that the estimator can take both positive and negative values.

Now we will present a theorem that catalogues the properties of the estimator for $Q_q > 0$. A sufficient condition to obtain $Q_q > 0$ is that the random variable X only takes positive values The proofs of the theoretical results in this paper is based on the theory of stochastic learning due to Norman [23].

Theorem 1. *Let $Q_q = {F_X}^{-1}(q)$ be the true quantile to be estimated and suppose that $Q_q > 0$. In addition, we suppose that $\widehat{Q_q}(0) > 0$. Applying the updating rules (2) to (4), we obtain*

$$\lim_{n\lambda\to\infty,\lambda\to 0} \widehat{Q_q}(n) = Q_q$$

We will first present a theorem due to Norman [23] that will be used for our proof. Norman [23] studied distance "diminishing models". The convergence of $\widehat{Q_q}(n)$ to Q_q is a consequence of this theorem.

Theorem 2. *Let $x(t)$ be a stationary Markov process dependent on a constant parameter $\theta \in [0,1]$. Each $x(t) \in I$, where I is a subset of the real line. Let $\delta x(t) = x(t+1) - x(t)$. The following are assumed to hold:*

1) *I is compact.*
2) *$E[\delta x(t)|x(t) = y] = \theta w(y) + O(\theta^2)$*
3) *$Var[\delta x(t)|x(t) = y] = \theta^2 s(y) + o(\theta^2)$*
4) *$E[\delta x(t)^3|x(t) = y] = O(\theta^3)$ where $sup_{y\in I}\frac{O(\theta^k)}{\theta^k} < \infty$ for $K = 2,3$ and $sup_{y\in I}\frac{o(\theta^2)}{\theta^2} \to 0$ as $\theta \to 0$.*
5) *$w(y)$ has a Lipschitz derivative in I.*
6) *$s(y)$ is Lipschitz I.*

If Assumptions (1)-(6) hold, $w(y)$ has a unique root y^* in I and $\left.\frac{dw}{dy}\right|_{y=y^*} \leq 0$ then

1) *$var[\delta x(t)|x(0) = x] = 0(\theta)$ uniformly for all $x \in I$ and $t \geq 0$. For any $x \in I$, the differential equation $\frac{dy(\tau)}{d\tau} = w(y(t))$ has a unique solution $y(\tau) = y(\tau, x)$ with $y(0) = x$ and $E[\delta x(t)|x(0) = x] = y(t\theta) + O(\theta)$ uniformly for all $x \in I$ and $t \geq 0$.*
2) *$\frac{x(t)-y(t\theta)}{\sqrt{\theta}}$ has a normal distribution with zero mean and finite variance as $\theta \to 0$ and $t\theta \to \infty$.*

Having presented Theorem 2, now we are ready to proof Theorem 1 which is the main result of this paper by resorting to Theorem 2.

Proof. Let $\delta\widehat{Q_q}(n) = \widehat{Q_q}(n+1) - \widehat{Q_q}(n)$.

$$\begin{aligned}
E[\delta\widehat{Q_q}(n)|\widehat{Q_q} = \widehat{Q_q}(n)] &= \lambda\widehat{Q_q}(n)qProb(\widehat{Q_q}(n) < X) \\
&\quad - \lambda\widehat{Q_q}(n)(1-q) \\
&\quad Prob(\widehat{Q_q}(n) \geq X) \\
&= \lambda\widehat{Q_q}(n)q(1 - Prob(X \leq \widehat{Q_q}(n)) \\
&\quad - \lambda\widehat{Q_q}(n)(1-q) \\
&\quad Prob(X \leq \widehat{Q_q}(n)) \\
&= \lambda\widehat{Q_q}(n)(q(1 - F_X(\widehat{Q_q}(n)) \\
&\quad - (1-q)F_X(\widehat{Q_q}(n)) \\
&= \lambda\widehat{Q_q}(n)(q - F_X(\widehat{Q_q}(n)) \qquad (5)
\end{aligned}$$

Let $w(\widehat{Q_q}(n)) = \widehat{Q_q}(n)(q - F_X(\widehat{Q_q}(n))$ and $\Delta\widehat{Q_q}(n+1) = E[\delta Q_q(n)|\widehat{Q_q} = \widehat{Q_q}(n)]$

We then get

$$\begin{aligned}
Var[\delta\widehat{Q_q}(n)|\widehat{Q_q} = \widehat{Q_q}(n)] &= E[\delta\widehat{Q_q}(n)^2|\widehat{Q_q} = \widehat{Q_q}(n)] \\
&\quad - \Delta\widehat{Q_q}(n)^2
\end{aligned}$$

or

$$\begin{aligned}
E[\delta\widehat{Q_q}(n)^2|\widehat{Q_q} = \widehat{Q_q}(n)] &= \lambda^2\widehat{Q_q}(n)^2qProb(\widehat{Q_q}(n) < X) \\
&\quad - \lambda^2\widehat{Q_q}(n)^2(1-q) \\
&\quad Prob(\widehat{Q_q}(n) \geq X) \\
&= \lambda^2\widehat{Q_q}(n)^2(q - F_X(\widehat{Q_q}(n)) \qquad (6)
\end{aligned}$$

Therefore, we can obtain

$$\begin{aligned}
Var[\delta\widehat{Q_q}(n)|\widehat{Q_q} = \widehat{Q_q}(n)] &= \lambda^2\widehat{Q_q}(n)^2(q - F_X(\widehat{Q_q}(n)) \\
&\quad - (\lambda\widehat{Q_q}(n)(q - F_X(\widehat{Q_q}(n)))^2 \\
&= \lambda^2\widehat{Q_q}(n)^2(q - F_X(\widehat{Q_q}(n)) \\
&\quad (1 - q + F_X(\widehat{Q_q}(n)) \qquad (7)
\end{aligned}$$

Let $s(\widehat{Q_q}(n)) = \widehat{Q_q}(n)^2(q - F_X(\widehat{Q_q}(n))(1 - q + F_X(\widehat{Q_q}(n))$.

We then get

$$\begin{aligned}
E[\widehat{Q_q}(n)^3|\widehat{Q_q} = \widehat{Q_q}(n)] &= \lambda^3\widehat{Q_q}(n)^3q(1 - F_X(\widehat{Q_q}(n))) \\
&\quad - \lambda^3\widehat{Q_q}(n)^3(1-q)F_X(\widehat{Q_q}(n)) \\
&= \lambda^3\widehat{Q_q}(n)^3(q - F_X(\widehat{Q_q}(n)) \qquad (8)
\end{aligned}$$

We will use the results of Norman to prove the convergence.

$$w(\widehat{Q_q}(n)) = \widehat{Q_q}(n)(q - F_X(\widehat{Q_q}(n))) \qquad (9)$$

It is easy to see that $w(\widehat{Q_q}(n))$ admits a to roots $Q_q = {F_X}^{-1}(q)$ and $Q_q = 0$. By introducing an arbitrarily small lower bound $Q_{min} > 0$ on estimate $\widehat{Q_q}(n)$, we can avoid the $Q_q = 0$. This is easily implemented by modifying the update rules and adding Q_{min} to the right term of equations (2) and (3). Therefore the unique root becomes $Q_q = {F_X}^{-1}(q)$.

Let us consider $\frac{dw(\widehat{Q_q})}{d\widehat{Q_q}}$

$$\frac{dw(\widehat{Q_q})}{d\widehat{Q_q}} = q - F_X(\widehat{Q_q}) - \widehat{Q_q}f_x(\widehat{Q_q}) \qquad (10)$$

We replace $\widehat{Q_q}$ by Q_q and get

$$\begin{aligned}
\left.\frac{w(\widehat{Q_q})}{d\widehat{Q_q}}\right|_{\widehat{Q_q}=Q_q} &= q - F_X(Q_q) - Q_qf_x(Q_q) \qquad (11) \\
&= 0 - Q_qf_x(Q_q) < 0. \qquad (12)
\end{aligned}$$

This gives $\lim_{n\lambda\to\infty,\lambda\to 0} E(\widehat{Q_q}) = Q_q + O(\lambda)$ and $Var(\widehat{Q_q}) = o(\lambda)$. Consequently $\lim_{n\lambda\to\infty,\lambda\to 0}\widehat{Q_q} = Q_q$. □

A. Estimating Negative Quantiles for the case of the RUMIQE algorithm

Now we will present a scheme for the randomized estimator (RUMIQE) when $Q_q < 0$. A sufficient condition to obtain $Q_q < 0$ is that the random variable X takes only negative values. To do this, we merely "invert" the sign of the update equation in the case of negative quantiles.

Then, the value of $\widehat{Q_q}(n)$ is updated as per the following simple rule:

$$\widehat{Q_q}(n+1) \leftarrow (1-\lambda)\widehat{Q_q}(n) \quad \text{if } \widehat{Q_q}(n) < x(n) \text{ and } rand() \leq q \tag{13}$$

$$\widehat{Q_q}(n+1) \leftarrow (1+\lambda)\widehat{Q_q}(n) \quad \text{if } \widehat{Q_q}(n) \geq x(n) \text{ and } rand() \leq (1-q) \tag{14}$$

$$\widehat{Q_q}(n+1) \leftarrow \widehat{Q_q}(n) \quad \text{else} \tag{15}$$

The convergence of the estimator to the true quantile based on this rule can be proven in exactly in the same way as Theorem 1 and thus the proof is omitted for the sake of brevity.

III. Deterministic Update Based Multiplicative Incremental Quantile Estimator (DUMIQE)

In this section, we present our second estimation algorithm which is the deterministic counter-part to the algorithm proposed in Section II in the sense that it does not involve the concept of randomization and rather implements deterministic updates.

Recall that we let $x(n)$ denote a realization of a stochastic variable $X \sim f_X(x)$ at time 'n'. We still like to estimate the q-th quantile, Q_q. Instead of the update rules (2) to (4) we now suggest the following update rules:

$$\widehat{Q_q}(n+1) \leftarrow (1+\lambda q)\widehat{Q_q}(n) \quad \text{if } \widehat{Q_q}(n) < x(n) \tag{16}$$

$$\widehat{Q_q}(n+1) \leftarrow (1-\lambda(1-q))\widehat{Q_q}(n) \quad \text{if } \widehat{Q_q}(n) \geq x(n) \tag{17}$$

where $0 < \lambda < 1$. The intuitive behind this update rule compared to RUMIQUE is that instead of updating the estimator probabilistically q and $1-q$ portions of the times when new sample arrives, we rather update the quantile *every* time and appropriately adjust how much we update the estimator. The estimator based on this rule also converges to the true quantile according to the following theorem:

Theorem 3. *Let $Q_q = {F_X}^{-1}(q)$ be the true quantile to be estimated and suppose that $Q_q > 0$. In addition, we suppose that $\widehat{Q_q}(0) > 0$. Applying the updating rules (16) and (17), we obtain*

$$\lim_{n\lambda \to \infty, \lambda \to 0} \widehat{Q_q}(n) = Q_q$$

The proof of Theorem 3 is straightforward by following the same steps as the proof of theorem 1 and therefore is omitted here.

In the same manner as in Section II-A, negative quantiles can be estimated by simply "inverting" the sign of the updates:

$$\widehat{Q_q}(n+1) \leftarrow (1-\lambda q)\widehat{Q_q}(n) \quad \text{if } \widehat{Q_q}(n) < x(n) \tag{18}$$

$$\widehat{Q_q}(n+1) \leftarrow (1+\lambda(1-q))\widehat{Q_q}(n) \quad \text{if } \widehat{Q_q}(n) \geq x(n) \tag{19}$$

A. Remark regarding Frugal [6]

RUMIQE bears similarity to Frugal [6] in the sense that it uses randomization as inherent part of the update procedure [6].

According to Frugal [6], $\widehat{Q_q}(n)$ is updated as per the following simple rule:

$$\widehat{Q_q}(n+1) \leftarrow \widehat{Q_q}(n) + 1 \quad \text{if } \widehat{Q_q}(n) < x(n) \text{ and } rand() \leq q \tag{20}$$

$$\widehat{Q_q}(n+1) \leftarrow \widehat{Q_q}(n) - 1 \quad \text{if } \widehat{Q_q}(n) \geq x(n) \text{ and } rand() \leq (1-q) \tag{21}$$

$$\widehat{Q_q}(n+1) \leftarrow \widehat{Q_q}(n) \quad \text{else} \tag{22}$$

In the experimental section, we will use a slightly more generalized version of Frugal by replacing 1 with a positive constant Δ.

Please note that DUMIQE possesses deterministic updates in contrasts to RUMIQE and Frugal.

IV. Getting "around zero" in the case of the RUMIQE and DUMIQE algorithms

The update equations for the randomized or the deterministic update schemes are such that whenever $\widehat{Q_q}(0)$ is initialized to a positive value, then $\widehat{Q_q}(n)$ will remain positive for all subsequent time instants n. The parameter λ is chosen on the interval $(0,1)$ which yields that $(1-\lambda) < 1$ and $(1-\lambda(1-q)) < 1$ while $(1+\lambda) > 1$ and $(1+\lambda q) > 1$. Thus the rational of our schemes is to increase the estimate by multiplying with a number larger than 1, or to decrease by multiplying with a number smaller than 1.

A consequence of the multiplicative updating rules, our presented estimators are unable to handle the case when the quantile changes sign during the estimation process. This case might emerge in dynamic environments where the true quantile might drift from positive values to negative values and vice-versa. In addition, even in a stationary environments, usually the sign of the true quantile is unknown and thus a wrong initialization of the quantile estimate will hinder convergence. This can happen for instance in the case where the true unknown quantile is positive and we initialize our incremental estimator to a negative value and vice-versa.

In order to cope with the case of changing the sign of the estimate while performing the estimation, we extend both estimators using two different approaches:

- The first approach exploits a subtle idea in which we reckon with so-called phantom quantiles. The idea is based on the concept inspired by simple observation that $Prob(X \leq a) = Prob(X + b \leq a + b)$ for any real numbers a and b.
- We merge the update operations designed for positive quantiles together with the counter-part operations for updating negative quantile so that to build an estimator that is able to estimate any quantile, and most importantly, a one that is able to change the sign of the quantile estimate in an online-manner.

A. Getting "around zero": Introducing Phantom Quantile

In this section, we introduce our first idea for getting around zero and allowing our schemes (RUMIQE and DUMIQE) to be able to estimate any quantile without any prior knowledge of its sign.

1) Introducing Phantom Quantiles with a Fixed Shift: Let a be a positive real number. We suppose that we are dealing with estimating the quantile of a distribution $f_X(n)$ defined over $[-a, \infty]$, where X is known to be in $[-a, \infty]$. Thus, f_X admits positive and negative quantiles. The question that we try to address in this section is how to allow to estimate the negative quantiles of f_X. Our solution is based on the simple property that

$$Prob(\widehat{Q_q}(n) < X) = Prob(\widehat{Q_q}(n) + a < X + a)$$

Thus, by making a subtle modification, we can estimate any quantile of the the distribution f_X, whether it is positive or negative, by resorting to what we call phantom quantile estimate, which is merely a shifted quantile.

We introduce a phantom quantile estimator, $\widehat{Q'_q}(n+1)$, that gets updated by considering the phantom sample $x(n) + a$ and using the afore-mentioned equations, equations (2) to (4) for the case of randomized update (RUMQE), and equations (16) and (17) for the case of deterministic update (DUMQE). Now the true quantile can be estimated by shifting the phantom quantile estimator $\widehat{Q'_q}(n+1)$, i.e.

$$\widehat{Q_q}(n+1) = \widehat{Q'_q}(n+1) - a$$

It is easy to note that $\widehat{Q_q}$ converges to the true quantile Q_q on the interval $[-a, \infty]$. Thus, we call $\widehat{Q'_q}$ as a phantom quantile estimator with a fixed shift. In the next section, we will generalize the latter result for estimating the quantile of any distribution.

2) Generalizing Phantom Quantile: Algorithms 1 and 2 describe the operations needed for updating the quantile estimate $\widehat{Q_q}$ using the phantom quantile estimate, $\widehat{Q'_q}$ for the updating rules RUMIQUE and DUMIQUE, respectively. This permits us to generalize the process of estimating the positive or negative quantile of any distribution. The basic idea is to make sure that the phantom estimate always is above some positive value Q_{min}.

Algorithm 1 Phantom Based Algorithm for RUMIQE

```
Δ ← 0
Q̂_q(0) ← Q_min
while Stream of Data do
    Get sample x(n)
    x'(n) ← x(n) + Δ
    if Q̂'_q(n) < x'(n) and q ≤ rand() then
        Q̂'_q(n+1) ← (1 + λ)Q̂'_q(n)
    else if Q̂'_q(n) ≥ x'(n) and rand() ≤ (1 − q) then
        Q̂'_q(n+1) ← (1 − λ)Q̂'_q(n)
    else
        Q̂'_q(n+1) ← Q̂'_q(n)
    end if
    if Q̂'_q(n+1) < Q_min then
        Δ ← Δ + (Q_min − Q̂'_q(n+1))
        Q̂'_q(n+1) ← Q_min
    end if
    Q̂_q(n+1) ← Q̂'_q(n+1) − Δ
end while
```

Upon receiving a sample $x(n)$, we consider a phantom sample $x'(n) = x(n) + \Delta$. We then update $\widehat{Q'_q}(n+1)$ using the phantom sample $x'(n)$, as per equations (2) to (4) or (16) and (17). As a consequence of the update, we might violate the constraint that $\widehat{Q'_q}(n+1) \geq Q_{min}$. We, therefore, add a shift to $\widehat{Q'_q}(n+1)$, i.e., a positive quantity $Q_{min} - \widehat{Q'_q}(n)$, so as to ensure that $\widehat{Q'_q}(n+1) \geq Q_{min}$. Note that we sum up all the shifts obtained so far (up to instant n) whenever a violation takes place. The non-phantom quantile (target quantile estimate) is obtained from the phantom quantile by subtracting the total shift, Δ, so far. Note that the phantom quantile $\widehat{Q'_q}(n)$ will always lie in $[Q_{min}, \infty)$, while the estimate $\widehat{Q_q}(n)$ will converge to the true estimate.

B. Getting Around Zero: "Creating a bridge"

We explain this idea based on the RUMIQUE scheme, but it extends naturally also to the DUMIQUE scheme. The idea is to use the update equation for the positive quantile whenever the quantile is positive. As the quantile approaches 0, we can introduce an artificial "jump" over zero in order to make a transition to negative values, and then use the update equations for negative quantiles given by (13) to (15). In simple terms, let Q_{min} be a positive value. We use the positive update form (rules in equations (2) to (4)) whenever $\widehat{Q_q}(n) > 0$ and the negative update form whenever $\widehat{Q_q}(n) < 0$ (rules in equations (13) to (15)).

If $\widehat{Q_q}(n) > 0$ and $\widehat{Q_q}(n+1)$ falls in the interval $[-Q_{min}, Q_{min}]$, we operate a "jump" over zero and assign $-Q_{min}$ to $\widehat{Q_q}(n+1)$. Similarly, whenever $\widehat{Q_q}(n) < 0$

Algorithm 2 Phantom Based Algorithm for DUMIQE

$\Delta \leftarrow 0$
$\widehat{Q_q}(0) \leftarrow Q_{min}$
while Stream of Data **do**
  Get sample $x(n)$
  $x'(n) \leftarrow x(n) + \Delta$
  if $\widehat{Q'_q}(n) < x'(n)$ **then**
    $\widehat{Q'_q}(n+1) \leftarrow (1 + \lambda q)\widehat{Q'_q}(n)$
  else
    $\widehat{Q'_q}(n+1) \leftarrow (1 - \lambda(1-q))\widehat{Q'_q}(n)$
  end if
  if $\widehat{Q'_q}(n+1) < Q_{min}$ **then**
    $\Delta \leftarrow \Delta + (Q_{min} - \widehat{Q'_q}(n+1))$
    $\widehat{Q'_q}(n+1) \leftarrow Q_{min}$
  end if
  $\widehat{Q_q}(n+1) \leftarrow \widehat{Q'_q}(n+1) - \Delta$
end while

and $\widehat{Q_q}(n+1)$ falls in the interval $[-Q_{min}, Q_{min}]$, we "jump" over zero and assign Q_{min} to $\widehat{Q_q}(n+1)$. The schemes for enhancing the RUMIQE and DUMIQE with the bridge idea is described in Algorithm 3 and Algorithm 4, respectively.

Algorithm 3 Enhancing RUMIQE with a Bridge

$\Delta \leftarrow 0$
$\widehat{Q_q}(0) \leftarrow Q_{min}$
while Stream of Data **do**
  Get sample $x(n)$
  $S \leftarrow sign(\widehat{Q_q}(n))$
  if $\widehat{Q_q}(n) < x(n)$ and $q \leq rand()$ **then**
    $\widehat{Q_q}(n+1) \leftarrow (1 + S\lambda)\widehat{Q_q}(n)$
  else if $\widehat{Q_q}(n) \geq x(n)$ and $rand() \leq (1-q)$ **then**
    $\widehat{Q_q}(n+1) \leftarrow (1 - S\lambda)\widehat{Q_q}(n)$
  else
    $\widehat{Q_q}(n+1) \leftarrow \widehat{Q_q}(n)$
  end if
  if $\widehat{Q_q}(n+1) \in [-Q_{min}, Q_{min}]$ **then**
    $\widehat{Q_q}(n+1) \leftarrow -SQ_{min}$
  end if
end while

V. Experiments

In this section we compare our proposed family of multiplicative incremental quantile algorithms, namely, RUMIQE and DUMIQE to four of the state-of-the-art incremental quantile estimators, namely, the Stochastic Approximation (SA)-based quantile estimator due to Tierney [1], the exponential Weighted Stochastic Approximation proposed by Chen et al. [2], the estimator due to Cao et al. [3] and the Frugal approach by Ma et al. [6]. To tackle both negative and positive quantiles we present results for the phantom variable approach in Section IV-A. The other approach, creating a bridge

Algorithm 4 Enhancing DUMIQE with a Bridge

$\Delta \leftarrow 0$
$\widehat{Q_q}(0) \leftarrow Q_{min}$
while Stream of Data **do**
  Get sample $x(n)$
  $S \leftarrow sign(\widehat{Q_q}(n))$
  if $\widehat{Q_q}(n) < x(n)$ **then**
    $\widehat{Q_q}(n+1) \leftarrow (1 + S\lambda q)\widehat{Q_q}(n)$
  else
    $\widehat{Q_q}(n+1) \leftarrow (1 - S\lambda(1-q))\widehat{Q_q}(n)$
  end if
  if $\widehat{Q_q}(n+1) \in [-Q_{min}, Q_{min}]$ **then**
    $\widehat{Q_q}(n+1) \leftarrow -SQ_{min}$
  end if
end while

(Section IV-B), was also evaluated and resulted in very similar results as the phantom approach and thus is not shown.

We focus on four different cases in which the quantiles changes with time. In the first two cases we assume that $x(n)$, $n = 1, 2, 3, \ldots$ are independent outcomes from a normal distribution with expectation μ_n (varies with time n) and standard deviation σ. In order to simulate a dynamic environment, we assume that the expectation varies periodically with n

$$\mu_n = a \sin\left(\frac{2\pi}{T} n\right)$$

which is the sinus function with period T. In the first and the second case we estimate the 0.7 and 0.95 quantiles, respectively. We denote the two cases NORM_0.7 and NORM_0.95. For the third and the forth cases we assume that $x(n)$, $n = 1, 2, 3, \ldots$ are independent outcomes from a χ^2 distribution where the number of degrees of freedom, ν_n, varies periodically with n

$$\nu_n = a \sin\left(\frac{2\pi}{T} n\right) + b$$

where $b > a$ such that $\nu_n > 0$ for all n. In the third and the fourth case we estimate the 0.7 and 0.95 quantiles, respectively, and denote the two cases CHI_0.7 and CHI_0.95. Figures $1-4$ show the estimation of the true quantile at every time step for the four cases described above using the different estimation methods presented in this paper. To generate the results in the figures we used $\sigma = 1$, $Q_{min} = 2$, $a = 2$, $b = 5$ and $T = 2000$. For the method in Chen et al we sat $M = 10$. Using lower values of M resulted in numerical issues. We see that for all of the cases Tierney performs poorly, which is as expected since the estimator is constructed for a stationary system. Chen et al. performs better than Tierney, but due the batches, the estimator is lagging behind the true estimator. We observe that all the four estimators: Cao et al., Ma et al., RUMIQE and DUMIQE yield high performance. This is quite an impressive due to the simplicity of the estimators presented in this

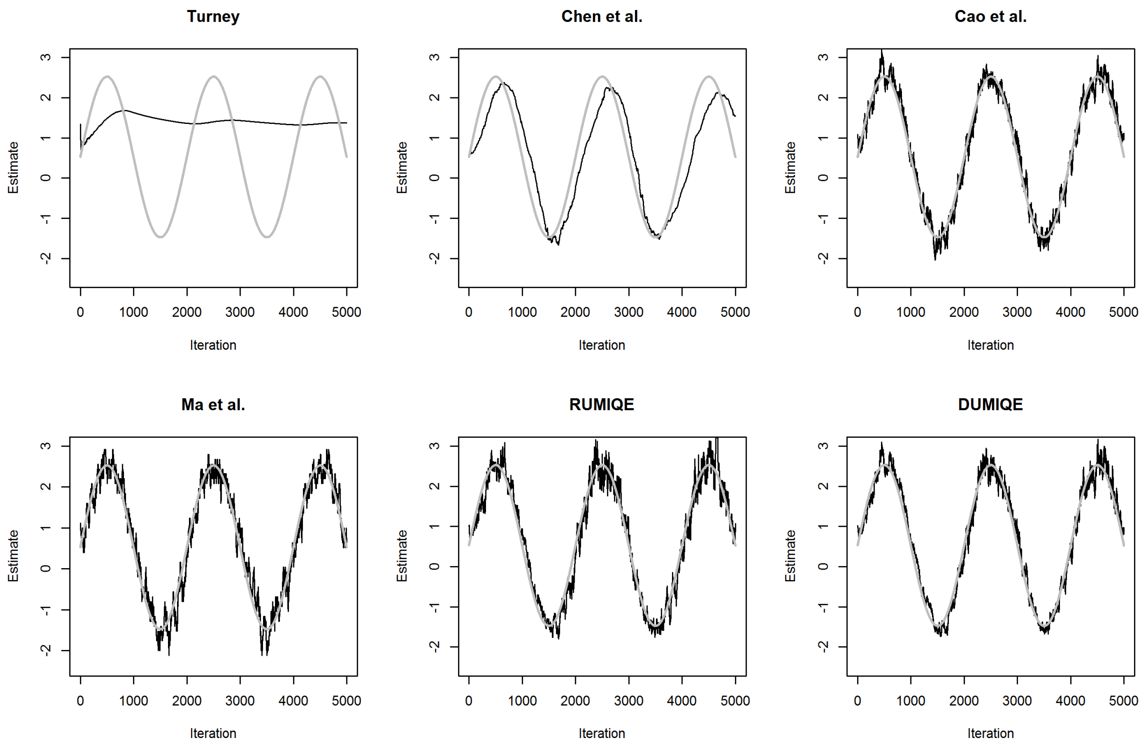

Fig. 1. Case: NORM_0.7. In each panel the gray and the black curves represent the true quantile as a function of iteration and the estimate respectively.

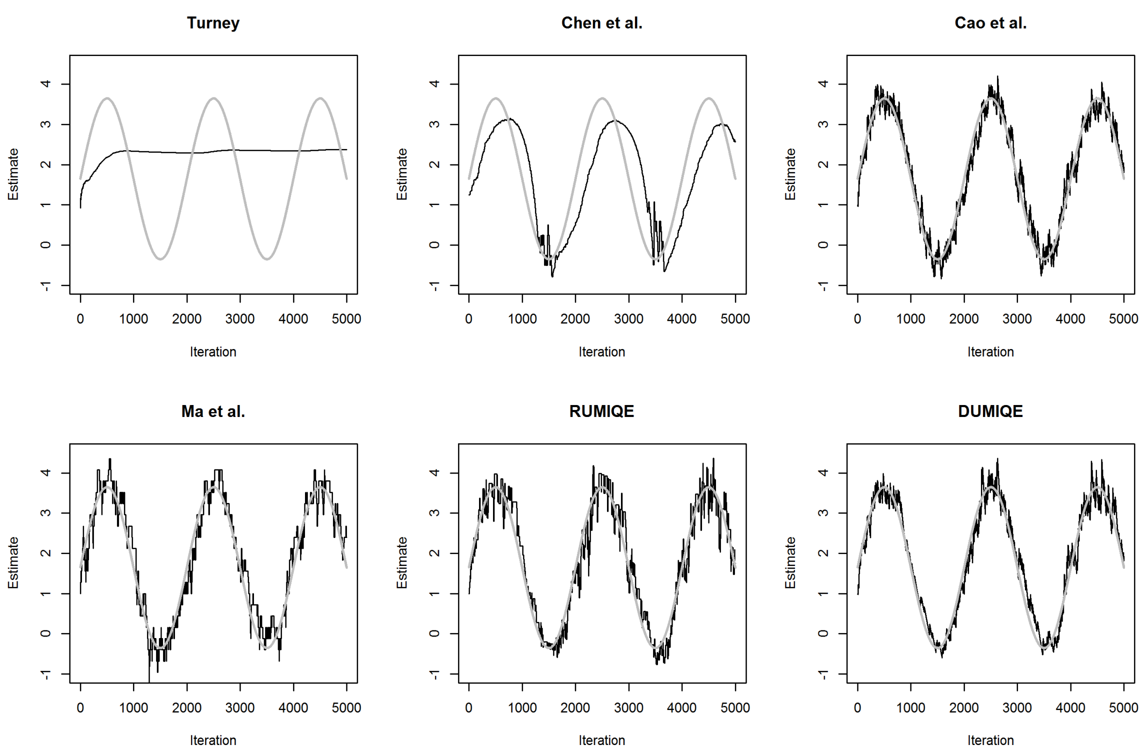

Fig. 2. Case: NORM_0.95. In each panel the gray and the black curves represent the true quantile as a function of iteration and the estimate respectively.

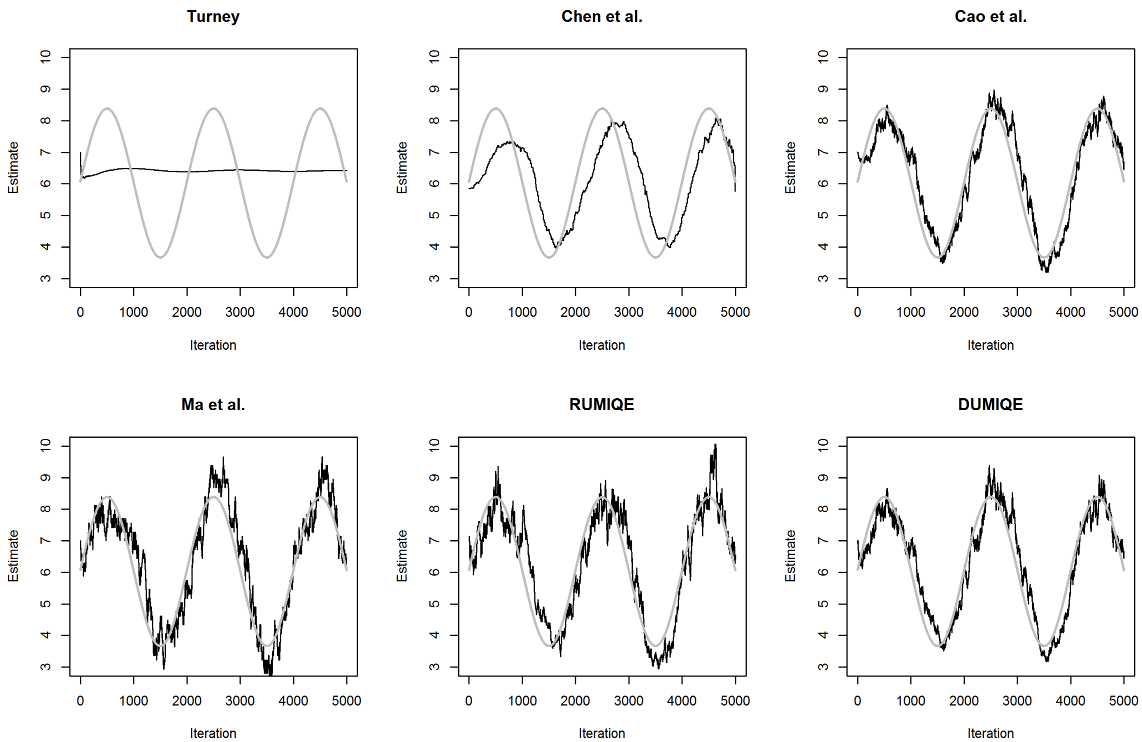

Fig. 3. Case: CHI_0.7. In each panel the gray and the black curves represent the true quantile as a function of iteration and the estimate respectively.

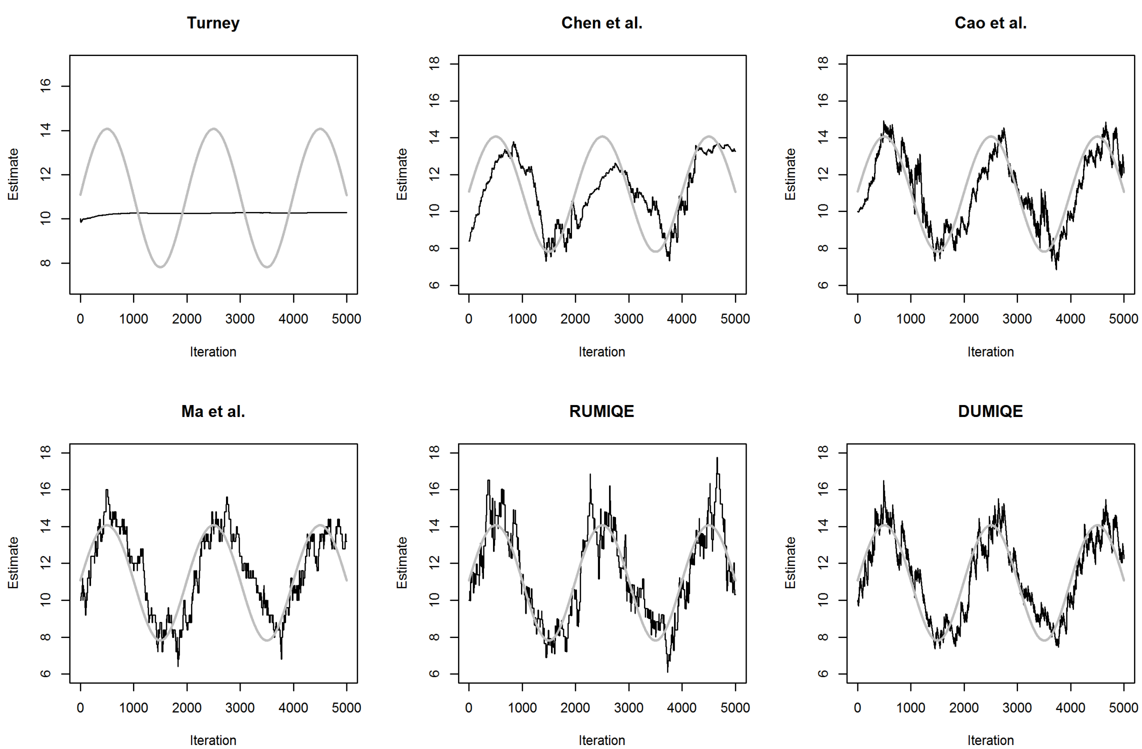

Fig. 4. Case: CHI_0.95. In each panel the gray and the black curves represent the true quantile as a function of iteration and the estimate respectively.

paper compared to the Cao et al. estimator. We also observe that RUMIQE and DUMIQE have no problem switching between a positive and a negative estimate of the quantile using the phantom approach (Section IV-A).

Next we do a more systematic comparison of the most promising estimators above, namely Cao et al., Ma et al., RUMIQE and DUMIQE. Ma et al. in fact has two algorithms, which we call Frugal 1 and Frugal 2 in the rest of the example. We started by generating 10^6 samples from both the time varying normal and χ^2 distribution using $\sigma = 1$, $a = 2$, $b = 5$ and two values of T, namely $T = 800$ (rapid variations) and $T = 8000$ (slow variations). For each of the four generated data sets we estimated both the 0.7 and 0.95 quantile in each iteration. We computed the estimation error using the root mean squared difference (RMSE) between the true quantile and the estimate for every iteration. We computed the estimation error for a large set of different values of the tuning parameters λ and c. Figures 5 and 6 show the results. The black, blue and red curves refer to RUMIQE, DUMIQUE and Ma et al. (Frugal 1 and 2), respectively and the gray curves refer to Cao et al. for different choices of c. The x-axis below the curves refers to the value of λ in the algorithms RUMIQUE, DUMIQUE and Cao et al. while the x-axis above the curves refers to the resolution used in Frugal 1 and 2. We see that for all the estimators the estimation error increases when the period, T, decreases or when estimating a quantile further into the tail of the distribution. We also see that the estimation error depends on the choice of the tuning parameters λ, Δ and c. For Cao et al. the estimation error seems to decrease using a lower value of c (except for the lower panels in Figure 6), but by using lower values of c than 2 we struggled with numerical issues. In fact, also for $c = 2, 5$ and 10 we got some numerical issues and typically by choosing a low value of c combined with a high value of λ for the χ^2 distribution cases. This is shown by the incomplete curves in Figures 5 and 6 were the missing results are due to the numerical issues.

Comparing RUMIQE and DUMIQE we see that DUMIQE systematically performs better than RUMIQE. For the case NORM_0.7 (upper panels in Figure 5) we see that DUMIQE performs about equally well as Cao et al. with $c = 2$ and about the same value of λ gives the optimal results. We see that Frugal 1 and 2 perform a little better than RUMIQUE, but poorer than DUMIQUE. Also for NORM_0.95 (lower panels in Figure 5) Cao et al. and DUMIQE perform about equally well, but that different choices of λ result in the best results ($\lambda \approx 0.05$ for Cao et al. and $\lambda \approx 0.1$ for DUMIQUE for $T = 800$). Further we see that Frugal 1 and 2 and RUMIQUE perform about equally well, but poorer than DUMIQUE.

For the case CHI_0.7 (upper panels in Figure 6) we see that $c = 2$ and DUMIQE perform about equally well using the optimal values of λ. For the case CHI_0.95 (lower panels in Figure 6) DUMIQE outperforms Cao et al. for all choices of c. We also see that DUMIQE seems to be more robust against estimation error when using a suboptimal value of λ (the curves changes less rapidly with λ). In a practical situation with a dynamical system, it is often hard to use an optimal value of λ so that a robustness vis-a-vis choice of the update parameter λ, is a great advantage. Further we see that $c = 10$ outperforms $c = 2$ and $c = 5$ which is in contrast with the other cases where $c = 2$ gave the best results. An other substantial disadvantage of Cao et al. compared to DUMIQE and RUMIQE is therefore the fact that we need to tune two parameters (λ and c) in contrast to only one for DUMIQUE and RUMIQE (λ). Finally we see that for both CHI_0.7 and CHI_0.95, RUMIQUE and Frugal 1 and 2 perform about equally well and poorer than DUMIQUE and Cao et al.

We also tested the Selection algorithm presented in [15]. The Selection algorithm operates without knowledge of the length of the data stream which is also the same underlying assumption as the family of Frugal algorithms [6] as well as our devised estimators DUMIQE and RUMIQE. The Selection algorithm returns the quantile of a data stream with at least $1 - \delta$ probability. We use the same parameter $\delta = 0.99$ as in [6] for the Selection algorithm. Apart from δ, the Selection algorithm does not have any tuning parameter. It resulted in root mean squared estimation errors as given by Tables I and II. We see that the algorithm performs poorer than the algorithms evaluated in Figures 5 and 6.

VI. Conclusion

In this paper, we have designed two novel incremental quantile estimators based on the theory of stochastic learning. The DUMIQE estimator is shown to outperform the state-of-the-art of incremental estimators in terms of convergence speed and accuracy. We emphasize that our estimators can be easily implemented and are far simpler than the Tierney family of estimators [1]–[5] as RUMIQUE and DUMIQUE do not require estimation of the density at the quantile.

We have shown how to extend the new estimators in order to handle negative quantiles by using two different methods. The first method is based on the idea of using phantom quantiles and simultaneously using the update equation designed for the positive quantile case. The second idea relies on modifying the update equation originally devised for estimating a positive quantile in order to accommodate the case of negative quantiles by exploiting the symmetry of the update equation for the positive quantile.

There are different extensions that can be envisaged for future work:

- Our algorithms for quantile estimation is designed for data elements that are added one by one. A possible extension is to generalize our schemes, namely, RUMIQE and DUMIQE to handle not only data insertions, but also dynamic data operations such as deletions and updates such as in [3].

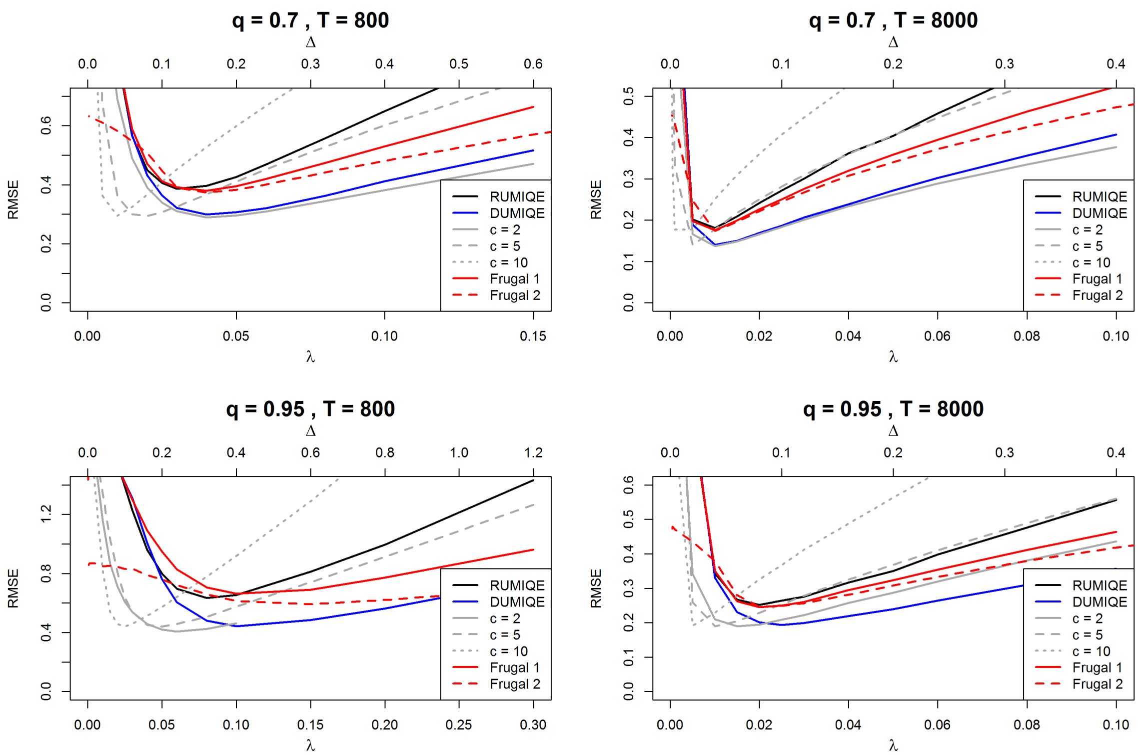

Fig. 5. Root mean squared estimation error for the case with outcomes from the normal distribution.

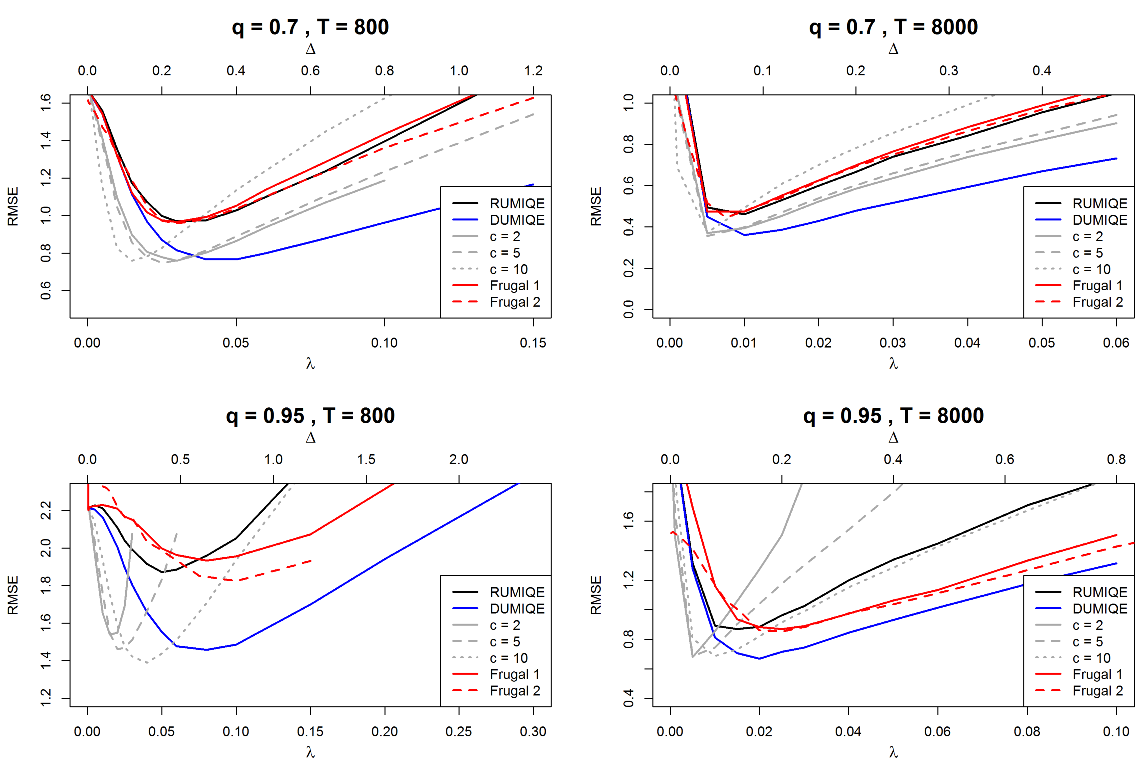

Fig. 6. Root mean squared estimation error for the case with outcomes from the χ^2 distribution.

$q = 0.7,\ T = 800$	$q = 0.7,\ T = 8000$	$q = 0.95,\ T = 800$	$q = 0.95,\ T = 8000$
1.545	1.606	1.746	1.724

TABLE I
NORMAL DISTRIBUTION CASE: ROOT MEAN SQUARED ESTIMATION ERROR FOR THE SELECTION ALGORITHM [15]

$q = 0.7,\, T = 800$	$q = 0.7,\, T = 8000$	$q = 0.95,\, T = 800$	$q = 0.95,\, T = 8000$
1.759	1.729	3.201	3.288

TABLE II
χ^2 DISTRIBUTION CASE: ROOT MEAN SQUARED ESTIMATION ERROR FOR THE SELECTION ALGORITHM [15]

- We are currently investigating how to extend our estimators in order to handle data arriving in a batch mode.
- An interesting research direction is to simultaneously estimate more than a single quantile value. To achieve this, our present schemes will have to be modified so as to guarantee the monotonicity property of the quantiles, i.e, maintaining multiple quantile estimates while simultaneously ensuring that the estimates do not violate the monotonicity property.
- An intriguing characteristic of our estimators is their multiplicative update form which is radically different from previous incremental estimators that resort to the additive update forms. We believe that this form of multiplicative update can be extended to other types of estimators such as binomial estimators.
- We submit that multiplicative increase-decrease estimator are faster than additive increase-decrease estimator, however at the cost of slightly higher variance [2]. There is a possibility to combine both schemes, i.e, multiplicative increase-decrease for approaching the optimal quantile and then additive increase decrease (similar to Tierney) for converging with less fluctuations to the optimal value. By virtue of the multiplicative updates, the quantile estimate can be adjusted in a ”geometric” manner.

REFERENCES

[1] L. Tierney, "A space-efficient recursive procedure for estimating a quantile of an unknown distribution," *SIAM Journal on Scientific and Statistical Computing*, vol. 4, no. 4, pp. 706–711, 1983.

[2] F. Chen, D. Lambert, and J. C. Pinheiro, "Incremental quantile estimation for massive tracking," in *Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2000, pp. 516–522.

[3] J. Cao, L. Li, A. Chen, and T. Bu, "Tracking quantiles of network data streams with dynamic operations," in *INFOCOM, 2010 Proceedings IEEE*. IEEE, 2010, pp. 1–5.

[4] J. Cao, L. E. Li, A. Chen, and T. Bu, "Incremental tracking of multiple quantiles for network monitoring in cellular networks," in *Proceedings of the 1st ACM workshop on Mobile internet through cellular networks*. ACM, 2009, pp. 7–12.

[5] J. M. Chambers, D. A. James, D. Lambert, and S. V. Wiel, "Monitoring networked applications with incremental quantile estimation," *Statistical Science*, pp. 463–475, 2006.

[6] Q. Ma, S. Muthukrishnan, and M. Sandler, "Frugal streaming for estimating quantiles," in *Space-Efficient Data Structures, Streams, and Algorithms*. Springer, 2013, pp. 77–96.

[7] B. J. Oommen and L. Rueda, "Stochastic learning-based weak estimation of multinomial random variables and its applications to pattern recognition in non-stationary environments," *Pattern Recogn.*, vol. 39, no. 3, pp. 328–341, 2006.

[8] L. Rueda and B. J. Oommen, "Stochastic automata-based estimators for adaptively compressing files with nonstationary distributions," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 36, no. 5, pp. 1196–1200, 2006.

[9] A. Yazidi and B. J. Oommen, "Novel discretized weak estimators based on the principles of the stochastic search on the line problem," *IEEE Transactions on Cybernetics*, vol. 46, no. 12, pp. 2732–2744, Dec 2016.

[10] J. I. Munro and M. S. Paterson, "Selection and sorting with limited storage," *Theoretical computer science*, vol. 12, no. 3, pp. 315–323, 1980.

[11] G. Cormode and S. Muthukrishnan, "An improved data stream summary: the count-min sketch and its applications," *Journal of Algorithms*, vol. 55, no. 1, pp. 58–75, 2005.

[12] B. Weide, "Space-efficient on-line selection algorithms," in *Computer Science and Statistics: Proceedings of the Eleventh Annual Symposium on the Interface*, 1978, pp. 308–311.

[13] A. Arasu and G. S. Manku, "Approximate counts and quantiles over sliding windows," in *Proceedings of the twenty-third ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*. ACM, 2004, pp. 286–296.

[14] M. Greenwald and S. Khanna, "Space-efficient online computation of quantile summaries," in *ACM SIGMOD Record*, vol. 30, no. 2. ACM, 2001, pp. 58–66.

[15] S. Guha and A. McGregor, "Stream order and order statistics: Quantile estimation in random-order streams," *SIAM Journal on Computing*, vol. 38, no. 5, pp. 2044–2059, 2009.

[16] H. Robbins and S. Monro, "A stochastic approximation method," *The annals of mathematical statistics*, pp. 400–407, 1951.

[17] H. Kushner and G. G. Yin, *Stochastic approximation and recursive algorithms and applications*. Springer Science & Business Media, 2003, vol. 35.

[18] V. R. Joseph, "Efficient robbins–monro procedure for binary data," *Biometrika*, vol. 91, no. 2, pp. 461–470, 2004.

[19] O. Arandjelovic, D.-S. Pham, and S. Venkatesh, "Two maximum entropy-based algorithms for running quantile estimation in non-stationary data streams," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 25, no. 9, pp. 1469–1479, 2015.

[20] B. W. Schmeiser and S. J. Deutsch, "Quantile estimation from grouped data: The cell midpoint," *Communications in Statistics-Simulation and Computation*, vol. 6, no. 3, pp. 221–234, 1977.

[21] V. Naumov and O. Martikainen, "Exponentially weighted simultaneous estimation of several quantiles," *World Academy of Science, Engineering and Technology*, vol. 8, pp. 563–568, 2007.

[22] R. Jain and I. Chlamtac, "The p 2 algorithm for dynamic calculation of quantiles and histograms without storing observations," *Communications of the ACM*, vol. 28, no. 10, pp. 1076–1085, 1985.

[23] M. F. Norman, *Markov processes and learning models*. Academic Press New York, 1972, vol. 84.

[24] M. Vojnović, J.-Y. Le Boudec, and C. Boutremans, "Global fairness of additive-increase and multiplicative-decrease with heterogeneous round-trip times," in *Proceedings of the IEEE INFOCOM 2000*, vol. 3. IEEE, 2000, pp. 1303–1312.

ACKNOWLEDGMENT

The authors gratefully acknowledge the help of Qiang Ma who provided us with an implementation of Frugal and the Selection algorithm.

[2]Please note that the terminology: multiplicative/additive increase-decrease is also used in the context of TCP congestion control algorithms [24].