Smooth estimates of multiple quantiles in dynamically varying data streams

Abstract In this paper, we investigate the problem of estimating multiple quantiles when samples are received online (data stream). We assume a dynamical system, i.e. the distribution of the samples from the data stream changes with time. A major challenge of using incremental quantile estimators to track multiple quantiles is that we are not guaranteed that the monotone property of quantiles will be satisfied, i.e, an estimate of a lower quantile might erroneously overpass that of a higher quantile estimate.

Surprisingly, we have only found two papers in the literature that attempt to counter these challenges, namely the works of Cao et al. [3] and Hammer and Yazidi [7] where the latter is a preliminary version of the work in this paper. Furthermore, the state-of-the-art incremental quantile estimator called Deterministic Update based Multiplicative Incremental Quantile Estimator (DUMIQE), due to Yazidi and Hammer [16], fails to guarantee the monotone property when estimating multiple quantiles.

A challenge with the solutions in [3] and [7], is that even though the estimates satisfy the monotone property of quantiles, the estimates can be highly irregular relative to each other which usually is unrealistic from a practical point of view. In this paper we suggest to generate the quantile estimates by inserting the quantile probabilities (e.g. $0.1, 0.2, \ldots, 0.9$) into a monotonically increasing and infinitely smooth function (can be differentiated infinitely many times). The function is incrementally updated from the data stream. The monotonicity and smoothness of the function ensure that both the monotone property and regularity requirement of the quantile estimates are satisfied.

The experimental results show that the method perform very well and estimate multiple quantiles more precisely than the original DUMIQE [16] and the approaches reported in [7] and [3].

This is a post-peer-review, pre-copyedit version of an article published in Pattern Analysis and Applications.
The final authenticated version is available online at: http://dx.doi.org/10.1007/s10044-019-00794-3.

Keywords Dynamically changing data stream · Incremental estimator · Multiple quantiles · Smooth quantile estimates

1 Introduction

Suppose that data samples from a probability distribution are arriving sequentially (data stream) and we are interested in estimating a quantile related to some probability q. The most natural estimator is to use the q quantile of the sample distribution. Unfortunately, such quantile estimators has a clear disadvantage as computation time and memory requirement are linear to the number of samples received from the data stream. Such methods thus are infeasible for large data streams. Several algorithms have been proposed to deal with those challenges. Most of the methods fall in to the category of what can be called histogram based methods. The methods are based on efficiently maintaining a histogram estimate of the data stream distribution such that small storage footprint is required. A representative work in this perspective is due to Schmeiser and Deutsch [12]. In fact, they proposed to use equidistant bins where the boundaries are adjusted online. Arandjelovic et al. [1] resort to 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.

Q-digest is probably among the most popular histogram based methods for computing quantile in data stream [13]. Q-digest maintains a binary tree where nodes in the tree store the frequency of the elements falling in the range specified by the corresponding subtree. Values whose counter is small are pushed up in the tree, and thus, Q-digest obtains an acceptable precision for high frequency values. In [15], Tschumitschew and Klawonn propose a so-called incremental quantile estimator based on the idea of maintaining an interval of m values around the target quantile and two counters: L and R where L points should fall on the left of the interval and R to the right of the interval. The authors suggest to balance the counters so that qR is approximately equal to $(1-q)L$. The major disadvantage of the approach is the need of change detection mechanism to deal with changes in the distribution. Furthermore, the algorithm is not incremental in contrast to DUMIQE and thus is more computationally involved. Jain and Chlamtac [8] are able to estimate a given single quantile by maintaining a set of target points using equi-width and equi-depth histograms from the data distribution and then dynamically updating their positions using quadratic interpolation.

For a thorough review and comparison of state-of-the-art histogram and batch methods, we refer the reader to [9] and references therein. It is worth mentioning that histogram methods, in fact, use less information from the past than incremental quantile estimators. The histograms only record in which bin a received sample should be assigned. This is particularly critical if the data

stream distribution changes with time. The optimal sliding window (in which the histogram is computed from) may then be short (few observations) and the resulting optimal histogram will consist of wide bins and a lot of information from each received sample is lost when converting the observations to a histogram. Naturally, again estimating quantiles from such a histogram will be poor and biased. In comparison, incremental quantile estimators estimate the needed quantiles directly without going the detour over sliding windows and histograms. Another fundamental challenge with histogram methods is how to maintain the bins. It is challenging to decide on the suitable width of the bins relative to the dynamics of the data stream. Further, if the data stream distribution changes, observations may fall outside the current bins and new bins must be created and other bins removed which substantially complicates the algorithms. Similar to the sliding window approach, histogram methods are more memory and computationally demanding than incremental quantile estimators since they are based on an underlying sliding window observations that need to be stored in the memory.

Another ally of methods are so called incremental update methods. The latter methods are based on performing small updates of the quantile estimate every time a new sample is received from the data stream. One of the first and prominent examples of this family of methods is the algorithm of Tierney (1983) [14] which is based on the stochastic learning theory. A few modifications of the Tierney method have been suggested, see e.g. [6, 2, 4, 5].

Nevertheless, a main shortcoming of Tierney based methods is the additional cost of building a local approximation of the distribution in the neighborhood of the quantile. The Frugal type of algorithms have been proposed in [10] which falls under the family of incremental quantile estimation method. Nevertheless, Frugal [10] suffers from limited accuracy due mainly to two facts: discretized search space and randomized updates which further increase the estimation error.

In [16], Yazidi and Hammer proposed DUMIQE which is an incremental quantile estimator and its randomized counter-part denoted by RUMIQE. The appealing property of DUMIQE is that the quantile estimate is updated by multiplying the current estimate with a suitable factor as seen in Equation (1) that is greater or less than 1 depending on wether the observation is bigger or smaller than the estimate. Such a multiplicative estimator thus will exponentially forget old sale estimates which is known to be the state-of-the-art strategy for adaptation to dynamically varying data streams, e.g. like the exponentially weighed average of observation are used to track data stream expectations. The informed reader would observe that classical incremental quantile estimators possess additive update form which means that the quantile estimate is updated by adding a positive quantity or a negative quantity according to wether the observation is bigger or smaller than the estimate. Through comprehensive simulation results, DUMIQE was shown to outperform other legacy quantile estimators including Frugal [10].

In data stream applications, a common situation is that the data stream distribution varies with time, referred to as a dynamical systems. Unfortu-

nately, histogram based methods, as referred to above, usually perform poorly in estimating quantiles in such systems and we are left with incremental methods as typically the only viable lightweight alternatives [3].

From a practical point of view it is often useful to estimate many quantiles of the dynamic data stream. A simple approach is to estimate the different quantiles independently of each other by running incremental estimators in parallel, one for each quantile to be estimated. Unfortunately, all incremental quantile update methods without any exception [6, 2, 4, 5, 10] face serious challenges when estimating multiple quantiles. The methods are not able to preserve the monotone property of quantiles, e.g. the estimate of the 50% quantile might overpass the estimate of the 70% quantile. The reason is the limited and local information that is used in the update, namely, current estimate and the last observation. The informed reader can notice that monotone violation is not an issue in the case of histogram based methods [11] as the quantile estimator share the same global knowledge, namely, the access to the histogram. In this paper we investigate the problem of estimating multiple quantiles from a dynamically changing data stream. The work on this topic is extremely sparse. To the best of our knowledge, Cao et al. [3] and the preliminary version of this paper [7] are the only solution found in the literature that propose a viable method to this problem. The work of Cao et al. [3] resorts to the idea of interpolation in order to avoid violation of the monotone property. Nevertheless, the approach of Cao et al. inherits the same disadvantages of Tierney method, namely, the need to build an approximation of the density in the neighborhood of the quantile. In [7], Hammer and Yazidi suggest to sort the quantiles or adjust the incremental step length if the monotone property is violated. A disadvantage with these methods is that, even though the methods will satisfy the monotone property of quantiles, the estimates will be highly irregular relative to each others which is unrealistic for most data stream distributions. In this paper, we suggest to estimate a monotonically increasing and infinitely smooth function (can be differentiated infinitely many times) and generate the quantile estimates as outputs of this function resulting in smooth, or regular, estimates relative to each others.

2 Estimation of multiple quantiles

Let X_n denote a stochastic variable representing the possible outcomes from a data stream at time n and let x_n denote a random sample of X_n. We assume that X_n is distributed according to some distribution $f_n(x)$ that varies dynamically with time n. Further let $Q_n(q)$ denote the quantile associated with probability q, i.e $P(X_n \leq Q_n(q)) = F_{X_n}(Q_n(q)) = q$.

In this paper we focus on simultaneously estimating the quantiles for K different probabilities $q_1, q_2, \ldots, q_K$ at each time step. We assume an increasing order of the probabilities, i.e. $q_1 < q_2 < \cdots < q_K$. The straight forward approach to estimate the quantiles would be to simply compute the DUMIQE

[16] (or some other online estimation procedure) for every probability

$$\begin{aligned} \widehat{Q}_{n+1}(q_k) &\leftarrow (1+\lambda q_k)\widehat{Q}_n(q_k) && \text{if } \widehat{Q}_n(q_k) < x_n \\ \widehat{Q}_{n+1}(q_k) &\leftarrow (1-\lambda(1-q_k))\widehat{Q}_n(q_k) && \text{if } \widehat{Q}_n(q_k) \geq x_n \end{aligned} \tag{1}$$

for $k = 1, 2, \ldots, K$. Unfortunately, this may lead to a violation of the monotone property of quantiles, i.e. we may not satisfy

$$\widehat{Q}_{n+1}(q_1) \leq \widehat{Q}_{n+1}(q_2) \leq \ldots \leq \widehat{Q}_{n+1}(q_K) \tag{2}$$

This can be explained as follows. Assume at time n that the monotone property is satisfied and that the sample x_n is between $\widehat{Q}_n(q_k)$ and $\widehat{Q}_n(q_{k+1})$, i.e.

$$\widehat{Q}_n(q_1) \leq \cdots \leq \widehat{Q}_n(q_k) < x_n < \widehat{Q}_n(q_{k+1}) \leq \cdots \leq \widehat{Q}_n(q_K) \tag{3}$$

Then according to (1) the estimates are updated as follows

$$\begin{aligned} \widehat{Q}_{n+1}(q_j) &\leftarrow (1+\lambda q_j)\widehat{Q}_n(q_j) && \text{for } j = 1, 2, \ldots, k \\ \widehat{Q}_{n+1}(q_j) &\leftarrow (1-\lambda(1-q_j))\widehat{Q}_n(q_j) && \text{for } j = k+1, \ldots, K \end{aligned} \tag{4}$$

which means that the estimates are increased for the quantiles with an estimate below x_n and decreased for the estimates above x_n. Consequently the monotone property might be violated. Next we present the two approaches suggested in [7], namely sorting the quantiles and adjusting the incremental step length.

2.1 Sorting the quantiles

Every time we receive a new sample x_n, the procedure consisted of the three following steps:

1. Update the quantile estimates according to (1) and get the estimates $\widehat{Q}_{n+1}(q_k), k = 1, 2, \ldots, K$
2. Sort the updated estimates and denote them $\widetilde{Q}_{n+1}(q_k), k = 1, 2, \ldots, K$. The estimates after sorting naturally will satisfy the monotone property.
3. Here we have two alternatives. Next time we received a sample (x_{n+1}), we updated according to Equation (1) using
   (a) the estimates from before the sorting, i.e. $\widehat{Q}_{n+1}(q_k), k = 1, 2, \ldots, K$
   (b) or the estimates after the sorting, i.e. $\widetilde{Q}_{n+1}(q_k), k = 1, 2, \ldots, K$

Alternative (a) means that we do *not* feed the information from the sorting back in to the estimation process, while in (b) we do. Using alternative (a) means that we only used sorting to "repair" the estimates from the original estimation process based on Equation (1).

2.2 Reduce the value of $\boldsymbol{\lambda}$

The next strategy was based on reducing the value of λ in a given iteration if the updates resulted in monotone property violation. Assume that we were in the situation where the sample x_n got a value between $\widehat{Q}_n(q_k)$ and $\widehat{Q}_n(q_{k+1})$ as given by (3). The first observation is that after the update, the monotone property always will be satisfied on each side of x_n, i.e.

$$\begin{aligned}
&\widehat{Q}_{n+1}(q_1) \leq \widehat{Q}_{n+1}(q_2) \leq \cdots \leq \widehat{Q}_{n+1}(q_k) \text{ and} \\
&\widehat{Q}_{n+1}(q_{k+1}) \leq \widehat{Q}_{n+1}(q_{k+2}) \leq \cdots \leq \widehat{Q}_{n+1}(q_K)
\end{aligned}$$

This follows from Equation (4). Therefore a sufficient criterion to satisfy the monotone property was to make sure to use a sufficiently small λ such that $\widehat{Q}_{n+1}(q_k) \leq \widehat{Q}_{n+1}(q_{k+1})$. We were able to find such a λ, denoted $\widetilde{\lambda}$, by making sure that the distance between $\widehat{Q}_{n+1}(q_k)$ and $\widehat{Q}_{n+1}(q_{k+1})$ was some portion, α, of the distance from the previous iteration, i.e.

$$\begin{aligned}
\widehat{Q}_{n+1}(q_{k+1}) - \widehat{Q}_{n+1}(q_k) &= \alpha \left(\widehat{Q}_n(q_{k+1}) - \widehat{Q}_n(q_k) \right) \\
(1 - \widetilde{\lambda}(1 - q_{k+1}))\widehat{Q}_n(q_{k+1}) - (1 + \widetilde{\lambda} q_k)\widehat{Q}_n(q_k) &= \alpha \left(\widehat{Q}_n(q_{k+1}) - \widehat{Q}_n(q_k) \right)
\end{aligned} \tag{5}$$

with $\alpha \in [0, 1)$. By solving (5) with respect to $\widetilde{\lambda}$ we got

$$\begin{aligned}
\widetilde{\lambda} &= (1 - \alpha) \frac{\widehat{Q}_n(q_{k+1}) - \widehat{Q}_n(q_k)}{(1 - q_{k+1})\widehat{Q}_n(q_{k+1}) + q_k \widehat{Q}_n(q_k)} \\
&= (1 - \alpha) H\Big(\widehat{Q}_n(q_k), \widehat{Q}_n(q_{k+1}) \Big)
\end{aligned} \tag{6}$$

We substituted λ with $\widetilde{\lambda}$ in (1) if using the originally chosen λ resulted in a monotone property violation. We then get the following updates

$$\widehat{Q}_{n+1}(q_k) \leftarrow (1+\lambda q_k)\widehat{Q}_n(q_k) \qquad \text{if } \widehat{Q}_n(q_k) < x_n \cap \widehat{Q}_n(q_{k+1}) < x_n \tag{7}$$

$$\begin{aligned}\widehat{Q}_{n+1}(q_k) \leftarrow{}& (1+\lambda q_k)\widehat{Q}_n(q_k)\\ &\text{if } \widehat{Q}_n(q_k) < x_n \cap \widehat{Q}_n(q_{k+1}) \geq x_n \cap \lambda < H\Big(\widehat{Q}_n(q_k), \widehat{Q}_n(q_{k+1})\Big)\end{aligned} \tag{8}$$

$$\begin{aligned}\widehat{Q}_{n+1}(q_k) \leftarrow{}& \Big(1+(1-\alpha)H\Big(\widehat{Q}_n(q_k), \widehat{Q}_n(q_{k+1})\Big)\, q_k\Big)\, \widehat{Q}_n(q_k)\\ &\text{if } \widehat{Q}_n(q_k) < x_n \cap \widehat{Q}_n(q_{k+1}) \geq x_n \cap \lambda > H\Big(\widehat{Q}_n(q_k), \widehat{Q}_n(q_{k+1})\Big)\end{aligned} \tag{9}$$

$$\widehat{Q}_{n+1}(q_k) \leftarrow (1-\lambda(1-q_k))\widehat{Q}_n(q_k) \;\text{ if } \widehat{Q}_n(q_k) \geq x_n \cap \widehat{Q}_n(q_{k-1}) \geq x_n \tag{10}$$

$$\begin{aligned}\widehat{Q}_{n+1}(q_k) \leftarrow{}& (1-\lambda(1-q_k))\widehat{Q}_n(q_k)\\ &\text{if } \widehat{Q}_n(q_k) \geq x_n \cap \widehat{Q}_n(q_{k-1}) < x_n \cap \lambda < H\Big(\widehat{Q}_n(q_{k-1}), \widehat{Q}_n(q_k)\Big)\end{aligned} \tag{11}$$

$$\begin{aligned}\widehat{Q}_{n+1}(q_k) \leftarrow{}& \Big(1-(1-\alpha)H\Big(\widehat{Q}_n(q_{k-1}), \widehat{Q}_n(q_k)\Big)\,(1-q_k)\Big)\, \widehat{Q}_n(q_k)\\ &\text{if } \widehat{Q}_n(q_k) \geq x_n \cap \widehat{Q}_n(q_{k-1}) < x_n \cap \lambda > H\Big(\widehat{Q}_n(q_{k-1}), \widehat{Q}_n(q_k)\Big)\end{aligned} \tag{12}$$

for $k = 2, \ldots, K-1$. The special cases for $k = 1$ and $k = K$ are shown below. Equation (7) shows the case when x_n takes a value above $\widehat{Q}_n(q_{k+1})$ and therefore there is no risk of monotone property violation. The update therefore is as in Equation (1). Equation (8) shows the case when x_n takes a value between $\widehat{Q}_n(q_k)$ and $\widehat{Q}_n(q_{k+1})$ and it is a potential risk of violating the monotone property. But since $\lambda < H\Big(\widehat{Q}_n(q_k), \widehat{Q}_n(q_{k+1})\Big)$ we do not get a monotone property violation using λ and thus also this update is as in (1). Equation (9) shows the case when x_n takes a value between $\widehat{Q}_n(q_k)$ and $\widehat{Q}_n(q_{k+1})$ and $\lambda > H\Big(\widehat{Q}_n(q_k), \widehat{Q}_n(q_{k+1})\Big)$ which results in a monotone property violation using λ and the update uses $\widetilde{\lambda}$ from Equation (6) instead of λ in this update. Equations (10) to (12) show the similar updates when x_n takes a value below $\widehat{Q}_n(q_k)$.

For the smallest and largest quantile estimates, we only got potential monotone violations upwards and downwards, respectively resulting in the following

updates

$$\widehat{Q}_{n+1}(q_1) \leftarrow (1+\lambda q_1)\widehat{Q}_n(q_1) \qquad \text{if } \widehat{Q}_n(q_1) < x_n \cap \widehat{Q}_n(q_2) < x_n \tag{13}$$

$$\begin{aligned}\widehat{Q}_{n+1}(q_1) \leftarrow{}& (1+\lambda q_1)\widehat{Q}_n(q_1) \\ &\text{if } \widehat{Q}_n(q_1) < x_n \cap \widehat{Q}_n(q_2) \geq x_n \cap \lambda < H\Big(\widehat{Q}_n(q_1), \widehat{Q}_n(q_2)\Big)\end{aligned} \tag{14}$$

$$\begin{aligned}\widehat{Q}_{n+1}(q_1) \leftarrow{}& \Big(1+(1-\alpha)H\Big(\widehat{Q}_n(q_1), \widehat{Q}_n(q_2)\Big)\, q_1\Big)\, \widehat{Q}_n(q_1) \\ &\text{if } \widehat{Q}_n(q_1) < x_n \cap \widehat{Q}_n(q_2) \geq x_n \cap \lambda > H\Big(\widehat{Q}_n(q_1), \widehat{Q}_n(q_2)\Big)\end{aligned} \tag{15}$$

$$\widehat{Q}_{n+1}(q_1) \leftarrow (1-\lambda(1-q_1))\widehat{Q}_n(q_1) \;\; \text{if } \widehat{Q}_n(q_1) \geq x_n \tag{16}$$

and

$$\widehat{Q}_{n+1}(q_K) \leftarrow (1+\lambda q_K)\widehat{Q}_n(q_K) \qquad \text{if } \widehat{Q}_n(q_K) < x_n \tag{17}$$

$$\widehat{Q}_{n+1}(q_K) \leftarrow (1-\lambda(1-q_K))\widehat{Q}_n(q_K) \text{ if } \widehat{Q}_n(q_K) \geq x_n \cap \widehat{Q}_n(q_{K-1}) \geq x_n \tag{18}$$

$$\begin{aligned}\widehat{Q}_{n+1}(q_K) \leftarrow{}& (1-\lambda(1-q_K))\widehat{Q}_n(q_K) \\ &\text{if } \widehat{Q}_n(q_K) \geq x_n \cap \widehat{Q}_n(q_{K-1}) < x_n \cap \lambda < H\Big(\widehat{Q}_n(q_{K-1}), \widehat{Q}_n(q_K)\Big)\end{aligned} \tag{19}$$

$$\begin{aligned}\widehat{Q}_{n+1}(q_K) \leftarrow{}& \Big(1-(1-\alpha)H\Big(\widehat{Q}_n(q_{K-1}), \widehat{Q}_n(q_K)\Big)\,(1-q_K)\Big)\, \widehat{Q}_n(q_K) \\ &\text{if } \widehat{Q}_n(q_K) \geq x_n \cap \widehat{Q}_n(q_{K-1}) < x_n \cap \lambda > H\Big(\widehat{Q}_n(q_{K-1}), \widehat{Q}_n(q_K)\Big)\end{aligned} \tag{20}$$

By estimating the quantiles using the rules in (7) − (20), we ensured that the monotone property in (2) was satisfied in every iteration $n = 1, 2, 3, \ldots$.

2.3 Distributional assumption

As described in the introduction, the only incremental approaches to simultaneously track multiple quantiles are the two approaches described above and the approach due to Cao et al. [3]. A challenge with all the these approaches is that even though they satisfy the monotone property of quantiles, they do not ensure that the estimates are smooth, or regular, relative to each others. In this section, we suggest a novel approach that both satisfies the monotone property of quantiles and the smoothness/regularity of the estimates relative to each other.

Assume that $f_n(x)$ is a parametric distribution such that the dynamic variations over time is determined by some parameter θ_n, i.e. $f_n(x) = f(x;\theta_n)$. Further let $F(x;\theta_n)$ denote the cumulative distribution, i.e. $F(x;\theta_n) = P(X_n \leq$

x). Due to the properties of quantiles we have that $F(Q_n(q);\theta_n) = q$ which means that

$$F\left(\widehat{Q}_n(q_k);\theta_n\right) \approx q_k,\ k = 1, 2, \ldots, K$$

and

$$\widehat{Q}_n(q_k) \approx F^{-1}\left(q_k;\theta_n\right),\ k = 1, 2, \ldots, K$$

We can now estimate the unknown parameters of the distribution to get a good fit to the estimates in the current iteration, e.g. using least squares

$$\widetilde{\theta}_n = \arg\min_{\theta_n} \left\{ \sum_{k=1}^{K} \left(\widehat{Q}_n(q_k) - F^{-1}\left(q_k;\theta_n\right) \right)^2 \right\} \tag{21}$$

Finally we can update the estimates $\widehat{Q}_n(q_k)$ with output from the inverse cumulative distribution

$$\widetilde{Q}_n(q_k) = F^{-1}(q_k;\widetilde{\theta}_n) \tag{22}$$

where $\widetilde{Q}_n(q_k)$ denote updated estimate of the estimate $\widehat{Q}_n(q_k)$. The appealing part of this procedure is that since $F^{-1}(q;\theta)$ is a monotonically increasing function in q for every θ, we are guaranteed that the updated estimates $\widetilde{Q}_n(q_1), \widetilde{Q}_n(q_2), \ldots, \widetilde{Q}_n(q_K)$ satisfy the monotone property. In addition, if $F^{-1}(q;\theta)$ is smooth as a function of q, the quantile estimates $\widetilde{Q}_n(q_1), \widetilde{Q}_n(q_2), \ldots, \widetilde{Q}_n(q_K)$ will be smooth, or regular, relative to each others.

An example for the normal distribution is as follows. Let $\Phi(\cdot)$ denote the cumulative distribution function for the standard normal distribution. Assuming that X_n is normally distributed with expectation μ_n and standard deviation σ_n, it is well known that

$$\Phi\left(\frac{Q_n(q) - \mu_n}{\sigma_n}\right) = q$$

and therefore that

$$\Phi\left(\frac{\widehat{Q}_n(q_k) - \mu_n}{\sigma_n}\right) \approx q_k,\ k = 1, 2, \ldots, K$$

Solving with respect to $\widehat{Q}_n(q_k)$ we get

$$\widehat{Q}_n(q_k) = \mu_n + \sigma_n \Phi^{-1}(q_k),\ k = 1, 2, \ldots, K$$

which is in fact a simple linear regression problem with unknown parameters μ_n and σ_n (with the constraint that $\sigma_n > 0$). The least squares estimates of μ_n and σ_n (Equation (21) can be computed analytically in $O(K)$ time which is of the same order as initially updating the quantiles when a new sample arrives.

The procedure is then as follows.

For(n = 1,2,3,...)

1. After receiving x_n, update the quantile estimates using (1) to get $\widehat{Q}_{n+1}(q_k), k = 1, 2, \ldots, K$.
2. Estimate $\widetilde{\theta}_{n+1}$ using (21).
3. Compute $\widetilde{Q}_{n+1}(q_k), k = 1, 2, \ldots, K$ using (22) to satisfy the monotone property.
4. Similar to step 3. of the sorting approach in Section 2.1, this approach can be run in two different ways as well. We can continue to next iteration using $\widehat{Q}_{n+1}(q_k), k = 1, 2, \ldots, K$ or using the updated quantiles $\widetilde{Q}_{n+1}(q_k), k = 1, 2, \ldots, K$.

3 Experiments

We evaluate the suggested method for both synthetic and real-life data examples.

3.1 Synthetic data experiments

It is possible to prove that the DUMIQE approach in (1) converges to the true quantiles [16]. For the suggested method in this paper (Section 2.3), we know that if we use $\widehat{Q}_n(q_k), k = 1, 2, \ldots, K$ as input to the next iteration, the underlying quantile estimates will converge to the true quantiles. Intuitively it makes sense to use $\widetilde{Q}_n(q_k), k = 1, 2, \ldots, K$ as input to the next iteration, but makes it intrinsically hard to formally prove convergence. Proofs of convergence are hard (or impossible) for the methods in Cao et al. [3] and Hammer and Yazidi [7] as well. We thus resort to simulations to compare the performance of the different approaches.

The experiments focus on the ability of the methods to track quantile estimates when the distribution of the data stream changes with time. We consider the two different cases were we assume that the data are outcomes from a normal distribution or a χ^2 distribution. For the normal distribution case we assume that the expectation of the distribution varies with time

$$\mu_n = a \sin\left(\frac{2\pi}{T} n\right), \quad n = 1, 2, 3, \ldots$$

which is the sinus function with period T. Further, we assume that the standard deviation of the distribution does not vary with time but is equal to one. For the χ^2 distribution case we assume that the number of degrees of freedom varies with time as follows:

$$\nu_n = a \sin\left(\frac{2\pi}{T} n\right) + b, \quad n = 1, 2, 3, \ldots$$

where $b > a$ such that $\nu_n > 0$ for all n. In the results below we used $a = 2$ and $b = 6$.

Figure 1 shows a small section of the estimation processes using DUMIQE as given by (1) and the methods in Section 2. The light gray dots show the samples from the data stream and is the same in all the four panels. The data are generated from the normal distribution above with $T = 800$. The gray and the black curves show estimates of the 0.4 and the 0.6 quantiles of the data, respectively. The text bring = TRUE above the upper right panel, means that we fed the sorted quantiles back into the estimation procedure. Similarly for lower right panel, bring = TRUE means that the updated estimates $\widetilde{Q}_n(q_k)$ were fed back in to the estimation procedure. We can observe that in the case of DUMIQE, the monotone property is violated in several iterations (upper left panel). For the other methods, the monotone property is, as expected, satisfied in every iteration.

Assume the χ^2 distribution case given above with $T = 800$. Figure 2 shows quantile estimates for the probabilities $q_1 = 0.1, \ldots, q_9 = 0.9$ for an arbitrary iteration. In the figure (and the figures below), the abbreviations SORT, PREV and GAUSS refer to the estimation approaches presented in Sections 2.1, 2.2 and 2.3, respectively. We see that the DUMIQE (gray dotted line) represent the most irregular estimates. The slope of the curve changes dramatically as a function of the quantile probabilities. The slope is even negative at some parts showing that DUMIQE do not satisfy the monotone property of quantiles. Further, we see that the methods suggested in this paper (blue curve) is far smoother than the other approaches (gray curves) and thus give more realistic estimates.

Now we turn to doing a thorough analysis of how well the proposed methods in Section 2 estimate quantiles of data streams. We estimated quantiles of both the normally and χ^2 distributed data streams above using two different periods, namely $T = 800$ (rapid variation) and $T = 8000$ (slow variation), i.e. in total four different data streams. In addition, for each of the four data streams we estimated quantiles that were centred around the median or in the tail of the distribution, i.e. eight different cases. We chose the quantiles close enough to get a fair amount of monotone property violations. Naturally, if we chose the quantiles far from each other we rarely or never got any violations. In more detail, we estimated the following quantiles for the different cases.

- For the normal distribution and the quantiles around the median, we estimated the quantiles related to the following probabilities $q_k = \Phi(-0.8 + 0.2(k-1))$, $k = 1, 2, \ldots, 9$ where $\Phi(\cdot)$ refers to the cumulative distribution function of the standard normal distribution. Recall that in dynamical systems, as in these experiments, the value of a quantile related to a specific probability varies with time.
- For the normal distribution and the quantiles in the tail of the distribution, we use $q_k = \Phi(0.8 + 0.2(k-1))$, $k = 1, 2, \ldots, 9$.
- For the χ^2 distribution and the quantiles around the median, we estimated the quantiles related to the following probabilities $q_k = F(4.2 + 0.3(k-1); \nu = 6)$, $k = 1, 2, \ldots, 9$ where $F(\cdot; \nu)$ refers to the cumulative distribution function of the χ^2 distribution with ν degrees of freedom.

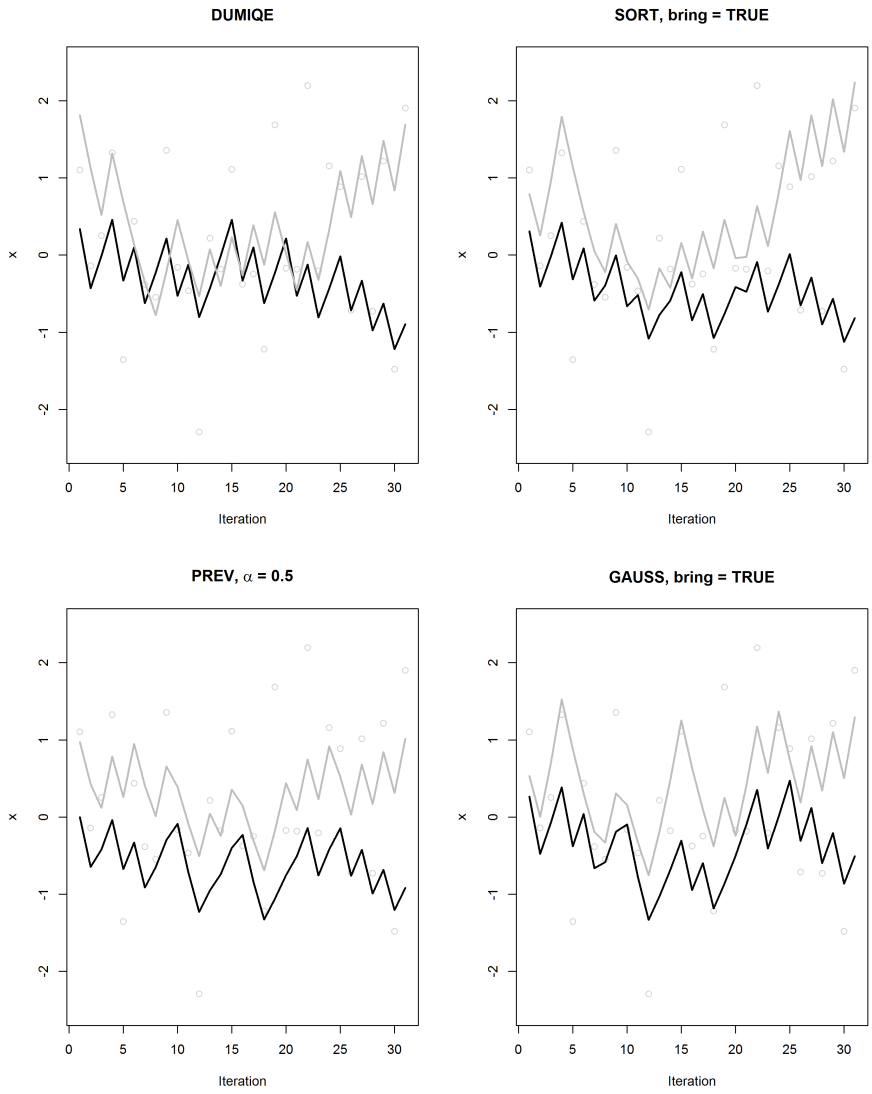

Fig. 1 Estimation processes using DUMIQE and the methods in Section 2. The light gray dots show that samples from the distribution (the data stream) and the black and the gray curves show estimates of the 0.4 and the 0.6 quantiles of the data, respectively.

– Finally, for the χ^2 distribution and the quantiles in the tail of the distribution, we estimated the quantiles related to the following probabilities $q_k = F(12 + 0.4(k-1); \nu = 6), \;\; k = 1, 2, \ldots, 9.$

The probabilities related to quantiles in the median and around the tail of the distribution are centred around the probabilities 0.5 and 0.95, respectively.

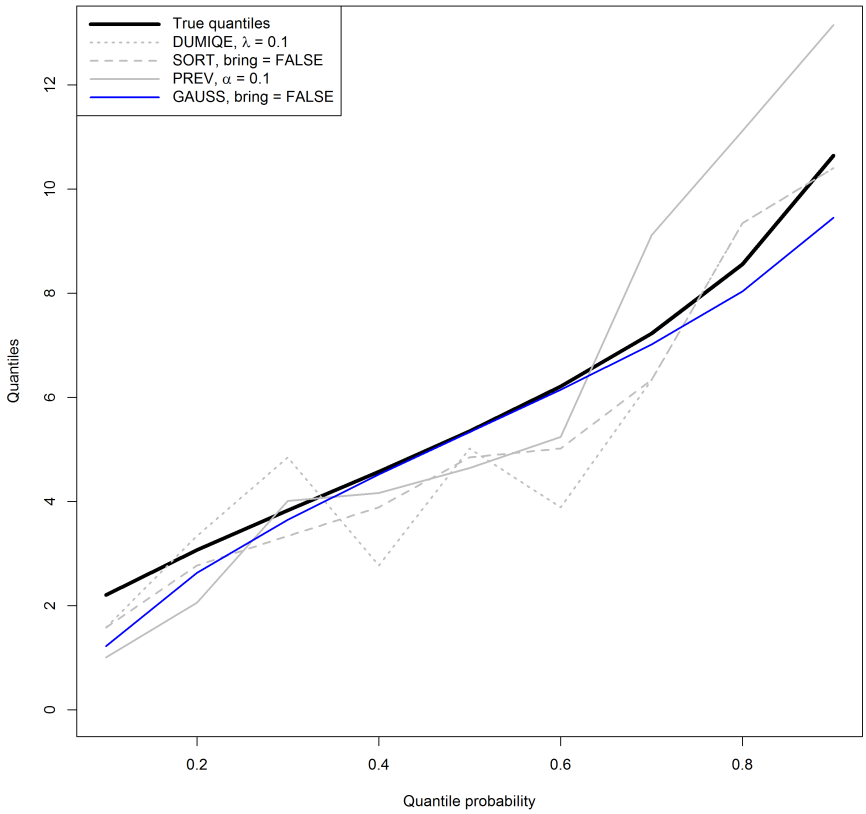

Fig. 2 Quantile estimate for the probabilities $q_1 = 0.1, \ldots, q_9 = 0.9$ for an arbitrary iteration using the methods presented in this paper. The abbreviations SORT, PREV and GAUSS refer to the estimation approaches presented in Sections 2.1, 2.2 and 2.3, respectively

The choices above resulted in a monotone property violation in about every third iteration using a typical value $\lambda = 0.05$ in (1).

To measure estimation error, we use the average of the root mean squares error (RMSE) for each quantile

$$RMSE = \frac{1}{K}\sum_{k=1}^{K}\sqrt{\frac{1}{N}\sum_{n=1}^{N}\left(Q_n(q_k) - \widehat{Q}_n(q_k)\right)^2}$$

where N is the total number of samples in the data stream. We investigate the estimation error for a large set of different values of the parameter λ. In the experiments we used $N = 10^7$ which efficiently removed any Monte Carlo errors in the experimental results.

The results for the normal and χ^2 cases are shown in Figures 3 and 4, respectively. For the suggested method in this paper (Section 2.3), we assume

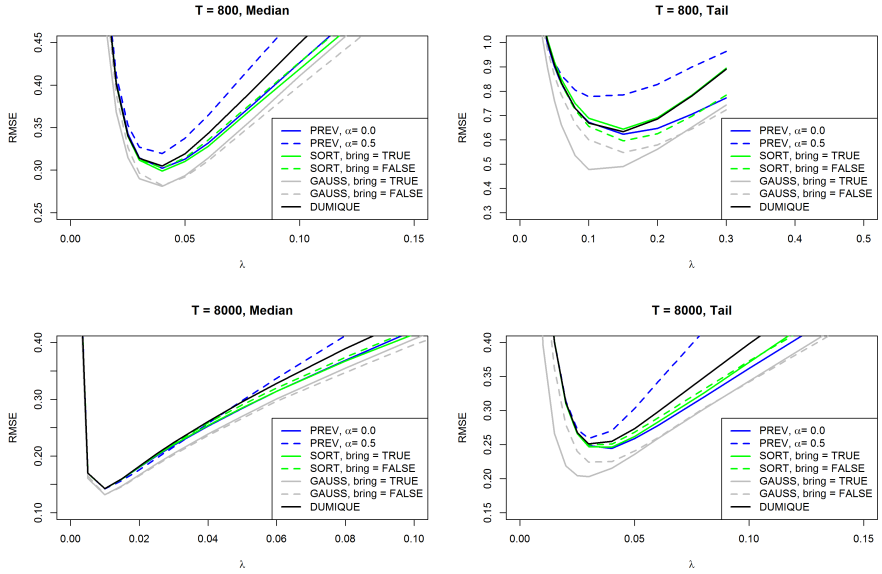

Fig. 3 Estimation error for data from the normal distribution.

that the outcomes from the data stream are normally distributed and thus denote the approach GAUSS. For the SORT approach (Section 2.1), bring = TRUE means that we fed the sorted quantiles back into the estimation procedure. Similarly for the GAUSS approach, bring = TRUE means that the updated estimates $\widetilde{Q}_n(q_k)$ were fed back in to the estimation procedure. DUMIQE refers to updating the quantiles using (1) and ignoring that the monotone property may get violated.

We start by discussing the normal distribution cases. For all the estimation methods, we see that the estimation error increases when the period decreases or when estimating further into the tail of the distribution. Further we see that the approach suggested in this paper (GAUSS) outperforms the other approaches. It seems also that feeding the updated estimates $\widetilde{Q}_n(q_k)$ back into the estimation process further improves the estimation compared to not doing it. For the approach in Section 2.2 (PREV) we observe that using $\alpha = 0.5$ (making small updates) performs poor in all the experiments. Using $\alpha = 0$ means that we update as much as possible without violating the monotone property. The latter case performs about equally well to sorting the quantiles (Section 2.1). For the sorting approach, feeding the sorted estimates back in the estimation process or not affects the estimation results minimally. A nice observation is that almost all the approaches performs better than updating

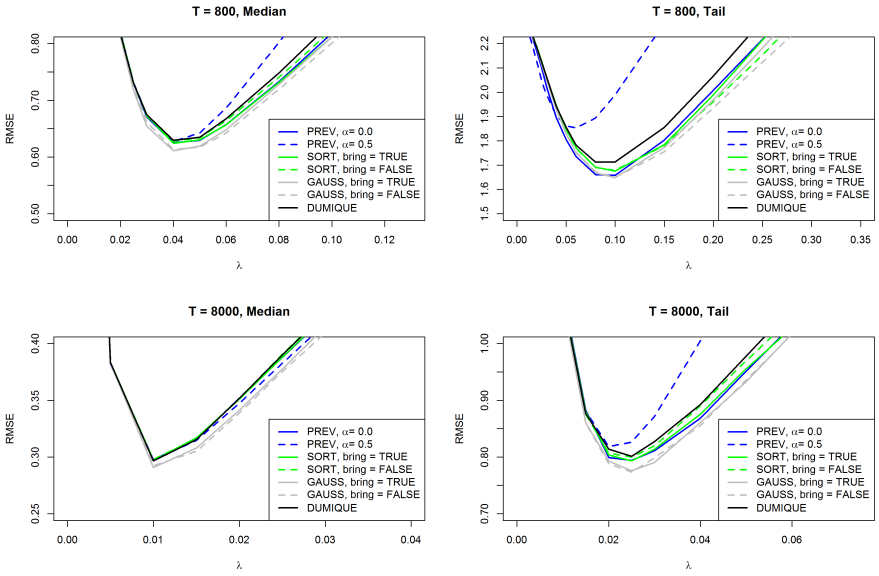

Fig. 4 Estimation error for data from the χ^2 distribution.

without caring about satisfying the monotone property (DUMIQE). In other words, we are able to both satisfy the monotone property and improve estimation precision and since the different approaches are computationally efficient this is achieved with minimal extra computational costs. In fact, the approach suggested in this paper may be appealing even when we only want to estimate one quantile. We can simply track several other (auxiliary) quantiles and use the approaches to improve the estimation of the quantile of interest. Please note that this comes with an additional computational cost compared to just using DUMIQE since several quantiles need to be tracked.

We now turn to the χ^2 distribution experiments. Also for these experiments, we see that the suggested method in this paper (Section 2.3) outperforms the other methods. Also here we assume that the outcomes from the data stream are from a normal distribution even though they in reality are from a χ^2 distribution. It is a quite interesting and counter-intuitive result that this approach still performs the best. The normal and the χ^2 distribution are quite different which means that the approach seems robust to erroneous assumptions about the distribution of the data stream. Similar to the normal distribution cases, PREV with $\alpha = 0.5$ performs poor (Section 2.2) and SORT (Section 2.1) and PREV with $\alpha = 0$ perform about equally well. Again we see that, except for PREV with $\alpha = 0.5$, all the approaches outperform DUMIQE.

For comparison we also tested the method in [3] for the eight estimation tasks described above. This is the only incremental estimation method we have found in the literature that attempts to estimate multiple quantiles in a dynamical system. The method has two tuning parameters, a weight parameter

Distrib.	$T = 800$, Med	$T = 800$, Tail	$T = 8000$, Med	$T = 8000$, Tail
Normal	0.312	0.630	0.259	0.370
	$T = 800$, Med	$T = 800$, Tail	$T = 8000$, Med	$T = 8000$, Tail
χ^2	0.79	2.40	0.445	1.611

Table 1 Estimation error using the method in Cao et al. (2009) [3].

similar to λ in the methods in this paper, and a parameter that controls the width of intervals to estimate the distribution of the data stream around a quantile. To achieve as good results as possible, we ran the method for a large set of values for the two parameters. The best estimation results are shown in Table 1. We see that for the normal distribution and $T = 800$ Cao et al. performs well, but poorer than the method suggested in this paper (gray curves). For the normal distribution and $T = 8000$, the method in this paper outperforms Cao et al. (2009) [3]. For all the cases related to the χ^2 distribution, the methods in this paper outperforms Cao et al. (2009) [3] with a clear margin. Not only does the methods in this paper outperform Cao et al. (2009) [3], they are also far simpler to implement and only contain only one tuning parameters which makes it easier to tune the method to perform well. The experiments also showed that the methods in this paper is less sensitive to the choice of the tuning parameter compared to Cao et al. (2009) [3].

3.2 Real-life data example

It is difficult to do a systematic evaluation of the performance of the algorithms for real-life data since we do not know the values of the true quantiles, i.e, the ground truth. If we were in a static system we could compare with state-of-the art offline estimators using all the data, but offline estimators do not cope with the case of dynamically changing data streams considered in this paper. Therefore we show the usefulness of the suggested algorithms by visualizing how well they are able to track a dynamically changing real-life data stream.

We consider the problem of tracking the number of tweets posted on Twitter. Figures 5 to 7 shows the results. The gray circles in the figures show the number of tweets posted by Norwegian Twitter users every minute in the time period before and after the Oslo bombing and Utøya massacre in Norway July 22 2011. The terror attach started by a bomb going off in Oslo at July 22 3:25 p.m, and as expected, we see a rapid increase in the number of posted tweets after that time.

For each of the suggested methods we tracked quantiles related to the $K = 9$ probabilities $q_1 = 0.1, q_2 = 0.2, \ldots, q_9 = 0.9$. The black, blue and red curves show the tracking of the quantiles related to the probabilities q_2, q_5 and q_8 of the distribution of the number of tweets posted. We see that the methods efficiently track the quantiles of the time-varying distribution. We also see that the monotone property of quantiles is satisfied in every iteration. Inspecting the figures more carefully we see that the GAUSS method suggested in this

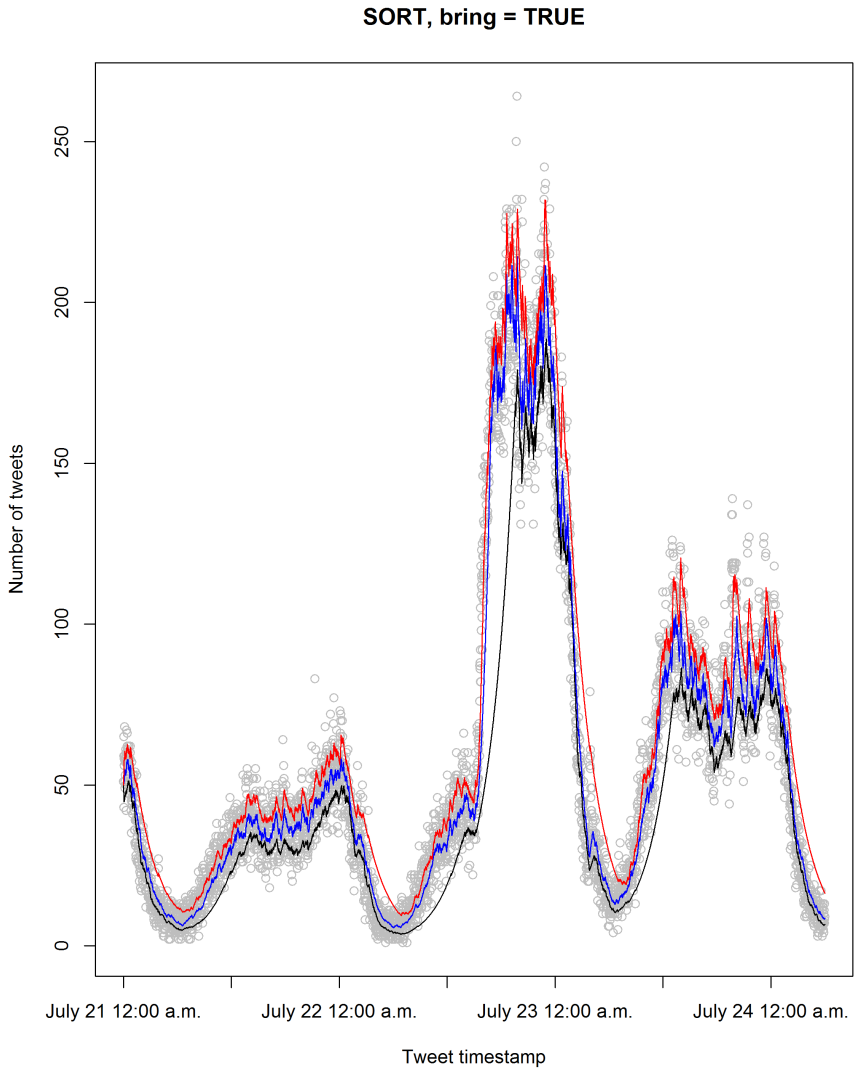

Fig. 5 Using the SORT method: The gray circles show the number of tweets posted by Norwegian Twitter users every minute from July 21 2011 to July 24 2011. The black, blue and red curves show running estimates of the 20, 50 and 80% quantiles of the distribution of the number of tweets posted.

paper is more efficient at tracking the quantiles in the time periods where the distribution changed rapidly and seems to be the method that in total performs the best. This is in accordance with our findings in the synthetic experiments (Figures 3 and 4).

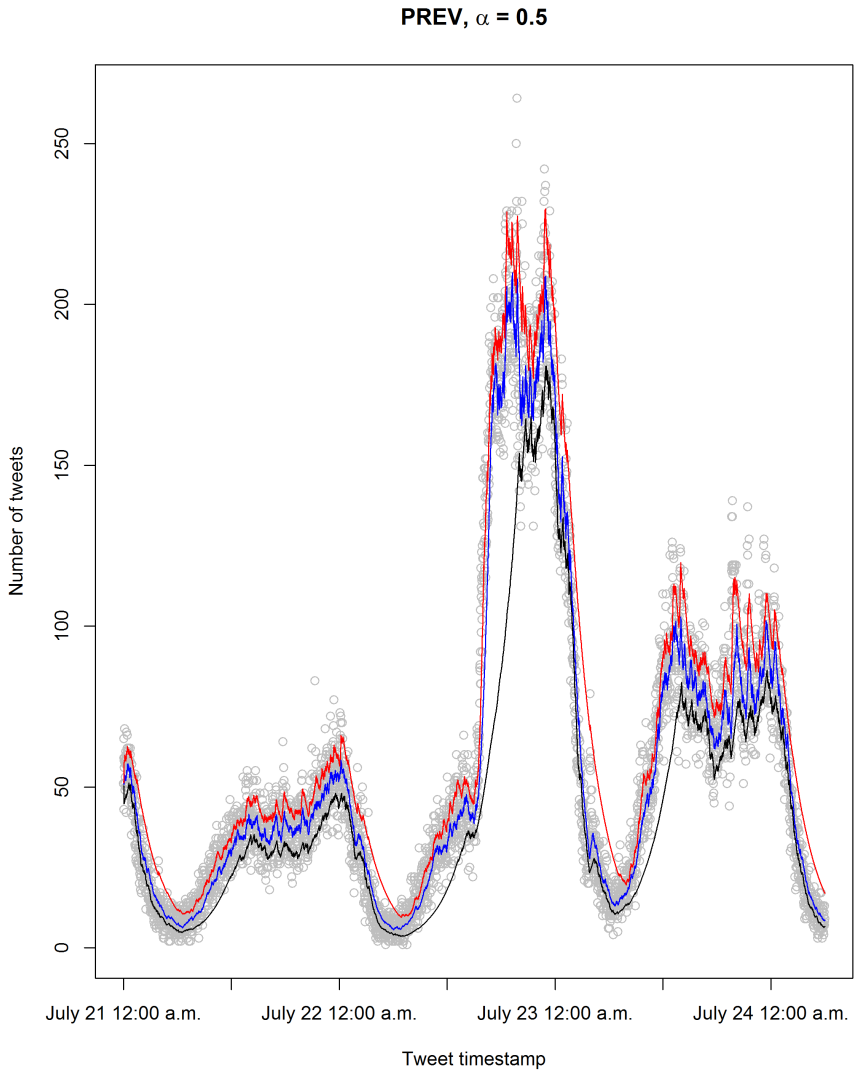

Fig. 6 Using the PREV method with $\alpha = 0.5$: The gray circles show the number of tweets posted by Norwegian Twitter users every minute from July 21 2011 to July 24 2011. The black, blue and red curves show running estimates of the 20, 50 and 80% quantiles of the distribution of the number of tweets posted.

4 Closing remarks

In this paper we have investigated the problem of estimating multiple quantiles from a data stream when the distribution of the data varies with time. Unfortunately, so called histogram methods do not perform well for such data

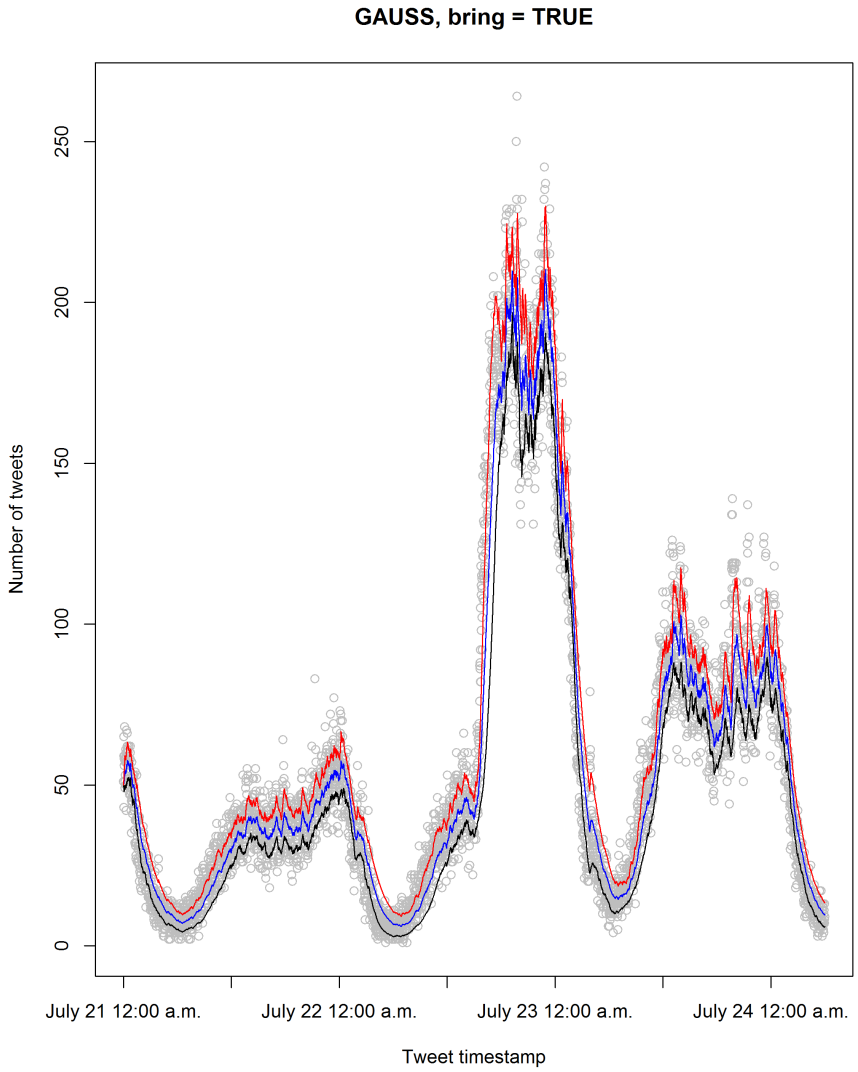

Fig. 7 Using the GAUSS method: The gray circles show the number of tweets posted by Norwegian Twitter users every minute from July 21 2011 to July 24 2011. The black, blue and red curves show running estimates of the 20, 50 and 80% quantiles of the distribution of the number of tweets posted.

streams and incremental estimators typically are the only viable alternative [3]. Incremental estimators that estimate multiple quantiles is surprisingly sparse. In fact, we have only found two papers in the literature that try to counter these challenges, namely the works of Cao et al. [3] and Hammer and Yazidi [7] where the latter is a preliminary version of the work in this paper.

A challenge when applying incremental methods to estimate multiple quantiles, is that we are not guaranteed that the quantile estimates satisfy the monotone property of quantiles. The methods in [3] and [7] satisfy this requirement, but the estimates often become highly irregular compared to each other. In addition to satisfying the monotone property of quantiles, the suggested method in this paper generates realistic quantile estimates in the sense that the estimates become smooth, or regular, relative to each other.

In addition to generating smooth and realistic estimates, the suggested method in this paper documents higher estimation precision than both [3] and [7]. Computationally, the suggested method is of the same order of computational complexity as DUMIQE. In other words, we are able to both satisfy the monotone property and generate smooth estimates with minimal extra computational costs. Furthermore, the approaches suggested in this paper may be appealing even when we only want to estimate a single quantile. In this perspective, we can simply track several other (auxiliary) quantiles and use the approach proposed in this paper to improve the estimation of the single quantile of interest. Obviously, this comes with an additional computational cost compared to using DUMIQE since several quantiles need to be tracked compared to just one.

As shown in Figure 4 the performance of the smoothing approach is robust to the differences between the chosen smoothing function and the data stream distribution. However, improved results could probably be achieved by equipping the algorithm with a handful of smoothing functions based on some well-known distributions like the normal, chi square, gamma, Cauchy etc. The final quantile estimates are based on the smoothing function that best fits to the individual quantile estimates. Thus is a possible direction for future research.

References

1. Ognjen Arandjelovic, Duc-Son Pham, and Svetha Venkatesh. Two maximum entropy-based algorithms for running quantile estimation in non-stationary data streams. *IEEE Transactions on Circuits and Systems for Video Technology*, 25(9):1469–1479, 2015.
2. Jin Cao, Li Li, Aiyou Chen, and Tian Bu. Tracking quantiles of network data streams with dynamic operations. In *2010 Proceedings of IEEE INFOCOM*, pages 1–5. IEEE, 2010.
3. Jin Cao, Li Erran Li, Aiyou Chen, and Tian 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*, pages 7–12. ACM, 2009.
4. Jin Cao, Li Erran Li, Aiyou Chen, and Tian 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*, pages 7–12. ACM, 2009.

5. John M Chambers, David A James, Diane Lambert, and Scott Vander Wiel. Monitoring networked applications with incremental quantile estimation. *Statistical Science*, pages 463–475, 2006.
6. Fei Chen, Diane Lambert, and José C Pinheiro. Incremental quantile estimation for massive tracking. In *Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 516–522. ACM, 2000.
7. Hugo Lewi Hammer and Anis Yazidi. Incremental quantiles estimators for tracking multiple quantiles. In *Proceedings of the 30th International Conference on Industrial Engineering and Other Applications of Applied Intelligent Systems (IEA/AIE)*, pages 202–210, France, 2017. Springer.
8. Raj Jain and Imrich Chlamtac. The p 2 algorithm for dynamic calculation of quantiles and histograms without storing observations. *Communications of the ACM*, 28(10):1076–1085, 1985.
9. Ge Luo, Lu Wang, Ke Yi, and Graham Cormode. Quantiles over data streams: experimental comparisons, new analyses, and further improvements. *The VLDB Journal*, 25(4):449–472, 2016.
10. Qiang Ma, S Muthukrishnan, and Mark Sandler. Frugal streaming for estimating quantiles: One (or two) memory suffices. *arXiv preprint arXiv:1407.1121*, 2014.
11. James P McDermott, G Jogesh Babu, John C Liechty, and Dennis KJ Lin. Data skeletons: simultaneous estimation of multiple quantiles for massive streaming datasets with applications to density estimation. *Statistics and Computing*, 17(4):311–321, 2007.
12. Bruce W Schmeiser and Stuart Jay Deutsch. Quantile estimation from grouped data: The cell midpoint. *Communications in Statistics-Simulation and Computation*, 6(3):221–234, 1977.
13. Nisheeth Shrivastava, Chiranjeeb Buragohain, Divyakant Agrawal, and Subhash Suri. Medians and beyond: new aggregation techniques for sensor networks. In *Proceedings of the 2nd international conference on Embedded networked sensor systems*, pages 239–249. ACM, 2004.
14. Luke Tierney. A space-efficient recursive procedure for estimating a quantile of an unknown distribution. *SIAM Journal on Scientific and Statistical Computing*, 4(4):706–711, 1983.
15. Katharina Tschumitschew and Frank Klawonn. Incremental quantile estimation. *Evolving Systems*, 1(4):253–264, 2010.
16. Anis Yazidi and Hugo Hammer. Multiplicative Update Methods for Incremental Quantile Estimation. *IEEE Transactions on Cybernetics*, 2017.