Information Sciences 563 (2021) 40–58

Contents lists available at ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

Joint tracking of multiple quantiles through conditional quantiles

Hugo Lewi Hammer [a,b,*], Anis Yazidi [a], Håvard Rue [c]

[a] OsloMet – Oslo Metropolitan University, Norway
[b] Simula Metropolitan Center for Digital Engineering, Norway
[c] King Abdullah University of Science & Technology, Saudi Arabia

ARTICLE INFO

Article history:
Received 13 February 2019
Received in revised form 8 February 2021
Accepted 12 February 2021
Available online 20 February 2021

Keywords:
Data mining
Data Stream
Joint estimates
Quantile tracking
Real time analytics

ABSTRACT

The estimation of quantiles is one of the most fundamental data mining tasks. As most real-time data streams vary dynamically over time, there is a quest for adaptive quantile estimators. The most well-known type of adaptive quantile estimators is the incremental one which documents the state-of-the art performance in tracking quantiles. However, the absolute vast majority of incremental quantile estimators fail to jointly estimate multiple quantiles in a consistent manner without violating the monotone property of quantiles. In this paper, first we introduce the concept of *conditional quantiles* that can be used to extend incremental estimators to jointly track multiple quantiles. Second, we resort to the concept of conditional quantiles to propose two new estimators. Extensive experimental results, based on both synthetic and real-life data, show that the proposed estimators clearly outperform legacy state-of-the-art joint quantile tracking algorithms in terms of accuracy while achieving faster adaptivity in the face of dynamically varying data streams.

1. Introduction

As a result of the steadily increase in the volume of data produced daily [35], the need for efficient algorithms for real-time analysis of such large amounts of data is becoming stringent [24]. Since conventional statistical and data mining techniques are mainly designed for offline situations, they are usually not adequate for real-time analysis [25]. Thus, a wide range of streaming algorithms were proposed in the literature for dealing with different data mining tasks such as clustering, filtering, cardinality estimation, estimation of moments or quantiles, predictions and anomaly detection [24] in online settings.

Given a stream of data, probably the first and most arguably foundational problem is to describe its data distribution. Quantiles are useful to describe the distribution in a flexible and nonparametric way [30]. The estimation of quantiles of data streams encompasses a wide range of applications such as portfolio risk measurement in the stock market [10,1], fraud detection [47], signal processing and filtering [41], climate change monitoring [48], SLA violation monitoring [39,40], network monitoring [7], Monte Carlo simulation [44], structural health monitoring [12], non-parametric statistical testing [28] and Tukey depth estimation [19]. Motivated by this wide range of applications of streaming quantile estimation, in this paper, we will investigate advancing the state-of-the-art when it comes to joint quantile estimation.

* Corresponding author.
E-mail address: hugo.hammer@oslomet.no (H.L. Hammer).

https://doi.org/10.1016/j.ins.2021.02.014
0020-0255/© 2021 The Author(s). Published by Elsevier Inc.

Suppose that we are interested in estimating the quantile related to some probability q. The straightforward approach is to store all the samples and use the q quantile of the sample distribution. Unfortunately, this straightforward approach suffers from clear shortcomings since the computation time and memory requirement are linear to the number of samples received so far. Thus, this family of methods is impractical for the case of large data streams.

Several algorithms have been proposed to deal with those computational and memory challenges. Most of the proposed methods fall under the so-called category of histogram based methods. Those methods are based on efficiently maintaining a histogram estimate of the data stream distribution such that only a small storage footprint is required. Another ally of methods are the so-called incremental update methods. The methods are based on performing small updates of the quantile estimate every time a new sample is received from the data stream. Generally, the current estimate is a convex combination of the estimate at the previous time step and a quantity depending on the current observation. A thorough review of state-of-the-art streaming quantile estimation methods is given in the related work section (Section 2).

In data stream mining applications, it is common that the distribution of the data stream varies with time. In such settings, the system is commonly referred to as dynamical system in the literature. In the context of a dynamical system, the problem most commonly addressed is to dynamically update estimates of quantiles based on all data received from the data stream so far. Histogram based methods are well suited to address this problem. However, an under-investigated and yet crucial problem is to estimate quantiles of the current distribution of the data stream typically referred to as quantile tracking. Incremental methods can document state-of-the-performance for the quantile tracking problem [46,18], while histogram methods are not well suited for efficient quantile tracking [5].

To address the tracking problem, several incremental quantile estimators have been suggested, for some representative examples see e.g. [5,4,46,31,43]. The intuition behind this family of estimators is rather simple. If the received sample has a value below some threshold, e.g. the current quantile estimate, the estimate is decreased. Alternatively, whenever the received sample has a value above the same threshold, the estimate is increased. Even though the estimators document state-of-the-art tracking performance [46], none of them to the best of our knowledge uses the values of the received samples directly to update the estimate, but only whether the values of the samples are above or below some varying threshold. Intuitively, this seems like a loss of information received from the data stream as the magnitude of the observations are not directly used. Recently, Hammer et al. [18,16] presented an incremental estimator that uses the values of the received samples directly which makes it distinct from *all* incremental estimators previously presented in the literature. The estimator is in fact a generalized exponentially weighted average of previous observations received from the data stream and documents state-of-the-art performance [18,16].

The incremental estimators above are constructed to track a single quantile. However, from a practical point of view, it is often more useful to jointly track multiple quantiles in order for example to be able to approximate the current data stream distribution. A naive approach is to run multiple incremental estimators in parallel, but this would result in a loss of control over the joint properties of the estimates. Even the monotone property of quantiles[1] will be most likely violated. Surprisingly, little research has been devoted to developing incremental quantile estimators that jointly can track multiple quantiles for dynamically varying data streams. To the best of our knowledge, a handful of methods found in the literature is due to Cao et al. [5] and Hammer et al. [15,20]. The method by Cao et al. is based on first running an incremental update of each quantile estimate and then computing a monotonically increasing approximation of the cumulative distribution of the data stream distribution using a form of a linear interpolation. Finally, the quantile estimates are computed from the approximate cumulative distribution. A disadvantage of the latter method is that the "monotonization" step is quite ad hoc and does not provide guarantee that the resulting quantile estimates converge to the true ones. The methods by Hammer et al. [15,20] are based on adjusting the update step size in each iteration to ensure that the monotone property of quantiles is always satisfied. A disadvantage with these methods is that the required step size for maintaining the monotone property of the quantiles might become too small, which might hinder tracking the changes of the data stream distribution in an efficient manner. We will shed further the light on this aspect in the experiments reported in the paper.

In this paper, we devise two algorithms to address the shortcomings of the current state-of-the-art multiple quantiles tracking estimators described above. We show that the proposed estimators converge to the true quantiles and that they can efficiently track multiple quantiles even under rapidly changing environment. Compared to most of the streaming quantile estimators reported in the literature, the propounded algorithms are extremely computational and memory efficient. In fact, our suggested algorithms require only storing a single value per quantile estimate.

The two algorithms can be seen as a crucial extension of [46,18] by virtue of invoking the subtle concept of conditional quantiles. The concept of conditional quantiles is general and can also be used to derive joint estimates based on other families of incremental estimators.

The remainder of this paper is organized as follows. Section 2 is dedicated to a comprehensive review of the related works. In Section 3, we motivate the quantile estimators developed in this paper by first shedding the light on how the state-of-the-art incremental estimators fall short in maintaining the monotone property when jointly estimating multiple quantiles. In Section 4, we present the concept of conditional quantiles and show how it can be applied to obtain joint estimates. In Sections 5 and 6, we put forward two new algorithms for joint quantile estimation. In Section 7, we present comprehensive experimental results that catalogue the properties of our estimators. Section 8 concludes the article.

[1] E.g. that the 70% quantile estimate must have a value above the 60% quantile estimate.

2. Related work

The most representative example of an incremental quantile estimator is due to the seminal work of Munro and Paterson [32]. In [32], Munro and Paterson described a p-pass algorithm for selection using $O(n^{1/(2p)})$ space for any $p \geqslant 2$. Cormode and Muthukrishnan [8] 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 [45]. Examples of these works include [45,32,11,29].

In [34], 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 to stationary data. The approximation of the quantiles 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]. 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 to keep it in its ideal position by counting the 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 by using a piecewise parabolic prediction function.

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% quantile of the 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 [36] which established the field of research called "stochastic approximation" [26] has 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 [36] was further developed in [23] in order to handle the case of "extreme quantiles". Moreover, the estimator provided by Tierney [42] falls under the same umbrella of the example given in [36], and thus can be seen as an extension of it.

As Arandjelovic remarks [2], 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.

Given dynamically varying data stream, two main problems are considered namely to i) dynamically update estimates of quantiles of all data received from the stream so far or ii) estimate quantiles of the current distribution of the data stream (tracking). To address problem i), histogram based methods form an important class of memory efficient methods. A representative work in this perspective is due to Schmeiser and Deutsch [37]. In fact, Schmeiser and Deutsch proposed to use equidistant bins where the boundaries are adjusted online. Arandjelovic et al. [2] 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. Nevertheless, histogram based methods have problems addressing problem ii) of tracking quantiles of the current data stream distribution [5].

To address the dynamic tracking problem ii) incremental algorithms represent an important class of methods. However, the research on incremental methods is quite sparse. As described in the introduction the methods are based on making small updates of the quantile estimates every time a new sample is received. Tierney [42] uses a sample mean update from previous quantile estimates, while [6,5,4] introduce modifications based on exponential decay when taking into account the old estimates and thus these works are useful to track quantiles of non-stationary data stream distributions. More recent incremental quantile estimation approaches are the Frugal algorithm by Ma et al. [31], the DUMIQE algorithm by Yazidi and Hammer [46], and the DQTRE and DQTRSE algorithms by Tiwari and Pandey [43]. An appealing property of the DUMIQE, DQTRE and DQTRSE and the estimators suggested in this paper is that the update size is automatically adjusted in accordance to the scale/range of the data. This makes the estimators robust to substantial changes in the data stream.

3. Monotone property violations for incremental quantile estimators

Let X_t denote a stochastic variable representing possible outcomes from a data stream at time t and let x_t denote a random sample (realization). Further, let $f_t(x)$ represent the distribution of X_t and $Q_t(q)$ the quantile associated with probability q, i.e $P(X_t \leqslant Q_t(q)) = q$.

This paper will focus on joint tracking of quantiles for K different probabilities $q_1, q_2, \ldots, q_K$. A straight-forward approach would be to run K quantile tracking algorithms independently of each other, but in this case, the joint properties of the quantiles will not be be enforced and even the monotone property of quantiles may get violated in this case. We illustrate this for the deterministic based multiplicative incremental quantile estimator (DUMIQE) approach found in [46].[2] Assume that the monotone property is satisfied and that a sample x_t admits a value between $\widehat{Q}_t(q_k)$ and $\widehat{Q}_t(q_{k+1})$, i.e

$$\widehat{Q}_t(q_1) \leqslant \cdots \leqslant \widehat{Q}_t(q_k) < x_t < \widehat{Q}_t(q_{k+1}) \leqslant \cdots \leqslant \widehat{Q}_t(q_K) \tag{1}$$

The estimates are then updated as follows

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

which means that the estimates are increased for the quantiles with an estimate below x_t and decreased for the estimates above x_t. Consequently, the monotone property might get violated.

Please note that the DUMIQE algorithm assumes that $\widehat{Q}_t(q_k) > 0 \forall k, t$ which is not useful if the true quantiles are negative for some t. To be able to efficiently track any quantile, [46] suggested two simple solutions. The first is based on tracking a phantom variable $D_t = X_t + \Delta_t$ where Δ_t is iteratively updated such that $D_t > Q_{\min} > 0$. The second approach is based on combining the DUMIQE above for positive quantiles and a modified version for negative quantiles.

4. Joint tracking of multiple quantiles

In Sections 5 and 6 we present a general framework to extend incremental quantile algorithms to perform joint tracking of multiple quantiles. The extensions are based on first tracking a central quantile of the distribution, say the median, before tracking the other quantiles in *relative* manner to the central quantile by taking advantage of *conditional* quantiles. E.g. tracking the 25% quantile reduces to tracking the median of observations below the estimates of the median.

We adopt this concept in order to extend the DUMIQE and QEWA algorithms [18], respectively. We will take advantage of the following general properties.

Property 1. Define Y as a shifted variable of $X, X = Y + \delta$ and let $Q_Y(q)$ denote the q quantile of Y, then $Q_X(q) = Q_Y(q) + \delta$. This means that if Y is a shifted variable of X, a similar shift is observed in the quantiles. This follows from

$$q = P(X < Q_X(q)) = P(Y < Q_Y(q)) = P(Y + \delta < Q_Y(q) + \delta) = P(X < Q_Y(q) + \delta)$$

Property 2. Let $q_1 < q_2$. From the conditional probability

$$P(X < Q_X(q_1) | X < Q_X(q_2)) = \frac{P(X < Q_X(q_1))}{P(X < Q_X(q_2))} = \frac{q_1}{q_2} \tag{3}$$

We see that the q_1 quantile of $X, Q_X(q_1)$, is equal to the q_1/q_2 quantile of the conditional variable $X|X < Q_X(q_2)$ with truncated distribution $f(x|x < Q_X(q_2))$. Applying Eq. 3 on a shifted variable $Y = X - Q_X(q_2)$ gives

$$P(Y < Q_Y(q_1) | Y < 0) = \frac{q_1}{q_2} \tag{4}$$

using that $Q_Y(q_2) = 0$ (Property 1). This means that the q_1 quantile of Y is equal to the q_1/q_2 quantile of the conditional variable $Y|Y < 0$. Conditioning in the opposite direction

$$P(X < Q_X(q_2) | X > Q_X(q_1)) = \frac{P(X < Q_X(q_2) \cap X > Q_X(q_1))}{P(X > Q_X(q_1))} = \frac{q_2 - q_1}{1 - q_1} \tag{5}$$

Similarly, we also observe that the q_2 quantile of X is equal to the $(q_2 - q_1)/(1 - q_1)$ quantile of the conditional variable $X|X > Q(q_1)$ with truncated distribution $f(x|x > Q(q_1))$. Finally, applying Eq. 5 on a shifted variable $Y = X - Q_X(q_1)$ gives

$$P(Y < Q_Y(q_2) | Y > 0) = \frac{q_2 - q_1}{1 - q_1} \tag{6}$$

[2] Cao et al. [5] illustrate this issue.

Property 3. For the DUMIQE algorithm, we observe from Eq. 2 that the update rule is multiplicative, and consequently whenever $\widehat{Q}_0(q) > 0$ all following estimates will be strictly positive.

5. Extension of the DUMIQE algorithm

We start by introducing the algorithm for $K = 2$ before extending it to the general case where $K > 2$.

5.1. Tracking of two quantiles

Assume that $q_1 < q_2$. The idea behind the extension is to first estimate $Q_t(q_1)$ and then estimate $Q_t(q_2)$ relative $Q_t(q_1)$ in such a way that the monotone property will be satisfied. Define a shifted variable $Y_{t,1} = X_t - \widehat{Q}_{t+1}(q_1)$ and let $\widehat{Q}_{Y,t}(q_2)$ denote an estimate of the q_2 quantile of $Y_{t,1}$. The algorithm is initiated with $\widehat{Q}_0(q_1) > 0$[3] and $\widehat{Q}_{Y,0}(q_2) > 0$ and consists of the following steps:

1.1 Update $\widehat{Q}_t(q_1)$ using the DUMIQE update rule

$$\begin{aligned}\widehat{Q}_{t+1}(q_1) &\leftarrow (1 + \lambda q_1)\widehat{Q}_t(q_1) \quad \text{if} \quad \widehat{Q}_t(q_1) < x_t \\ \widehat{Q}_{t+1}(q_1) &\leftarrow (1 - \lambda(1 - q_1))\widehat{Q}_t(q_1) \quad \text{if} \quad \widehat{Q}_t(q_1) \geqslant x_t\end{aligned}$$

1.2 Compute the shifted observation $y_{t,1} \leftarrow x_t - \widehat{Q}_{t+1}(q_1)$.

1.3 Update the q_2 quantile of the shifted variable

$$\begin{aligned}\widehat{Q}_{Y,t+1}(q_2) &\leftarrow (1 + \gamma q_2)\widehat{Q}_{Y,t}(q_2) \quad \text{if} \quad \widehat{Q}_{Y,t}(q_2) < y_{t,1} \\ \widehat{Q}_{Y,t+1}(q_2) &\leftarrow (1 - \gamma(1 - q_2))\widehat{Q}_{Y,t}(q_2) \quad \text{if} \quad \widehat{Q}_{Y,t}(q_2) \geqslant y_{t,1}\end{aligned}$$

1.4 Finally get an estimate for $Q_{t+1}(q_2)$ by shifting back

$$\widehat{Q}_{t+1}(q_2) \leftarrow \widehat{Q}_{Y,t+1}(q_2) + \widehat{Q}_{t+1}(q_1) \tag{7}$$

From Property 1 it follows that $\widehat{Q}_{t+1}(q_2)$ will be an estimate of $Q_{t+1}(q_2)$. Further, due to the multiplicative update form of DUMIQE, it follows that $\widehat{Q}_{Y,t}(q_2)$ is positive for every t and therefore from Eq. 7 it follows that $\widehat{Q}_t(q_1) < \widehat{Q}_t(q_2)$ which means that the monotone property will be satisfied in every iteration.

The quantiles can also be estimated in the opposite order, i.e. by first estimating $Q_t(q_2)$ and then $Q_t(q_1)$ relative to $Q_t(q_2)$. Define the shifted variable $Y_{t,2} = \widehat{Q}_{t+1}(q_2) - X_t$ and let $\widehat{Q}_{Y,t}(q_1)$ denote an estimate of the q_1 quantile of $Y_{t,2}$. The algorithm is initiated with $\widehat{Q}_0(q_2) > 0$ and $\widehat{Q}_{Y,0}(q_1) > 0$ and consists of the following updates:

2.1 Update $\widehat{Q}_t(q_2)$ using the DUMIQE update rule

$$\begin{aligned}\widehat{Q}_{t+1}(q_2) &\leftarrow (1 + \lambda q_2)\widehat{Q}_t(q_2) \quad \text{if} \quad \widehat{Q}_t(q_2) < x_t \\ \widehat{Q}_{t+1}(q_2) &\leftarrow (1 - \lambda(1 - q_2))\widehat{Q}_t(q_2) \quad \text{if} \quad \widehat{Q}_t(q_2) \geqslant x_t\end{aligned}$$

2.2 Compute the shifted observation $y_{t,2} \leftarrow \widehat{Q}_{t+1}(q_2) - x_t$.

2.3 Update the q_1 quantile of the shifted variable

$$\begin{aligned}\widehat{Q}_{Y,t+1}(q_1) &\leftarrow (1 + \gamma q_1)\widehat{Q}_{Y,t}(q_1) \quad \text{if} \quad \widehat{Q}_{Y,t}(q_1) < y_{t,2} \\ \widehat{Q}_{Y,t+1}(q_1) &\leftarrow (1 - \gamma(1 - q_1))\widehat{Q}_{Y,t}(q_1) \quad \text{if} \quad \widehat{Q}_{Y,t}(q_1) \geqslant y_{t,2}\end{aligned}$$

2.4 Finally get an estimate for $Q_{t+1}(q_1)$ by shifting back

$$\widehat{Q}_{t+1}(q_1) \leftarrow \widehat{Q}_{t+1}(q_2) - \widehat{Q}_{Y,t+1}(q_1) \tag{8}$$

Again since $\widehat{Q}_{Y,t}(q_2)$ is positive it follows from Eq. 8 that $\widehat{Q}_t(q_1) < \widehat{Q}_t(q_2)$ for every t.

[3] We refer to [46] for modifications of DUMIQE to estimate quantiles with negative values.

5.2. Tracking of multiple quantiles

The algorithm for a general K is given in Algorithm 1. In step 2, q_c refers to a central probability of the data stream distribution and typically $q_c = 0.5$ which means that $\widehat{Q}_t(q_c)$ is an estimate of the median. The function $\text{DUMIQE}\left(\widehat{Q}_t(q_c), x_t, q_c, \lambda\right)$ refers to one update with the DUMIQE algorithm.

In steps 3 to 7, the procedure 2.1 to 2.4 in the previous section is repeatedly performed. First $\widehat{Q}_{t+1}(q_{c-1})$ is updated such that $\widehat{Q}_{t+1}(q_{c-1}) < \widehat{Q}_{t+1}(q_c)$, then $\widehat{Q}_{t+1}(q_{c-2})$ such that $\widehat{Q}_{t+1}(q_{c-2}) < \widehat{Q}_{t+1}(q_{c-1})$ and so on. In steps 8 to 12, the procedure 1.1 to 1.4 is repeatedly performed. In this way, the monotone property $\widehat{Q}_t(q_1) \leqslant \cdots \leqslant \widehat{Q}_t(q_K)$ will be maintained.

Algorithm 1. Joint quantiles tracking algorithm based on an extension of DUMIQE.

Input:
$x_1, x_2, x_3, \ldots$ // Data stream
λ, γ, K, c
$0 < \widehat{Q}_0(q_1) < \cdots < \widehat{Q}_0(q_K)$
$0 < \widehat{Q}_{Y,0}(q_1) < \cdots < \widehat{Q}_{Y,0}(q_K)$
Method:
1: **for** $n \in 1, 2, \ldots$ **do**
2: $\widehat{Q}_{t+1}(q_c) \leftarrow \text{DUMIQE}\left(\widehat{Q}_t(q_c), x_t, q_c, \lambda\right)$
3: **for** $k \in c-1, \ldots, 1$ **do**
4: $y_{t,k+1} \leftarrow \widehat{Q}_{t+1}(q_{k+1}) - x_t$
5: $\widehat{Q}_{Y,t+1}(q_k) \leftarrow \text{DUMIQE}\left(\widehat{Q}_{Y,t}(q_k), y_{t,k+1}, q_k, \gamma\right)$
6: $\widehat{Q}_{t+1}(q_k) \leftarrow \widehat{Q}_{t+1}(q_{k+1}) - \widehat{Q}_{Y,t+1}(q_k)$
7: **end for**
8: **for** $k \in c+1, \ldots, K$ **do**
9: $y_{t,k+1} \leftarrow x_t - \widehat{Q}_{t+1}(q_{k-1})$
10: $\widehat{Q}_{Y,t+1}(q_k) \leftarrow \text{DUMIQE}\left(\widehat{Q}_{Y,t}(q_k), y_{t,k-1}, q_k, \gamma\right)$
11: $\widehat{Q}_{t+1}(q_k) \leftarrow \widehat{Q}_{Y,t+1}(q_k) + \widehat{Q}_{t+1}(q_{k-1})$
12: **end for**
13: **end for**

6. Extension of the QEWA algorithm

The QEWA algorithm consists of the following updates [18]:

- $\widehat{Q}_{t+1}(q) \leftarrow \left(1 - \widehat{b}_t\right)\widehat{Q}_t(q) + \widehat{b}_t x_t$ (9)
- If$x_t > \widehat{Q_t}(q)$

$-\widehat{\mu}^+_{t+1} \leftarrow \widehat{Q}_{t+1}(q) - \widehat{Q}_t(q) + (1-\rho)\widehat{\mu}^+_t + \rho x_t$ (10)
$-\widehat{\mu}^-_{t+1} \leftarrow \widehat{Q}_{t+1}(q) - \widehat{Q}_t(q) + \widehat{\mu}^-_t$ (11)

- Else

$-\widehat{\mu}^+_{t+1} \leftarrow \widehat{Q}_{t+1}(q) - \widehat{Q}_t(q) + \widehat{\mu}^+_t$ (12)
$-\widehat{\mu}^-_{t+1} \leftarrow \widehat{Q}_{t+1}(q) - \widehat{Q}_t(q) + (1-\rho)\widehat{\mu}^-_t + \rho x_t$ (13)

- $\widehat{a}_{t+1} \leftarrow \dfrac{q}{\widehat{\mu}^+_{t+1} - \widehat{Q}_{t+1}(q)} \Bigg/ \left(\dfrac{q}{\widehat{\mu}^+_{t+1} - \widehat{Q}_{t+1}(q)} + \dfrac{1-q}{\widehat{Q}_{t+1}(q) - \widehat{\mu}^-_{t+1}}\right)$ (14)
- $\widehat{b}_{t+1} \leftarrow \lambda\left(\widehat{a}_{t+1} + I\left(x_t \leqslant \widehat{Q}_{t+1}(q)\right)(1 - 2\widehat{a}_{t+1})\right)$ (15)

where $\widehat{\mu}^+_{t+1}$ and $\widehat{\mu}^-_{t+1}$ represent estimates of the conditional expectations $\mu^+ = E\left(X_t | X_t > \widehat{Q}_t(q)\right)$ and $\mu^- = E\left(X_t | X_t < \widehat{Q}_t(q)\right)$, respectively. From Eq. 9, we see that the estimator is in fact a generalized exponentially weighted average with weights $0 < b_t < 1$. The weights, b_t, are computed such that the estimator tracks the quantile $Q_t(q)$ and not the expectation $E(X_t)$ of the data stream distribution [3][4]. We refer to [18] for further details.

We start by presenting the algorithm for $K = 2$ quantiles before extending to the general case of $K > 2$.

6.1. Tracking of two quantiles

The procedures will consist of the same steps as in Section 5.1, except that steps 1.3 and 2.3 invoke the conditional quantiles property (Property 2) and thus are slightly more involved. The two procedures will be presented in the opposite order. Let $\widehat{Q}_{Y,t}(q_1)$ denote an estimate of the q_1 quantile of the shifted variable $Y_{t,2} = X_t - \widehat{Q}_{t+1}(q_2)$ (opposite sign from above). The algorithm consists of the following updates[5]:

1.1 Update $\widehat{Q}_t(q_2)$ using the QEWA update rules in Eqs. (9)–(13), (15) to obtain an estimate $\widehat{Q}_{t+1}(q_2)$.
1.2 Compute the shifted observation $y_{t,2} \leftarrow x_t - \widehat{Q}_{t+1}(q_2)$.
1.3 Update $Q_{Y,t}(q_1)$ by updating the q_1/q_2 quantile of the conditional variable $Y_{t,2} | Y_{t,2} < 0$ (recall Eq. 4). Thus if $y_{t,2} < 0$, apply the update rules in Eqs. (9)–(13), (15) with $q = q_1/q_2$ to obtain a quantile estimate $\widehat{Q}_{Y,t+1}(q_1)$. Further, if $y_{t,2} > 0$, then we are outside of the support of the conditional variable and therefore skip updating the quantile estimate, i.e. $\widehat{Q}_{Y,t+1}(q_1) \leftarrow \widehat{Q}_{Y,t}(q_1)$.
1.4 Finally shift back

$$\widehat{Q}_{t+1}(q_1) \leftarrow \widehat{Q}_{Y,t+1}(q_1) + \widehat{Q}_{t+1}(q_2)$$

We shall now prove that the above described algorithm ensures the monotone property, $\widehat{Q}_t(q_1) < \widehat{Q}_t(q_2)$. We start proving, by induction, that if we initialise the algorithm with $\widehat{Q}_{Y,0}(q_1) < 0$, then every $\widehat{Q}_{Y,t}(q_1)$ will also be negative. Assume that $\widehat{Q}_{Y,t}(q_1) < 0$. If $y_{t,2} < 0$, $\widehat{Q}_{Y,t+1}(q_1)$ is computed using Eq. 9. Since both $\widehat{Q}_{Y,t}(q_1) < 0$ and $y_{t,2} < 0$, consequently, $\widehat{Q}_{Y,t+1}(q_1) < 0$ (convex combination of two negative values). If $y_{t,2} > 0$, then $\widehat{Q}_{Y,t+1}(q_1) \leftarrow \widehat{Q}_{Y,t}(q_1)$, which again implies $\widehat{Q}_{Y,t+1}(q_1) < 0$. From Step 1.4, this again implies that the monotone property $\widehat{Q}_t(q_1) < \widehat{Q}_t(q_2)$ is satisfied.

The same quantiles can also be updated in the opposite order. Let $\widehat{Q}_{Y,t}(q_2)$ denote an estimate of the q_2 quantile of the shifted variable $Y_{t,1} = X_t - \widehat{Q}_{t+1}(q_1)$. The algorithm consists of the following updates:

2.1 Update $\widehat{Q}_t(q_1)$ using the QEWA update rules in Eqs. (9)–(13), (15) to obtain an estimate $\widehat{Q}_{t+1}(q_1)$.
2.2 Compute the shifted observation $y_{t,1} \leftarrow x_t - \widehat{Q}_{t+1}(q_1)$.
2.3 Update $Q_{Y,t}(q_2)$ by updating the $(q_2 - q_1)/(1 - q_1)$ quantile of the conditional variable $Y_{t,1} | Y_{t,1} > 0$ (recall Eq. 6). Thus if $y_{t,1} > 0$, we update using Eqs. (9)–(13), (15) with $q = (q_2 - q_1)/(1 - q_1)$ to obtain a quantile estimate $\widehat{Q}_{Y,t+1}(q_2)$. If $y_{t,1} < 0$, no update: $\widehat{Q}_{Y,t+1}(q_2) \leftarrow \widehat{Q}_{Y,t}(q_2)$.
2.4 Shift back (Property 1).

$$\widehat{Q}_{t+1}(q_2) \leftarrow \widehat{Q}_{Y,t+1}(q_2) + \widehat{Q}_{t+1}(q_1)$$

Following the same lines of reasoning as the first algorithm, this algorithm can be easily shown to ensure the monotone property.

6.2. Tracking of multiple quantiles

The algorithm for a general K is described in Algorithm 2 where $\widehat{\mu}^-_{c,t}$ and $\widehat{\mu}^+_{c,t}$ refer to estimates of the conditional expectations $E\left(X_t | X_t < \widehat{Q}_t(q_c)\right)$ and $E\left(X_t | X_t > \widehat{Q}_t(q_c)\right)$, respectively. Further $\widehat{\mu}^-_{Y,k,t}$ and $\widehat{\mu}^+_{Y,k,t}$ refer to estimates of the conditional expectations $E\left(Y_{t,k} | Y_{t,k} < \widehat{Q}_{Y,t}(q_k)\right)$ and $E\left(Y_{t,k} | Y_{t,k} > \widehat{Q}_{Y,t}(q_k)\right)$, respectively. The function $\text{QEWA}\left(\widehat{Q}_t(q_c), \widehat{\mu}^-_{c,t}, \widehat{\mu}^+_{c,t}, x_t, q_c, \lambda, \rho\right)$ refers to one update with the QEWA algorithm (Eqs. 9–15). During steps 3 to 11 and steps 12 to 21, the procedures in 1.1 to 1.4 and 2.1 to 2.4 in the previous section, are repeatedly performed and thus the monotone property $\widehat{Q}_t(q_1) \leqslant \cdots \leqslant \widehat{Q}_t(q_K)$ will be ensured.

[4] If constants weights were used, i.e. $b_t = b$, Eq. 9 would track the expectation $E(X_t)$ and not $Q_t(q)$

[5] Please note that to be able to track a quantile using the QEWA algorithm, the conditional expectations must also be tracked to obtain the weights b_t. For easing the explanation, we only focus on the quantile tracking problem for now.

Algorithm 2. Joint quantiles tracking algorithm based on an extension of QEWA.

Input:
$x_1, x_2, x_3, \ldots$ // Data stream
λ, γ, ρ, K, c
$\widehat{\mu}^-_{c,0} < \widehat{Q}_0(q_c) < \widehat{\mu}^+_{c,0}$
$\widehat{\mu}^-_{Y,k,0} < \widehat{Q}_{Y,0}(q_k) < \widehat{\mu}^+_{Y,k,0} < 0,\ k < c$
$0 < \widehat{\mu}^-_{Y,k,0} < \widehat{Q}_{Y,0}(q_k) < \widehat{\mu}^+_{Y,k,0},\ k > c$
Method:
for $n \in 1, 2, \ldots$ **do**
  $\left(\widehat{Q}_{t+1}(q_c), \widehat{\mu}^-_{c,t+1}, \widehat{\mu}^+_{c,t+1}\right) \leftarrow \text{QEWA}\left(\widehat{Q}_t(q_c), \widehat{\mu}^-_{c,t}, \widehat{\mu}^+_{c,t}, x_t, q_c, \lambda, \rho\right)$
  for $k \in c-1, \ldots, 1$ **do**
    if $x_t < \widehat{Q}_{t+1}(q_{k+1})$
      $y_{t,k+1} \leftarrow x_t - \widehat{Q}_{t+1}(q_{k+1})$
      $\left(\widehat{Q}_{Y,t+1}(q_k), \widehat{\mu}^-_{Y,k,t+1}, \widehat{\mu}^+_{Y,k,t+1}\right) \leftarrow \text{QEWA}\left(\widehat{Q}_{Y,t}(q_k), \widehat{\mu}^-_{Y,k,t}, \widehat{\mu}^+_{Y,k,t}, y_{t,k+1}, q_{k+1}/q_k, \gamma, \rho\right)$
    else
      $\left(\widehat{Q}_{Y,t+1}(q_k), \widehat{\mu}^-_{Y,k,t+1}, \widehat{\mu}^+_{Y,k,t+1}\right) \leftarrow \left(\widehat{Q}_{Y,t}(q_k), \widehat{\mu}^-_{Y,k,t}, \widehat{\mu}^+_{Y,k,t}\right)$
    end if
      $\widehat{Q}_{t+1}(q_k) \leftarrow \widehat{Q}_{Y,t+1}(q_k) + \widehat{Q}_{t+1}(q_{k+1})$
  end for
  for $k \in c+1, \ldots, K$ **do**
    if $x_t > \widehat{Q}_{t+1}(q_{k-1})$
      $y_{t,k-1} \leftarrow x_t - \widehat{Q}_{t+1}(q_{k-1})$
      $\left(\widehat{Q}_{Y,t+1}(q_k), \widehat{\mu}^-_{Y,k,t+1}, \widehat{\mu}^+_{Y,k,t+1}\right) \leftarrow$
        $\text{QEWA}\left(\widehat{Q}_{Y,t}(q_k), \widehat{\mu}^-_{Y,k,t}, \widehat{\mu}^+_{Y,k,t}, y_{t,k-1}, (q_k - q_{k-1})/(1 - q_{k-1}), \gamma, \rho\right)$
  else
    $\left(\widehat{Q}_{Y,t+1}(q_k), \widehat{\mu}^-_{Y,k,t+1}, \widehat{\mu}^+_{Y,k,t+1}\right) \leftarrow \left(\widehat{Q}_{Y,t}(q_k), \widehat{\mu}^-_{Y,k,t}, \widehat{\mu}^+_{Y,k,t}\right)$
  end if
    $\widehat{Q}_{t+1}(q_k) \leftarrow \widehat{Q}_{Y,t+1}(q_k) + \widehat{Q}_{t+1}(q_{k-1})$
  end for
end for

We end this section with few pertinent remarks.

Remark 1. The estimate $\widehat{Q}_t(q_c)$, in Algorithms 1 and 2, tracks the overall trend of the data stream, while the other quantiles are updated relative to $\widehat{Q}_t(q_c)$ and only need to track changes in *shape* of the data stream distribution. Thus, for most dynamic data streams it is natural to use a step size λ that is on a larger scale than γ. As we will see later, our experiments shows that the performance of the algorithm is not sensitive to the value of the γ parameter. Any value of γ somewhere between $1/10$ and $1/10,000$ performed well in all our experiments. Furthermore, the performance of Algorithm 2 is not sensitive to the value of ρ. Following the recommendation in [18] of using $\rho = 0.01\lambda$, Algorithm 2 performed well in all our experiments.

Remark 2. The theoretical results reported in [46,18] prove that the DUMIQE and QEWA estimators converge to the true quantiles for static data streams. Although the latter two algorithms are designed to cope with dynamic environments, it is a fundamental requirement that the they converge to the true quantile for static data streams. These theoretical results again imply that Algorithms 1 and 2 will converge to the true quantiles for static data streams.

7. Experiments

In this section, we evaluate the performance of Algorithms 1 and 2 for both synthetic and real life data sets. We will denote Algorithms 1 and 2 by ShiftQ and CondQ, respectively (since computation of quantiles are based on SHIFTed and

CONDitional stochastic variables). We compare against the legacy multiple quantiles tracking algorithm that we are aware of in the literature, namely the method of Cao et al. in [5] and the MDUMIQE by Hammer et al. in [20].

7.1. Synthetic experiments

Fig. 1 shows a small section of the tracking processes for the algorithm DUMIQE, MDUMIQE, CondQ and ShiftQ. The gray dots show the data stream, that is the same in all the four panels. The black, gray and green curves illustrate tracking of the 0.4, 0.5 and the 0.6 quantiles of the data, respectively. We observe that when using DUMIQE, the monotone property is violated while the three other algorithms are able to satisfy the monotone property. The idea behind the MDUMIQE is to reduce the step size to avoid monotone property violations. Consequently, as shown on the upper right panel, if the data stream changes rapidly, the MDUMIQE is not able to track efficiently since the adjusted step sizes become too small. We see that both CondQ and ShiftQ are able to efficiently track the dynamics of the data stream. However, CondQ is able to track the dynamics with less noise and thus in conclusion seems the most efficient algorithm.

We now proceed to a more thorough evaluation of the suggested algorithms. The algorithms are designed to perform well for dynamically changing data streams and the experiments will focus on testing this hypothesis. We considered four different data cases. For the first case, the data stream distributions were normally distributed and the expectations, μ_t, varied smoothly as follows

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

which is a sinus function with period T. For the second case, the data stream distributions were also normally distributed, but the expectation switched between values a and $-a$

$$\mu_t = \begin{cases} a & \text{if} \quad n \mod T \leqslant T/2 \\ -a & \text{else} \end{cases}$$

The standard deviation was set to one. For the two remaining cases, the data stream distributions were χ^2 distributed, one with smooth changes and one with rapid switches. For the smooth case the number of degrees of freedom, v_t, varied with time as follows

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

where $b > a$ such that $v_t > 0$ for all t. For the switch case, the number of degrees of freedom switched between values $a + b$ and $-a + b$

$$\mu_t = \begin{cases} a + b & \text{if} \quad n \bmod T \leqslant T/2 \\ -a + b & \text{else} \end{cases}$$

In the experiments we used $a = 2$ and $b = 6$. The χ^2 distribution is quite heavy tailed and both the scale and the shape of the distribution change with time rendering this a challenging quantile tracking problem.

We considered the two periods $T = 100$ (rapid variation) and $T = 1000$ (slow variation). For each data stream, either $K = 3$ or $K = 19$ quantiles were tracked. For the $K = 3$ case, quantiles associated with the probabilities $q_1 = 0.2, q_2 = 0.5$ and $q_3 = 0.8$ quantiles were tracked. For the $K = 19$ case, quantiles associated with the probabilities $q_k = 0.05k, k = 1, \ldots, 19$ were tracked.

Tracking error was measured using the average root mean squared error for the different quantiles

$$\text{RMSE} = \frac{1}{K}\sum_{k=1}^{K}\sqrt{\frac{1}{N}\sum_{t=1}^{N}\left(Q_t(q_k) - \widehat{Q}_t(q_k)\right)^2}$$

where N is the total number of samples received from the data stream. In the experiments, we used $N = 10^6$ to remove any Monte Carlo errors. Following the recommendations in [18], for the QEWA estimator in the CondQ algorithm we used a ρ/λ ratio equal to 1/100.

In a practical situation, the history of the data stream can be used to find fairly optimal values of the tuning parameters in algorithms as performed in [14]. Thus, the main focus will be on the performance of the algorithms under optimal values of the tuning parameters. Complete results, showing estimation errors for a wide range of values of the tuning parameters, are given in Figs. 3–6 in Appendix A.

Tables 1–4 show estimation error for the different algorithms using an optimal step length. We see that the CondQ consistently outperforms all the other algorithms in each of the 16 cases (data streams) and mostly with a clear margin. For most of the cases, both CondQ and ShiftQ outperform the algorithm by Cao et al. and the MDUMIQE. Further, we see that the performance of CondQ and ShiftQ are not sensitive to the choice of the tuning parameter γ. Any of the value of γ between 0.0001 and 0.1, performed well in all the experiments. This makes it easy to tune the CondQ and ShiftQ algorithms.

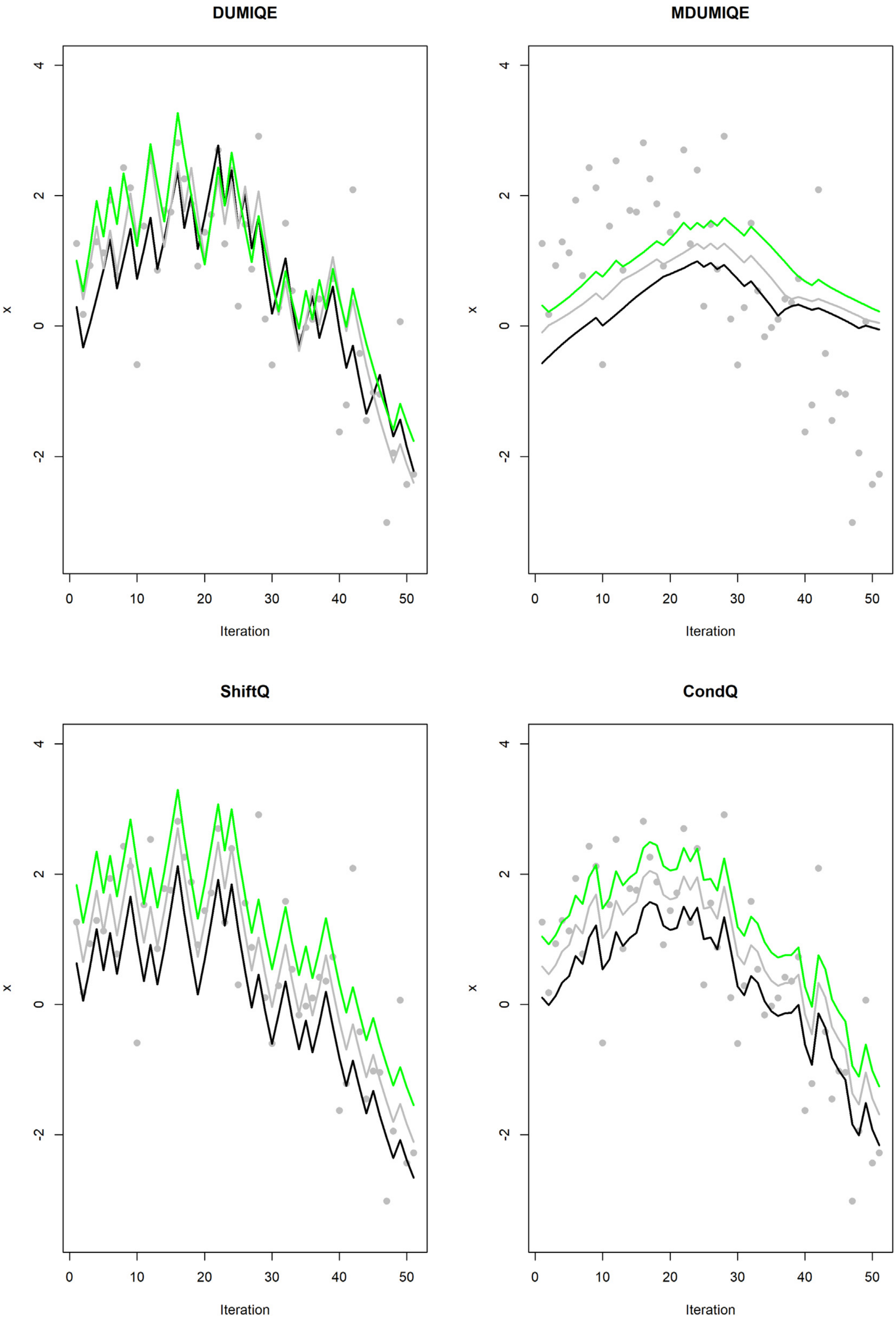

Fig. 1. Estimation processes for DUMIQE, MDUMIQE, CondQ and ShiftQ. The gray dots show samples from the data stream distribution while the black, gray and green curves show estimates of the 0.4, 0.5 and the 0.6 quantiles of the data stream, respectively.

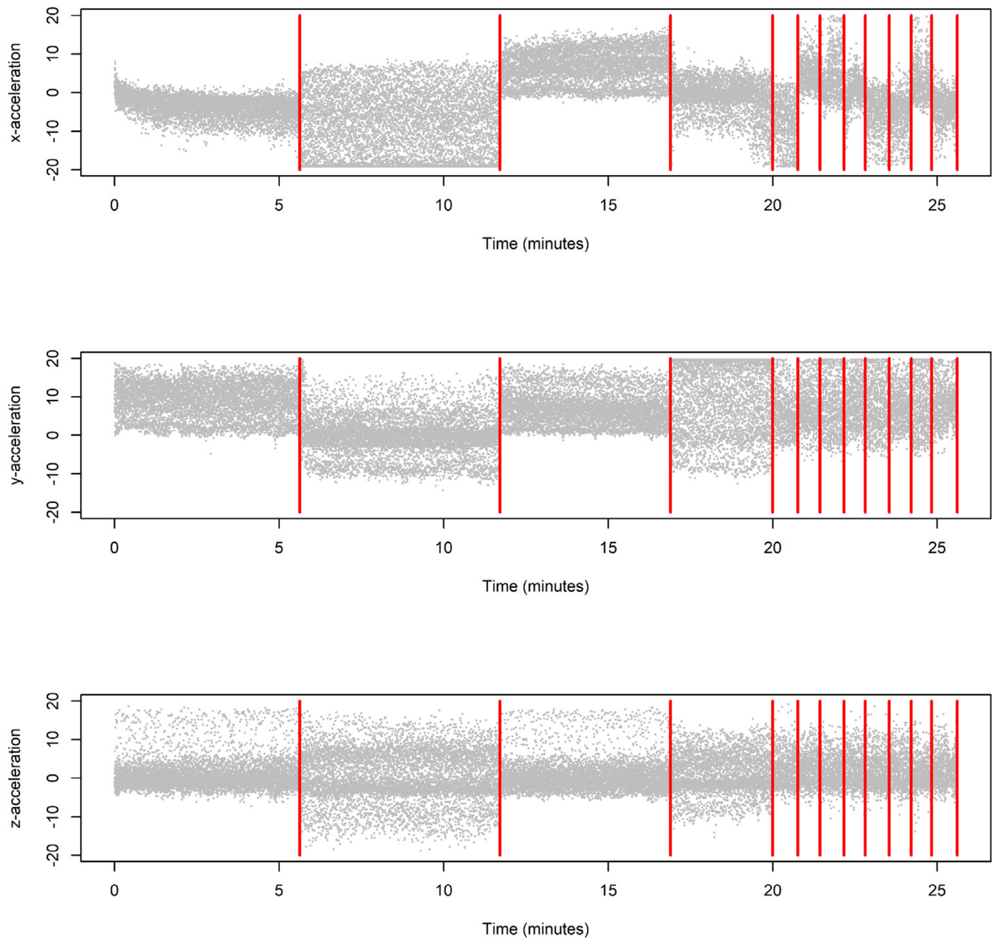

Fig. 2. The gray dots show accelerometer observations for an arbitrary user. The red lines mark the time instants when the user changes activity.

7.2. Real-life data streams – activity change detection

Activity recognition is a popular machine learning task where the goal is to use sensors to automatically detect and identify the activity of a user. For instance, one could use activity recognition to assess whether a person is performing a healthy amount of exercise, and to detect falls. In this experiment, we will focus on identifying changes in activities using accelerometer data which is available on almost any smart cell phone today.

We consider an accelerometer dataset from the Wireless Sensor Data Mining (WISDM) project [27]. Accelerations in x, y and z directions were observed, with a frequency of 20 observations per second, while users where performing the activities walking, jogging, walking up a stairway and walking down a stairway. The dataset corresponds to activities from 36 users and contains a total of 989 875 observations.

Current research focuses on supervised approaches where historic and annotated activity observations are used to train an activity classification model [3]. For instance, the work reported in [27] trained models like decision trees and neural networks. In this example, we rather adopt an unsupervised approach where the goal is to detect whenever a user changes activity. Since we receive 20 accelerometer observations per second, it is important that the streaming approach is computationally efficient.

Change detection is useful as part of a sequential supervised scheme. Whenever a change is detected, the observations from the last activity can be classified and the supervised classifier should be retrained. If the supervised learner is sufficiently uncertain about the activity type of the last activity, the system may ask the user for feedback in an online manner to gradually improve performance.

Fig. 2 shows in gray x, y and z acceleration for an arbitrary user. The red lines mark the time instants when the user changed activity. Mostly, the acceleration distributions are fairly stationary within an activity, but with some gradual and abrupt changes. The users often changed activities as often as every 30 s making the problem in hand a challenging tracking and change detection problem.

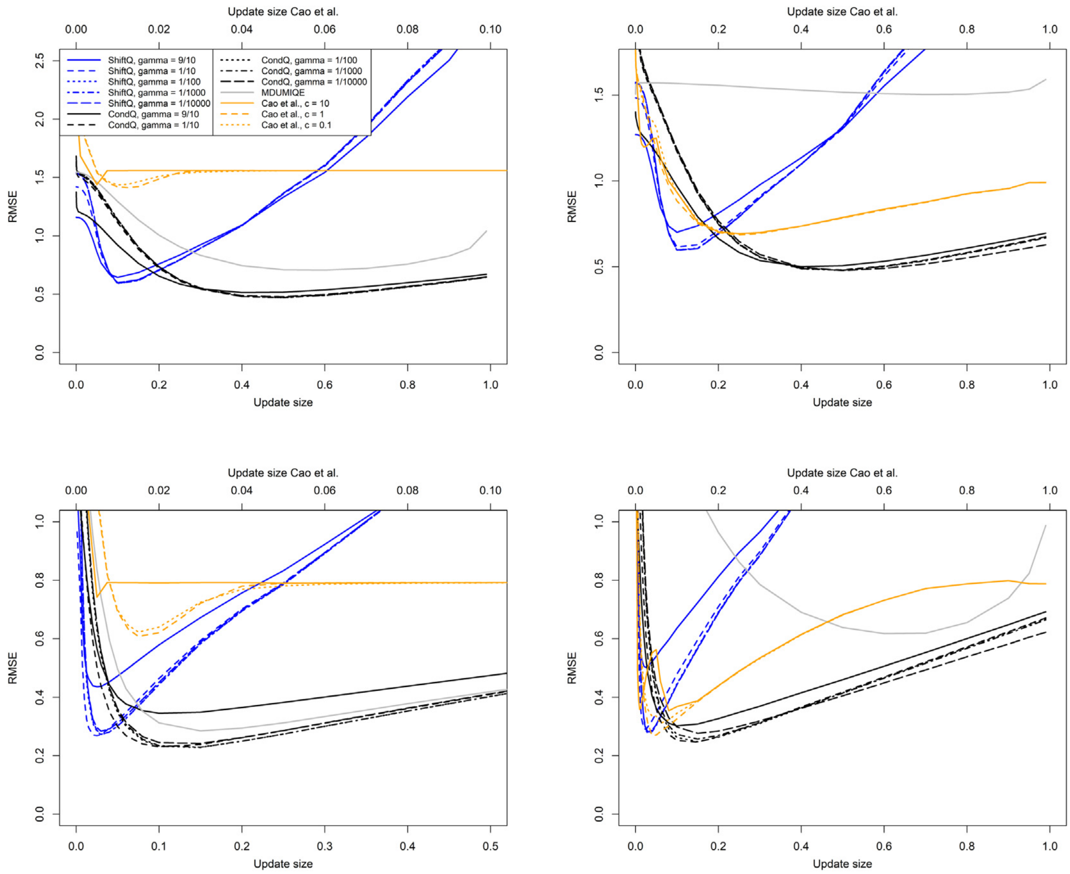

Fig. 3. Normal distribution periodic case: The left and right columns show results for $K = 3$ and $K = 19$, respectively. The top and bottom rows show results for periods $T = 100$ and 1000, respectively.

We suggest the following change detection procedure. Let $\widehat{Q}_{t,w}(q)$ represent the q quantile of the data stream accelerometer distribution at time t and in dimension $w \in \{x, y, z\}$.

1. Use a multiple quantiles tracking algorithm to obtain estimates $\widehat{Q}_{t,w}(q_k),\ k = 1, \ldots, K, w = x, y, z$.
2. In each dimension, compute the Euclidean distance between the current quantile estimates and the estimates h seconds back in time

$$\mathrm{ED}_{t,w} = \sqrt{\sum_{k=1}^{K} \left(\widehat{Q}_{t,w}(q_k) - \widehat{Q}_{t-h,w}(q_k) \right)^2}$$

3. In each dimension, characterize the main distributional properties of $\mathrm{ED}_{t,w}$ by tracking the first two moments using exponentially weighted moving averages

$$\hat{\mu}(\mathrm{ED}_{t,w}) = (1 - \xi)\hat{\mu}(\mathrm{ED}_{t-1,w}) + \xi \mathrm{ED}_{t,w}$$
$$\hat{\mu}\left(\mathrm{ED}_{t,w}^2\right) = (1 - \xi)\hat{\mu}\left(\mathrm{ED}_{t-1,w}^2\right) + \xi \mathrm{ED}_{t,w}^2$$
$$\hat{\sigma}(\mathrm{ED}_{t,w}) = \sqrt{\hat{\mu}(\mathrm{ED}_{t,w})^2 - \hat{\mu}\left(\mathrm{ED}_{t,w}^2\right)}$$

4. When the user changes activity, we expect $\mathrm{ED}_{\mathrm{it}\ t,w}$ to rapidly increase in at least one dimension w. Thus, a new activity is detected when $\mathrm{ED}_{t,w}$ is more than η standard deviations higher then $\hat{\mu}(\mathrm{ED}_{t,w})$ in at least one dimension, i.e. if

$$\max_{w} \left\{ \frac{\mathrm{ED}_{t,w} - \hat{\mu}(\mathrm{ED}_{t,w})}{\hat{\sigma}(\mathrm{ED}_{t,w})} \right\} \geqslant \eta$$

5. When a new activity is detected, restart the tracking of the quantile estimates and go back to step 1.

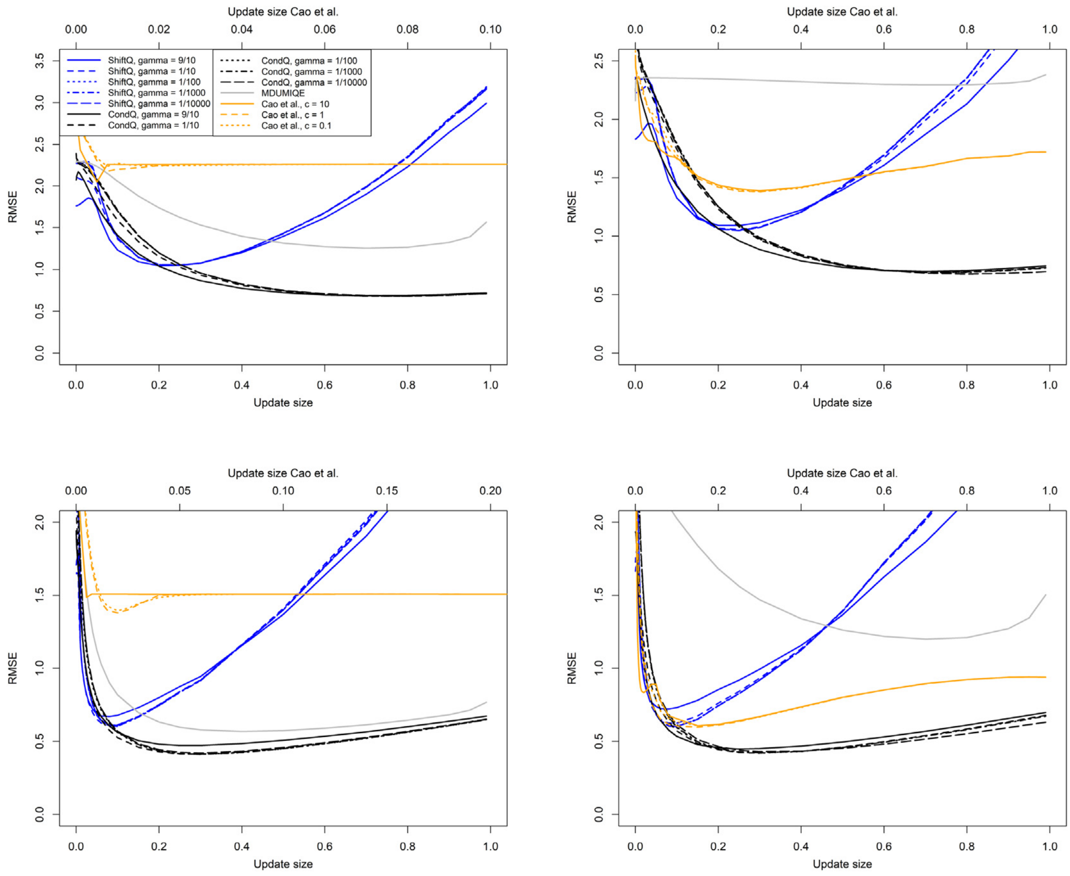

Fig. 4. Normal distribution switch case: The left and right columns show results for $K = 3$ and $K = 19$, respectively. The top and bottom rows show results for periods $T = 100$ and 1000, respectively.

We compare the quantile tracking approach above with tracking the first two moments of the acceleration distribution in each dimension $w \in \{x, y, z\}$ leading to the following approach:

1. Let $x_{t,w}$ denote the observed accelerations at time t in dimension $w \in \{x, y, z\}$. Track the mean and standard deviation in each dimension using exponentially weighted moving average

$$\hat{\mu}(X_{t,w}) = (1 - v)\hat{\mu}(X_{t,w}) + v x_{t,w} \tag{16}$$

$$\hat{\mu}\left(X_{t,w}^2\right) = (1 - v)\hat{\mu}\left(X_{t,w}^2\right) + v x_{t,w}^2 \tag{17}$$

$$\hat{\sigma}(X_{t,w}) = \sqrt{\hat{\mu}(\mathrm{ED}_{t,w})^2 - \hat{\mu}\left(\mathrm{ED}_{t,w}^2\right)} \tag{18}$$

2. In each dimension, compute the Mahalanobis distance (MD) between the current estimate of the mean and the estimate h seconds back in time, $\mathrm{MD}_{t,w} = |\hat{\mu}(X_{t,w}) - \hat{\mu}(X_{t-h,w})|/\hat{\sigma}(X_{t,w})$.
3. In each dimension, characterize the main distributional properties of $\mathrm{MD}_{t,w}$ by tracking the first two moments using exponentially weighted moving averages

$$\hat{\mu}(\mathrm{MD}_{t,w}) = (1 - \xi)\hat{\mu}(\mathrm{MD}_{t-1,w}) + \xi \mathrm{MD}_{t,w}$$

$$\hat{\mu}\left(\mathrm{MD}_{t,w}^2\right) = (1 - \xi)\hat{\mu}\left(\mathrm{MD}_{t-1,w}^2\right) + \xi \mathrm{MD}_{t,w}^2$$

$$\hat{\sigma}(\mathrm{MD}_{t,w}) = \sqrt{\hat{\mu}(\mathrm{MD}_{t,w})^2 - \hat{\mu}\left(\mathrm{MD}_{t,w}^2\right)}$$

4. When the user changes activity, we expect $\mathrm{MD}_{t,w}$ to rapidly increase in at least one dimension w. Thus a new activity is detected when $\mathrm{MD}_{t,w}$ is more than η standard deviations higher then $\hat{\mu}(\mathrm{MD}_{t,w})$ in at least one dimension, i.e. if

$$\max_w \left\{ \frac{\mathrm{MD}_{t,w} - \hat{\mu}(\mathrm{MD}_{t,w})}{\hat{\sigma}(\mathrm{MD}_{t,w})} \right\} \geqslant \eta$$

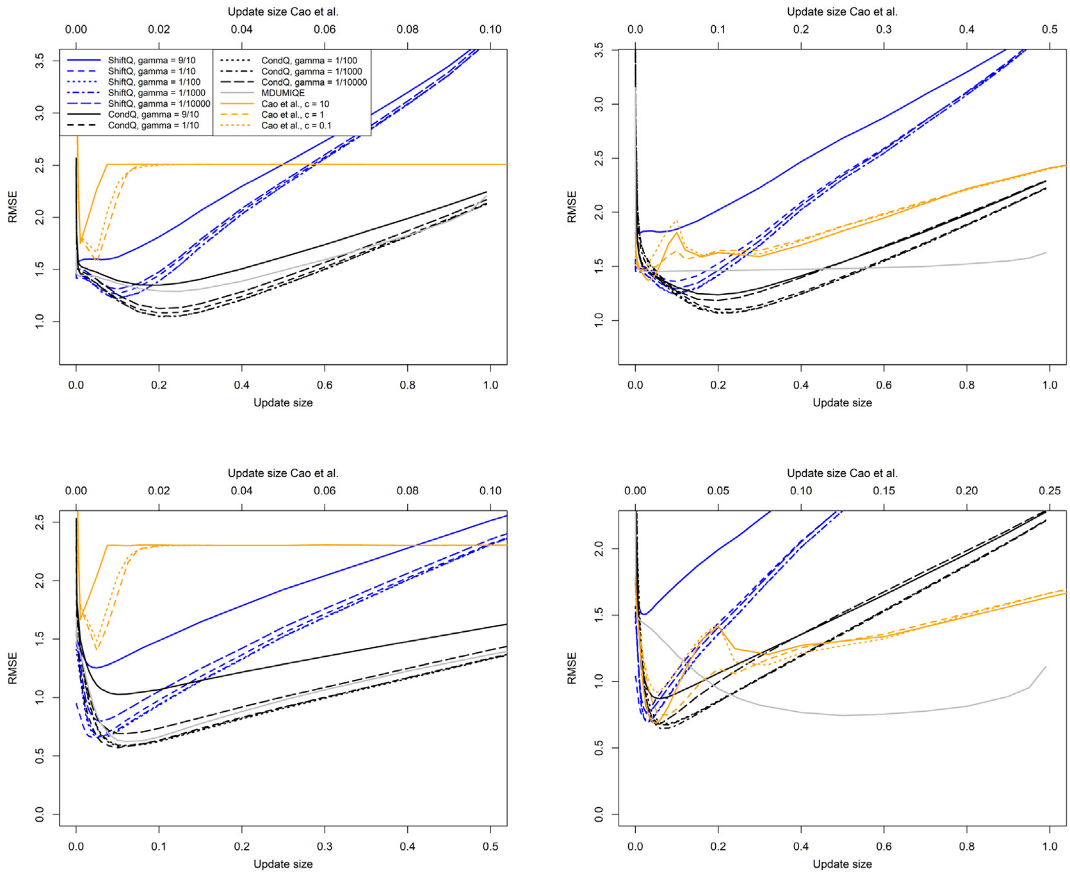

Fig. 5. χ^2 distribution periodic case: The left and right columns show results for $K = 3$ and $K = 19$, respectively. The top and bottom rows show results for periods $T = 100$ and 1000, respectively.

5. When a new activity is detected, restart the tracking of the quantile estimates and go back to step 1.

A disadvantage with the MD approach above is that it only can detect changes in the first two moments of the acceleration distributions, while the quantile tracking approach can detect *any* changes in the distributions.

We measured the performance of the approaches for a wide range of values of the tuning parameters. To properly characterize the acceleration distributions, we tracked a total of $K = 9$ quantiles, namely quantiles associated with the probabilities $0.1, 0.2, \ldots, 0.9$. Given the scale of the observations, choosing small values of λ, β and ν equal to 0.001, 0.005 and 0.01 respectively is a reasonable choice. Further, we used 0.005, 0.01 and 0.05 for ξ, 5, 10 and 15 s for h we used and 10, 15 and 25 for η. Finally, following the recommendation in [18], we used $\rho/\lambda = 0.01$. We ran the change detection approaches for the whole dataset for all the combinations of the tuning parameters resulting in a total of 162 and 81 experiments for the quantile and MD approaches, respectively.

To measure detection performance, we used the well-known measures precision, recall and the F1 score [38]which are commonly been used for event detection tasks, see e.g. [13,9] (biological signals) and [21] (computer vision). If the approach detects more than one change between two true changes, we characterize the first change as a correct detection and the rest as false detections. Then define precision, recall and F1 score

$$\text{Precision} = \frac{\text{No. of correct detections}}{\text{No. of detections}}$$
$$\text{Recall} = \frac{\text{No. of correct detections}}{\text{No. true changes}}$$
$$\text{F1 score} = \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$$

where the F1 score is the harmonic mean of precision and recall.

Tables 5–7 show the top ten results with respect to the F1 score for the different approaches. We see that the CondQ and ShiftQ outperform MD. Further, we see that CondQ detects true changes more rapidly than ShiftQ (last column). Using CondQ for change detection seems to be the best alternative.

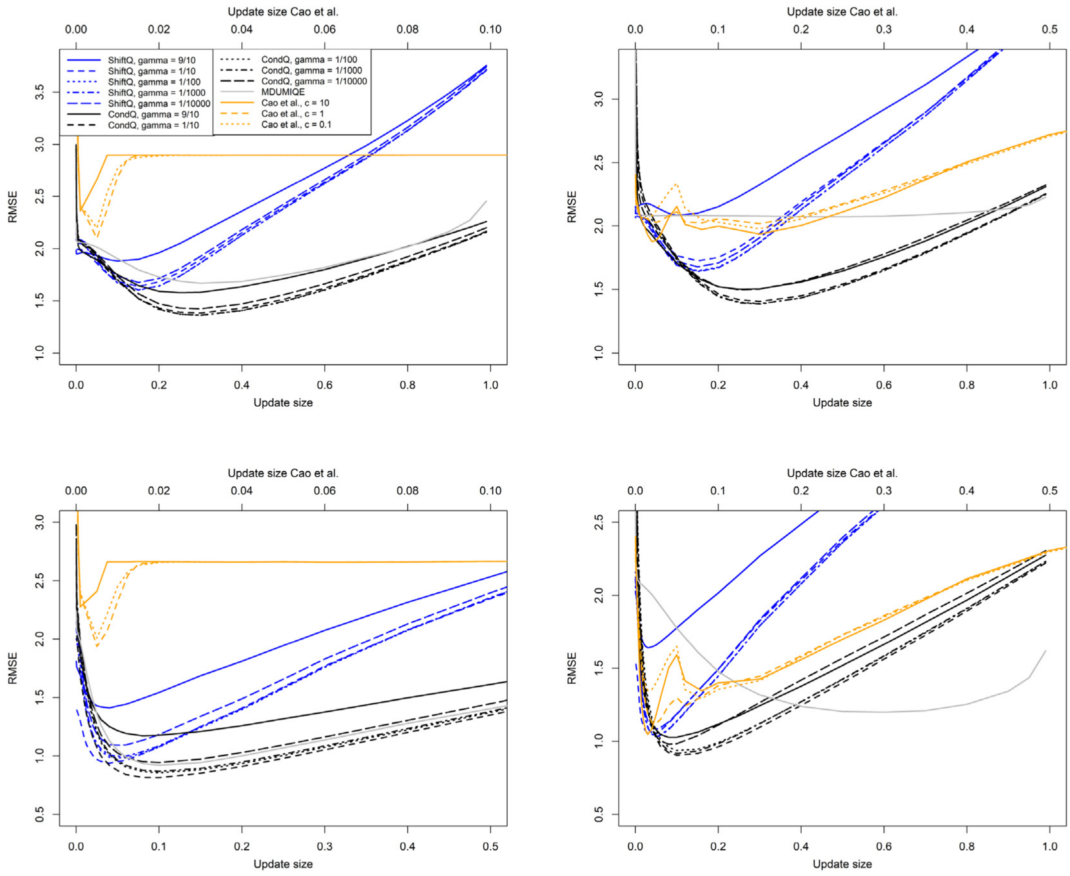

Fig. 6. χ^2 distribution switch case: The left and right columns show results for $K = 3$ and $K = 19$, respectively. The top and bottom rows show results for periods $T = 100$ and 1000, respectively.

Table 1
Normal distribution periodic case: Estimation error for the different algorithms using an optimal value of the update size λ. The values in boldface show for each case the algorithm that is performing the best.

	$K = 3$		$K = 19$	
	$T = 100$	$T = 1000$	$T = 100$	$T = 1000$
ShiftQ, $\gamma = 0.9$	0.644	0.436	0.701	0.502
ShiftQ, $\gamma = 0.1$	0.598	0.269	0.615	0.290
ShiftQ, $\gamma = 0.01$	0.592	0.272	0.597	0.281
ShiftQ, $\gamma = 0.001$	0.592	0.274	0.595	0.276
ShiftQ, $\gamma = 0.0001$	0.597	0.284	0.597	0.282
CondQ, $\gamma = 0.9$	0.514	0.346	0.500	0.302
CondQ, $\gamma = 0.1$	0.475	0.230	0.482	**0.247**
CondQ, $\gamma = 0.01$	**0.471**	**0.229**	0.479	0.248
CondQ, $\gamma = 0.001$	0.472	**0.229**	0.480	0.257
CondQ, $\gamma = 0.0001$	0.479	0.242	**0.478**	0.276
MDUMIQE	0.706	0.285	1.504	0.618
Cao et al., $c = 10$	1.438	0.735	0.693	0.354
Cao et al., $c = 1$	1.412	0.608	0.685	0.269
Cao et al., $c = 0.1$	1.435	0.623	0.692	0.311

8. Closing remark

Incremental quantile estimators document the state-of-the performance in tracking quantiles of dynamically varying data streams. However, due to their incremental nature, they fail to jointly estimate multiple quantiles without violating the monotone property of quantiles. In this paper, we present a new paradigm that enables incremental quantile estimators to obtain joint estimates of multiple quantiles. The fundamental idea relies on first tracking a central quantile and then track-

Table 2
Normal distribution switch case: Estimation error for the different algorithms using an optimal value of the update size λ. The values in boldface show the algorithms resulting in minimum estimation error.

	$K = 3$		$K = 19$	
	$T = 100$	$T = 1000$	$T = 100$	$T = 1000$
ShiftQ, $\gamma = 0.9$	1.045	0.670	1.090	0.722
ShiftQ, $\gamma = 0.1$	1.051	0.608	1.054	0.624
ShiftQ, $\gamma = 0.01$	1.050	0.604	1.047	0.603
ShiftQ, $\gamma = 0.001$	1.050	0.603	1.046	0.601
ShiftQ, $\gamma = 0.0001$	1.054	0.609	1.047	0.601
CondQ, $\gamma = 0.9$	0.685	0.472	0.698	0.448
CondQ, $\gamma = 0.1$	**0.680**	0.412	0.690	**0.420**
CondQ, $\gamma = 0.01$	**0.680**	**0.411**	0.690	**0.420**
CondQ, $\gamma = 0.001$	0.681	**0.411**	0.689	0.422
CondQ, $\gamma = 0.0001$	0.686	0.420	**0.677**	0.430
MDUMIQE	1.255	0.567	2.160	1.200
Cao et al., $c = 10$	2.049	1.487	1.387	0.607
Cao et al., $c = 1$	2.181	1.380	1.377	0.597
Cao et al., $c = 0.1$	2.237	1.395	1.379	0.607

Table 3
χ^2 distribution periodic case: Estimation error for the different algorithms using an optimal value of the update size λ. The values in boldface show the algorithms resulting in minimum estimation error.

	$K = 3$		$K = 19$	
	$T = 100$	$T = 1000$	$T = 100$	$T = 1000$
ShiftQ, $\gamma = 0.9$	1.573	1.252	1.791	1.506
ShiftQ, $\gamma = 0.1$	1.265	0.656	1.362	0.751
ShiftQ, $\gamma = 0.01$	1.220	0.653	1.254	0.702
ShiftQ, $\gamma = 0.001$	1.222	0.670	1.246	0.693
ShiftQ, $\gamma = 0.0001$	1.310	0.798	1.293	0.768
CondQ, $\gamma = 0.9$	1.350	1.026	1.237	0.870
CondQ, $\gamma = 0.1$	1.085	**0.572**	1.103	**0.675**
CondQ, $\gamma = 0.01$	**1.052**	0.584	1.077	0.683
CondQ, $\gamma = 0.001$	1.053	0.591	**1.069**	0.647
CondQ, $\gamma = 0.0001$	1.128	0.692	1.187	0.680
MDUMIQE	1.291	0.625	1.453	0.745
Cao et al., $c = 10$	1.750	1.669	1.374	0.687
Cao et al., $c = 1$	1.569	1.403	1.465	0.748
Cao et al., $c = 0.1$	1.662	1.487	1.477	0.918

Table 4
χ^2 distribution switch case: Estimation error for the different algorithms using an optimal value of the update size λ. The values in boldface show the algorithms resulting in minimum estimation error.

	$K = 3$		$K = 19$	
	$T = 100$	$T = 1000$	$T = 100$	$T = 1000$
ShiftQ, $\gamma = 0.9$	1.882	1.411	2.081	1.640
ShiftQ, $\gamma = 0.1$	1.638	0.939	1.725	1.045
ShiftQ, $\gamma = 0.01$	1.603	0.985	1.647	1.037
ShiftQ, $\gamma = 0.001$	1.605	0.998	1.642	1.032
ShiftQ, $\gamma = 0.0001$	1.676	1.093	1.678	1.083
CondQ, $\gamma = 0.9$	1.578	1.173	1.502	1.028
CondQ, $\gamma = 0.1$	1.383	**0.815**	1.407	**0.905**
CondQ, $\gamma = 0.01$	**1.361**	0.857	1.389	0.938
CondQ, $\gamma = 0.001$	1.362	0.868	**1.386**	0.917
CondQ, $\gamma = 0.0001$	1.424	0.944	1.498	0.980
MDUMIQE	1.669	0.922	2.072	1.199
Cao et al., $c = 10$	2.359	2.277	1.878	1.048
Cao et al., $c = 1$	2.100	1.937	2.017	1.124
Cao et al., $c = 0.1$	2.196	2.009	1.977	1.293

Table 5
Change detection example. Results for the MD approach. Detection delay (last column) is given in seconds.

ν	ξ	h	η	Precision	Recall	F1 score	Det. delay
0.010	0.010	5	25	0.633	0.639	**0.636**	**15.469**
0.005	0.005	10	25	0.707	0.578	**0.636**	**25.072**
0.010	0.005	10	25	0.686	0.592	**0.636**	**22.637**
0.005	0.010	5	25	0.617	0.642	**0.629**	**16.739**
0.010	0.010	15	25	0.668	0.592	**0.628**	**30.128**
0.010	0.010	10	25	0.616	0.639	**0.627**	**16.795**
0.010	0.005	5	25	0.690	0.566	**0.622**	**19.326**
0.005	0.010	15	25	0.630	0.595	**0.612**	**36.359**
0.005	0.010	10	25	0.581	0.642	**0.610**	**24.175**
0.005	0.005	5	25	0.666	0.558	**0.607**	**15.683**

Table 6
Change detection example. Results using ShiftQ to track quantiles. Detection delay (last column) is given in seconds.

λ	γ	ξ	h	η	Precision	Recall	F1 score	Det. delay
0.001	0.2	0.05	10	15	0.694	0.662	**0.678**	**18.615**
0.001	0.1	0.05	10	25	0.781	0.587	**0.670**	**16.229**
0.001	0.1	0.05	15	25	0.798	0.572	**0.667**	**13.993**
0.001	0.2	0.01	10	10	0.795	0.572	**0.666**	**18.441**
0.001	0.1	0.01	5	10	0.727	0.607	**0.661**	**26.102**
0.001	0.1	0.05	10	15	0.592	0.746	**0.660**	**16.727**
0.001	0.2	0.05	15	15	0.714	0.613	**0.659**	**26.152**
0.001	0.1	0.01	10	10	0.690	0.604	**0.644**	**18.833**
0.001	0.1	0.05	5	15	0.621	0.662	**0.641**	**21.371**
0.001	0.1	0.01	15	10	0.700	0.561	**0.623**	**23.274**

Table 7
Change detection example. Results using CondQ to track quantiles. Detection delay (last column) is given in seconds.

λ	γ	ξ	h	η	Precision	Recall	F1 score	Det. delay
0.010	0.1	0.05	10	25	0.785	0.613	**0.688**	**15.457**
0.010	0.2	0.05	10	15	0.705	0.662	**0.683**	**13.400**
0.010	0.1	0.01	10	10	0.723	0.633	**0.675**	**19.437**
0.010	0.1	0.05	15	25	0.786	0.584	**0.670**	**18.470**
0.005	0.1	0.05	10	25	0.792	0.572	**0.664**	**17.855**
0.010	0.2	0.05	15	15	0.711	0.618	**0.662**	**19.957**
0.005	0.1	0.05	15	25	0.779	0.569	**0.658**	**22.672**
0.005	0.1	0.01	10	10	0.720	0.595	**0.652**	**20.829**
0.005	0.2	0.05	15	15	0.723	0.590	**0.650**	**22.142**
0.005	0.1	0.05	15	15	0.624	0.668	**0.645**	**27.182**

ing other quantiles in a relative manner to the central one. The properties of the propounded joint estimators are derived based on using the properties of the shifted and conditional quantiles.

We invoke the concept of conditional quantiles to devise the ShiftQ and CondQ algorithms which are extensions of the DUMIQE and QEWA algorithms, respectively. The experiments show that both the ShiftQ and CondQ algorithms outperform state-of-the-art multiple quantiles tracking algorithms in both synthetic and real-life data streams. Further, CondQ outperforms ShiftQ which is expected since the QEWA documents better performance than DUMIQE [18].

Usually, the mean and standard deviation are the most foresought statistical quantities summarizing a data stream, but these quantities have limited ability to capture distributional properties as well as other properties such as asymmetry or multimodality. Joint estimates of multiple quantiles yield a more general and flexible representation of the data stream distribution. The suggested estimators are able to process large quantities of raw data in real time. The resulting quantiles were shown useful in real time decision making, for instance classification [17,33] or concept drift detection in machine learning [18]. Streaming quantile estimators have also found applications in engineering features in machine learning [33,49].

As a future work, we aim to generalize the concept of conditional quantiles to also extend other incremental quantile estimators.

CRediT authorship contribution statement

Hugo Lewi Hammer: Conceptualization, Methodology, Software, Validation, Formal analysis, Writing - original draft, Writing - review & editing, Visualization, Project administration. **Anis Yazidi:** Conceptualization, Methodology, Supervision, Writing - review & editing. **Håvard Rue:** Conceptualization, Methodology, Supervision, Writing - review & editing.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Appendix A. Synthetic experiments – complete results

Figs. 3–6 show the complete error curves for the synthetic experiments in Section 7.

References

[1] Babak Abbasi, Montserrat Guillen, Bootstrap control charts in monitoring value at risk in insurance, Expert Syst. Appl. 40 (15) (2013) 6125–6135.
[2] Ognjen Arandjelovic, Duc-Son Pham, Svetha Venkatesh, Two maximum entropy-based algorithms for running quantile estimation in nonstationary data streams, IEEE Trans. Circ. Syst. Video Technol. 25 (9) (2015) 1469–1479.
[3] Andreas Bulling, Ulf Blanke, Bernt Schiele, A tutorial on human activity recognition using body-worn inertial sensors, ACM Comput. Surveys (CSUR) 46 (3) (2014) 33.
[4] Jin Cao, Li Li, Aiyou Chen, Tian Bu, Tracking quantiles of network data streams with dynamic operations, in: INFOCOM, 2010 Proceedings IEEE, IEEE, 2010, pp. 1–5..
[5] Jin Cao, Li Erran Li, Aiyou Chen, 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, 2009, pp. 7–12.
[6] Fei Chen, Diane Lambert, José 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..
[7] Baek-Young Choi, Sue Moon, Rene Cruz, Zhi-Li Zhang, Christophe Diot, Quantile sampling for practical delay monitoring in internet backbone networks, Comput. Netw. 51 (10) (2007) 2701–2716.
[8] Graham Cormode, S. Muthukrishnan, An improved data stream summary: the count-min sketch and its applications, J. Algorithms 55 (1) (2005) 58–75.
[9] Zhen Gao, Guoliang Lu, Peng Yan, Chen Lyu, Xueyong Li, Wei Shang, Zhaohong Xie, Wanming Zhang, Automatic change detection for real-time monitoring of eeg signals, Front. Physiol. 9 (2018) 325.
[10] Manfred Gilli et al, An application of extreme value theory for measuring financial risk, Comput. Econ. 27 (2–3) (2006) 207–228.
[11] Michael Greenwald, Sanjeev Khanna, Space-efficient online computation of quantile summaries, in: ACM SIGMOD Record, vol. 30, ACM, 2001, pp. 58–66..
[12] Alastair Gregory, F. Lau, Liam Butler, A quantile-based approach to modelling recovery time in structural health monitoring. arXiv preprint arXiv:1803.08444, 2018..
[13] Agnes Grünerbl, Amir Muaremi, Venet Osmani, Gernot Bahle, Stefan Oehler, Gerhard Tröster, Oscar Mayora, Christian Haring, Paul Lukowicz, Smartphone-based recognition of states and state changes in bipolar disorder patients, IEEE J. Biomed. Health Inf. 19 (1) (2014) 140–148.
[14] Hugo L. Hammer, Anis Yazidi, Michael A. Riegler, Håvard Rue, Efficient quantile tracking using an oracle. arXiv preprint arXiv:2004.12588, 2020..
[15] Hugo Lewi Hammer, Anis Yazidi, Incremental quantiles estimators for tracking multiple quantiles, in: International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems, Springer, 2017, pp. 202–210.
[16] Hugo Lewi Hammer, Anis Yazidi, A novel incremental quantile estimator using the magnitude of the observations, in: 2018 26th Mediterranean Conference on Control and Automation (MED), IEEE, 2018, pp. 290–295..
[17] Hugo Lewi Hammer, Anis Yazidi, B. John Oommen, On the classification of dynamical data streams using novel anti-bayesian techniques, Pattern Recogn. 76 (2018) 108–124.
[18] Hugo Lewi Hammer, Anis Yazidi, Håvard Rue, A new quantile tracking algorithm using a generalized exponentially weighted average of observations, Appl. Intell. 49 (4) (2019) 1406–1420.
[19] Hugo Lewi Hammer, Anis Yazidi, Håvard Rue, Estimating Tukey depth using incremental quantile estimators. arXiv preprint arXiv:2001.02393, 2020..
[20] Hugo Lewi Hammer, Anis Yazidi, Håvard Rue, Tracking of multiple quantiles in dynamically varying data streams, Pattern Anal. Appl. 23 (1) (2020) 225–237.
[21] Chung-Lin Huang, Bing-Yao Liao, A robust scene-change detection method for video segmentation, IEEE Trans. Circ. Syst. Video Technol. 11 (12) (2001) 1281–1288.
[22] Raj Jain, Imrich Chlamtac, The p 2 algorithm for dynamic calculation of quantiles and histograms without storing observations, Commun. ACM 28 (10) (1985) 1076–1085.
[23] V. Roshan Joseph, Efficient robbins–monro procedure for binary data, Biometrika 91 (2) (2004) 461–470.
[24] Arun Kejariwal, Sanjeev Kulkarni, Karthik Ramasamy, Real time analytics: algorithms and systems, Proc. VLDB Endowment 8 (12) (2015) 2040–2041.
[25] Georg Krempl, Indre Žliobaite, Dariusz Brzeziński, Eyke Hüllermeier, Mark Last, Vincent Lemaire, Tino Noack, Ammar Shaker, Sonja Sievi, Myra Spiliopoulou, et al, Open challenges for data stream mining research, ACM SIGKDD Explor. Newslett. 16 (1) (2014) 1–10.
[26] Harold Kushner, G. George Yin, Stochastic Approximation and Recursive Algorithms and Applications,, vol. 35, Springer Science & Business Media, 2003.
[27] Jennifer R. Kwapisz, Gary M. Weiss, Samuel A. Moore, Activity recognition using cell phone accelerometers, ACM SigKDD Explor. Newslett. 12 (2) (2011) 74–82.
[28] Ashwin Lall, Data streaming algorithms for the kolmogorov-smirnov test, in: 2015 IEEE International Conference on Big Data (Big Data), IEEE, 2015, pages 95–104..
[29] Chunquan Liang, Mei Li, Bin Liu, Online computing quantile summaries over uncertain data streams, IEEE Access 7 (2019) 10916–10926.
[30] Ge Luo Lu, Ke Yi Wang, Graham Cormode, Quantiles over data streams: experimental comparisons, new analyses, and further improvements, VLDB J. (2016) 1–24.
[31] Qiang Ma, S. Muthukrishnan, Mark Sandler, Frugal streaming for estimating quantiles, in: Space-Efficient Data Structures, Streams, and Algorithms, Springer, 2013, pp. 77–96..
[32] J. Ian Munro, Mike S Paterson, Selection and sorting with limited storage, Theor. Comput. Sci. 12 (3) (1980) 315–323.
[33] Fatemeh Nargesian, Horst Samulowitz, Udayan Khurana, Elias B. Khalil, Deepak S. Turaga, Learning feature engineering for classification, in: IJCAI, 2017, pp. 2529–2535..
[34] Valeriy Naumov, Olli Martikainen, Exponentially weighted simultaneous estimation of several quantiles, World Acad. Sci. Eng. Technol. 8 (2007) 563–568.
[35] Sergio Ramírez-Gallego, Bartosz Krawczyk, Salvador García, Michał Woźniak, Francisco Herrera, A survey on data preprocessing for data stream mining: current status and future directions, Neurocomputing 239 (2017) 39–57.
[36] Herbert Robbins, Sutton Monro, A stochastic approximation method, Ann. Math. Stat. (1951) 400–407..
[37] Bruce W. Schmeiser, Stuart Jay Deutsch, Quantile estimation from grouped data: the cell midpoint, Commun. Stat. Simul. Comput. 6 (3) (1977) 221–234.
[38] Marina Sokolova, Guy Lapalme, A systematic analysis of performance measures for classification tasks, Inf. Process. Manage. 45 (4) (2009) 427–437.

[39] Joel Sommers, Paul Barford, Nick Duffield, Amos Ron, Accurate and efficient sla compliance monitoring, in: ACM SIGCOMM Computer Communication Review, vol. 37, 2007, ACM, pp. 109–120..
[40] Joel Sommers, Paul Barford, Nick Duffield, Amos Ron, Multiobjective monitoring for sla compliance, IEEE/ACM Trans. Network. (TON) 18 (2) (2010) 652–665.
[41] Volker Stahl, Alexander Fischer, Rolf Bippus, Quantile based noise estimation for spectral subtraction and wiener filtering, in: 2000 IEEE International Conference on Acoustics, Speech, and Signal Processing, 2000. ICASSP'00. Proceedings, vol. 3, 2000, IEEE, pp. 1875–1878..
[42] Luke Tierney, A space-efficient recursive procedure for estimating a quantile of an unknown distribution, SIAM J. Sci. Stat. Comput. 4 (4) (1983) 706–711.
[43] Nitya Tiwari, Prem C. Pandey, A technique with low memory and computational requirements for dynamic tracking of quantiles, J. Signal Process. Syst. (2018) 1–12.
[44] Wei Wang, Wai-Ki Ching, Shouyang Wang, Lean Yu, Quantiles on stream: an application to monte carlo simulation, J. Syst. Sci. Inf. 4 (4) (2016) 334–342.
[45] 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.
[46] Anis Yazidi, Hugo Hammer, Multiplicative update methods for incremental quantile estimation, IEEE Trans. Cybern. (2017).
[47] Linfeng Zhang, Yong Guan, Detecting click fraud in pay-per-click streams of online advertising networks, in: The 28th International Conference on Distributed Computing Systems, 2008. ICDCS'08, IEEE, 2008, pp. 77–84..
[48] Xuebin Zhang, Lisa Alexander, Gabriele C. Hegerl, Philip Jones, Albert Klein Tank, Thomas C. Peterson, Blair Trewin, Francis W. Zwiers, Indices for monitoring changes in extremes based on daily temperature and precipitation data, Wiley Interdisc. Rev. Clim. Change 2 (6) (2011) 851–870.
[49] Alice Zheng, Amanda Casari, Feature Engineering for Machine Learning: Principles and Techniques for Data Scientists, O'Reilly Media Inc, 2018.