Efficient quantile tracking using an oracle

Hugo L. Hammer[1] ⓘ · Anis Yazidi[2] · Michael A. Riegler[3] · Håvard Rue[4]

Abstract

Concept drift is a well-known issue that arises when working with data streams. In this paper, we present a procedure that allows a quantile tracking procedure to cope with concept drift. We suggest using expected quantile loss, a popular loss function in quantile regression, to monitor the quantile tracking error, which, in turn, is used to efficiently adapt to concept drift. The suggested procedures adapt efficiently to concept drift, and the tracking performance is close to theoretically optimal. The procedures were further applied to three real-life streaming data sets related to Twitter event detection, activity recognition, and stock trading. The results show that the procedures are efficient at adapting to concept drift, thereby documenting the real-world applicability of the procedures. We further used asymptotic theory from statistics to show the appealing theoretical property that, if the data stream distribution is stationary over time, the procedures converge to the true quantile.

Keywords Concept drift · Data mining · Incremental quantile estimator · Real-time tracking

1 Introduction

The volume of automatically generated data is constantly increasing [39], which leads to more urgent demand for real-time analysis [34]. Conventional statistical and data mining techniques are usually designed for offline situations and are therefore often inappropriate for real-time analysis [21]. Thus, a wide range of streaming algorithms is continuously being developed by the data mining community that address a range of real-time tasks, such as clustering, filtering, cardinality estimation, estimation of moments or quantiles, predictions, dimensionality reduction, and anomaly detection [19, 34].

Concept drift, a well-known issue when working with data streams, refers to unforeseeable changes in the underlying

distribution of streaming data over time [28]. Therefore, the prediction error for a machine learning method trained on historic data usually increases when faced with concept drift. Consequently, machine learning methods should be retrained on more recent data to adapt to the concept drift. A range of different methods has been developed to detect and adapt to concept drift, and a review is provided in Section 2.

In this paper, we address the problem of adapting to concept drift when the objective is to track quantiles of the data stream distribution. A data stream that changes from slow to rapid variations or experiences changes in the scale of the data over time is a typical example of concept drift. To the best of our knowledge, this is the first work in the literature to address this research problem. When quantiles are estimated in real time, the problem is usually referred to as quantile tracking [16]. Quantile tracking has found a wide range of applications, a review of which is provided in Section 2. In this paper, we focus on a family of lightweight and efficient estimators called incremental quantile estimators. The estimators perform small updates of the quantile estimate every time a new sample is received from the data stream. Incremental quantile estimators document state-of-the-art performance on quantile tracking [14, 16].

Learning under concept drift is traditionally based on two steps. The first step involves a method for detecting concept drift, while the second step involves a method for adaptation when concept drift is detected, which is typically

✉ Hugo L. Hammer
hugo.hammer@oslomet.no

1 Oslo Metropolitan University and Simula Metropolitan Center, Oslo, Norway

2 Oslo Metropolitan University and Oslo University Hospital, Oslo, Norway

3 Simula Metropolitan Center and University of Tromsø, Oslo and Tromsø, Norway

4 King Abdullah University of Science & Technology, Thuwal, Saudi Arabia

based on retraining the machine learning method [28]. However, this two-step procedure is not optimal for making quantile tracking algorithms adapt to concept drift. First, the retraining of incremental quantile tracking algorithms is computationally heavier than running the algorithms. Secondly, incremental quantile algorithms usually only have very few parameters. This opens up for a single-step, fast approach instead of the two steps outlined above. More specifically, in this paper, we suggest a novel approach to optimize the parameters of the quantile tracking algorithms in each iteration based on monitoring the expected quantile loss, a loss function commonly used in quantile regression [20]. The main contributions of the paper can be summarized as follows:

– We present two new methods for efficient quantile tracking under concept drift. The methods can be coupled with any quantile tracking algorithm to improve tracking performance under concept drift. To the best of our knowledge, these are the first methods in the literature to address the issue of quantile tracking under concept drift.
– The methods have appealing theoretical properties in the sense that, for stationary data streams, the resulting procedures will converge to the true quantile. We prove convergence using the asymptotic theory.
– Our experiments show that the suggested procedures clearly improve the tracking performance of existing quantile estimators for data streams with concept drift. The performance is close to the theoretically optimal performance, i.e., the performance when using the optimal values of the parameters in the quantile tracking algorithm in every iteration.
– The real-life applicability of the procedures is demonstrated using three large datasets related to Twitter event detection, activity recognition, and stock trading.

The paper is organized as the following: Section 2 presents related work. Section 3 gives a short presentation of incremental quantile estimators. Section 4 explains how to estimate the current quantile tracking error, and how to use it to adapt to concept drift. Sections 6 and 7 evaluate the suggested procedures using synthetic and real-life data streams, while Section 8 consists of some closing remarks.

2 Related work

2.1 Incremental quantile algorithms

Quantiles are useful for characterizing data stream distribution in a flexible and non-parametric way [29]. They have been used for a range of applied intelligence tasks, such as real-time classification [13], concept drift detection

[14], anomaly detection [11], portfolio risk measurement in the stock market [1, 8], fraud detection [49], signal processing and filtering [44], climate change monitoring [50], Service Level Agreement (SLA) violation monitoring [42, 43], network monitoring [6, 27], structural health monitoring [9], non-parametric statistical testing [23], concept drift detection [14], and Tukey depth estimation [11, 33].

Incremental algorithms are an important class of methods for addressing the problem of quantile tracking, but the research on such methods is still quite sparse. Tierney [46] introduced the concept of incremental quantile estimators, which was originally designed to work on static data streams. A more recent incremental quantile estimation approach is the Frugal algorithm of Ma et al. [30], which can also be applied to dynamically varying data streams. Yazidi and Hammer [48] suggested using the DUMIQE algorithms, while Tiwari and Pandey [47] proposed DQTRE and DQTRSE, which can all be seen as multiplicative variants of the Frugal algorithm. The multiplicative variants are more efficient at adapting to changes in the scale of the data. A weakness of the above estimators is that they do not use the magnitude of the observations when updating the current estimate, which can result in slow convergence. To address this issue, Hammer et al. [14] suggested the QEWA algorithm, where the update size is proportional to the current tracking error. This algorithm has documented efficient performance, but it has a disadvantage compared to most other incremental quantile estimators because it is not robust to outliers in the data stream. The reason for this is the fact that the observations are used directly in the updating of the running estimate. One limitation of the estimators above is that they can only track a single quantile. Obviously, the estimators can be run in parallel to track many quantiles, but the monotone property of quantiles can then be violated. Some methods have recently been developed to address this issue by allowing joint tracking of multiple quantiles [15, 16]. We will not consider tracking of multiple quantiles in this paper, although our suggested concept drift adaptation methods can also be applied to these algorithms. There is a gap in the current research within incremental quantile estimation related to finding suitable values for the tuning parameters, and how to adjust the values if the properties of the data stream change (concept drift) which is the specific issue addressed in this paper.

2.2 Concept drift

Methods for adapting to concept drift mainly consist of two parts: i) a method to detect concept drift and ii) a method to adapt after concept drift is detected.

One of the first methods for concept drift detection was the Drift Detection Method (DDM) of [7], which is based on testing whether the prediction error significantly increased

within a recent time window. However, a disadvantage of window-based approaches is that every sample in the window is weighted equally. Another computationally efficient approach that addresses this issue relies on tracking the current prediction error using the exponentially weighted moving average [12, 40].

For concept drift adaptation, the most natural approach is probably to retrain the machine learning methods based on recent data. An example is the paired learner, which uses a stable learner and a reactive learner using only recent data [4]. There are also researchers who attempt to integrate concept drift detection with retraining, like [26] for Extreme Learning Machine. However, a weakness of these approaches is that it is computationally demanding to repeatedly retrain a machine learning method. To address this issue, Sun et al. [45] suggest a strategy to build up an ensemble of trained models to reduce the amount of retraining required when faced with recurring drift. Pratama et al. [37] suggested only updating parts of the parameters of the machine learning method, thereby saving computational resources.

Class imbalance is another well-known issue in machine learning. Recent works like [2, 24, 38] address this challenge when faced with concept drift. Concept drift methods usually assume that class labels are immediately available. Mahdi et al. [31] addressed the problem of concept drift detection in the opposite case when class labels are not available, whereas the work reported in [25] suggests a method for making an ensemble of trained models as efficient as possible by introducing diversity measures.

3 Incremental quantile algorithms

Let $X_t \sim f_t(x)$ represent possible outcomes from a data stream at time t, x_t a random sample, and $Q_{t,q}$ the quantile associated with probability q, i.e., $P(X_t \leq Q_{t,q}) = F_t(Q_{t,q}) = q$.

Incremental quantile algorithms update a quantile estimate every time a new observation is received. The algorithms are initiated with an estimate $\widehat{Q}_{0,q}(\lambda_t)$ that is further recursively updated

$$
\begin{aligned}
\widehat{Q}_{t+1,q}(\lambda_t) &\leftarrow \widehat{Q}_{t,q}(\lambda_t) + \lambda_t D_1\left(q, \widehat{Q}_{t,q}(\lambda_t)\right) \text{ if } x_t \\
&\geq \widehat{Q}_{t,q}(\lambda_t) \\
\widehat{Q}_{t+1,q}(\lambda_t) &\leftarrow \widehat{Q}_{t,q}(\lambda_t) - \lambda_t D_2\left(q, \widehat{Q}_{t,q}(\lambda_t)\right) \text{ if } x_t \\
&< \widehat{Q}_{t,q}(\lambda_t)
\end{aligned}
\tag{1}
$$

where the functions D_1 and D_2 are positive and can be deterministic or random. The estimation procedure is intuitive in the sense that, if the received observation is above (below) the current estimate, the estimate is increased (decreased). The functions are typically further constructed

to ensure that the estimator converges to the underlying true quantile [46]. A prominent example is the deterministic-based multiplicative incremental quantile estimator (DUMIQE) [48] where $D_1(q, \widehat{Q}_{t,q}(\lambda_t)) = q\widehat{Q}_{t,q}(\lambda_t)$ and $D_2(q, \widehat{Q}_{t,q}(\lambda_t)) = (1 - q)\widehat{Q}_{t,q}(\lambda_t)$. Another example is the Frugal estimator [30], where $D_1(q, \widehat{Q}_{t,q}(\lambda_t)) = I(q < U)$ and $D_2(q, \widehat{Q}_{t,q}(\lambda_t)) = I(1 - q < U)$, and where U denotes a uniformly distributed number on $[0, 1]$ and $I(\cdot)$ the indicator function.

The tuning parameter λ_t determines the update size in each iteration. If the data stream distribution changes rapidly (slowly) with time, a high (small) value should be used. Furthermore, the step size should be adjusted to the scale of the data.

4 Adaptive quantile tracking under concept drift

We consider the problem of predicting $Q_{t,q}$ in every iteration using an incremental quantile algorithm. However, such an algorithm consists of one or more parameters, e.g., the λ_t from the previous section. If the properties of the data stream change (concept drift), the values of these parameters must be adjusted to maintain efficient tracking. See the bottom panel of Fig. 2 for an example where the data stream concept changes from rapid variations to slow variations on iteration 10 000.

To adapt to concept drift, we suggest monitoring the current quantile tracking error using expected quantile loss (EQL), a popular loss function in quantile regression. The quantile loss on iteration t is

$$
\mathrm{QL}_t(\lambda_t) = \begin{cases} (1-q)\left(x_t - \widehat{Q}_{t,q}(\lambda_t)\right) & \text{if } x_t \geq \widehat{Q}_{t,q}(\lambda_t) \\ q\left(\widehat{Q}_{t,q}(\lambda_t) - x_t\right) & \text{if } x_t < \widehat{Q}_{t,q}(\lambda_t) \end{cases}
\tag{2}
$$

To estimate the current expected quantile loss, we use the exponentially weighted moving average [12]

$$
\widehat{\mathrm{EQL}}_t(\lambda_t) = (1 - \gamma)\widehat{\mathrm{EQL}}_{t-1}(\lambda_t) + \gamma \mathrm{QL}_t(\lambda_t)
\tag{3}
$$

A common strategy for learning under concept drift is to 1) use the error measure $\widehat{\mathrm{EQL}}_t(\lambda_t)$ to detect concept drift, and 2) to adjust the values of the parameter of the algorithm. However, since the number of parameters of quantile tracking algorithms is usually small, we suggest avoiding the detection part and instead continuously updating the values of the parameters of the quantile tracking algorithms in every iteration. The method will therefore adapt more rapidly to concept drift, especially in cases where the drift occurs gradually. We call the method *Oracle*. It is described below.

Oracle Let the values $\lambda_1^* < \lambda_2^* < \cdots < \lambda_L^*$ span all reasonable values for the tuning parameter and track $Q_{t,q}(\lambda_l^*)$, $l = 1, \ldots, L$ using (1). For each $\widehat{Q}_{t,q}(\lambda_l^*)$,

compute the associated EQL, $\widehat{\text{EQL}}_t(\lambda^*)$, $l = 1, \ldots, L$, using (3). Let $\lambda_t = \arg\min_{\lambda_l^*, l \in 1, \ldots, L} \widehat{\text{EQL}}_t(\lambda_l^*)$ and let the current quantile estimate be given by $\widehat{Q}_{t,q}(\lambda_t)$.

Naturally, there may be large fluctuations in the values of λ used, and updates may be limited to only neighbouring values (friction), but we have not explored this further in this paper. We denote this approach the *Oracle* approach, since we can imagine an *Oracle* that monitors the individual quantile tracking procedures and uses the estimated EQLs to select the best current quantile estimate, without disturbing the quantile or EQL tracking procedures. The procedure is illustrated in Fig. 1, where the objective is to track the $q = 0.7$ quantile of the data stream (gray dots). In the bottom left panel, the quantile is tracked with a small update size, λ_1^*, resulting in a high estimation bias (the proportion of observations below the quantile estimates is far below the target $q = 0.7$). Further, in the bottom right panel, the quantile is tracked with a large update size, λ_L^*, resulting in high tracking variance. The estimator in the middle panel seems to have both a fairly small bias and a variance resulting in a small EQL.

A potential challenge if this approach is used is that, with limited knowledge about the data stream, it may be difficult to select the range of λ^*'s. The solution is that, if the *Oracle* selects the quantile estimate for a λ close to λ_1^* (or λ_L^*), additional estimators using values of λ less than λ_1^* (or above λ_L^*) are included.
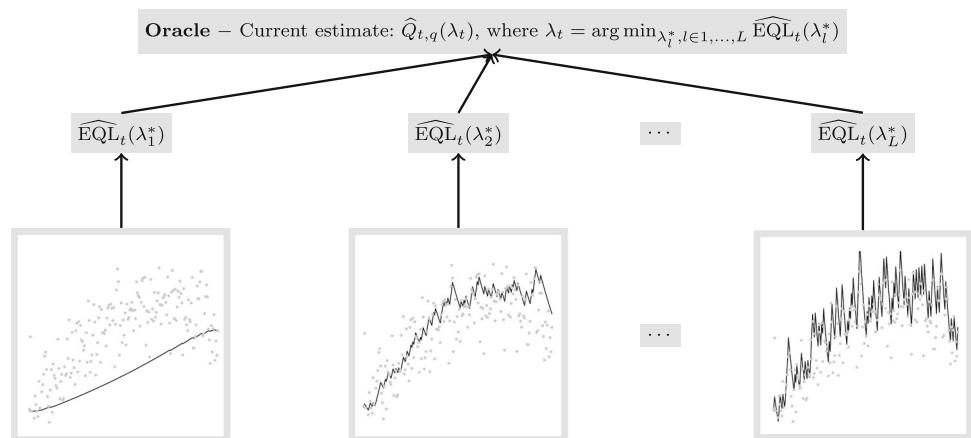
The approach entails storing a total of $2L$ values and performing a total of $2L$ operations (storing and updating the quantile estimate and the expected quantile loss) for each sample received from the data stream. A typical value for L is 100, and thus for extremely massive data streams, this may be a computational challenge. An example is when incremental quantile estimators are used to track depth contours, which requires the tracking of thousands or even millions of quantile estimates [11].

HIL This approach follows the traditional approach to handling concept drift, namely to first detect and then adapt to concept drift. The method will be computationally less demanding than the *Oracle* approach, but at the cost of adapting less efficiently to concept drift. The approach only tracks the quantile for a high, intermediate and low (HIL) value of the update size, and therefore only entails storing a total of $2 \cdot 3$ values and performing a total of $2 \cdot 3$ operations for each received data stream sample. For convenience of notation, the subscript t is avoided.

– Run three quantile tracking estimators in parallel using tuning parameters $\lambda_{\text{low}} = \lambda/a$, $\lambda_{\text{intermediate}} = \lambda$ and $\lambda_{\text{high}} = a\lambda$, $a > 1$, and track the EQL for each of them.
– Every M iterations, update λ:

  – If EQL is smallest for λ_{low} (or λ_{high}), reduce (or increase) the value of the tuning parameter for the three estimators by setting $\lambda \leftarrow \lambda/a$ (or $\lambda \leftarrow a\lambda$). Restart the three quantile estimators initialized with the currently best quantile estimate, i.e., for λ_{low} (or λ_{high}).
  – If EQL is smallest using $\lambda_{\text{intermediate}}$, no updates are done.

The current quantile estimate is given by $\widehat{Q}_{t,q}(\lambda)$. Concept drift usually happen slowly or rarely, and it makes sense to use a high value of M, say 10^3. Further, the values of the tuning parameter γ should be chosen so that, when an update of λ is performed, the estimate of the EQL has converged at the same time as most of the information since the last update of λ is used. A simple rule of thumb is to set $\gamma = 1 - \sqrt[M]{0.01}$, which means that the weight of the Mth term of the exponentially weighted sum in (3) is 0.01. This worked well in our experiments. Data streams follow different periodic patterns most of the time, thus randomly selecting when to update λ can sometimes be useful.

Fig. 1 An overview of the *Oracle* approach. In the bottom row, the gray dots show observations from the data stream. The black curves show tracking of the $q = 0.7$ quantile for different step sizes (λ_1^*, λ_2^* and λ_L^*). The information in the panels is used to estimate expected quantile loss for each value of the step size using the procedure in Section 4. The expected quantile loss estimates are sent to the *Oracle*, which selects the currently best quantile estimate in each iteration

$\textbf{Oracle} - \text{Current estimate: } \widehat{Q}_{t,q}(\lambda_t), \text{ where } \lambda_t = \arg\min_{\lambda_l^*, l \in 1, \ldots, L} \widehat{\text{EQL}}_t(\lambda_l^*)$

$\widehat{\text{EQL}}_t(\lambda_1^*)$ $\widehat{\text{EQL}}_t(\lambda_2^*)$ $\cdots$ $\widehat{\text{EQL}}_t(\lambda_L^*)$

A challenge posed by the *HIL* approach is that the quantile estimates and associated EQLs must be restarted after an update and must have converged before a new update is performed. This limits how rapidly and smoothly the procedure can adapt to concept drift. Further, since λ is rarely updated, a fairly large value of a, say 2, must be used, thereby limiting any fine tuning of λ. The *Oracle* approach is not burdened with these challenges.

5 Asymptotics

A natural requirement of the suggested procedures is that, if the data stream distribution does not change with time (stationary stream), the quantile estimate should converge to the true quantile as time goes to infinity. This is confirmed by the following theorem for the *HIL* approach (a similar theorem can also be set up for the *Oracle* approach, which is explained below).

Theorem 1 *Assume a stationary data stream, and let Q_q be the true quantile to be estimated. Further assume that D_1 and D_2 are selected, such that $\widehat{Q}_{t,q}(\lambda_t)$ satisfies the assumptions for convergence of Markov processes* [35, 36]. *Using the HIL approach with $a_t = (t + M)^p / t^p, 0 < p < 1$, then*

$$\lim_{t \to \infty} \lambda_t = 0 \tag{4}$$

$$\lim_{t \to \infty} \widehat{Q}_{t,q}(\lambda_t) = Q_q \tag{5}$$

The proof is provided in Appendix A. The overall intuition is that, if the data stream is stationary, the *HIL* procedure will iteratively select a smaller and smaller step size, and the estimator will converge to the true quantile. For the *Oracle* approach, we suggested including values less than λ_1^* if the *Oracle* selected the quantile estimate for a λ_t close to λ_1^*. Thus, a similar argument can be set up to prove the convergence of the *Oracle* approach.

Most of the recently developed incremental quantile estimators satisfy the assumptions for convergence of Markov processes referred to in the theorem, see, e.g., the proofs in [14, 15, 48]. Therefore, by using one of these incremental quantile estimators in combination with either the *HIL* or the *Oracle* procedures, convergence is guaranteed.

6 Synthetic experiments

Consider a normally distributed data stream where the expectation changes between slow and rapid dynamics (concept drift)

$$f_t(x) = N\left(\mu + b \sin\left(\frac{2\pi}{\tau(n)}n\right), \sigma\right)$$

$$\tau(n) = \tau_1 I(\bmod n < T) + \tau_2 I(T \le \bmod n < 2T) \tag{6}$$

with $\tau_1 = 500$, $\tau_2 = 10^4$, $\mu = 8$, $b = 2$, $\sigma = 1$, $T = 10^4$. We tracked this data stream using the DUMIQE algorithm. To estimate the step length λ in each iteration, we used the the *Oracle* approach with the following values for the update sizes $\lambda_1^* = \exp(-7), \lambda_2^* = \exp(-6.95), \ldots, \lambda_L^* = \exp(0)$. We estimate the expected quantile loss using (3) with $\gamma = 1 - \sqrt[1000]{0.01} = 0.005$, following the rule of thumb suggested above.

Figure 2 shows the estimated expected quantile loss (top left) and the estimated values of the step length (top right) in each iteration and the actual quantile tracking (bottom panel). In the upper left panel, we see that quantile loss quickly decreases and then stabilizes. The initial value of the quantile tracking algorithm is outside the data stream distribution. Therefore, in the initial phase, the quantile loss is high and a high value of λ (update size) is used for the tracking algorithm to quickly get within the data stream distribution. When the quantile tracking algorithm is within the data stream distribution, a smaller λ is optimal and the value of λ is reduced. The expected quantile loss is reduced accordingly. The gray lines in the top right panel show the optimal value of λ for the data stream dynamics before and after the concept change at iteration 10 000. We see that a larger step size is needed to efficiently track rapid variations. We further see that the *Oracle* approach uses step lengths close to the optimal step lengths.

We now analyze the performance of the suggested procedures in more detail for data streams with concept drift. In particular, we analyze how the efficiency of the concept drift adaptation depends on the parameter γ in (3). Consider again the data stream in (6) as well as a χ^2 distributed stream

$$f_t(x) = \chi^2\left(\nu + b \sin\left(\frac{2\pi}{\tau(n)}n\right)\right)$$

$$\tau(n) = \tau_1 I(\bmod n < T) + \tau_2 I(T \le \bmod n < 2T) \tag{7}$$

using the same values τ_1, τ_2, T and b as above, and $\nu = 6$. $\chi^2(\nu)$ represents the χ^2 distribution with ν degrees of freedom. The χ^2 distributed stream is challenging since both the expectation and variance change with time. For the *HIL* approach, $a = 1.5$ and $M = 1000 + U$ were used, where U was uniformly distributed on the interval $[0, 1000]$, i.e., λ was updated on average every 1500 iterations. We tracked the $q = 0.5, 0.7$ and 0.9 quantiles. To remove any Monte Carlo error, data streams were run for a total of $N = 10^7$ iterations and the observed tracking MSE was computed:

$$\text{MSE} = \frac{1}{N} \sum_{t=1}^{N} \left(\widehat{Q}_{t,q} - Q_{t,q}\right)^2 \tag{8}$$

Figure 3 shows the tracking error as a function of γ for the data streams above, which contain concept drift.
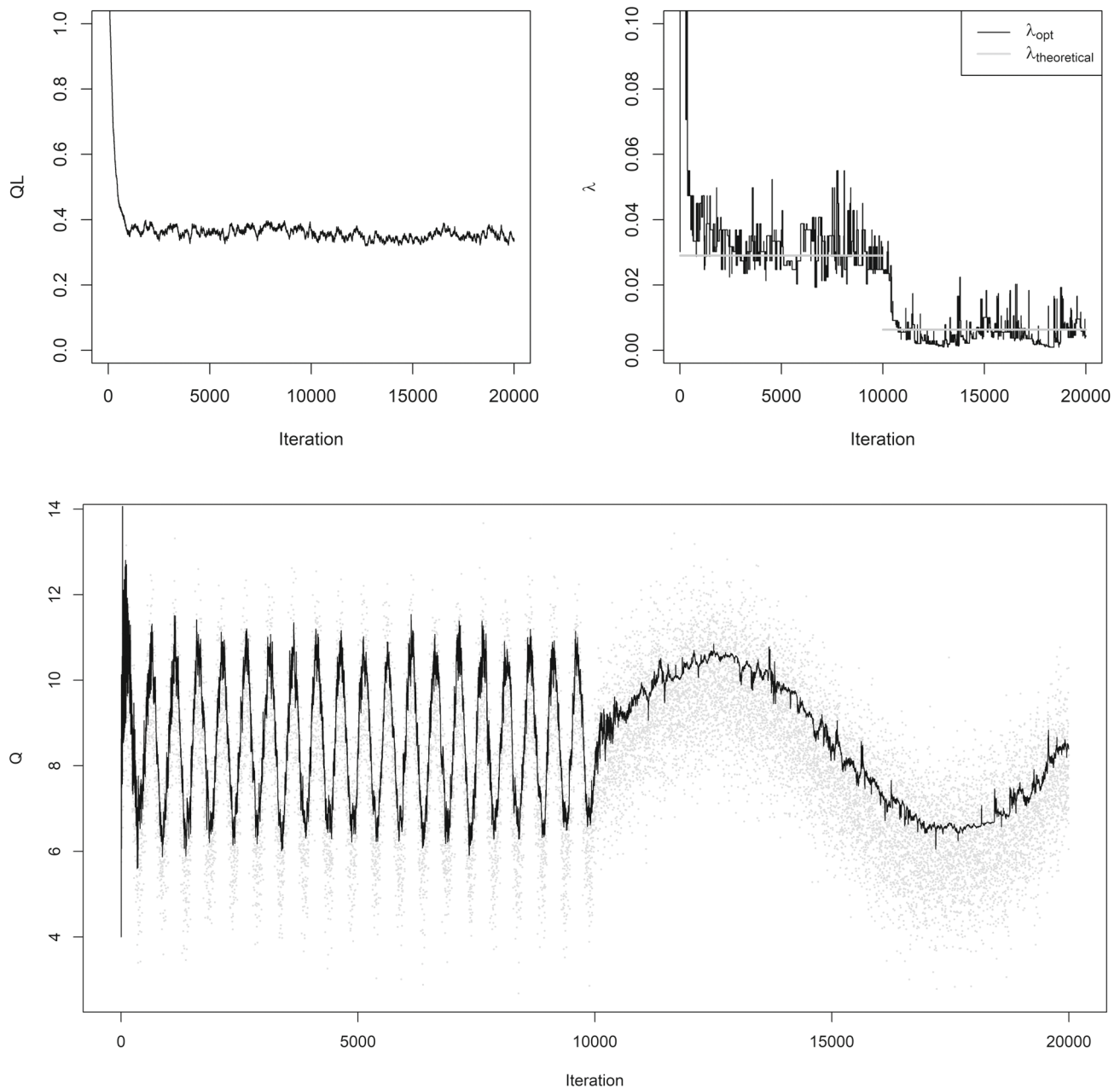
Fig. 2 Diagnostics plots when tracking the $q = 0.7$ quantile using the *Oracle* approach. The top left panel shows the estimated QL in every iteration. The top right panel shows the resulting recursive updating of λ. The symbol λ_{opt} refers to the value resulting in the minimal estimated QL. The gray lines refer to theoretically optimal values of λ. The bottom panel shows the resulting tracking. The gray dots represent the data stream in (6) and the black line is tracking

Let $\text{MSE}_{\tau_1}^{\text{theo}}$ and $\text{MSE}_{\tau_2}^{\text{theo}}$ represent the theoretically minimum tracking MSE for a data stream with constant fast dynamics (τ_1) and constant slow dynamics (τ_2), respectively. These errors were found by running the DUMIQE algorithm for a range of values of λ to find the values that minimized the tracking error for fast dynamics and slow dynamics. The data streams in examples (6) and (8) consist of an equal amount of fast and slow dynamics. By tracking with optimal values of λ for fast and slow dynamics and instantaneous adaptation to concept drift, the minimum tracking error will be $0.5\,\text{MSE}_{\tau_1}^{\text{theo}} + 0.5\,\text{MSE}_{\tau_2}^{\text{theo}}$, which is shown as gray solid lines in Fig. 3. The gray lines therefore represent the theoretically minimal tracking error. We further computed the minimum tracking MSE using a constant value of λ, which is shown as gray dashed lines. The results show that, for all the cases, the *Oracle* approach tracks the true quantile with an error that is just slightly above the theoretically optimal tracking error. The *HIL*
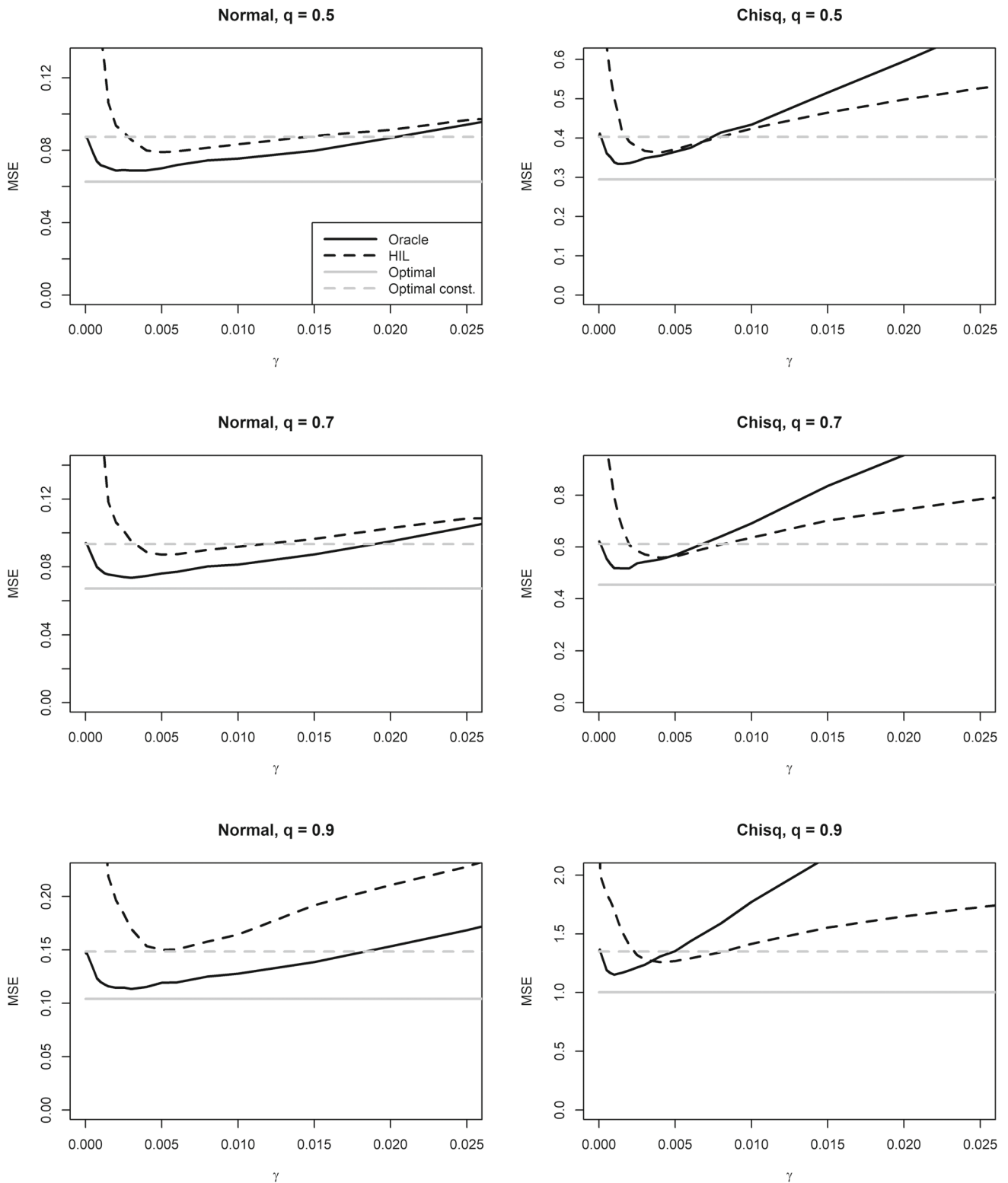
Fig. 3 The left and right panels show results for the normal and χ^2 distributed data streams. The rows from top to bottom show results from tracking of the $q = 0.5$, 0.7 and 0.9 quantiles, respectively

approach also performs well, but not as well as the *Oracle* approach, which is as expected given the discussion at the end of Section 4. Optimal values of the tuning parameter, γ, are close to the suggested rule of thumb $1 - \sqrt[M]{0.01} = 0.0031$.

7 Real-life data examples

In this section, we present three real-world data examples for benchmarking the procedures. The examples are related to Twitter streaming data, activity monitoring, and stock trading.

7.1 Twitter

Twitter data streams have been used for many interesting applications, such as predicting election outcomes and detecting natural disasters and other real world events [3, 17].

We consider a dataset consisting of the time stamp for every tweet posted by Norwegian users before and after the terrorist attack on July 22, 2011 [41]. The time stamps were given in seconds, and to reconstruct a representation of the true time stamps a uniformly drawn value between zero and one second was added to each time stamp.

Let T_t represent the time stamp when tweet number t was posted. We consider the problem of tracking quantiles of the quantity $R_t = (T_t - T_{t-1})^{-1}$, which can be interpreted as the frequency of posted tweets. The quantity can, for instance, be used for real world event detection in the sense that if the number of received tweets is increasing, R_t will increase.

The terrorist attack was initiated by a bomb exploding in Oslo on July 22 at 3:25 p.m. local time, which created a significant concept drift in the data stream distribution. We will evaluate how well the methods in this paper are able to adapt to the concept drift in the dataset. We tracked quantiles of R_t using the Frugal estimator [30] and the *Oracle* approach was used for adaptation. We used $\gamma = 0.005$ in accordance with the rule of thumb in Section 4.

The results are shown in Fig. 4. The black curve in the upper panel shows the tracking of the $q = 0.7$ quantile of R_t. The gray dots show the observed R_t. The gray and black curves in the bottom panel show the value of λ in every iteration and a moving average, respectively. The Frugal estimator was initiated with $\lambda = 1$ and the procedure rapidly adjusted the step length to a more suitable value.

For the period until the bomb exploded, it is a trend that λ is adjusted to lower values during night time, which makes sense since both the scale and dynamics of R_t are smaller. When the bomb exploded, the values of R_t rapidly increased (concept drift), and the value λ therefore rapidly increased to be able to efficiently track under the increased scale and dynamics.

7.2 Activity monitoring

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, activity recognition could be used to assess whether a person is performing a healthy amount of exercises or to detect accidents such as falls. In this experiment, we will track
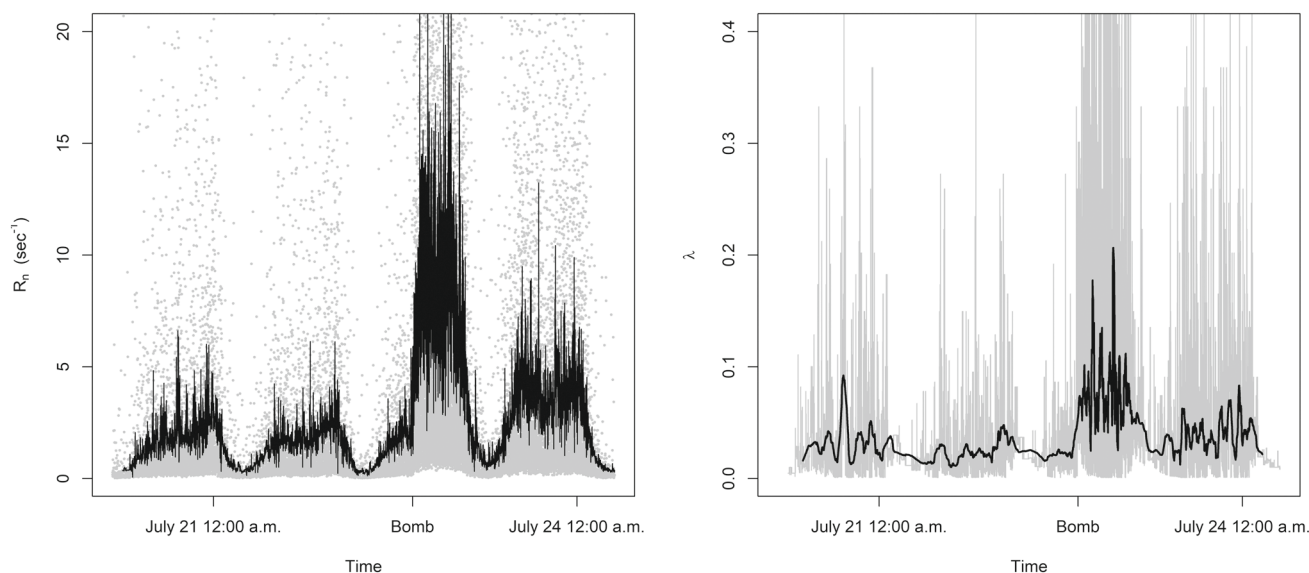

Fig. 4 Twitter data example. Upper panel: Gray dots show the observed data stream R_t (every 10th shown) and the black curve tracking of the $q = 0.7$ quantile using the Frugal estimator. Bottom panel: The values of λ are shown in gray, and the black curve is a moving average of the λ values
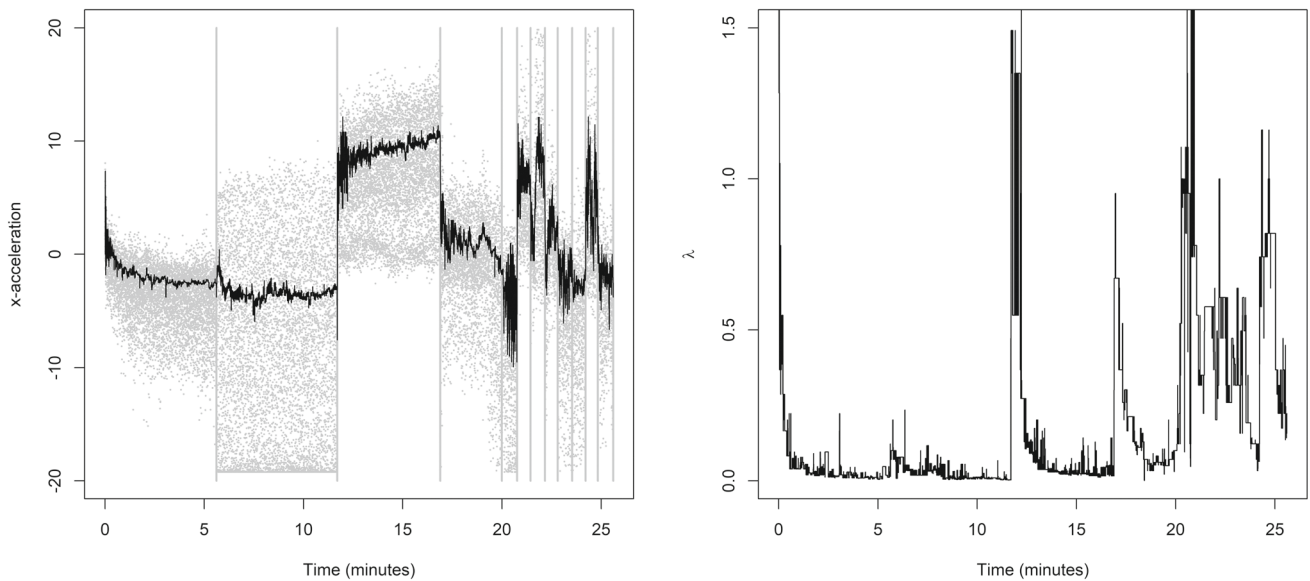
Fig. 5 Accelerometer data example. Upper panel: Gray dots show the accelerometer observations and the black curve the tracking of the $q = 0.7$ quantile using the Frugal estimator. The gray vertical lines show when the user changed activity. Bottom panel: The values of λ

quantiles of accelerometer data that are available on almost any modern cell phone or smart watch device. Tracking quantiles can be used to detect when a user changes activity [10] or to classify the current activity of the user [5].

We consider an accelerometer dataset from the Wireless Sensor Data Mining (WISDM) project [22]. Accelerations in x, y, and z directions were observed, with a frequency of 20 observations per second, while users were performing the following activities: walking, jogging, walking up a stairway and walking down a stairway.

Figure 5 shows the tracking of the x acceleration of an arbitrary user. We see that, when the accelerometer distribution is stable over time, the Oracle approach uses a small value of λ, e.g, for the two first activities. When the user changed activity, an immediate change in acceleration was observed. The value of λ (step length) was rapidly increased for the quantile tracking to efficiently adapt to the new accelerometer distribution. Further, the value of λ was gradually reduced when the accelerometer distribution stabilized for the new activity. From around 20 minutes, the user
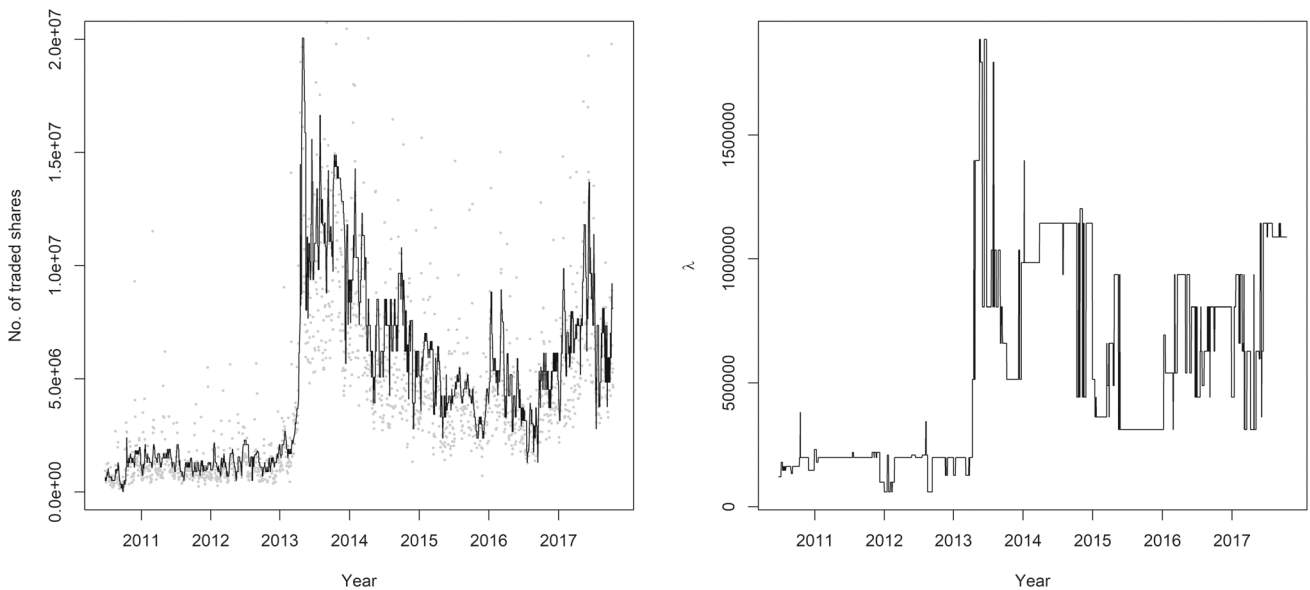

Fig. 6 Stock trading example. Upper panel: Gray dots show the number of traded Tesla, Inc. shares, while the black curve shows the tracking of the $q = 0.7$ quantile using the Frugal estimator. Bottom panel: The values of λ

changed activity far more frequently, and higher values of λ were therefore used to ensure efficient quantile tracking.

7.3 Stock trading

High frequency data are highly abundant in stock trading, and they need to be analyzed efficiently in real-time [18]. Figure 6 shows tracking of the $q = 0.7$ quantile of the number of traded Tesla Inc. shares on the The New York Stock Exchange [32]. When the number of traded shares increased rapidly, the value of λ was increased to rapidly adapt to the rapid change in scale and dynamics (concept drift). The value of λ further decreased slightly when the trade volume (dynamics) stabilized at a high level.

The three real-life data experiments demonstrate that the suggested procedure is able to rapidly adjust λ to maintain efficient quantile tracking under concept drifts.

8 Closing remarks

Surprisingly little attention has been paid to automatic adjustment of the values of the parameters of incremental quantile algorithms. In this paper, we develop two simple procedures to address this problem. Both procedures are based on estimating the current tracking quantile loss and using this to efficiently track the true quantiles. The *Oracle* approach tracks the quantile and associated quantile loss for a wide range of values of the algorithm parameters and, in each iteration, selects the quantile estimate with the minimal estimated quantile loss. The second approach only tracks the quantile for three values of the tuning parameter and repeatedly forgets the estimate with the highest estimated quantile loss and adds a quantile estimator for another value of the tuning parameter. Both methods are computationally and memory efficient, since only a limited set of quantities needs to be computed and stored in each iteration.

The results show that the methods are highly efficient for adjusting the value of λ to achieve efficient tracking. The synthetic experiments showed that the resulting tracking error is close to the theoretical minimum. The *Oracle* performs best, but at a higher computational cost. The real-life data examples demonstrated that the procedures were able to adapt to concept drift for complex and massive real-life data streams. In future work, we would like to test the methods on data with different properties, such as mental health activity data that often show rapid changes in the behaviour of the patient.

Appendix A: Proof of Theorem 1

In this section, we provide a proof of Theorem 1.

Proof Since the data stream is stationary, we avoid the time subscript for the data stream distribution. Using Taylor expansion

$$
\begin{aligned}
L(\widehat{Q}_{t,q}(\lambda)) &= L(Q_q) + L'(Q_q)(\widehat{Q}_{t,q}(\lambda) - Q_q) \\
&\quad + \frac{1}{2}L''(Q_q)(\widehat{Q}_{t,q}(\lambda) - Q_q)^2 \\
&\quad + O((\widehat{Q}_{t,q}(\lambda) - Q_q)^3) \quad\quad (9) \\
&= L(Q_q) + \frac{1}{2}L''(Q_q)(\widehat{Q}_{t,q}(\lambda) - Q_q)^2 \\
&\quad + O((\widehat{Q}_{t,q}(\lambda) - Q_q)^3) \quad\quad (10)
\end{aligned}
$$

where $L(\cdot)$ refer to expected quantile loss.

According to [36], $\widehat{Q}_{t,q}(\lambda)$ will be approximately normally distributed with $\text{Bias}^2 = (\widehat{Q}_{t,q}(\lambda) - Q_q)^2 = O(\lambda^2)$ and $\text{Var}(\widehat{Q}_{t,q}(\lambda)) = O(\lambda)$ (further details can be found in [35]). Thus according to (10), $L(\widehat{Q}_{t,q}(\lambda)) = O(\lambda^2)$, which implies that $L(\lambda_{\text{low}}) < L(\lambda_{\text{intermediate}})$ since $\lambda_{\text{low}} < \lambda_{\text{intermediate}}$.

By using a sufficiently small value for γ in (3) and a sufficiently large M, the approximation of the expected quantile loss will be sufficiently good to ensure that $\widehat{\text{EQL}}(\lambda_{\text{low}}) < \widehat{\text{EQL}}(\lambda_{\text{intermediate}})$. Consequently, the *HIL* procedure will iteratively select smaller and smaller values for the step size.

A final requirement for convergence is that λ_t approaches zero, such that $t\lambda_t \to \infty$ [35]. This is ensured by using a_t as given in Theorem 1. The algorithm is initiated using a step size of $\lambda_1 = \lambda$. At time $t = M$, the step size is reduced by dividing by a_M: $\lambda_M = \lambda M^p/(2M)^p = \lambda/2^p$. At time $t = 2M$: $\lambda_{2M} = \lambda/2^p \cdot (2M)^p/(3M)^p = \lambda/3^p$. Thus $t\lambda_t = kM \cdot \lambda_{kM} = kM \cdot \lambda/(k+1)^p \to \infty$ since $0 < p < 1$.

Convergence follows

$$
\lim_{\lambda \to 0, t\lambda_t \to \infty} \widehat{Q}_{t,q}(\lambda_t) = Q_q \quad\quad (11)
$$

$\square$

References

1. Abbasi B, Guillen M (2013) Bootstrap control charts in monitoring value at risk in insurance. Expert Syst Appl 40(15):6125–6135

2. Ancy S, Paulraj D (2020) Handling imbalanced data with concept drift by applying dynamic sampling and ensemble classification model. Comput Commun 153:553–560

3. Atefeh F, Khreich W (2015) A survey of techniques for event detection in twitter. Comput Intell 31(1):132–164

4. Bach SH, Maloof MA (2008) Paired learners for concept drift. In: 2008 Eighth IEEE international conference on data mining. IEEE, pp 23–32

5. Bulling A, Blanke U, Schiele B (2014) A tutorial on human activity recognition using body-worn inertial sensors. ACM Computing Surveys (CSUR) 46(3):33

6. Choi B-Y, Moon S, Cruz R, Zhang Z-L, Diot C (2007) Quantile sampling for practical delay monitoring in internet backbone networks. Comput Netw 51(10):2701–2716

7. Gama J, Medas P, Castillo G, Rodrigues P (2004) Learning with drift detection. In: Brazilian symposium on artificial intelligence. Springer, pp 286–295

8. Gilli M et al (2006) An application of extreme value theory for measuring financial risk. Comput Econ 27(2-3):207–228

9. Lau AGF, Butler L (2018) A quantile-based approach to modelling recovery time in structural health monitoring. arXiv:1803.08444

10. Hammer HL, Yazidi A, Riegler MA, Rue H (2020) Efficient quantile tracking using an oracle. arXiv:2004.12588

11. Hammer HL, Yazidi A, Rue H (2022) Estimating Tukey depth using incremental quantile estimators. Pattern Recogn 122:108339

12. Hammer HL, Yazidi A (2018) Parameter estimation in abruptly changing dynamic environments using stochastic learning weak estimator. Appl Intell 48(11):4096–4112

13. Hammer HL, Yazidi A, Oommen BJ (2018) On the classification of dynamical data streams using novel "anti-bayesian" techniques. Pattern Recogn 76:108–124

14. Hammer HL, Yazidi A, Rue H (2019) A new quantile tracking algorithm using a generalized exponentially weighted average of observations. Appl Intell 49(4):1406–1420

15. Hammer HL, Yazidi A, Rue H (2020) Tracking of multiple quantiles in dynamically varying data streams. Pattern Anal Applic 23(1):225–237

16. Hammer HL, Yazidi A, Rue H (2021) Joint tracking of multiple quantiles through conditional quantiles. Inf Sci 563:40–58

17. Hasan M, Orgun MA, Schwitter R (2017) A survey on real-time event detection from the twitter data stream. J Inform Sci, p 0165551517698564

18. Nikolaus H (2011) Econometrics of financial high-frequency data. Springer Science & Business Media, Berlin

19. Kejariwal A, Kulkarni S, Ramasamy K (2015) Real time analytics: algorithms and systems. Proceedings of the VLDB Endowment 8(12):2040–2041

20. Koenker R, Chernozhukov V, He X, Peng L (2017) Handbook of quantile regression

21. Krempl G, Žliobaite I, Brzeziński D, Hüllermeier E, Last M, Lemaire V, Noack T, Shaker A, Sievi S, Spiliopoulou M et al (2014) Open challenges for data stream mining research. ACM SIGKDD Explorations Newsletter 16(1):1–10

22. Kwapisz JR, Weiss GM, Moore SA (2011) Activity recognition using cell phone accelerometers. ACM SigKDD Explorations Newsletter 12(2):74–82

23. Ashwin L (2015) Data streaming algorithms for the kolmogorov-smirnov test. In: Big data (big data), 2015 IEEE international conference on. IEEE, pp 95–104

24. Li Z, Huang W, Xiong Y, Ren S, Zhu T (2020) Incremental learning imbalanced data streams with concept drift: the dynamic updated ensemble algorithm. Knowl-Based Syst 195:105694

25. Liu A, Lu J, Zhang G (2020) Diverse instance-weighting ensemble based on region drift disagreement for concept drift adaptation. IEEE Transactions on Neural Networks and Learning Systems 32(1):293–307

26. Liu D, Wu Y, He J (2016) Fp-elm: An online sequential learning algorithm for dealing with concept drift. Neurocomputing 207:322–334

27. Liu J, Zheng W, Zheng L, Lin N (2018) Accurate quantile estimation for skewed data streams using nonlinear interpolation. IEEE Access

28. Lu J, Liu A, Dong F, Gu F, Gama J, Zhang G (2018) Learning under concept drift: A review. IEEE Trans Knowl Data Eng 31(12):2346–2363

29. Ge L, Lu W, Ke Yi, Cormode G (2016) Quantiles over data streams: Experimental comparisons, new analyses, and further improvements. The VLDB Journal, 1–24

30. Ma Q, Muthukrishnan S, Sandler M (2013) Frugal streaming for estimating quantiles. In: Space-efficient data structures, streams, and algorithms. Springer, pp 77–96

31. Mahdi OA, Pardede E, Ali N, Cao J (2020) Fast reaction to sudden concept drift in the absence of class labels. Appl Sci 10(2):606

32. Marjanovic B (2022) Huge stock market dataset, historical daily prices and volumes of all u.s. stocks and etfs. Accessed: 2022-01-20

33. Maunu T, Lerman G (2020) Depth descent synchronization in SO(D). arxiv: Optimization and Control

34. Naeem M, Jamal T, Diaz-Martinez J, Butt SA, Montesano N, Tariq MI, De-la-Hoz-Franco E, De-La-Hoz-Valdiris E (2022) Trends and future perspective challenges in big data. In: Advances in intelligent data analysis and applications. Springer, pp 309–325

35. Nicopolitidis P, Papadimitriou GI, Pomportsis AS (2003) Learning automata-based polling protocols for wireless lans. IEEE Trans Commun 51(3):453–463

36. Frank Norman M (1972) Markov processes and learning models, vol 84. Academic Press, New York

37. Pratama M, Lu J, Zhang G (2015) Evolving type-2 fuzzy classifier. IEEE Trans Fuzzy Syst 24(3):574–589

38. Priya S, Annie Uthra R (2021) Comprehensive analysis for class imbalance data with concept drift using ensemble based classification. Journal of Ambient Intelligence and Humanized Computing 12(5):4943–4956

39. Ramírez-Gallego S, Krawczyk B, García S, Woźniak M, Herrera F (2017) A survey on data preprocessing for data stream mining: Current status and future directions. Neurocomputing 239:39–57

40. Ross GJ (2014) Sequential change detection in the presence of unknown parameters. Stat Comput 24(6):1017–1030

41. Sollid SJM, Rimstad R, Rehn M, Nakstad AR, Tomlinson A-E, Strand T, Heimdal HJ, Nilsen JE, Sandberg M (2012) Oslo government district bombing and utøya island shooting july 22, 2011: the immediate prehospital emergency medical service response. Scandinavian Journal of Trauma, Resuscitation and Emergency Medicine 20(1):3

42. Sommers J, Barford P, Duffield N, Ron A (2007) Accurate and efficient sla compliance monitoring. In: ACM SIGCOMM Computer communication review, vol 37. ACM, pp 109–120

43. Sommers J, Barford P, Duffield N, Ron A (2010) Multiobjective monitoring for sla compliance. IEEE/ACM Transactions on Networking (TON) 18(2):652–665

44. Stahl V, Fischer A, Bippus R (2000) Quantile based noise estimation for spectral subtraction and wiener filtering. In: Acoustics, speech, and signal processing, 2000. ICASSP'00. Proceedings. 2000 IEEE international conference on, vol 3. IEEE, pp 1875–1878

45. Yu S, Ke T, Zhu Z, Yao X (2018) Concept drift adaptation by exploiting historical knowledge. IEEE Transactions on Neural Networks and Learning Systems 29(10):4822–4832

46. Tierney L (1983) A space-efficient recursive procedure for estimating a quantile of an unknown distribution. SIAM J Sci Stat Comput 4(4):706–711

47. Tiwari N, Pandey PC (2018) A technique with low memory and computational requirements for dynamic tracking of quantiles. Journal of Signal Processing Systems, 1–12

48. Yazidi A, Hammer H (2017) Multiplicative update methods for incremental quantile estimation. IEEE Transactions on Cybernetics 49(3):746–756

49. Zhang L, Guan Y (2008) Detecting click fraud in pay-per-click streams of online advertising networks. In: Distributed computing systems, 2008. ICDCS'08. The 28th international conference on. IEEE, pp 77–84

50. Zhang X, Alexander L, Hegerl GC, Jones P, Tank AK, Peterson TC, Trewin B, Zwiers FW (2011) Indices for monitoring changes in extremes based on daily temperature and precipitation data. Wiley Interdisciplinary Reviews: Climate Change 2(6):851–870

Hugo L. Hammer received the M.Sc. and Ph.D. degrees from the Norwegian University of Science and Technology, in 2003 and 2008, respectively. He is currently a Professor with the Department of Computer Science, Oslo Metropolitan University, Oslo, Norway and an Adjunct Chief Research Scientist at Simula Metropolitan Center, Oslo, Norway. Before joining OsloMet, he worked as a researcher with Norwegian Computer Center, Oslo, Norway. His research interests include computer intensive statistical methods, uncertainty quantification and transparent machine learning methods.

Anis Yazidi received the M.Sc. and Ph.D. degrees from the University of Agder, Grimstad, Norway, in 2008 and 2012, respectively. He is currently a Professor with the Department of Computer Science, Oslo Metropolitan University, Oslo, Norway. Before joining HIOA, he worked as a researcher with Teknova AS, Grimstad, Norway. His current research interests include machine learning, learning automata, stochastic optimization, software-defined networks and cloud computing. He led the research group Autonomous Systems and Networks at OsloMet from 2015 to 2018. He is currently the leader of the research group Applied Artificial Intelligence at OsloMet.

Michael A. Riegler received the Ph.D. degree from the Department of Informatics, University of Oslo, Oslo, Norway, in 2015. He is currently working as a Chief Research Scientist at SimulaMet, Oslo, Norway and holds a position as associate Professor at the University of Tromsø, Norway. His research interests include machine learning, multi modal analysis and understanding, image processing, image retrieval, explainable machine learning and applications of artificial intelligence in medicine and social sciences.

Håvard Rue received the M.Sc. and Ph.D. degrees from the Norwegian Institute of Technology, in 1988 and 1993, respectively. He is currently a professor in statistics with the Computer, Electrical and Mathematical Science and Engineering Division, King Abdullah University of Science and Technology (KAUST), Saudi Arabia. Before joining KAUST, he worked as a Professor in statistics at Norwegian University of Science and Technology. His research interests lie in computational Bayesian statistics and Bayesian methodology such as priors, sensitivity and robustness. His main body of research is built around the R-INLA project (www.r-inla.org), which aims to provide a practical tool for approximate Bayesian analysis of latent Gaussian models, often at extreme data scales. This project also includes efforts to use stochastic partial differential equations to represent Gaussian fields, for the use in spatial statistics. He is the leader of the research group Bayesian Computational Statistics and Modeling.