Change Detection in Multivariate data streams: Online Analysis with Kernel-QuantTree

Michelangelo Olmo Nogara Notarianni[1](✉), Filippo Leveni[1], Diego Stucchi[2], Luca Frittoli[1], and Giacomo Boracchi[1]

[1] Politecnico di Milano (DEIB), Milan, Italy
{michelangeloolmo.nogara, filippo.leveni, luca.frittoli, giacomo.boracchi}@polimi.it
[2] STMicroelectronics, Agrate Brianza, Italy
diego.stucchi@st.com

Abstract. We present Kernel-QuantTree Exponentially Weighted Moving Average (KQT-EWMA), a non-parametric change detection algorithm which combines the Kernel-QuantTree (KQT) histogram as a model for the data distribution and the EWMA statistic to monitor multivariate data streams online. KQT-EWMA performs monitoring without any assumptions on the stationary or post-change distribution and, most remarkably, it is supported by theoretical results that enable controlling false alarms by operating at a pre-determined Average Run Length (ARL_0), i.e., the average number of stationary samples monitored before a false alarm is triggered. The ability to control ARL_0, a feature that is missing in most of the non-parametric change-detection models, provides flexibility and numerous practical advantages to KQT-EWMA. Our experiments, conducted on both synthetic and real-world datasets, demonstrate KQT-EWMA's ability to maintain a target ARL_0 while achieving detection delays comparable to or lower than existing state-of-the-art methods designed to work in the same conditions.

Keywords: Online Change Detection · Nonparametric Monitoring · Multivariate Data Streams

1 Introduction

Change detection is a frequently faced challenge in data stream analysis, where the statistical properties of some monitored variable, e.g. a measurement acquired by a sensor, may change over time. In machine learning, changes in data distribution are known as concept drifts and pose challenges for classifiers and learning systems in general, requiring continuous adaptation to evolving data-generating processes.

In many application domains such as industrial monitoring, communication networks, and computer security, data come in virtually unlimited streams and need to be monitored *online*. In particular, each new observation needs to be processed immediately after being acquired, and must be evaluated considering the whole data stream seen so far, while using a limited amount of memory and performing a fixed number of operations. In this study, we focus on online change-detection methods for multivariate data streams, which require algorithms capable of handling multidimensional data within these computational requirements and storage limitations. Another important challenge is posed by monitoring in a non-parametric manner, i.e., without any assumption on the initial data distribution. Non-parametric methods are particularly useful in real-world scenarios where the distribution of data is typically unknown. Unfortunately, most of the non-parametric change-detection algorithms are designed to monitor univariate data streams [15]. On top of that, controlling false alarms is a significant concern when each detected change can trigger costly interventions. Unfortunately, most online change-detection algorithms for multivariate data streams, particularly the non-parametric ones, struggle to effectively control false alarms. This paper addresses both these challenges by proposing a method that is non-parametric and capable of maintaining a target false alarm rate.

Online change detection monitoring techniques can be grouped into two categories: *one-shot* methods, which evaluate fixed-size batches of data points, and *sequential* methods, which do not require a fixed sample size and take into account the whole data stream. QuantTree (QT) is a *one-shot* non-parametric solution which, supported by theoretical results, guarantees a pre-set constant false positive rate (FPR). First presented in [2], QT algorithm defines a histogram, partitioning the d-dimensional input space. Non-parametric statistics can be computed over it, enabling change detection in multivariate data streams batch-wise. A fundamental limitation of QT is discussed in [12]: its splits are defined along the space axis, resulting in a hyper-rectangular partitioning that does not always adhere to the input distribution. To address this problem, a preprocessing stage is typically introduced to align the split directions to the principal components of the training set. However, it was observed [12] that this preprocessing can also worsen the control over false alarms. Hence, Kernel-QuantTree (KQT), a generalized version of QT which partitions the space using kernel functions learned from data, was introduced. The increased flexibility of the histogram in modeling the data distribution results in *one-shot* monitoring of multivariate data streams with increased detection power. Moreover, KQT is invariant to roto-translations of the data set, thus it is invariant to standard preprocessing techniques such as PCA [12]. However, KQT is a batch-wise monitoring scheme using fixed-size windows and it fails to leverage the knowledge of the entire data stream distribution, thereby hindering fast detection of changes in an online scenario.

A *sequential* version of QT algorithm, QT-EWMA, was presented in [4]. QT-EWMA computes the Exponentially Weighted Moving Average (EWMA) statistic on a QT histogram, thus considering the entire data stream acquired up to the current time instant t to monitor the data distribution. Moreover, QT-EWMA can control the average time elapsed before a false alarm is triggered (ARL_0). Although being a truly *sequential* extension of QT, QT-EWMA inherits the same weaknesses of its *one-shot* counterpart.

We propose Kernel-QuantTree Exponentially Weighted Moving Average (KQT-EWMA), a novel *sequential* non-parametric change-detection algorithm for multivariate data streams that extends KQT to the online scenario, following the approach of QT-EWMA. The theoretical properties of KQT guarantee that KQT-EWMA is completely non-parametric since the distribution of our statistic does not depend on the data distribution, hence the thresholds controlling the ARL_0 can be set a priori, as in QT-EWMA. These thresholds guarantee by design a constant false alarm probability over time, thus a fixed false alarm rate at any time instant during monitoring.

Our extensive experimental analysis on both synthetic and real datasets shows that KQT-EWMA outperforms state-of-the-art existing methods, successfully extending KQT properties to the online scenario. Specifically, KQT-EWMA achieves control over the ARL_0 at a lower detection delay compared to competitors. We will show that, by relying on a precise partition of the space, KQT-EWMA outperforms QT-EWMA in complex scenarios, e.g., when analyzing data from multimodal distributions.

2 Problem formulation

We consider a virtually unlimited multivariate data stream $x_1, x_2, \ldots$ in $\mathbb{R}^d$ where data samples x_t are i.i.d. realizations of a random variable with unknown distribution ϕ_0. We define the *change-point* $t = \tau$ as the unknown time instant when the distribution ϕ_0 experiences a change to ϕ_1, i.e.:

$$x_t \sim \begin{cases} \phi_0 & \text{if } t < \tau \\ \phi_1 & \text{if } t \geq \tau. \end{cases} \tag{1}$$

We consider a training set TR of N stationary realizations from ϕ_0 to be provided to fit a model $\hat{\phi}_0$ of the initial distribution.

After estimating $\hat{\phi}_0$, an online change-detection algorithm typically computes a statistic T_t at each observation x_t to assess whether the new sequence $\{x_1, \ldots, x_t\}$ contains a change point or not. The decision rule usually involves checking whether $T_t > h_t$, where h_t is a given threshold. The detection time t^* is identified as the earliest time instant when sufficient statistical evidence indicates a change in the distribution, i.e.:

$$t^* = \min\{t : T_t > h_t\}. \tag{2}$$

A desirable requirement of change-detection algorithms is that the sequence of thresholds $\{h_t\}_t$ can be set *a priori* to guarantee a predefined $ARL_0 = \mathbb{E}_{\phi_0}[t^*]$, where the expectation is taken assuming that the whole data stream is drawn from ϕ_0. ARL_0, i.e. the average time before a false alarm occurs, is similar to Type I error probability control in hypothesis testing. The goal is to detect a distribution change as soon as possible, thereby minimizing the detection delay $t^* - \tau$, while aiming for an empirical ARL_0 that closely approximates the predefined target value set beforehand.

3 Related work

Change-detection algorithms are often *parametric* since they are based on certain hypotheses about the data distribution ϕ_0. As an example, the Change Point Model (CPM) [15] based on the Hotelling test relies on the assumption that the initial distribution ϕ_0 of data conforms to Gaussian distribution. CPM performs online monitoring of data streams with theoretical guarantees regarding ARL_0 control. It can also detect changes given an unknown non-Gaussian distribution when implemented with the Lepage statistic [10], but although straightforward to obtain assurance regarding false alarm control, its descriptive potential for the data generating process is limited. Another significant limitation of rank-based statistics, such as the Lepage and Mann-Whitney tests, is their difficulty in extension to the multivariate case.

A *semi-parametric* change-detection strategy, Semi-Parametric Log-Likelihood (SPLL), was presented in [8]. SPLL models the initial data distribution ϕ_0 by fitting a Gaussian Mixture Model (GMM) $\hat{\phi}_0$ on a training set and then compares new incoming batches with batches from the training set using a likelihood test. Since SPLL does not provide a way to set the detection threshold a priori to control the ARL_0, it was combined with CPM [4]. In SPLL-CPM [4], SPLL reduces the dimensionality of incoming samples by computing their log-likelihood, then the resulting univariate sequence is monitored by a non-parametric extension of CPM leveraging the Lepage test statistic [10]. Again, the main limitation of both SPLL and SPLL-CPM is the assumption that ϕ_0 can be well approximated by a probability distribution of a known family (a GMM), which is not true in general.

There are only a few other multivariate methods that perform *non-parametric* change-detection. Scan-B [9] employs a Maximum Mean Discrepancy (MMD) statistic and can be configured in order to achieve a target ARL_0. However, the thresholds for this method are defined by analyzing the asymptotic behavior of ARL_0 when the size B of the sliding window is large [9]. Therefore, it cannot guarantee an accurate ARL_0 control. The NEWMA algorithm [7], also based on MMD, examines the relationship between two EWMA statistics with distinct forgetting factors. A limitation of this approach is that setting the ARL_0 thresholds requires the known analytical expression of ϕ_0. The Kernel-CUSUM [14] algorithm avoids assumptions about data distribution ϕ_0, but relies on a truncated approximation for ARL_0, which results in the underestimation of the thresholds [14]. QT [2] and QT-EWMA [4, 5] are histogram-based change-detection methods having the desirable property that the distribution of test statistics, defined over bin probabilities, does not depend on the initial distribution ϕ_0. This allows to set detection thresholds a priori via Monte Carlo procedures, allowing for efficient false alarm control. KQT [12] defines histogram bins via nonlinear partition of the input space, resulting in a powerful change-detection algorithm in multivariate data streams. However, as a *one-shot* method, it cannot be directly employed for online change detection. Our proposal, namely KQT-EWMA, aims to extend KQT to the *sequential* scenario while retaining the capability of controlling false alarms given a target ARL_0 defined beforehand.

4 Kernel-QuantTree EWMA

We present KQT-EWMA, a novel change-detection algorithm which combines a KQT histogram [12], used as a model $\hat{\phi}_0$ of the stationary distribution ϕ_0, and a time-dependent statistic T_t based on an Exponential Weighted Moving Average [4]. In Section 4.1, we illustrate the KQT-EWMA algorithm, describing the histogram construction, the threshold computation to control ARL_0, and the online monitoring process. In Section 4.2, we compare the computation complexity of KQT-EWMA against the alternatives considered in the experimental section. Finally, in Section 4.3 we discuss the limitations of KQT-EWMA.

4.1 The KQT-EWMA algorithm

Algorithm 1 illustrates the training and inference phases of the KQT-EWMA algorithm. First, the KQT histogram $h = \{(S_k, \pi_k)\}_{k=1}^K$ is constructed over the training set $TR \subset \mathbb{R}^d$ to match a set of target probabilities $\{\pi_j\}_{j=1}^K$ (line 1), as described in [12]. The histogram construction process consists in iteratively splitting the input space into K bins defined as sub-level sets of a measurable function $\{f_k : \mathbb{R}^d \to \mathbb{R}\}_{k=1}^K$, which measures distances between points and fixed centroids using a kernel function. We remark that $K-1$ bins defined by KQT are compact subsets of $\mathbb{R}^d$. A sample that does not fall into any of these is assigned to the *residual* bin, which covers the unbounded remaining part of the space. We consider the Mahalanobis and the Weighted Mahalanobis (WM) distances as kernel functions, as in [12], but the properties of KQT hold for *any* measurable function projecting a multivariate vector in $\mathbb{R}^d$ on a single dimension.

As in [4], KQT-EWMA computes the weighted averages $\{Z_{j,t}\}$, which keep track of the percentage of data stream samples $\{x_1, \ldots, x_t\}$ falling in each bin S_j; to this purpose, we define K binary statistics $\{y_{j,t}\}_j$ as:

$$y_{j,t} = \mathbb{1}(x_t \in S_j), \tag{3}$$

for each $j \in \{1, \ldots, K\}$ and $t \geq 1$. As discussed in [4], under the assumption that the monitored samples $x_t \sim \phi_0$ are stationary, the expected values of the binary statistics in (3) can be approximated (line 2) as:

$$\mathbb{E}[y_{j,t}] \approx \hat{\pi}_j := \frac{N\,\pi_j}{N+1}, \quad j < K \quad \text{and} \quad \mathbb{E}[y_{K,t}] \approx \hat{\pi}_K := \frac{N\,\pi_K + 1}{N+1}. \tag{4}$$

During monitoring, each incoming sample x_t is used to update the weighted averages $\{Z_{j,t}\}$ and to compute the test statistic T_t. First, the sample is processed by the histogram h to obtain the binary statistics $\{y_{j,t}\}$ (line 6), which in turn are used to update the weighted averages $\{Z_{j,t}\}$ (line 7) as

$$Z_{j,t} = (1-\lambda)\, Z_{j,t-1} + \lambda\, y_{j,t} \quad \text{where} \quad Z_{j,0} = \hat{\pi}_j. \tag{5}$$

The past samples are weighted by an exponential curve which decreases with time constant λ. The expected value of the $Z_{j,t}$ statistic under ϕ_0 approximates $\hat{\pi}_j$,

Algorithm 1: KQT-EWMA

```
   Input: training set $TR \subset \mathbb{R}^d$, target probabilities $\{\pi_j\}_{j=1}^K$, thresholds $\{h_t\}_t$,
          data stream to be monitored $x_1, x_2, \dots, x_t, \dots \subset \mathbb{R}^d$
   Output: detection flag $CD$, detection time $t^*$
 1 Construct the KQT histogram $\{S_j, \pi_j\}_{j=1}^K$ over $TR$ as in [12]
 2 Calculate the expected probabilities $\{\hat{\pi}_j\}_{j=1}^K$ as in (4)
 3 Initialize the weighted averages $Z_{j,0} \leftarrow \hat{\pi}_j$ for each bin $j \in \{1, \dots, K\}$
 4 Initialize the detection flag $CD \leftarrow$ False and the detection time $t^* \leftarrow \infty$
 5 for $t = 1 \dots$ do
 6     Compute the binary mask $y_{j,t} \leftarrow \mathbb{1}(x_t \in S_j)$
 7     Update the random variables $Z_{j,t} \leftarrow (1-\lambda)\, Z_{j,t-1} + \lambda\, y_{j,t},\ \forall j = 1 \dots, K$
 8     Compute the test statistic $T_t \leftarrow \sum_{j=1}^K (Z_{j,t} - \hat{\pi}_j)^2 / \hat{\pi}_j$
 9     if $T_t > h_t$ then
10         $CD \leftarrow$ True, $t^* \leftarrow t$
11         break
12 return $CD$, $t^*$
```

i.e. $\mathbb{E}[Z_{j,t}] \approx \hat{\pi}_j$ for $j = 1, ..., K$, thus the change-detection statistic is computed (line 8) as follows:

$$T_t = \sum_{j=1}^{K} \frac{(Z_{j,t} - \hat{\pi}_j)^2}{\hat{\pi}_j}. \tag{6}$$

T_t measures the overall difference between the proportion of points in each bin S_j, represented by $Z_{j,t}$, and their approximated expected values $\hat{\pi}_j$ under ϕ_0, thus corresponds to the Pearson statistic. The statistic naturally increases as a consequence of a change $\phi_0 \to \phi_1$ that modifies the probability of at least one bin. Finally, the statistic T_t is compared against the corresponding threshold h_t to detect a change (line 10).

False Alarms and Threshold computation strategy. KQT-EWMA algorithm inherits from Kernel-QuantTree the fundamental property that the distribution of the statistics in (6) does not depend on ϕ_0, so the thresholds $\{h_t\}_t$ can be defined a priori to control ARL_0 on any data stream, which is defined as:

$$ARL_0 = \mathbb{E}_{\phi_0}[t^*] = \frac{1}{\alpha}. \tag{7}$$

As explained in [5], to guarantee the constant false alarm probability, the thresholds must satisfy the following equation:

$$\mathbb{P}(T_t > h_t \mid T_k \leq h_k \ \forall k < t) = \alpha \quad \forall t \geq 1. \tag{8}$$

Moreover, since t^* is a Geometric random variable with parameter α, the probability of encountering a false alarm before time t can be determined through the geometric sum:

$$\mathbb{P}\left(t^* \leq t\right) = \sum_{k=1}^{t} \alpha(1-\alpha)^{k-1} = \alpha \cdot \frac{1-(1-\alpha)^t}{\alpha} = 1-(1-\alpha)^t. \tag{9}$$

We leverage results in [5] which guarantees that, to estimate the thresholds, one can directly simulate the construction of QT histograms on a training set $TR \sim \phi_0$ of size N by drawing its bin probabilities from the Dirichlet distribution, $(p_1, \dots, p_K) \sim \mathcal{D}(\pi_1 N, \pi_2 N, \dots, \pi_K N+1)$. This approach holds with KQT given any kernel function, i.e. we can use the same threshold sequences given any measurable function, including linear split functions, which produce a QT histogram. Therefore, the same thresholds, computed in a Monte Carlo scheme, can be used for QT-EWMA and KQT-EWMA to guarantee a constant false alarm probability over time. The thresholds do not depend on the data distribution ϕ_0 nor the data dimension d. The entire simulation procedure must be repeated when changing the λ parameter of the EWMA statistic, the target bin probabilities $\{\pi_j\}_{j=1}^{K}$, or the training set size N.

4.2 Computational complexity

Since efficiency is key in online monitoring, we analyze the computational complexity of KQT-EWMA in comparison with QT, QT-EWMA, KQT, and SPLL, SPLL-CPM, and Scan-B. The results are summarized in Table 1. Further explanations can be found in [5, 12]

The training of a KQT given a training set TR of N points comprises *i*) the projection of TR by f_k, whose cost depends on the specific kernel function, *ii*) the computation of the split value, which costs $\mathcal{O}(N)$, and *iii*) the centroid selection. The cost of computing the Euclidean distance - or other distances based on l_p norms - is $\mathcal{O}(d)$, while the Mahalanobis costs $\mathcal{O}(d^2)$ and the Weighted Mahalanobis (WM) costs $\mathcal{O}(M\ d^2)$, where M is the number of Gaussian components fitted to TR and d is the data dimension. The centroid selection criteria is based on the information gain, which estimate is dominated by the computation of the determinant of the sample covariance matrix, which costs $\mathcal{O}(d^3)$. Overall, the cost of the index computation is multiplied by the number of centroids V tested during the selection procedure; therefore, an upper bound for the cost of KQT construction is $\mathcal{O}(K\ V\ (N + M\ N\ d^2 + d^3))$ when using the WM distance and the information gain criteria. During monitoring, the only operation performed is the projection by f_k of the samples, resulting in a cost of $\mathcal{O}(K\ M\ d^2)$ in case of the WM distance.

4.3 Discussion and limitations

The main limitation of KQT-EWMA is its dependence on the sample covariance matrix, which estimation is challenging in high-dimensional data streams, requiring an increasing number of samples. In KQT, given any kernel function, the centroid selection criteria is the maximum information gain: the best split lowers the

Table 1: Training and inference costs of KQT-EWMA with Weighted Mahalanobis (WM) distance and distances derived from l_p norms (e.g. Euclidean distance when $p = 2$), compared against the other considered methods. V is the number of centroids tested to build each bin, M is the number of Gaussian components fit on the dataset, K is the number of bins, and N is the training set size. As for the other methods, m is the number of Gaussian components and w is the window length used by SPLL; n is the number of windows of B samples employed by Scan-B.

Method	Training Cost	Inference Cost (per sample)
KQT-EWMA (WM)	$O(K\,V(N + M\,N\,d^2 + d^3))$	$O(K\,M\,d^2)$
KQT-EWMA (l_p)	$O(K\,V(N + N\,d + d^3))$	$O(K\,d)$
QT-EWMA	$O(K\,N\,\log N)$	$O(K)$
SPLL (online)	$O(m\,N\,d^2)$	$O(m\,d + w\,\log w)$
Scan-B	N.A.	$O(n\,B\,d)$

data entropy [12], which is computed as $H(B) = (1/2)\log\left((2\pi e)^d \det(\text{cov}[B])\right)$, where cov$[B]$ is the sample covariance matrix computed over a set of points B. Moreover, the sample covariance matrix estimated from the training set TR is used to define the Mahalanobis and the WM distances. The problem of determining the minimal sample size N that guarantees that the sample covariance matrix approximates the actual covariance matrix depends on the data distribution, as well explained in [13]. Our experiments shows that KQT-EWMA can lose control over ARL_0 when few training points are available.

QT-EWMA-update Algorithm [5] is an effective monitoring scheme when N is relatively small, i.e. when there are a few training samples, as this estimates the bin probabilities incrementally as new observations are available, as long as no changes are detected. While an incremental variant of KQT-EWMA can be implemented, this would be impractical due to computational and memory requirements, as it would require re-computing covariance matrices and centroids (possibly in a high dimensional space) at each update.

5 Experiments

The goal of our experiments is to show that KQT-EWMA controls the false alarms while achieving state-of-the-art detection delays. To do this, we will show empirical results obtained on both synthetic and real-world data streams. In KQT-EWMA, as in QT-EWMA [4], we set the number of bins to $K = 32$ and uniform target probabilities $\pi_j = 1/K$. The exponential decay of the EWMA statistic is given by a time constant $\lambda = 0.05$. To monitor with QT and SPLL we set the batch size $\nu = 32$ as in [4], and we employ the original configuration of the Scan-B algorithm [9] ($n = 5$ windows of $B = 100$ samples), if not specified otherwise. We set window length $w = 1000$ for SPLL-CPM. SPLL is built fitting m Gaussian components, as specified in the experimental section; KQT-EWMA is always built fitting $M = 4$ Gaussian components on the dataset; the number of centroids tested to build each bin is $V = 250$.

5.1 Datasets

Synthetic: As in [5], we generate synthetic data streams in spaces of increasing dimension $d \in \{2, 4, 8, 16, 32, 64\}$. We use Gaussian distributions ϕ_0 with a random covariance matrix, and then we compute the post-change distribution $\phi_1 = \phi_0(Q + v)$ as a random roto-translation of ϕ_0. The roto-translation parameters Q and v are generated using the CCM framework [3] to guarantee a fixed distance between the two distributions computed as the symmetric Kullback-Leibler divergence $sKL(\phi_0, \phi_1) \in \{0.5, 1, 1.5, 2, 2.5, 3\}$. We expand our analysis to datasets sampled from bi-modal and tri-modal Gaussians to show the benefits of the distribution estimation with a KQT histogram for change detection.

Real-world: As in [4], we also test our change-detection method on seven multivariate classification datasets of varying dimensionality: El Niño Southern Oscillation ("niño", $d = 5$), Physicochemical Properties of Protein Ternary Structure ("protein", $d = 9$), two of the Forest Covertype datasets ("spruce" and "lodgepole", $d = 10$), Credit Card Fraud Detection ("credit", $d = 28$), Sensorless Drive Diagnosis ("sensorless", $d = 48$), and MiniBooNE particle identification ("particle", $d = 50$). We preprocess these datasets from the UCI Machine Learning Repository [6] as in [4]: datasets are standardized and we sum to each component of "sensorless", "particle", "spruce" and "lodgepole" imperceptible Gaussian noise to avoid repeated values, which harm the construction of QT histograms. The distributions of these datasets are considered to be stationary [4]. We randomly sample the data streams and introduce changes $\phi_0 \to \phi_1$ by shifting the each distribution by a random vector drawn from a d-dimensional Gaussian scaled by the total variance of the dataset. We show the analysis of UCI datasets as the average results obtained over all the datasets.

Our experiments also include the INSECTS dataset [11], which contains $d = 33$ attributes derived from the wing-beat frequency of various insects, captured via an optical sensor. This dataset, tailor-made for change-detection techniques, contains records under diverse environmental conditions impacting insect flight behaviors. We focus on the abrupt-change variant of this dataset, which includes five distinct distribution changes $\phi_0 \to \phi_1 \to \cdots \to \phi_5$. We sample data points from these distributions to build our training set TR and test data streams. Results obtained over the five changes present in the INSECTS datasets are averaged and shown all together.

5.2 Figures of merit

Empirical $\mathbf{ARL_0}$. To assess whether KQT-EWMA and the other considered methods control the target ARL_0 (see (7)), we compute its empirical value as the average time before raising a false alarm on data streams we sample from ϕ_0. Empirical ARL_0 values are measured on 4000 data streams drawn from ϕ_0, given a target ARL_0 taking values in {500, 1000, 2000, 5000}. We generate stationary data streams of length $L = 6 \cdot ARL_0$ - the corresponding probability to detect a false alarm in each sequence is thus $\mathbb{P}(t^* \leq L) \approx 0.9975$, as in [4].

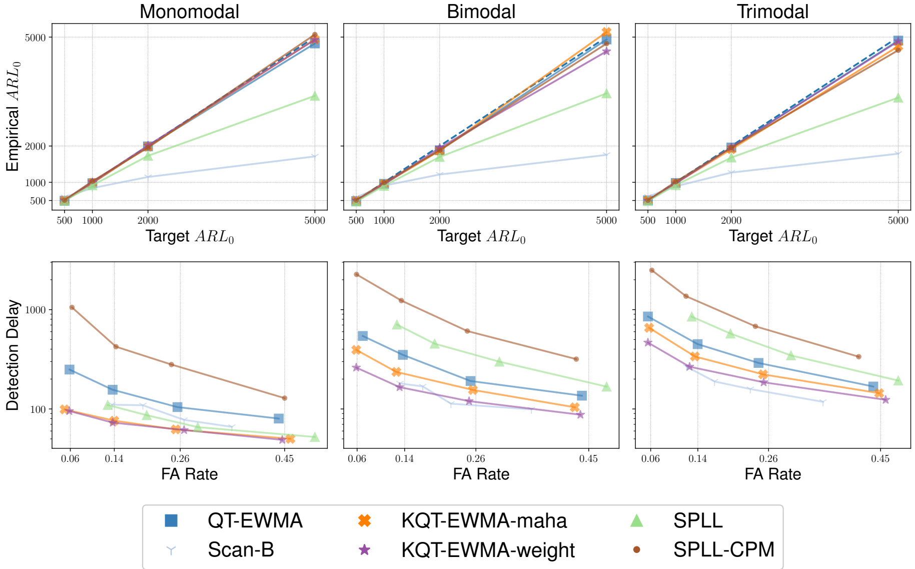

Fig. 1: Empirical ARL_0 and detection delay achieved by the considered methods monitoring data streams generated by Gaussian mixtures with increasing number of components (1, 2, 3). We show that as the number of components increases, KQT-EWMA with Weighted Mahalanobis (WM) distance advantage in terms of detection delay increases, achieving in general the lowest delays while controlling false alarms. In all the experiments, the GMM used to compute the WM distance fits $M = 4$ components.

Detection Delay. We evaluate the detection power of KQT-EWMA and the other considered methods by their detection delay, i.e., $ARL_1 = \mathbb{E}[t^* - \tau]$, where the expectation is taken assuming that a change point τ is present. Again, we set the target ARL_0 a priori, $ARL_0 \in \{500, 1000, 2000, 5000\}$. Results are averaged over 4000 data streams of length $6 \cdot ARL_0$, each containing a change point at $\tau = 300$. This is a difference compared to the analysis in [4,5], where the average detection delay is computed at any given target ARL_0 on sequences of fixed length. We use sequences of the same length to estimate detection delay and ARL_0 to achieve a fair comparison between these two quantities.
False Alarm rate. The False Alarms (FA) rate is computed as the number of alarms raised at some $t < \tau$, averaged over 4000 experiments. By setting the target ARL_0 to $\{500, 1000, 2000, 5000\}$, we expect the percentage of false alarms to be $\{45\%, 26\%, 14\%, 6\%\}$, respectively, as stated by (9). The target FA rates are indicated in the plots by vertical dotted lines.

5.3 Results and Discussion

False alarms control. To show the control over false alarms, we plot the empirical ARL_0 obtained in our experiments against the target ARL_0 set a priori. We compare results obtained over data sampled from monomodal Gaus-

Results averaged over UCI+Credit datasets

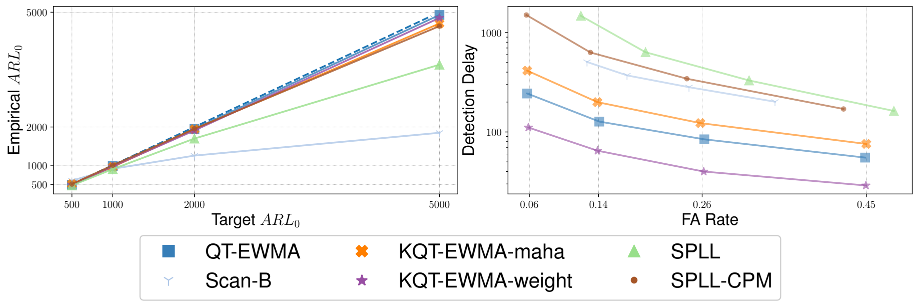

Fig. 2: Average empirical ARL_0 and detection delay on data streams sampled from the UCI datasets, excluding the highest-dimensional ones (i.e., "particle" and "sensorless"), where $N = 4096$ training samples are not enough for KQT-EWMA based on Mahalanobis and WM distances to properly control ARL_0. In this setting, KQT-EWMA with WM distance achieves by far the best performance, halving the detection delay of QT-EWMA while controlling the target ARL_0.

Results averaged over INSECTS dataset ($d = 33$)

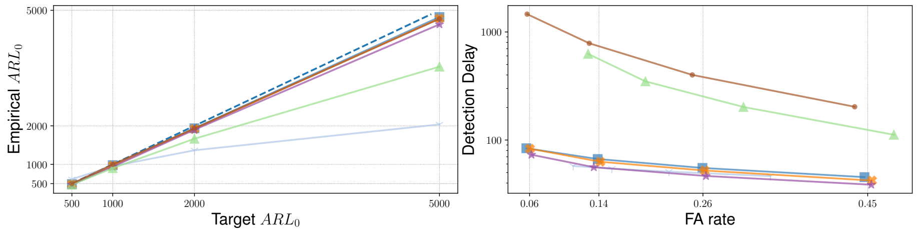

Fig. 3: Average empirical ARL_0 and detection delay on data streams sampled from the INSECTS dataset [11], with different combinations of initial distribution ϕ_0 and post-change distribution ϕ_1. QT-EWMA and KQT-EWMA achieve similar detection delays, while KQT-EWMA with WM distance struggles in controlling higher values of ARL_0.

sians with the same performance measures computed over multimodal Gaussian datasets (bimodal and trimodal, in Figure 1, first row). In all the experiments, the number of components used to fit the GMM and compute the WM distance is $M = 4$. QT-EWMA and SPLL-CPM can control all the target values chosen for ARL_0, while in general SPLL and Scan-B struggle in achieving high target $ARL_0 \in \{2000, 5000\}$. This is true also considering the results obtained with others real-world data sets (see Figures 2 and 3).

KQT-EWMA can control target values of ARL_0, while achieving the lowest detection delays in these scenarios, given a target FA rate (see Figure 1, second row). Figure 2 (first row) shows experimental results averaged over data streams sampled from UCI datasets “protein” ($d = 9$), “credit” ($d = 28$), “niño” ($d = 5$), “spruce” ($d = 10$) and “lodgepole” ($d = 10$) given $N = 4096$ training points. We show two high-dimensional datasets (“particle” and “sensorless”, $d = 50$ and

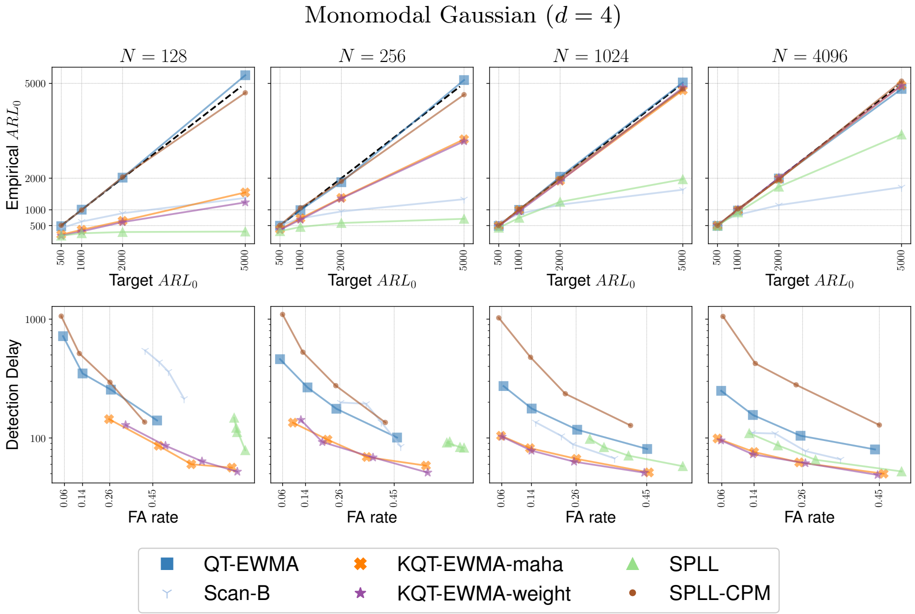

Fig. 4: Empirical ARL_0 and detection delay on data streams drawn from a Gaussian distribution in $d = 4$ dimensions, for varying training set sizes $N \in \{128, 256, 1024, 4096\}$. The empirical ARL_0 (first row) of QT-EWMA and SPLL-CPM always approaches the target values $(500, 1000, 2000, 5000)$, while the other methods cannot control the ARL_0. The False Alarm (FA) rates corresponding to the target ARL_0 are computed as in (9) and are shown as dotted vertical lines in the figures (second row). When the training set size N is sufficiently large ($N \in \{1024, 4096\}$), KQT-EWMA can control the FA rate, and achieves the lowest detection delay when using the Mahalanobis or the WM distance.

$d = 48$ respectively, see Figure 6) separately, since 4096 training points are not enough to estimate the sample covariance matrix in high dimensions, and KQT-EWMA based on Mahalanobis and WM distances do not properly control the ARL_0. In particular, KQT-EWMA with WM distance achieves low empirical ARL_0 values due to the difficulty in fitting a Gaussian Mixture Model in those settings.

Figure 7 plots the change magnitude $sKL(\phi_0, \phi_1)$ against the detection delay achieved by the considered methods monitoring Gaussian data sequences with a target $ARL_0 = 1000$. While all methods successfully control the false alarms, KQT-EWMA achieves the lowest detection delay, even in this setting where the parametric assumptions of SPLL are met (number of Gaussian components is set to $m = 1$). The advantage of KQT-EWMA over the alternatives is especially noticeable when the divergence between the pre- and post-change distributions is low ($sKL = 0.5$). As expected, the detection delay of all methods decreases when the change magnitude increases.

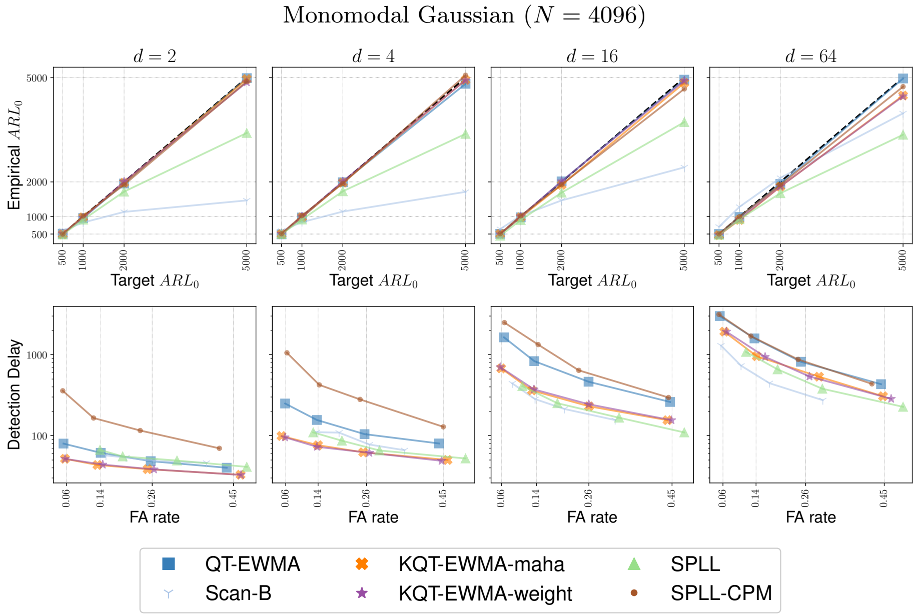

Fig. 5: Empirical ARL_0 and detection delay on data streams drawn from a Gaussian distribution with dimension $d \in \{2, 4, 16, 64\}$, trained over $N = 4096$ stationary samples. The empirical ARL_0 (first row) of KQT-EWMA, QT-EWMA, and SPLL-CPM always match the target, while Scan-B and SPLL fail. However, KQT-EWMA using the Mahalanobis distance cannot control the ARL_0 well when $d = 64$, as $N = 4096$ training points are not sufficient to estimate such a high-dimensional covariance matrix. Otherwise, the detection delay (second row) achieved with KQT-EWMA with Mahalanobis distance is the lowest achieved among the methods controlling the ARL_0.

Detection Delay vs False Alarms. To represent the detection power of these models, we plot the average detection delay against the percentage of false alarms (FA), which target values are given by target ARL_0 values as defined by (9).

Figure 1 (second row) shows that KQT-EWMA with Mahalanobis and WM distances achieves the lowest detection delay regardless of the number of modalities, alongside Scan-B. However, Scan-B is unable to control higher values of ARL_0, resulting in higher FA rates than the theoretically computed values (9). Similarly, SPLL also shows a shift towards higher FA rates for the same reason. This shift is more pronounced in SPLL because it does not perfectly control ARL_0 even at low values, and this behavior is also evident in the results on real data (Figures 2 and 3). If we compare KQT-EWMA to QT-EWMA, which has the second-best detection delay values in the Gaussian scenario (Figure 1, second row), we can observe that KQT-EWMA more than halves the detection delay of QT-EWMA in the monomodal case, and the difference increases as the complexity of the underlying distribution rises (bimodal and trimodal cases). This result is confirmed by all the experiments on both real (Figures 2 and 3) and synthetic (Figures 4 and 5) datasets, suggesting that the histogram

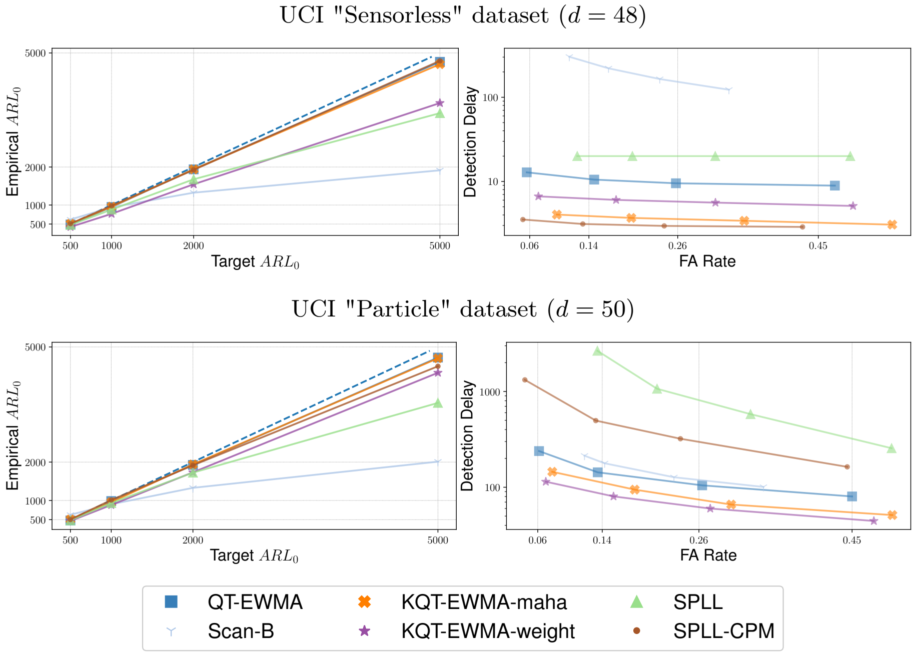

Fig. 6: Empirical ARL_0 and detection delay achieved by the considered methods monitoring the two high-dimensional UCI data sets "particle" ($d = 48$, above) and "sensorless" ($d = 50$, below). The $N = 4096$ training samples used in these experiments are not enough for KQT-EWMA based on Mahalanobis and WM distances to properly control ARL_0. Results are averaged over 4000 experiments.

construction strategy of KQT-EWMA, coupled with the Mahalanobis and WM distances, makes it more sensitive to changes in distribution, thereby achieving lower detection delays. Figure 5 shows the effects of *detectability loss* [1]: the ability to perceive a distribution change diminishes as the data dimensionality d increases while the distance between pre- and post-change distribution is fixed ($sKL = 1$), thus detection delays increase.

Figure 7 reports the relation between the detection delay and the change magnitude between pre- and post-change Gaussian sequences. We set target $ARL_0 = 1000$ for all methods. All methods successfully control the false alarms, and KQT-EWMA achieves the lowest detection delay, even when the parametric assumptions of SPLL are met (number of Gaussian components is set to $m = 1$). The advantage of KQT-EWMA over the alternatives is especially noticeable when the divergence between the pre- and post-change distributions is low ($sKL = 0.5$). As expected, the detection delay of all methods decreases when the change magnitude increases.

Our extensive analysis shows that KQT-EWMA consistently achieves the lowest detection delays across different scenarios. Overall, KQT-EWMA based on Mahalanobis and WM distances can detect distribution changes more effectively, especially when the complexity of the underlying distribution rises.

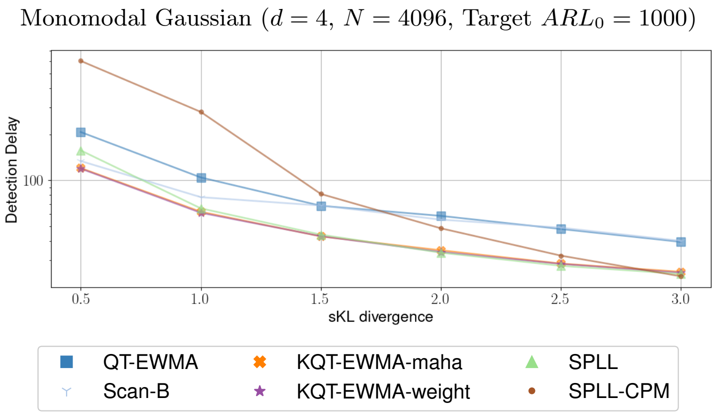

Fig. 7: Detection delay as a function of the magnitude of the change $\phi_0 \to \phi_1$ between pre- and post-change Gaussian sequences. We set the target $ARL_0 = 1000$. KQT-EWMA achieves the lowest detection delay, even in the challenging scenario when the change magnitude is low ($sKL = 0.5$). As expected, all methods decrease their detection delays when the change magnitude increases. We remark that the empirical ARL_0 achieved by SPLL and Scan-B is lower than the target.

6 Conclusion and Future Works

We introduce KQT-EWMA, a non-parametric online change-detection algorithm for multivariate data streams based on Kernel-QuantTree [12]. The theoretical results underpinning KQT-EWMA [5, 12] guarantee the control of false alarms independently on the initial data distribution. Our experiments on synthetic and real-world data streams show that KQT-EWMA achieves state-of-the-art detection delay while effectively controlling false alarms.

In particular, the algorithm can leverage any measurable kernel function and it is able to fit complex distributions, resulting in high detection power. The sequences of thresholds can be computed independently on the data distribution ϕ_0, the data dimension d, and the selected kernel function. Moreover, the monitoring scheme is invariant to roto-translation of the input data (when employing Mahalanobis and Weighted Mahalanobis distances, as shown in [12]), thus KQT-EWMA does not require any preprocessing step such as PCA.

Our experimental evaluation also delineates some limitations: while the computational complexity of QT-EWMA scales well with the data dimension d during both training and testing phases, KQT-EWMA's computational complexity does not, potentially impacting its practical utility in high-dimensional scenarios. Additionally, KQT-EWMA relies on the sample covariance matrix, whose estimation can be poor in high-dimensional scenarios where the training set TR is not sufficiently large. Nevertheless, our experiments show that KQT-EWMA achieves excellent performance compared to the other methods designed for online monitoring, including QT-EWMA, while effectively controlling the false alarms, especially when considering complex data distributions such as multi-modal Gaussians or real-world datasets.

In future work, we aim to address the limitations of KQT-EWMA with high-dimensional datasets. Specifically, we plan to design kernels that do not rely on covariance matrix computation and are specifically tailored for the *sequential* scenario, where high-throughput is crucial.

References

1. Alippi, C., Boracchi, G., Carrera, D., Roveri, M.: Change detection in multivariate datastreams: Likelihood and detectability loss. International Joint Conference on Artificial Intelligence (IJCAI) **2**, 1368–1374 (2016)
2. Boracchi, G., Carrera, D., Cervellera, C., Macciò, D.: QuantTree: Histograms for change detection in multivariate data streams. In: Proceedings of the 35th International Conference on Machine Learning. vol. 80, pp. 639–648. PMLR (2018)
3. Carrera, D., Boracchi, G.: Generating high-dimensional datastreams for change detection. Big Data Research **11**, 11–21 (2018)
4. Frittoli, L., Carrera, D., Boracchi, G.: Change detection in multivariate datastreams controlling false alarms. In: Machine Learning and Knowledge Discovery in Databases. Research Track. pp. 421–436. Springer (2021)
5. Frittoli, L., Carrera, D., Boracchi, G.: Nonparametric and online change detection in multivariate datastreams using QuantTree. IEEE Transactions on Knowledge and Data Engineering **25**(8), 8328–8342 (2022)
6. Kelly, M., Longjohn, R., Nottingham, K.: The uci machine learning repository, https://archive.ics.uci.edu
7. Keriven, N., Garreau, D., Poli, I.: Newma: A new method for scalable model-free online change-point detection. IEEE Transactions on Signal Processing **68**, 3515–3528 (2020). https://doi.org/10.1109/TSP.2020.2990597
8. Kuncheva, L.I.: Change detection in streaming multivariate data using likelihood detectors. IEEE Transactions on Knowledge and Data Engineering **25**(5), 1175–1180 (2013)
9. Li, S., Xie, Y., Dai, H., Song, L.: Scan B-statistic for kernel change-point detection. Sequential Analysis **38**(4), 503–544 (2019)
10. Ross, G.J., Tasoulis, D.K., Adams, N.M.: Nonparametric monitoring of data streams for changes in location and scale. Technometrics **53**(4), 379–389 (2011)
11. Souza, V.M.A., dos Reis, D.M., Maletzke, A.G., Batista, G.E.A.P.A.: Challenges in benchmarking stream learning algorithms with real-world data. Data Mining and Knowledge Discovery **34**, 1805 – 1858 (2020)
12. Stucchi, D., Rizzo, P., Folloni, N., Boracchi, G.: Kernel QuantTree. In: Proceedings of the 40th International Conference on Machine Learning (2023)
13. Vershynin, R.: How close is the sample covariance matrix to the actual covariance matrix? Journal of Theoretical Probability **25** (04 2010). https://doi.org/10.1007/s10959-010-0338-z
14. Wei, S., Xie, Y.: Online kernel cusum for change-point detection (11 2022). https://doi.org/10.48550/arXiv.2211.15070
15. Zamba, K.D., Hawkins, D.M.: A multivariate change-point model for statistical process control. Technometrics **48**(4), 539–549 (2006)