Concept Drift Detection in High-Dimensional Spaces

Michelangelo Olmo Nogara Notarianni

November 29, 2023

1 Introduction

In predictive modeling, we typically assume that the relationship between input and output data remains static. However, in real-world scenarios, data patterns change over time, rendering predictive models obsolete. Concept Drift Detection addresses this issue by identifying these changes, known as concept drifts. Concept drifts represent unforeseen alterations in the statistical properties of the target variable, making it crucial to continuously monitor inference data and detect departures from conditions seen in training data.

One notable challenge is dealing with high-dimensional multivariate data streams. Such data streams are prevalent in fields like IoT and text analysis. Due to the curse of dimensionality, it's challenging to determine what constitutes a meaningful change and where it occurs within an huge feature space; feature reduction techniques, such as Principal Component Analysis (PCA), can be employed to map high-dimensional data to lower-dimensional representations while preserving distance relationships; however, not only they may have important computational costs, but also might ruin the behavior of concept drift detection algorithms, thus should be chosen carefully. One important property that might be lost after dimensionality reduction (DR) is the control of the False Positive Rate (FPR). Ensuring that alarms are only triggered when there are genuine deviations, and having a clear understanding of the expected number of false alarms for appropriately sizing the monitoring system, is crucial; particularly in applications like manufacturing quality control and healthcare diagnostics, where a high FPR can lead to unnecessary resource allocation and costs, if not to the erroneous prescription of unnecessary medications and their associated side effects.

Our greatest challenge was to study the performances of QuantTree (QT) algorithm in high dimensional spaces with only a small number of training points provided. More specifically, we investigate different dimensionality reduction techniques, and extend the study to QT's generalized version, Kernel-QT (KQT), and its online version, QT-EWMA. Finally, we propose an online version of KQT algorithm (KQT-EWMA).

Our main contributions include:

1. Development of a benchmark designed for studying concept drift detection, specifically addressing challenges associated with high dimensionality and limited availability of training data, thus the use of conventional data processing methods such as PCA.

2. Utilization of distances derived from l_p quasi-norms, where $p \leq 1$, to adapt KQT to sparse environments.

3. Proposal of an online version of KQT, KQT-EWMA. We demonstrate that its theoretical properties are upheld and, under various conditions, it outperforms the current state-of-the-art methods.

2 Problem Formulation

A change detection algorithm has usually three main ingredients: a model $\hat{\phi}_0$ of the initial distribution, a statistic based on it, and a decision rule to report changes. We make the assumption that both ϕ_0 and $\phi_1 \neq \phi_0$ are unknown. To estimate ϕ_0, a training set TR, which consists of N stationary realizations from ϕ_0, is provided. In our case, we employ a QuantTree-like histogram $\hat{\phi}_0$ fitted on the distribution ϕ_0 given TR. A histogram is defined as:

$$h = \{(S_k, \hat{\pi}_k)\}_{k=1,\dots,K}$$

where the K subsets S_k form a partition of $\mathbb{R}^d$, i.e. $\bigcup_{k=1}^{K} S_k = \mathbb{R}^d$ and $S_j \cap S_i = \emptyset$ for $j \neq i$, and each $\hat{\pi}_k \in [0, 1]$ corresponds to the probability for data generated from ϕ_0 to fall inside S_k. Online and batch-wise are two modes for drift detection. In batch-wise drift detection (also referred to as two-sample test), the idea is to infer whether two sample sets have been selected from the same population. A batch represent a discrete chunk of data collected over a specific time interval or

event. In contrast, online drift detection continuously monitors data - sometimes, real-time - and some model is updated as each new point/vector arrives. The choices between these approaches depends on the nature of the data and the specific requirements of the application.

To evaluate the performance of concept drift detection algorithms, online or offline, both quantitative and qualitative criteria need to be considered. Quantitative criteria include metrics such as TPR and FPR, but also the execution time, and the memory usage. Qualitative criteria, such as robustness to roto-translations of the dataframe and to the presence of outliers, also play a significant role in evaluating algorithm performance.

2.1 Batch-wise monitoring

We process the incoming data in batches $W = \{x_1, ..., x_\nu\}$, where ν represents the number of samples in each batch. Our goal is to detect changes using a hypothesis test (HT) that assesses whether the data in W aligns with the reference histogram h learned from the training set TR. We formulate the hypothesis test HT as:

$$H_0 : W \sim \phi_0 \quad \text{vs} \quad H_1 : W \sim \phi_1 \neq \phi_0 \tag{1}$$

where H_0 represents the null hypothesis: "W follows the distribution ϕ_0"; and H_1 is the alternative hypothesis: "W follows a different distribution $\phi_1 \neq \phi_0$". These tests are based on a test statistic $\mathcal{T}_h$ defined over the histogram h. Thus, $\mathcal{T}_h$ solely depends on $y_{k(k=1,...,K)}$, where y_k denotes the number of samples in W falling in S_k. We detect a change in the incoming W when:

$$\mathcal{T}_h(W) = \mathcal{T}_h(y_1, .., y_k) > \tau, \tag{2}$$

where τ is the threshold that controls the FPR, namely the proportion of type I errors. For each given test statistic $\mathcal{T}_h$ and reference FPR value α, we define a threshold τ such that:

$$P_{\phi_0}(\mathcal{T}_h(W) > \tau) \leq \alpha, \tag{3}$$

where α is the reference FPR value, when P_{ϕ_0} denotes the probability under H_0 that W contains samples generated from ϕ_0.

2.2 Online monitoring

We consider a virtually unlimited multivariate datastream $x_1, x_2, \ldots$ in $\mathbb{R}^d$. We assume that, in the absence of changes, all the data samples are i.i.d. realizations of a random variable with an unknown distribution ϕ_0, which support is $\mathcal{X} \subseteq \mathbb{R}^d$. We define the changepoint τ as the unknown time instant when a change $\phi_0 \to \phi_1 \neq \phi_0$ takes place. The data x_t follows the distribution:

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

Here, x_t represents the random variable that follows the distribution ϕ_0 before the changepoint τ (in the so called *in control state*), and then follows the distribution ϕ_1 for t greater than or equal to τ (*out of control state*).

Ideally, the target *Average Run Length* ARL_0, i.e. the average number of samples arrived from the stationary distribution before a false alarm is given, is set a priori, as for the type I error probability in hypothesis testing. The goal is to detect a distribution change as soon as possible, minimizing the detection delay $t^* - \tau$, while controlling ARL_0 by the means of a target value established before monitoring. It is worth noting that controlling ARL_0 also provides an upper bound on the expected detection delay.

3 Related Work

3.1 QuantTree

The QuantTree (QT) algorithm was first proposed in 2018 [1] to handle concept drift detection in multivariate dataframes. We refer to the algorithms presented in the paper. QT is a recursive binary splitting scheme designed to dynamically adapt histogram bins for effective change detection. Its greatest advantage is that the distribution of any statistic defined over the resulting histogram does not depend on ϕ_0, i.e. that decision rules to be used do not depend on the data and can be numerically computed from synthetically generated univariate sequences, even in multivariate change detection problems. The fact that QT can have a pre-assigned number of bins and can be represented as a tree, enables a very efficient computation of test statistics.

Histogram computation: We denote by $\mathcal{X}_k \subseteq \mathcal{X}$ the subset of the input space that still has to be partitioned. The subset S_k is then defined by splitting $\mathcal{X}_k$ along a component $i \in \{1, \ldots, d\}$ that is randomly chosen with uniform probability. S_k contains L_k points among the N in $\mathcal{X}$, thus the estimated probability of S_k is $\widehat{\pi}_k = L_k/N$. This procedure is iterated until K subsets are extracted. QuantTree divides $\mathcal{X}$ in a given number of subsets, where each S_k has an estimated probability $\hat{\pi}_k \approx \pi_k$ (equality holds when $\pi_k N$ is integer). Since the probabilities π_k are set a priori, in what follows we use π_k in place of $\hat{\pi}_k$. Indexes i and parameter γ are randomly chosen to add variability to the histogram construction.

3.2 QT-EWMA

QT-EWMA algorithm was introduced in [2] together with the procedure to define its thresholds controlling the ARL_0. It leverages an online statistic T_t defined over a QuantTree histogram, which monitors the proportion of samples in the datastream that fall in each bin S_j. We evaluate the EWMA statistic $Z_{j,t}$, $j \in \{1, ..., K\}$, to monitor the proportion of data that falls in each bin S_j:

$$Z_{j,t} = (1-\lambda)Z_{j,t-1} + \lambda y_{j,t} \quad , \quad Z_{j,0} = \hat{\pi}_j \quad (4)$$

Since, under ϕ_0, the expected value $\mathbb{E}[Z_{j,t}] \approx \hat{\pi}_j$ for $j = 1, ..., K$, we define the QT-EWMA change-detection statistic as follows:

$$T_t = \sum_{j=1}^{K} \frac{(Z_{j,t} - \hat{\pi}_j)^2}{\hat{\pi}_j} \quad (5)$$

The statistic is computed at each incoming sample and then compared against the corresponding threshold h_t to detect changes. QT-EWMA algorithm inherits from QuantTree the fundamental property that the distribution of the statistics (4) and (5) - like any other statistic entirely defined over QuantTree bins - does not depend on ϕ_0, so the thresholds $\{h_t\}_t$ can be defined a priori to guarantee the ARL_0 on any datastream. The sequence of thresholds has to be properly defined to guarantee the given $ARL_0 = \mathbb{E}[t^*]$, where the expected value is computed assuming that the whole datastream is drawn from ϕ_0.

3.3 Kernel QuantTree

A fundamental limitation of QT is that splits are defined along the axis, resulting in a partitioning that does not always adhere to the input distribution. A preprocessing stage is typically introduced to align the split directions to the principal components of the training set. While this solution is in practice beneficial, still many bins have non-finite volumes, which can lead to poor estimation of bin probabilities. In [3] is thus introduced Kernel QuantTree (KQT), a non-parametric and multivariate CD algorithm that partitions the space in $K-1$ compact bins defined by kernel functions evaluated on the training data. An additional "residual" bin is non-compact and gathers all the remaining points. The distribution of the test statistic $\mathcal{T}_h$ computed from a KQT histogram h does not depend on the stationary distribution ϕ_0 and the detection thresholds τ can be set a priori via Monte Carlo simulations. Moreover, the monitoring performed by KQT using specific kernel functions is not influenced by preprocessing based on roto-translations, including alignment to principal components. KQT was studied with the Euclidean, the Mahalanobis, and the Weighted Mahalanobis distances [3]. We examine the behavior of l_p norms, and propose as alternative kernel functions for KQT these distances derived from l_p (quasi-) norms with $p \leq 1$, knowing that fractional distances may be suitable to preserve the meaningfullness of proximity measures in high dimensional spaces. It is worth noting that these distances are not invariant to rotations (in Euclidean spaces).

4 KQT-EWMA

We propose Kernel-QuantTree Exponentially Weighted Moving Average (KQT-EWMA), a novel online nonparametric change-detection algorithm for multivariate datastreams. It combines a KQT histogram [3], used as a model $\hat{\phi}_0$, and a statistic T_t based on EWMA, which turns KQT into a truly sequential monitoring scheme, i.e. a testing process conducted in an adaptive manner, continuously evaluating incoming data and making decisions based on cumulative information rather than fixed sample sizes.

The theoretical properties of Kernel-QuantTree guarantee that KQT-EWMA is completely nonparametric since the distribution of our statistic does not depend on ϕ_0, hence its thresholds $\{h_t\}_t$ controlling the ARL_0 can be set a priori. Moreover, these thresholds guarantee by design a constant false alarm probability over time and, consequently, a fixed false alarm rate at any time instant during monitoring. Thus, KQT-EWMA controls both ARL_0 and false alarm (FA) rate. This property is also exploited to compute detection thresholds by Monte Carlo simulations, such that the empirical ARL_0 matches any target.

The statistic T_t monitors the proportion of samples in the datastream that fall in each bin S_j. In particular, for each x_t we define K binary statistics $\{y_{j,t}\}_j$ as the indicator functions of each bin S_j. We compute the EWMA statistic $Z_{j,t}$, $j \in \{1, ..., K\}$, to monitor the proportion of data that falls in each bin S_j. Since, under ϕ_0, the expected value $\mathbb{E}[Z_{j,t}] \approx \hat{\pi}_j$ for $j = 1, ..., K$, we define the KQT-EWMA change-detection statistic as it is in QT-EWMA (Eq. 5).

Theoretical results are based on the following theorem, derived from previous works ([2], [3]) and proved in the thesis work.

Theorem 1. *Let T_t be defined as in (5) over the histogram h computed by KQT. When $x_t \sim \phi_0$, the distribution of T_t does not depend on ϕ_0 nor on data dimension d.*

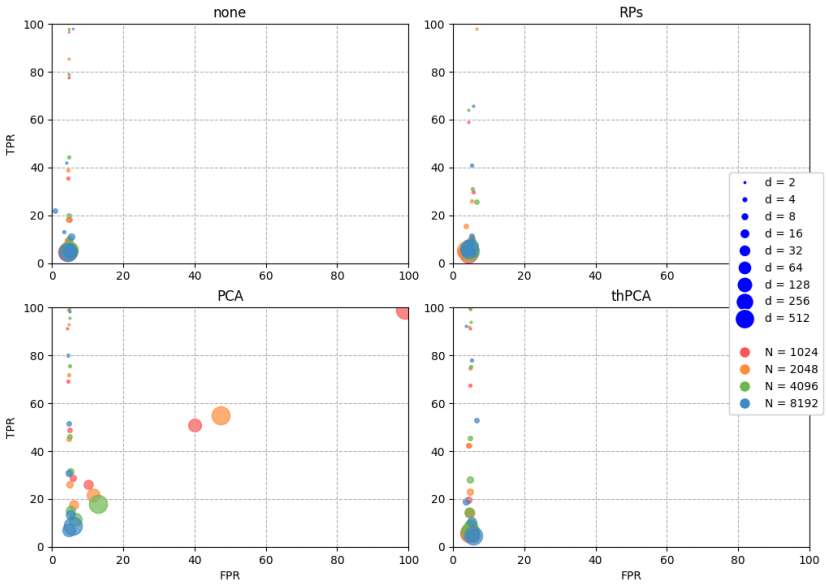

Figure 1: QT performances after projections with no DR. We scatter $x = FPR$ and $y = TPR$ given different combinations of dimension d and cardinality N, given by the radius and the color respectively. We aim to achieve vertical alignment of points at $x = 5\%$, signifying FPR control around the threshold we set. However, we notice (sample-based PCA) shifts through the bisector of the graph towards the upper-right corner, where $TPR = FPR = 1$; this phenomenon is reduced by an increased number of training points. Experiments were repeated 100 times over (512+512) test batches.

5 Experiments

5.1 QT in High Dimensional spaces

Obtaining real-world high dimensional datasets with fixed characteristics is challenging, thus experiments are typically conducted on either synthetically generated data or real-world datasets modified to introduce changes at known locations. Here, "Controlling Change Magnitude" framework (CCM) is used. It applies a rototranslation directly to the data to ensure precise control over change magnitudes measured by symmetric Kullback-Leibler (sKL). Datasets are mostly - always, for what concerns this summary - generated from a null-mean Gaussian distribution with a random covariance matrix before the CCM framework is applied. The sKL distance between pre- and post-change distribution is always fixed as the dimensionality of the space grows.

The study explores how dimensionality affects QT performances in terms of TPR and FPR, with and without a PCA-like preprocessing. In particular we focus on FPR control, which can be lost when the N/d ratio is small. Our first experiments were done with **PCA** and its randomized version - based on applying random projections before SVD decomposition - which requires less memory and allows a more computationally efficient DR from high-dimensional feature spaces. Dataframes are generated from monomodal Gaussians in $\mathbb{R}^d$.

Even if PCA preprocessed comes with no dimensionality reduction (DR), FPR control is lost when an insufficient number of training points is provided (see Fig. 1). *Is PCA genuinely failing, or are the limitations a byproduct of the (limited) training data?*. This is a central question in our study, and nearly the entirety of our experiments were dedicated to answer it. To know if the problem is given by the presence of outliers in too small training sets, we tried **RobustPCA** preprocessing, and a **theoretical version of PCA** which is not computed from TR but rather from the known distribution which data is sampled from - since the covariance matrix is known, we can derive the true principal components as its eigenvectors. Even though this approach cannot be used in practice, we adopt it to discern the root causes of PCA's issues with FPR control. Indeed, while a PCA rotation prevents QT to control the FPR, this covariance matrix-based projections increases TPR while not increasing FPR above the set threshold.

We tried several experiments using **convex combinations** of PCA matrices, namely: **1)** Combinations are represented by the equation $\lambda * M_{PCA} + (1-\lambda) * \mathbf{1}$, where $\lambda \in [0, 1]$, M_{PCA} represents the matrix transformation associated with PCA, and $\mathbf{1}$ is the identity matrix. **2)** Combinations are represented by the equation $\lambda * M_{PCA} + (1-\lambda) * M_{th}$, where M_{th} is the matrix associated with theoretical-PCA.

One hypothesis trying to explain the effects of PCA over the control of the FPR, even with no DR, revolves around the alignment of QT's bins with the PCs, especially in scenarios where data points collapse onto particular low-variance components. We also tried to compute the PCs, project the data onto them, and then apply **d-dimensional rotations**, generating stochastic rotation matrices around all axes in the space, intentionally introducing increasing misalignments

with the PCs before building the histogram. Regrettably, it seems that the rotation after PCA does not preserve the control of the FPR - indeed, there is no obvious trend of the results with the rotation angle.

Choosing components for DR: If we use PCA on our data sampled from d-dimensional distributions and keep a number $d' < d$ of principal components, what should be the choice? This is already discussed in literature: concept drifts may manifest in the "low variance components" which in stationary conditions we expect to exhibit low variance. However, sample-based computation might fail to "explain" a small amount of variance. The effectiveness of low-variance components in detecting concept drift is tightly bound to the robustness of their computation, emphasizing the importance of an adequately sized and representative training set.

Random projections (RPs) serve as a remarkably efficient and straightforward approach to DR. Leveraging the Johnson-Lindenstrauss lemma, RPs stand out for their speed and simplicity, aligning seamlessly with the demands of diverse algorithms. In the case of the QT algorithm, RPs enable effective control over the FPR, emphasizing the method's practicality and reliability in managing the intricacies of high-dimensional data while maintaining theoretical bounds.

In Fig. 1 we show the results of QT analysis of batches drawn from monomodal Gaussians with increasing dimension d, given N training points to build the histogram. We can see the effects of detectability loss as dots of greater radius are associated with lower TPR; still, QT achieves FPR control independently of d and N. This is also true after rotating data with random matrices (RPs) and with theoretical PCA matrices. When directions for the bins computation are learned from the training set through PCA, FPR control is lost and the scattered points shifts toward to the upper right corner of the graph, where $TPR = FPR = 100\%$. A greater N (green/blue dots) is sufficient to keep FPR below the threshold. In Fig. 1 data is preprocessed with no dimensionality reduction (DR). In Fig. 2, we illustrate the impact of DR while addressing the parallel problem in an online fashion.

5.2 Kernel QT and l_p norms as f_p

The Euclidean distances computation is $\mathcal{O}(d)$-complex, way less than for Mahalanobis (M) and Weighted Mahalanobis (WM) distances ($\mathcal{O}(d^2)$ and $\mathcal{O}(d^3)$ respectively). Between distances in [3], only the Euclidean provides FPR control when dimensionality increases. We propose to use (quasi-)distances derived from l_p norms, with $p \leq 0.1$, which are effective in preserving the meaningfullness of proximity measures in high-dimensional spaces. We remark that they are not invariant to (Euclidean) rotations.

We show the effects of reducing d with PCA in Fig. 2. Even with a sufficient number N of training points, KQT performances using l_p norms do not consistently surpass those of QT, which is more stable when keeping low variance components. Mahalanobis and Weighted Mahalanobis (´weighted') distances achieved higher TPR in small spaces but cannot in general control FPR when N/d is small, relying on the estimated covariance matrix of the distributions. Instead, l_p norms achieve FPR control when no DR is performed. QT here is the only model able to keep FPR stable and achieving a considerable detection power $\Delta(TPR - FPR)$ when projecting data on low variance components.

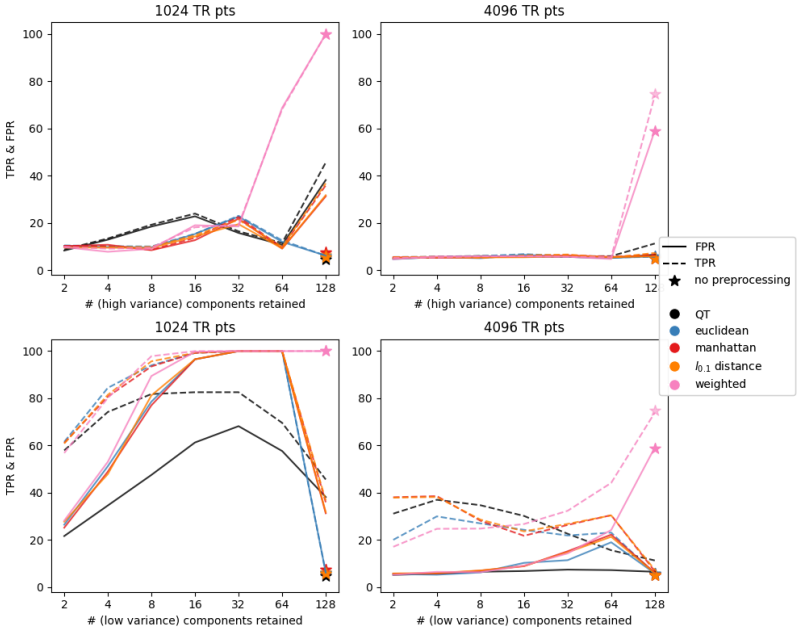

Figure 2: 128-dimensional dataset preprocessed with PCA and analyzed by QT and KQT with the three metrics $l_{0.1}$, l_1 and l_2 and Weighted Mahalanobis ('weighted') distance. We keep x dimensions, confronting high- (top) and low-variance (bottom) components. KQT achieves higher TPR in small spaces, but has no FPR control here. QT maintains stable FPR and achieves significant detection power $\Delta(TPR - FPR)$ when projecting data onto low-variance components if N/d is great enough. Experiments were repeated 100 times over (512+512) test batches.

5.3 Online: KQT-EWMA

Our aim is to show that the algorithm controls the false alarms comparably than competing methods, while achieving lower detection delays. We

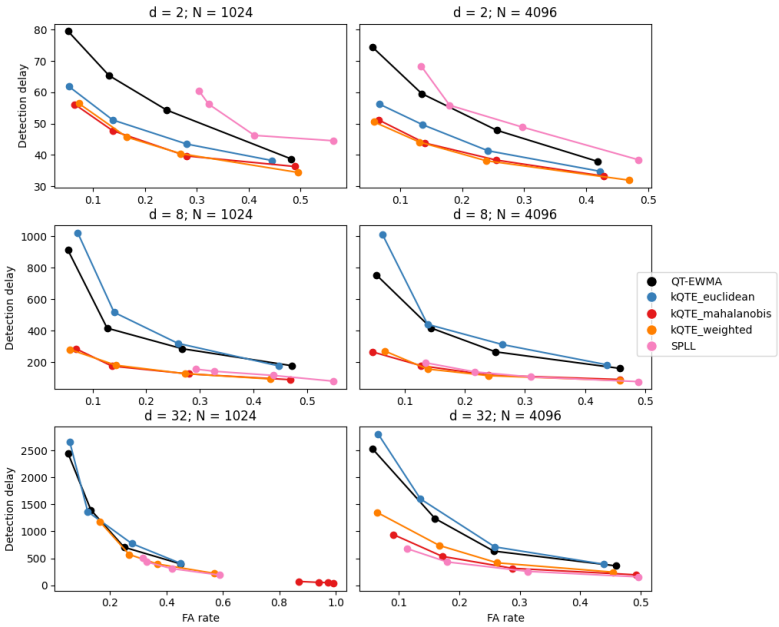

Figure 3: Experimental results over Gaussian datastreams with $d \in \{2, 8, 32\}$ and $N \in \{1024, 4096\}$ show that KQT-EWMA can outperforms the state of the art. While QT-EWMA and Euclidean KQT-EWMA can control the false alarm rates, this is not true with Mahalanobis (M) and Weighted Mahalanobis (WM) distances employed and dimensionality grows with N fixed. In small-dimensional frames, KQT-EWMA with M and WM distances achieves the lowest (best) detection delays, while controlling false alarms (SPLL delay is comparable here when $d = 8$, but FA rate is 2-3 times bigger. As dimensionality grows, only QT-EWMA and Euclidean KQT-EWMA can control the false alarms. Values were averaged over 1000 experiments.

compare KQT-EWMA implemented with different distances, QT-EWMA and an online version of SPLL, which relies on a Gaussian Mixture Model (GMM), together with the effect of PCA-like rotations. Again, we increase data dimensionality d controlling sKL, and decrease the number of training points N to study the algorithms' performance in when in trouble. While QT-EWMA and Euclidean KQT-EWMA can control the ARL_0 and SPLL cannot, KQT-EWMA with Mahalanobis and Weighted Mahalanobis distances loses this property with increasing dimensionality, N fixed (same discussion as for the FPR control batch-wise). When there is control over the false positives, i.e. with a sufficient number of training points, Mahalanobis and Weighted Mahalanobis KQT-EWMA achieves the lowest (best) detection delays. Some of the results are shown in Fig. 3.

6 Conclusions

Concept Drift Detection frameworks give a comprehensive view of the challenges and strategies inherent in monitoring predictive models under dynamic data conditions. The thesis work confronted the complexity of high-dimensional multivariate data streams, studying the performances of the QT algorithm, its generalized version Kernel-QT (KQT), and its online variant, QT-EWMA. This exploration finally included the proposal of a novel online algorithm, KQT-EWMA, which combines a generalized QT histogram with an exponentially weighted moving average statistic and outshines both QT-EWMA and the "oracle" SPLL, both in terms of controlling ARL_0 and in achieving impressively low detection delays when N/d ratio is large.

We considered the interplay between dimensionality, training data availability, and the choice of distance metrics, together with conventional data processing methods such as PCA; in particular we considered the significance of the choice of components for building the projected space, always keeping in mind the importance of FPR (or ARL_0, online) control.

The conclusions of this study show paths for future explorations and real-world applications: we tried to set a controlled but comprehensive framework for establishing the limits of these algorithms, but in doing so, we just discovered new questions, and the music is just starting.

References

[1] G. Boracchi, D. Carrera, C. Cervellera, and D. Macciò. QuantTree: Histograms for change detection in multivariate data streams. In J. Dy and A. Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 639–648. PMLR, 10–15 Jul 2018.

[2] L. Frittoli, D. Carrera, and G. Boracchi. Change detection in multivariate datastreams controlling false alarms. In N. Oliver, F. Pérez-Cruz, S. Kramer, J. Read, and J. A. Lozano, editors, *Machine Learning and Knowledge Discovery in Databases. Research Track*, pages 421–436, Cham, 2021. Springer International Publishing.

[3] D. Stucchi, P. Rizzo, N. Folloni, and G. Boracchi. Kernel quanttree. In *Proceedings of the 40th International Conference on Machine Learning*, 2023.