Online Change Point Detection with Kernels

André Ferrari, Cédric Richard, Anthony Bourrier, Ikram Bouchikhi

Online change-point detection with kernels

André Ferrari[a,*], Cédric Richard[a], Anthony Bourrier[b], Ikram Bouchikhi[a]

[a] *Université Côte d'Azur, Observatoire de la Côte d'Azur, CNRS, Lab. Lagrange, France*
[b] *Thales Alenia Space, Cannes la Bocca, France*

Abstract

Change-points in time series data are usually defined as the time instants at which changes in their properties occur. Detecting change-points is critical in a number of applications as diverse as detecting credit card and insurance frauds, or intrusions into networks. Recently the authors introduced an online kernel-based change-point detection method built upon direct estimation of the density ratio on consecutive time intervals. This paper further investigates this algorithm, making improvements and analyzing its behavior in the mean and mean square sense, in the absence and presence of a change point. These theoretical analyses are validated with Monte Carlo simulations. The detection performance of the algorithm is illustrated through experiments on real-world data and compared to state of the art methodologies.

Keywords: Non-parametric change-point detection, reproducing kernel Hilbert space, kernel least-mean-square algorithm, online algorithm, convergence analysis.

1. Introduction

From a statistical perspective, a change-point is defined as a time instant at which some properties of a signal change, that is, the observations belong to one state up to that point, and belong to an other state after it. This change can be caused by external events, as well as by sharp transitions in the dynamics of the signal, either way it can hold critical information. Among possible applications of change point detection (CPD) we can mention medical monitoring [1, 2, 3], finance [4] and network security [5, 6]. We refer interested readers to, e.g., [7] or [8] for comprehensive reviews of CPD algorithms.

CPD algorithms can be classified, based on what is assumed to be known about the data distribution, as *parametric* or *non-parametric*. *Parametric* approaches assume that a model describing the data distributions of the different

*Corresponding author
Email addresses: Andre.Ferrari@univ-cotedazur.fr (André Ferrari), Cedric.Richard@univ-cotedazur.fr (Cédric Richard), Anthony.Bourrier@thalesaleniaspace.com (Anthony Bourrier), Ikram.Bouchikhi@univ-cotedazur.fr (Ikram Bouchikhi)

states is available. For instance, cumulative sum (CUSUM) type algorithms [9] assume, in their simplest form, that the parameter that undergoes changes is known, but also require knowledge of its pre-change and sometimes post-change values, e.g., change in the mean or in the variance [10]. In case where the aforementioned parameters are unknown, the generalized likelihood ratio [11], which consists of substituting all the unknown parameters by their maximum likelihood estimates, can be used. Less restrictive approaches have also been devised. Among these, subspace identification techniques are built upon the idea that if, at a certain time instant, there is a change in the mechanism generating the time series, then the (linear) subspace spanned by the signal trajectory also changes. This principle is used in [12] where the authors explicitly model the observations via a discrete-time linear state-space system. Another example is the Singular Spectrum Transformation, which calculates distance-based change-point scores by comparing singular spectra of two trajectory matrices over consecutive windows [13, 14]. If all assumptions about the data model are met, these techniques can be robust and efficient. In practice though, stochastic models that properly describe the data are not often available. And, even when they are, data are susceptible to deviations from the assumed models. *Non-parametric* approaches were introduced to cope with these limitations. They can be used in a broader range of applications, since they do not require (strong) prior information.

Non-parametric algorithms are usually classified as *supervised* or *unsupervised* methods. In the first case, when the number of possible states is specified, and labeled data representing each state is available, machine learning algorithms can be used to train multi-class classifiers and then find each state boundary. If not, the nominal-state sequences represent the unique class and the problem can be solved using, e.g., a one-class algorithm such as [15] and [16] or alternative approaches such as the Hilbert-Schmidt Independence Criterion in [17]. However, in many practical situations, labeled data is not available and *unsupervised* algorithms that can adapt to different situations are required. This problem can be tackled by extending subspace identification techniques to non-linear subspaces using, e.g., nearest neighbors algorithms or, more generally, manifold learning methods, [18, 19]. An alternative approach consists of operating in a Reproducing Kernel Hilbert Space (RKHS) in order to extend the use of linear models and algorithms to nonlinear problems, [20]. This strategy is used in [21] which proposes an online implementation of the Maximum Mean Discrepancy (MMD) two-sample test based on the B-statistics [21]. Note that [22], where change-points are detected using a Kernel Fisher Discriminant, and [23], where the authors propose to monitor the mean of the process in the feature space, are both strongly related to MMD. Another class of *unsupervised* methods is based on the direct estimation of the ratio of probability densities of the data over consecutive segments. They include the Kullback-Leibler Importance Estimation Procedure (KLIEP), the Unconstrained Least Squares Importance Fitting (uLSIF) and the Relative Unconstrained Least Squares Importance Fitting (RuLSIF) [24]. The main contributions of this article lie in this class of methods.

Recently, an online version of a RuLSIF-based CPD algorithm, which con-

sists of estimating the density ratio over consecutive intervals of the time series data, was introduced [25]. In this algorithm, the model parameters are estimated in an online and adaptive way similar to the Kernel Least Mean Squares (KLMS) algorithm [26]. The methodology showed promising and reliable detection results. In [27] the authors proposed to modify the original cost function used in [25] in order to further improve the performance and achieve unbiasedness of the algorithm, referred to as NOUGAT (Nonparametric Online chanGepoint detection AlgoriThm).

The main contribution of this paper are as follows. After introducing the proposed algorithm (Section 2), we provide a theoretical analysis of its stochastic behavior by deriving models for the mean and the variance of the detection statistics, in the absence and the presence of a change-point (Section 3). These models are useful for several purposes: i) to assess the detection performance of the algorithm; ii) for detector design and optimization. Then we demonstrate the accuracy of these models, and we present performance comparisons with state-of-the-art algorithms on simulated and real data sets (Section 4). Finally we conclude this paper with recommendations for future research (Section 5).

2. The NOUGAT algorithm

In this section, we formulate the CPD problem. We review the proposed method and the online algorithm denoted as NOUGAT. Then we introduce the detection statistic. Finally we briefly discuss related works.

2.1. Problem formulation

We aim at detecting change-points in the distribution of independent random variables $\{\boldsymbol{y}_t\}_{t\in\mathbb{N}}$, $\boldsymbol{y}_t \in \mathbb{R}^k$, by estimating a model $g(\cdot)$ for $r(\boldsymbol{y})-1$, where $r(\boldsymbol{y}) = p_{\text{test}}(\boldsymbol{y})/p_{\text{ref}}(\boldsymbol{y})$ is the density ratio between the probability density $p_{\text{test}}(\boldsymbol{y})$ of the data on a test interval:

$$\boldsymbol{Y}_t^{\text{test}} = (\boldsymbol{y}_{t-(N_{\text{test}}-1)}, \ldots, \boldsymbol{y}_{t-1}, \boldsymbol{y}_t) \in \mathbb{R}^{k\times N_{\text{test}}} \tag{1}$$

and the probability density $p_{\text{ref}}(\boldsymbol{y})$ of the data on a reference interval:

$$\boldsymbol{Y}_t^{\text{ref}} = (\boldsymbol{y}_{t-(N_{\text{ref}}+N_{\text{test}}-1)}, , \ldots, \boldsymbol{y}_{t-N_{\text{test}}}) \in \mathbb{R}^{k\times N_{\text{ref}}} \tag{2}$$

where N_{test} and N_{ref} are the number of samples in the test and reference intervals, respectively. Note that, contrary to RuLSIF [24], $r(\boldsymbol{y}) - 1$ is preferred to $r(\boldsymbol{y})$ because it leads to an unbiased estimator under the no change-point hypothesis as we shall see later.

In the general case of a scalar time series $\{y_t\}_{t\in\mathbb{N}}$, as commonly reported in the literature, we propose to proceed by considering

$$\boldsymbol{y}_t = (y_t, y_{t+1}, \ldots, y_{t+k-1})^\top \in \mathbb{R}^k \tag{3}$$

to take into account any dependence that may exist between successive y_t.

2.2. Density-ratio estimation

The problem addressed in this paper consists of estimating a model $g(\cdot)$ for $r(\boldsymbol{y})-1$. It can be solved by fitting $g(\boldsymbol{y})$ to $r(\boldsymbol{y})-1$ with respect to the squared loss:

$$\mathcal{C}(g) = \frac{1}{2}\mathsf{E}_{p_{\text{ref}}(\boldsymbol{y})}\{(r(\boldsymbol{y}) - 1 - g(\boldsymbol{y}))^2\} \tag{4}$$

Note that, as in [24], the expectation operator is defined with respect to the reference interval. By expanding (4) and then using $r(\boldsymbol{y})p_{\text{ref}}(\boldsymbol{y}) = p_{\text{test}}(\boldsymbol{y})$, we obtain:

$$\mathcal{C}(g) = \frac{1}{2}\mathsf{E}_{p_{\text{ref}}(\boldsymbol{y})}\{g^2(\boldsymbol{y})\} - \mathsf{E}_{p_{\text{test}}(\boldsymbol{y})}\{g(\boldsymbol{y})\} + \mathsf{E}_{p_{\text{test}}(\boldsymbol{y})}\{g(\boldsymbol{y})\} + C \tag{5}$$

where C denotes a constant value. Approximating the expected values in (5) by their empirical averages over the reference and test intervals data $\boldsymbol{Y}_t^{\text{ref}}$ and $\boldsymbol{Y}_t^{\text{test}}$ for any fixed t, leads to the following empirical optimization problem:

$$\begin{aligned}\min_{g\in\mathcal{H}} \Bigg(& \frac{1}{2N_{\text{ref}}} \sum_{i=t-(N_{\text{ref}}+N_{\text{test}}-1)}^{t-N_{\text{test}}} g^2(\boldsymbol{y}_i) - \frac{1}{N_{\text{test}}} \sum_{i=t-(N_{\text{test}}-1)}^{t} g(\boldsymbol{y}_i) \\ & + \frac{1}{N_{\text{ref}}} \sum_{i=t-(N_{\text{ref}}+N_{\text{test}}-1)}^{t-N_{\text{test}}} g(\boldsymbol{y}_i) + \nu\,\Omega(\|g\|_{\mathcal{H}}) \Bigg)\end{aligned} \tag{6}$$

where $\mathcal{H}$ denotes an arbitrary reproducing kernel Hilbert space of real-valued functions on $\mathbb{R}$. Let $\kappa(\cdot\,,\cdot)$ be the reproducing kernel of $\mathcal{H}$. The term $\nu\,\Omega(\|g\|_{\mathcal{H}})$ with $\nu \geq 0$ is a regularization term added to promote smoothness of the solution. By virtue of the Representer Theorem [28], any function $g(\cdot)$ of $\mathcal{H}$ that minimizes (6) can be expressed as a kernel expansion in terms of available data:

$$g(\cdot, \boldsymbol{\theta}) = \sum_{i=t-(N_{\text{ref}}+N_{\text{test}}-1)}^{t} \theta_i\ \kappa(\cdot\,, \boldsymbol{y}_i) \tag{7}$$

where the θ_i are parameters to be learned. This model cannot be trained efficiently in an online framework, as it needs to update both $\{\boldsymbol{y}_i\}$ and $\boldsymbol{\theta}$ as time t progresses. A standard strategy in the literature is to substitute $\{\boldsymbol{y}_i\}$ in (7) by a fixed dictionary of size L, $\{\boldsymbol{y}_{\omega_i}\}_{i=1}^{L}$, whose elements are chosen according to some sparsification rule [29] to represent the input data space accurately, resulting in a fixed order model,

$$g(\cdot, \boldsymbol{\theta}) = \sum_{i=1}^{L} \theta_i \kappa_{\omega_i}(\cdot) \tag{8}$$

where $\kappa_{\omega_i}(\cdot) = \kappa(\cdot, \boldsymbol{y}_{\omega_i})$, for all $i \in \{1, \ldots, L\}$, are the elements of the dictionary, and $\boldsymbol{\kappa_\omega}(\cdot) = [\kappa_{\omega_1}(\cdot), \ldots, \kappa_{\omega_L}(\cdot)]^\top$.

Substituting (8) into (6), assuming a ridge parameter space regularization [28], and minimizing (6) w.r.t. $\boldsymbol{\theta}$, we find that the optimal parameter vector $\hat{\boldsymbol{\theta}}_t$ is the solution of the following strictly convex quadratic optimization problem:

$$\begin{aligned} &\hat{\boldsymbol{\theta}}_t = \arg\min_{\boldsymbol{\theta}\in\mathbb{R}^L} J_t(\boldsymbol{\theta}) \\ &\text{with} \quad J_t(\boldsymbol{\theta}) = \frac{1}{2}\boldsymbol{\theta}^\top \boldsymbol{H}_t^{\text{ref}}\boldsymbol{\theta} + \boldsymbol{\theta}^\top \boldsymbol{e}_t^\circ + \frac{\nu}{2}\|\boldsymbol{\theta}\|^2 \end{aligned} \tag{9}$$

where

$$\boldsymbol{e}_t^\circ = \boldsymbol{h}_t^{\text{ref}} - \boldsymbol{h}_t^{\text{test}} \tag{10}$$

and

$$\boldsymbol{h}_t^{\text{test}} = \frac{1}{N_{\text{test}}} \sum_{i=t-(N_{\text{test}}-1)}^{t} \boldsymbol{\kappa}_{\boldsymbol{\omega}}(\boldsymbol{y}_i) \tag{11}$$

$$\boldsymbol{h}_t^{\text{ref}} = \frac{1}{N_{\text{ref}}} \sum_{i=t-(N_{\text{test}}+N_{\text{ref}}-1)}^{t-N_{\text{test}}} \boldsymbol{\kappa}_{\boldsymbol{\omega}}(\boldsymbol{y}_i) \tag{12}$$

$$\boldsymbol{H}_t^{\text{ref}} = \frac{1}{N_{\text{ref}}} \sum_{i=t-(N_{\text{test}}+N_{\text{ref}}-1)}^{t-N_{\text{test}}} \boldsymbol{\kappa}_{\boldsymbol{\omega}}(\boldsymbol{y}_i)\boldsymbol{\kappa}_{\boldsymbol{\omega}}^\top(\boldsymbol{y}_i) \tag{13}$$

2.3. Online density-ratio estimation

Let $\boldsymbol{\theta}_t$ be an estimate of the parameter vector of the density ratio model at time instant t. When $t \to t+1$, according to (9), $\boldsymbol{\theta}_{t+1}$ should be computed, as proposed in RuLSIF [24], by updating first (11)–(13) and then minimizing the updated criterion $J_{t+1}(\boldsymbol{\theta})$. In order to reduce the computational cost, we propose as an alternative strategy to compute $\boldsymbol{\theta}_{t+1}$ by updating $\boldsymbol{\theta}_t$ based on a gradient descent step of $J_{t+1}(\boldsymbol{\theta})$:

$$\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t - \mu\nabla J_{t+1}(\boldsymbol{\theta}_t) \tag{14}$$

$$= \boldsymbol{\theta}_t - \mu\big[(\boldsymbol{H}_{t+1}^{\text{ref}} + \nu\boldsymbol{I})\boldsymbol{\theta}_t + \boldsymbol{e}_{t+1}^\circ\big] \tag{15}$$

where $\mu > 0$ is a small step size, and $\nabla J_{t+1}(\boldsymbol{\theta}_t)$ denotes the gradient of $J_{t+1}(\cdot)$ evaluated at $\boldsymbol{\theta}_t$. The resulting algorithm shares similarities with the KLMS algorithm [30]. The convergence behavior of the KLMS was analyzed in the case of a fixed dictionary in [31], and in a more general case in [32]. Additional constraints such as sparsity have been also considered [33].

In practice, updating model $g(\cdot, \boldsymbol{\theta}_t)$ at each time instant t is a two-stage process that consists of updating both the dictionary $\{\boldsymbol{y}_{\omega_i}\}_{i=1}^L$ and the order L of the kernel expansion (8), followed by the update of parameter vector $\boldsymbol{\theta}_t$.

2.4. Dictionary update

Numerous strategies of dictionary learning have been introduced in the online kernel filtering literature. They consist of building the dictionary $\{\boldsymbol{y}_{\omega_i}\}_{i=1}^L$

sequentially, by inserting selected samples $\boldsymbol{y}_i$ that improve the representation of input data according to some criterion. For instance, the Approximate Linear Dependency (ALD) [34] criterion checks whether, in feature space $\mathcal{H}$, the new candidate element $\kappa(\cdot\,, \boldsymbol{y}_{t+1})$ can be well approximated by a linear combination of the elements $\kappa(\cdot\,, \boldsymbol{y}_{\omega_i})$ which are already in the dictionary. If not, it is added to the dictionary. The coherence rule [30] was introduced to avoid the computational complexity inherent to ALD. It is now considered as a state-of-the-art strategy and widely used as such. Defined by:

$$\eta = \max_{i \neq j} |\kappa(\boldsymbol{y}_{\omega_i}, \boldsymbol{y}_{\omega_j})|,$$

coherence η reflects the largest correlation between the dictionary elements. The coherence rule for kernel-based dictionary selection consists of inserting $\boldsymbol{y}_{t+1}$ in dictionary $\{\boldsymbol{y}_{\boldsymbol{\omega}_i}\}_{i=1}^{L}$ provided that its coherence remains below a threshold η_0 preset by the user:

$$\max_{\boldsymbol{y}_{\omega_i} \in \{\boldsymbol{y}_{\boldsymbol{\omega}_i}\}_{i=1}^{L}} |\kappa(\boldsymbol{y}_{t+1}, \boldsymbol{y}_{\omega_i})| \leq \eta_0 \tag{16}$$

In [30] the authors show that the dimension of dictionaries determined with rule (16) is finite due to the compactness of the input space.

2.5. NOUGAT Algorithm

Depending on whether the new sample $\boldsymbol{y}_{t+1}$ has been inserted into the dictionary, or not, parameter vector $\boldsymbol{\theta}_t$ is updated similarly to [30]. At each time instant t, given $\boldsymbol{\theta}_t$, we propose as a test statistic to consider the average of the (shifted by 1) density ratio estimators over the test interval, namely:

$$g_t = \frac{1}{N_{\text{test}}} \sum_{i=t-(N_{\text{test}}-1)}^{t} g(\boldsymbol{y}_i, \boldsymbol{\theta}_t) = \boldsymbol{\theta}_t^{\top} \boldsymbol{h}_t^{\text{test}} \tag{17}$$

CPD is then performed by comparing $g_t + 1$ to a given threshold ξ. The corresponding NOUGAT algorithm is described in Alg. 1.

2.6. Related works

Iteration (15) turns out to be related to the classical Geometric Moving Average algorithm (GMA) proposed in [35]. GMA monitors a geometrically weighted estimate of the mean of $\boldsymbol{y}_t$ and detects a change when the estimated mean deviates from its nominal value. Without loss of generality, the mean in the observation space can be replaced by the mean $\mathsf{E}\{\boldsymbol{\kappa}_{\boldsymbol{\omega}}(\boldsymbol{y})\}$ in the feature space defined by mapping $\boldsymbol{\kappa}_{\boldsymbol{\omega}}(\cdot)$, leading to:

$$\boldsymbol{\vartheta}_{t+1} = (1 - \alpha)\boldsymbol{\vartheta}_t + \alpha \boldsymbol{\kappa}_{\boldsymbol{\omega}}(\boldsymbol{y}_{t+1}) \tag{18}$$

However, as pointed out in [23], a drawback of GMA is that it requires to know the nominal value of $\mathsf{E}\{\boldsymbol{\kappa}_{\boldsymbol{\omega}}(\boldsymbol{y})\}$ in order to be able to calculate the associated test statistic: $\|\boldsymbol{\vartheta}_t - \mathsf{E}\{\boldsymbol{\kappa}_{\boldsymbol{\omega}}(\boldsymbol{y})\}\|_2$.

Algorithm 1: NOUGAT Algorithm

Step size μ, initial dictionary $\boldsymbol{\omega}$, regularization ν, thresholds η_0 and ξ
for $t = 1, 2, \ldots$ **do**
  update $\boldsymbol{H}_t^{\text{ref}}$, $\boldsymbol{h}_t^{\text{test}}$ and $\boldsymbol{e}_t^{\circ}$ using (10)-(13)
  if $\max_{\boldsymbol{y}_{\omega_i} \in \{\boldsymbol{y}_{\omega_i}\}_{i=1}^{L}} |\kappa(\boldsymbol{y}_{t+1}, \boldsymbol{y}_{\omega_i})| > \eta_0$ **then**
    # the dictionary remains unchanged and $\boldsymbol{\theta}_t$ is updated using (15)

$$\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t - \mu\big[(\boldsymbol{H}_{t+1}^{\text{ref}} + \nu \boldsymbol{I})\boldsymbol{\theta}_t + \boldsymbol{e}_{t+1}^{\circ}\big]$$

  else
    # $\boldsymbol{y}_{t+1}$ is added to the dictionary and $\boldsymbol{\theta}_t$ is updated
    $L \leftarrow L + 1$, $\omega_{L+1} = t + 1$

$$\boldsymbol{\theta}_{t+1} = \begin{pmatrix} \boldsymbol{\theta}_t \\ 0 \end{pmatrix} - \mu \left[(\boldsymbol{H}_{t+1}^{\text{ref}} + \nu \boldsymbol{I}) \begin{pmatrix} \boldsymbol{\theta}_t \\ 0 \end{pmatrix} + \boldsymbol{e}_{t+1}^{\circ} \right]$$

  end if
  # compute the test statistic and test
  $g_{t+1} = \boldsymbol{\theta}_{t+1}^{\top} \boldsymbol{h}_{t+1}^{\text{test}}$
  if $|g_{t+1} + 1| > \xi$ **then**
    flag $t + 1$ as a change point
  end if
end for

To solve this problems in the GMA framework, a natural approach consists of comparing the estimates of $\mathsf{E}\{\boldsymbol{\kappa}_{\boldsymbol{\omega}}(\boldsymbol{y}_t)\}$ on two sliding windows, namely, the reference interval (2) and the test interval (1), as proposed in the Moving Average (MA) algorithm described in [23] which tracks:

$$\|\boldsymbol{e}_t^{\circ}\|_2 = \|\boldsymbol{h}_t^{\text{test}} - \boldsymbol{h}_t^{\text{ref}}\|_2 \tag{19}$$

The approach implemented by NOUGAT differs in so far as, instead of calculating a deviation between two quantities estimated over the test and reference intervals, it estimates a unique statistic $r(\boldsymbol{y})$ over the two intervals which is inherently equal to 1 under the null hypothesis. Note that an alternative approach, called NEWMA, proposed recently in [23], consists of testing the deviation between two GMA with different forgetting factors. The GMA with the smallest forgetting factor is used to provide an estimation of the in-control quantity. Contrarily to NOUGAT these algorithms do not explicitly take into account the covariance of the data in the feature space. From this point of view NOUGAT is similar to the Kernel Fisher Discriminant Ratio [22] but with a much lower computation footprint since it does not require the inversion of a covariance matrix for each t. Concerning the memory resources, NOUGAT requires to buffer $N_{\text{ref}} + N_{\text{test}}$ data points. The B-statistics CPD algorithm [21] also relies on a single sliding test window, but, conversely, on multiple reference windows which considerably increases the memory requirement.

In Section 4.2 we shall compare the detection performance and computational load of the three algorithms mentioned above with the same memory footprint, namely, MA, NOUGAT and RuLSIF.

3. Theoretical analysis

In this section we analyze the stochastic behavior of the proposed algorithm, and derive conditions for its stability in the mean and mean square sense, in the absence and presence of a change-point. To make the analysis tractable, we shall conduct it in the case of a *pre-tuned* dictionary, i.e., a fixed dictionary of size L is assumed to be available beforehand. This means that L is fixed and the $\{\boldsymbol{y}_{\omega_i}\}_{i=1}^{L}$ are assumed to be deterministic. The classical Modified Independence Assumption (MIA) [36], which assumes that $\boldsymbol{H}_{t+1}^{\text{ref}}$ and $\boldsymbol{\theta}_t$ are statistically independent, will also be considered. Although not true in general, this assumption is commonly used to analyze adaptive constructions since it allows to simplify the derivations without constraining the conclusions. There are several results in the adaptation literature that show that performance results that are obtained under this assumption match well the actual performance of the algorithms when the step-size is sufficiently small.

Using the update rule (15), we obtain the following recursion for $\boldsymbol{\theta}_t$:

$$\boldsymbol{\theta}_{t+1} = \big[\boldsymbol{I} - \mu(\boldsymbol{H}_{t+1}^{\text{ref}} + \nu\boldsymbol{I})\big]\boldsymbol{\theta}_t - \mu\boldsymbol{e}_{t+1}^{\circ} \tag{20}$$

Define:

$$\mathbf{h}_t^{\text{test}} = \mathsf{E}_{p_{\text{test}}(\boldsymbol{y})}\{\boldsymbol{h}_t^{\text{test}}\} \tag{21}$$

$$\mathbf{h}_t^{\text{ref}} = \mathsf{E}_{p_{\text{ref}}(\boldsymbol{y})}\{\boldsymbol{h}_t^{\text{ref}}\} \tag{22}$$

$$\mathbf{H}_t^{\text{ref}} = \mathsf{E}_{p_{\text{ref}}(\boldsymbol{y})}\{\boldsymbol{H}_t^{\text{ref}}\} \tag{23}$$

Taking the expected values on both sides of (20) and using the MIA we obtain the mean weight model:

$$\boldsymbol{m}_{\boldsymbol{\theta},t+1} = \big[\boldsymbol{I} - \mu(\mathbf{H}_{t+1}^{\text{ref}} + \nu\boldsymbol{I})\big]\ \boldsymbol{m}_{\boldsymbol{\theta},t} + \mu(\mathbf{h}_{t+1}^{\text{test}} - \mathbf{h}_{t+1}^{\text{ref}}) \tag{24}$$

We denote by $\boldsymbol{C}_{\boldsymbol{\theta},t}$ the correlation matrix of the weight vector $\boldsymbol{\theta}_t$:

$$\boldsymbol{C}_{\boldsymbol{\theta},t} = \mathsf{E}\{\boldsymbol{\theta}_t\boldsymbol{\theta}_t^{\top}\}$$

Estimating the variance of the test statistics requires a model for matrix $\boldsymbol{C}_{\boldsymbol{\theta},t}$. Post-multiplying (20) by its transpose, taking the expectation, and using the MIA, we obtain the following recursive expression:

$$\begin{aligned}\boldsymbol{C}_{\boldsymbol{\theta},t+1} = {} & (1-\mu\nu)^2\boldsymbol{C}_{\theta,t} - \mu(1-\mu\nu)(\mathbf{H}_{t+1}^{\text{ref}}\boldsymbol{C}_{\boldsymbol{\theta},t} + \boldsymbol{C}_{\boldsymbol{\theta},t}\mathbf{H}_{t+1}^{\text{ref}}) \\ & + \mu^2(\boldsymbol{T}+\boldsymbol{Q}+\boldsymbol{Z}+\boldsymbol{Z}^{\top}) - \mu(1-\mu\nu)(\boldsymbol{N}+\boldsymbol{N}^{\top})\end{aligned} \tag{25}$$

where:

$$\boldsymbol{T} = \mathsf{E}\{\boldsymbol{H}_{t+1}^{\text{ref}} \boldsymbol{\theta}_t \boldsymbol{\theta}_t^\top \boldsymbol{H}_{t+1}^{\text{ref}}\} \tag{26}$$

$$\boldsymbol{Q} = \mathsf{E}\{\boldsymbol{e}_{t+1}^\circ {\boldsymbol{e}_{t+1}^\circ}^\top\} \tag{27}$$

$$\boldsymbol{Z} = \mathsf{E}\{\boldsymbol{e}_{t+1}^\circ \boldsymbol{\theta}_t^\top \boldsymbol{H}_{t+1}^{\text{ref}}\} \tag{28}$$

$$\boldsymbol{N} = \mathsf{E}\{\boldsymbol{e}_{t+1}^\circ \boldsymbol{\theta}_t^\top\} \tag{29}$$

In the general, all these matrices can depend on t. To simplify the notations, this dependence is dropped.

3.1. Stochastic behavior analysis under the null hypothesis

3.1.1. Mean analysis

Under the null hypothesis we have:

$$\mathbf{h}_t^{\text{ref}} = \mathbf{h}_t^{\text{test}} = \mathsf{E}_{p_{\text{ref}}(\boldsymbol{y})}\{\boldsymbol{\kappa}_{\boldsymbol{\omega}}(\boldsymbol{y})\} = \mathsf{E}_{p_{\text{test}}(\boldsymbol{y})}\{\boldsymbol{\kappa}_{\boldsymbol{\omega}}(\boldsymbol{y})\} = \boldsymbol{h}$$
$$\mathbf{H}_t^{\text{ref}} = \mathsf{E}_{p_{\text{ref}}(\boldsymbol{y})}\{\boldsymbol{\kappa}_{\boldsymbol{\omega}}(\boldsymbol{y})\ \boldsymbol{\kappa}_{\boldsymbol{\omega}}^\top(\boldsymbol{y})\} = \boldsymbol{H}$$

and the mean weight model (24) simplifies to:

$$\boldsymbol{m}_{\boldsymbol{\theta},t+1} = \left[\boldsymbol{I} - \mu(\boldsymbol{H} + \nu\boldsymbol{I})\right]\ \boldsymbol{m}_{\boldsymbol{\theta},t} \tag{30}$$

The mean stability of the algorithm is then ensured by using a step size μ that satisfies:

$$\mu < \frac{2}{\zeta_{\max}\{\boldsymbol{H} + \nu\boldsymbol{I}\}} \tag{31}$$

where $\zeta_{\max}\{\cdot\}$ stands for the maximal eigenvalue of its matrix argument. Under this assumption $\boldsymbol{m}_{\theta,t} \to \mathbf{0}$. When $\boldsymbol{y}$ is Gaussian distributed, analytical expressions of $\boldsymbol{h}$ and $\boldsymbol{H}$ for a Gaussian reproducing kernel can be derived; see Appendix A.

Taking the expectation of (17) and assuming that $\boldsymbol{\theta}_t$ and $\boldsymbol{h}_t^{\text{test}}$ are independent, we get the mean of the test statistics g_t:

$$\mathsf{E}\{g_t\} = \boldsymbol{h}^\top \boldsymbol{m}_{\boldsymbol{\theta},t} \tag{32}$$

The necessary independence assumption together with the MIA will be validated by computer simulations.

Assuming (31) holds, under the null hypothesis, the asymptotic unbiasedness of the estimator implies $\lim_{t\to\infty} \mathsf{E}\{g_t\} = 0$. When initializing (20) with $\boldsymbol{\theta}_0 = \mathbf{0}$, namely, $\boldsymbol{m}_{\boldsymbol{\theta},0} = \mathbf{0}$, equation (30) implies $\boldsymbol{m}_{\boldsymbol{\theta},t} = \mathbf{0}$ for all t. As a consequence $\mathsf{E}\{g_t\} = 0$, which means that the estimation of the density ratio $r(\boldsymbol{y}_t) = 1$ is unbiased under the null hypothesis for all t.

3.1.2. Mean squared analysis

The general model of $\boldsymbol{C}_{\boldsymbol{\theta},t}$ in (25) depends on the matrices $\boldsymbol{T}$, $\boldsymbol{Q}$, $\boldsymbol{Z}$ and $\boldsymbol{N}$, defined in (26)–(29). These matrices can be computed under the null hypothesis as follows.

- Denoting $\boldsymbol{c}_{\boldsymbol{\theta},t} = \mathrm{vec}(\boldsymbol{C}_{\boldsymbol{\theta},t})$ where $\mathrm{vec}(\cdot)$ refers to the standard vectorization operator that stacks the columns of a matrix on top of each other, using the MIA and $\mathrm{vec}(\mathbf{ABC}) = (\mathbf{C}^\top \otimes \mathbf{A})\,\mathrm{vec}(\mathbf{B})$ with $\otimes$ the Kronecker product, we find:

$$\boldsymbol{T} = \frac{1}{N_{\mathrm{ref}}} \left(\mathrm{vec}^{-1}(\boldsymbol{\Gamma} \boldsymbol{c}_{\boldsymbol{\theta},t}) + (N_{\mathrm{ref}} - 1)\ \boldsymbol{H} \boldsymbol{C}_{\boldsymbol{\theta},t} \boldsymbol{H} \right) \tag{33}$$

  where $\boldsymbol{\Gamma}$ is the $(L^2 \times L^2)$ matrix defined by:

$$\boldsymbol{\Gamma} = \mathsf{E}_{p_{\mathrm{ref}}(\boldsymbol{y})} \{ \boldsymbol{\kappa}_{\boldsymbol{\omega}}(\boldsymbol{y}) \boldsymbol{\kappa}_{\boldsymbol{\omega}}^\top(\boldsymbol{y}) \otimes \boldsymbol{\kappa}_{\boldsymbol{\omega}}(\boldsymbol{y}) \boldsymbol{\kappa}_{\boldsymbol{\omega}}^\top(\boldsymbol{y}) \} \tag{34}$$

  The expression of $\boldsymbol{\Gamma}$ is given in Appendix B.

- Under the null hypothesis, $\boldsymbol{Q}$ is given by:

$$\boldsymbol{Q} = \frac{N_{\mathrm{ref}} + N_{\mathrm{test}}}{N_{\mathrm{ref}}\ N_{\mathrm{test}}} \ (\boldsymbol{H} - \boldsymbol{h}\boldsymbol{h}^\top) \tag{35}$$

- In the same way as $\boldsymbol{T}$, we find that:

$$\boldsymbol{Z} = \frac{1}{N_{\mathrm{ref}}} \left(\mathrm{vec}^{-1}(\boldsymbol{\Delta} \boldsymbol{m}_{\boldsymbol{\theta},t}) - \boldsymbol{h} \boldsymbol{m}_{\boldsymbol{\theta},t}^\top \boldsymbol{H} \right) \tag{36}$$

  where $\boldsymbol{\Delta}$ is the $(L^2 \times L)$ matrix defined by:

$$\boldsymbol{\Delta} = \mathsf{E}_{p_{\mathrm{ref}}(\boldsymbol{y})} \{ \boldsymbol{\kappa}_{\boldsymbol{\omega}}(\boldsymbol{y}) \boldsymbol{\kappa}_{\boldsymbol{\omega}}(\boldsymbol{y})^\top \otimes \boldsymbol{\kappa}_{\boldsymbol{\omega}}(\boldsymbol{y}) \} \tag{37}$$

  The expression of $\boldsymbol{\Delta}$ is given in Appendix B.

- Application of the MIA implies $\boldsymbol{N} = \mathbf{0}$.

The variance of the test statistics g_t in (17) can be calculated using the independence assumption required previously for the computation of its mean. In particular:

$$\begin{aligned} \mathrm{var}\{g_t\} &= \mathsf{E}\{g(\boldsymbol{y}_t)^2\} - \mathsf{E}\{g(\boldsymbol{y}_t)\}^2 \\ &= \frac{1}{N_{test}} \left(\mathrm{tr}(\boldsymbol{H} \boldsymbol{C}_{\boldsymbol{\theta},t}) - (\boldsymbol{h}^\top \boldsymbol{m}_{\boldsymbol{\theta},t})^2 \right) \end{aligned} \tag{38}$$

The mean term $\boldsymbol{m}_{\boldsymbol{\theta},t}$ in (36), (38) equals zero when $\boldsymbol{\theta}_0 = \mathbf{0}$ and can be neglected for large values of t according to the mean analysis in Section 3.1.1. As a consequence, setting $\boldsymbol{Z} = \mathbf{0}$, vectorizing (25) and using standard results on Kronecker product leads to the following proposition.

Proposition 1. *Under the null hypothesis, neglecting $\boldsymbol{m}_{\boldsymbol{\theta},t}$ and assuming the MIA holds, $\boldsymbol{c}_{\boldsymbol{\theta},t} = \mathrm{vec}(\boldsymbol{C}_{\boldsymbol{\theta},t})$ verifies:*

$$\boldsymbol{c}_{\boldsymbol{\theta},t+1} = \boldsymbol{S}\boldsymbol{c}_{\boldsymbol{\theta},t} + \mu^2 \,\mathrm{vec}(\boldsymbol{Q}) \tag{39}$$

with:

$$\boldsymbol{S} = (1-\mu\nu)^2\boldsymbol{I} + \frac{\mu^2}{N_{ref}}(\boldsymbol{\Gamma} + (N_{ref}-1)\boldsymbol{H}\otimes\boldsymbol{H}) - \mu(1-\mu\nu)(\boldsymbol{H}\oplus\boldsymbol{H})$$

where $\boldsymbol{H}\oplus\boldsymbol{H} = \boldsymbol{H}\otimes\boldsymbol{I} + \boldsymbol{I}\otimes\boldsymbol{H}$. The variance of the test statistics (17) *is given by:*

$$\mathrm{var}\{g_t\} = \frac{1}{N_{test}}\,\mathrm{tr}(\boldsymbol{H}\boldsymbol{C}_{\boldsymbol{\theta},t}) \tag{40}$$

The stability of matrix $\boldsymbol{S}$ then ensures the mean-square stability of the algorithm. If the algorithm is mean-square stable, then, $\boldsymbol{c}_{\boldsymbol{\theta},t}$ converges to:

$$\boldsymbol{c}_{\boldsymbol{\theta},\infty} = \mu^2(\boldsymbol{I}-\boldsymbol{S})^{-1}\,\mathrm{vec}(\boldsymbol{Q}) \tag{41}$$

The asymptotic variance of the test statistics directly derives from this result. Assuming a small step size μ we have:

$$\boldsymbol{S} = \boldsymbol{I} - 2\mu\nu\boldsymbol{I} - \mu\boldsymbol{H}\oplus\boldsymbol{H} + o(\mu)$$

Replacing in (40) and using classical properties of Kronecker products, the asymptotic variance simplifies to:

$$\mathrm{var}\{g_\infty\} = \frac{\mu}{N_{\mathrm{test}}}\,\mathrm{tr}\left(\boldsymbol{H}\,\mathrm{vec}^{-1}\left((2\nu\boldsymbol{I}+\boldsymbol{H}\oplus\boldsymbol{H})^{-1}\,\mathrm{vec}(\boldsymbol{Q})\right)\right) + o(\mu) \tag{42}$$

$$= \frac{\mu}{N_{\mathrm{test}}}\,\mathrm{vec}(\boldsymbol{H})(2\nu\boldsymbol{I}+\boldsymbol{H}\oplus\boldsymbol{H})^{-1}\,\mathrm{vec}(\boldsymbol{Q}) + o(\mu) \tag{43}$$

Note that the rightmost term in (42) can be efficiently computed as the solution of the Lyapunov equation $(\nu\boldsymbol{I}+\boldsymbol{H})\boldsymbol{X} + \boldsymbol{X}(\nu\boldsymbol{I}+\boldsymbol{H}) = \boldsymbol{Q}$, see [37, Proposition 7.2.4].

3.2. Stochastic behavior analysis in the presence of a change-point

Under the assumption of the presence of a single change-point t_0, the analysis is conducted by comparing each time instant t to t_0, as $\mathsf{h}_t^{\mathrm{ref}}$, $\mathsf{h}_t^{\mathrm{test}}$ and $\mathsf{H}_t^{\mathrm{ref}}$ defined by (21)–(23) depend on time t. We assume that input data $\boldsymbol{y}_i$ are i.i.d with $\boldsymbol{y}_i \sim p_0(\cdot)$ before the change, and i.i.d with $\boldsymbol{y}_i \sim p_1(\cdot)$ after the change.

- If $t < t_0$: $\mathsf{h}_t^{\mathrm{test}} = \mathsf{h}_t^{\mathrm{ref}} = \boldsymbol{h}_0$ and $\mathsf{H}_t^{\mathrm{ref}} = \boldsymbol{H}_0$.
- If $t_0 \le t \le t_0 + N_{\mathrm{test}} - 1$: the test interval contains samples from both distributions; see figure 1. According to (21):

$$\mathsf{h}_t^{\mathrm{test}} = \frac{1}{N_{\mathrm{test}}} \sum_{i=t-(N_{\mathrm{test}}-1)}^{t} \mathsf{E}\{\boldsymbol{\kappa}_{\boldsymbol{\omega}}(\boldsymbol{y}_i)\}$$

$$= \frac{1}{N_{\mathrm{test}}}\left(n_0\,\boldsymbol{h}_0 + n_1\,\boldsymbol{h}_1\right)$$

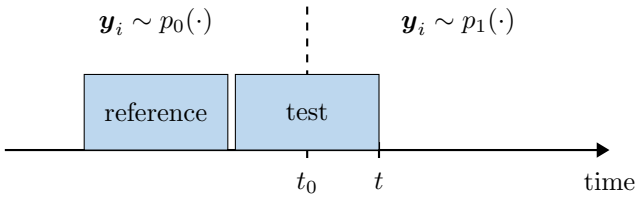

Figure 1: An illustration of CPD procedure when the test interval contains samples driven by $p_0(\cdot)$ and $p_1(\cdot)$.

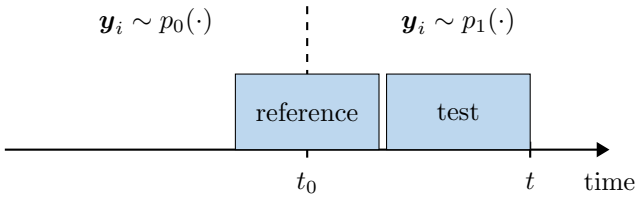

Figure 2: An illustration of CPD procedure when the reference interval contains samples driven by $p_0(\cdot)$ and $p_1(\cdot)$.

where $n_1 = t - t_0 + 1$, and $n_0 = N_{\text{test}} - n_1$. In that case: $\mathbf{h}_t^{\text{ref}} = \boldsymbol{h}_0$ and $\mathbf{H}_t^{\text{ref}} = \boldsymbol{H}_0$.

- If $t_0 + N_{\text{test}} \leq t \leq t_0 + N_{\text{test}} + N_{\text{ref}} - 1$: the reference interval contains samples from both distributions; see figure 2. In the same way we find:

$$\mathbf{h}_t^{\text{ref}} = \frac{1}{N_{\text{ref}}} \left(n_0' \, \boldsymbol{h}_0 + n_1' \, \boldsymbol{h}_1\right)$$
$$\mathbf{H}_t^{\text{ref}} = \frac{1}{N_{\text{ref}}} \left(n_0' \, \boldsymbol{H}_0 + n_1' \, \boldsymbol{H}_1\right)$$

  where: $n_1' = t - (t_0 + N_{\text{test}}) + 1$, and $n_0' = N_{\text{ref}} - n_1'$.

- If $t \geq t_0 + N_{\text{ref}} + N_{\text{test}}$, $\mathbf{h}_t^{\text{test}} = \mathbf{h}_t^{\text{ref}} = \boldsymbol{h}_1$ and $\mathbf{H}_t^{\text{ref}} = \boldsymbol{H}_1$.

$\boldsymbol{h}_0$, $\boldsymbol{h}_1$, $\boldsymbol{H}_0$ and $\boldsymbol{H}_1$ can be computed using the expressions in Appendix A when p_0 is $\mathcal{N}(\boldsymbol{\mu}_0, \boldsymbol{R}_0)$ and p_1 is $\mathcal{N}(\boldsymbol{\mu}_1, \boldsymbol{R}_1)$.

3.2.1. Mean analysis

A recursive model of $\boldsymbol{m}_{\boldsymbol{\theta},t}$ can be obtained by replacing $\mathbf{h}_t^{\text{ref}}$, $\mathbf{h}_t^{\text{test}}$, and $\mathbf{H}_t^{\text{ref}}$ by their expressions over time in (24).

- If $t < t_0$:

$$\boldsymbol{m}_{\boldsymbol{\theta},t+1} = \left[\boldsymbol{I} - \mu(\boldsymbol{H}_0 + \nu\boldsymbol{I})\right]\boldsymbol{m}_{\boldsymbol{\theta},t} \tag{44}$$

- If $t_0 \leq t \leq t_0 + N_{\text{test}} - 1$:

$$\boldsymbol{m}_{\boldsymbol{\theta},t+1} = \left[\boldsymbol{I} - \mu(\boldsymbol{H}_0 + \nu\boldsymbol{I})\right] \, \boldsymbol{m}_{\boldsymbol{\theta},t} + \mu \, \frac{n_1}{N_{\text{test}}} \, (\boldsymbol{h}_1 - \boldsymbol{h}_0) \tag{45}$$

- If $t_0 + N_{\text{test}} \leq t \leq t_0 + N_{\text{test}} + N_{\text{ref}} - 1$:

$$\begin{aligned}\boldsymbol{m}_{\boldsymbol{\theta},t+1} = \big[\boldsymbol{I} - \mu\ (\frac{n_0'}{N_{\text{ref}}}\ \boldsymbol{H}_0 + \frac{n_1'}{N_{\text{ref}}}\ \boldsymbol{H}_1 + \nu\boldsymbol{I})\big]\ \boldsymbol{m}_{\boldsymbol{\theta},t} \\ + \mu\ \frac{n_0'}{N_{\text{ref}}}\ (\boldsymbol{h}_1 - \boldsymbol{h}_0)\end{aligned} \tag{46}$$

- If $t \geq t_0 + N_{\text{ref}} + N_{\text{test}}$:

$$\boldsymbol{m}_{\boldsymbol{\theta},t+1} = \big[\boldsymbol{I} - \mu(\boldsymbol{H}_1 + \nu\boldsymbol{I})\big]\boldsymbol{m}_{\boldsymbol{\theta},t} \tag{47}$$

  and the mean stability of the algorithm is ensured by using a step size μ that satisfies:

$$\mu < \frac{2}{\zeta_{\max}\{\boldsymbol{H}_1 + \nu\boldsymbol{I}\}}$$

The mean of the test statistics (17) is, in the presence of a change-point, given by:

$$\mathsf{E}\{g_t\} = \begin{cases} \boldsymbol{h}_0^\top\ \boldsymbol{m}_{\boldsymbol{\theta},t} & t < t_0 \\ \dfrac{1}{N_{\text{test}}}(n_1\boldsymbol{h}_1 + n_0\boldsymbol{h}_0)^\top\boldsymbol{m}_{\boldsymbol{\theta},t} & t_0 \leq t < t_0 + N_{\text{test}} \\ \boldsymbol{h}_1^\top\ \boldsymbol{m}_{\boldsymbol{\theta},t} & t \geq t_0 + N_{\text{test}} \end{cases} \tag{48}$$

3.2.2. Mean squared analysis

The first step consists of the computation of the matrices $\boldsymbol{T}$, $\boldsymbol{Q}$, $\boldsymbol{Z}$ and $\boldsymbol{N}$ in the presence of a change point.

- Following the same steps as in (33), we find:

$$\boldsymbol{T} = \frac{1}{N_{\text{ref}}}\ \Big(\text{vec}^{-1}(\boldsymbol{\Gamma}\boldsymbol{c}_{\boldsymbol{\theta},t}) + (N_{\text{ref}} - 1)\ \mathbf{H}_{t+1}^{\text{ref}}\boldsymbol{C}_{\boldsymbol{\theta},t}\mathbf{H}_{t+1}^{\text{ref}}\Big) \tag{49}$$

  where $\boldsymbol{\Gamma}$ is defined in (34) and depends on t.

- $\boldsymbol{Q}$ can be decomposed as:

$$\boldsymbol{Q} = \boldsymbol{Q}_1 + \boldsymbol{Q}_2 - (\boldsymbol{Q}_3 + \boldsymbol{Q}_3^\top) \tag{50}$$

  where:

$$\boldsymbol{Q}_1 = \mathsf{E}\{\boldsymbol{h}_{t+1}^{\text{test}}\ \boldsymbol{h}_{t+1}^{\text{test}\top}\} \tag{51}$$

  Substituting (11) into (51) and expanding the expression we get:

$$\boldsymbol{Q}_1 = \frac{1}{N_{\text{test}}}\ \mathbf{H}_{t+1}^{\text{test}} + (1 - \frac{1}{N_{\text{test}}})\ \mathbf{h}_{t+1}^{\text{test}}(\mathbf{h}_{t+1}^{\text{test}})^\top$$

  where:

$$\mathbf{H}_t^{\text{test}} = \mathsf{E}_{p_{\text{test}}(\boldsymbol{y})}\{\boldsymbol{H}_t^{\text{test}}\}$$

In the same way, we find:

$$\begin{aligned}\boldsymbol{Q}_2 &= \mathsf{E}\{\boldsymbol{h}_{t+1}^{\text{ref}}\ \boldsymbol{h}_{t+1}^{\text{ref}\top}\} \\ &= \frac{1}{N_{\text{ref}}}\ \mathbf{H}_{t+1}^{\text{ref}} + (1 - \frac{1}{N_{\text{ref}}})\ \mathbf{h}_{t+1}^{\text{ref}}(\mathbf{h}_{t+1}^{\text{ref}})^\top\end{aligned}$$

and since the samples $\boldsymbol{y}_i$ in the reference and test intervals are independent,

$$\begin{aligned}\boldsymbol{Q}_3 &= \mathsf{E}\{\boldsymbol{h}^{\text{test}}\boldsymbol{h}^{\text{ref}\top}\} \\ &= \mathbf{h}_{t+1}^{\text{test}}(\mathbf{h}_{t+1}^{\text{ref}})^\top\end{aligned}$$

- The matrix $\boldsymbol{Z}$ can be expanded as:

$$\boldsymbol{Z} = \mathsf{E}\{\boldsymbol{h}_{t+1}^{\text{ref}}\boldsymbol{\theta}_t^\top \boldsymbol{H}_{t+1}^{\text{ref}}\} - \mathsf{E}\{\boldsymbol{h}_{t+1}^{\text{test}}\boldsymbol{\theta}_t^\top \boldsymbol{H}_{t+1}^{\text{ref}}\} \tag{52}$$

The first expectation term in (52) can be computed using the MIA and the vectorization operator:

$$\begin{aligned}\boldsymbol{Z} = &\frac{1}{N_{\text{ref}}}\ \left(\text{vec}^{-1}(\boldsymbol{\Delta}\boldsymbol{m}_{\boldsymbol{\theta},t}) + (N_{\text{ref}} - 1)\ \mathbf{h}_{t+1}^{\text{ref}}\boldsymbol{m}_{\boldsymbol{\theta},t}^\top\mathbf{H}_{t+1}^{\text{ref}}\right) \\ &- \mathbf{h}_{t+1}^{\text{test}}\boldsymbol{m}_{\boldsymbol{\theta},t}^\top\mathbf{H}_{t+1}^{\text{ref}}\end{aligned} \tag{53}$$

where $\boldsymbol{\Delta}$ is defined in (37) and depends on t.

- Using the MIA:

$$\boldsymbol{N} = (\mathbf{h}_{t+1}^{\text{test}} - \mathbf{h}_{t+1}^{\text{ref}})\ \boldsymbol{m}_{\boldsymbol{\theta},t}^\top$$

The last step consists in replacing the expressions of $\mathbf{h}_t^{\text{ref}}$, $\mathbf{h}_t^{\text{test}}$, and $\mathbf{H}_t^{\text{ref}}$ as a function of t in $\boldsymbol{T}$, $\boldsymbol{Q}$, $\boldsymbol{Z}$ and $\boldsymbol{N}$. We will denote by $\boldsymbol{\Gamma}_0$ (resp. $\boldsymbol{\Gamma}_1$) and $\boldsymbol{\Delta}_0$ (resp. $\boldsymbol{\Delta}_1$) matrices $\boldsymbol{\Gamma}$ and $\boldsymbol{\Delta}$ in (34,37) computed for $\boldsymbol{y}_i \sim p_0(\cdot)$ (resp. $\boldsymbol{y}_i \sim p_1(\cdot)$), see Appendix B. This leads to the following proposition.

Proposition 2. *Under the assumption of a change-point t_0 and assuming that the MIA holds, $\boldsymbol{C}_{\boldsymbol{\theta},t}$ is given by* (25) *where:*

- *If $t < t_0$:*

$$\begin{aligned}\boldsymbol{T} &= \frac{1}{N_{ref}}\ \left(\text{vec}^{-1}(\boldsymbol{\Gamma}_0\boldsymbol{c}_{\boldsymbol{\theta},t}) + (N_{ref} - 1)\ \boldsymbol{H}_0\boldsymbol{C}_{\boldsymbol{\theta},t}\boldsymbol{H}_0\right) \\ \boldsymbol{Q} &= \frac{N_{ref} + N_{test}}{N_{ref}\ N_{test}}\ (\boldsymbol{H}_0 - \boldsymbol{h}_0\boldsymbol{h}_0^\top) \\ \boldsymbol{Z} &= \frac{1}{N_{ref}}\ \left(\text{vec}^{-1}(\boldsymbol{\Delta}_0\boldsymbol{m}_{\boldsymbol{\theta},t}) - \boldsymbol{h}_0\boldsymbol{m}_{\boldsymbol{\theta},t}^\top\boldsymbol{H}_0\right) \\ \boldsymbol{N} &= \mathbf{0}\end{aligned}$$

- *If* $t_0 \le t \le t_0 + N_{test} - 1$*:*

$$
\begin{aligned}
\boldsymbol{T} &= \frac{1}{N_{ref}} \left(\mathrm{vec}^{-1}(\boldsymbol{\Gamma}_0 \boldsymbol{c}_{\boldsymbol{\theta},t}) + (N_{ref} - 1)\ \boldsymbol{H}_0 \boldsymbol{C}_{\boldsymbol{\theta},t} \boldsymbol{H}_0 \right) \\
\boldsymbol{Q} &= \big(\frac{n_0}{N_{test}^2} + \frac{1}{N_{ref}}\big)\boldsymbol{H}_0 + \frac{n_1}{N_{test}^2}\boldsymbol{H}_1 + \frac{n_1(n_1-1)}{N_{test}^2}\boldsymbol{h}_1\boldsymbol{h}_1^\top \\
&\quad + \Big(\frac{n_0\big(n_0-1\big)}{N_{test}^2} - \frac{2n_0}{N_{test}} - \frac{1}{N_{ref}} + 1\Big)\ \boldsymbol{h}_0\boldsymbol{h}_0^\top \\
&\quad + n_1\ \big(\frac{n_0}{N_{test}^2} - \frac{1}{N_{test}}\big)(\boldsymbol{h}_0\boldsymbol{h}_1^\top + \boldsymbol{h}_1\boldsymbol{h}_0^\top) \\
\boldsymbol{Z} &= \big(1 - \frac{1}{N_{ref}} - \frac{n_0}{N_{test}}\big)\ \boldsymbol{h}_0\boldsymbol{m}_{\boldsymbol{\theta},t}^\top\boldsymbol{H}_0 - \frac{n_1}{N_{test}}\ \boldsymbol{h}_1\boldsymbol{m}_{\boldsymbol{\theta},t}^\top\boldsymbol{H}_0 \\
&\quad + \frac{1}{N_{ref}}\ \mathrm{vec}^{-1}(\boldsymbol{\Delta}_0\boldsymbol{m}_{\boldsymbol{\theta},t}) \\
\boldsymbol{N} &= \frac{n_1}{N_{test}}\ (\boldsymbol{h}_1 - \boldsymbol{h}_0)\ \boldsymbol{m}_{\boldsymbol{\theta},t}^\top
\end{aligned}
$$

- *If* $t_0 + N_{test} \le t \le t_0 + N_{test} + N_{ref} - 1$

$$
\begin{aligned}
\boldsymbol{T} &= \frac{1}{N_{ref}^2}\ \Big(n_0'\ \ \mathrm{vec}^{-1}(\boldsymbol{\Gamma}_0\boldsymbol{c}_{\boldsymbol{\theta},t}) + n_1'\ \ \mathrm{vec}^{-1}(\boldsymbol{\Gamma}_1\boldsymbol{c}_{\boldsymbol{\theta},t}) \\
&\quad + n_0'(n_0'-1)\boldsymbol{H}_0\boldsymbol{C}_{\boldsymbol{\theta},t}\boldsymbol{H}_0 + n_1'(n_1'-1)\boldsymbol{H}_1\boldsymbol{C}_{\boldsymbol{\theta},t}\boldsymbol{H}_1 \\
&\quad + n_0'n_1'\ (\boldsymbol{H}_0\boldsymbol{C}_{\boldsymbol{\theta},t}\boldsymbol{H}_1 + \boldsymbol{H}_1\boldsymbol{C}_{\boldsymbol{\theta},t}\boldsymbol{H}_0)\Big) \\
\boldsymbol{Q} &= \frac{n_0'}{N_{ref}^2}\ \boldsymbol{H}_0 + \big(\frac{n_1'}{N_{ref}^2} + \frac{1}{N_{test}}\big)\ \boldsymbol{H}_1 + \frac{n_0'(n_0'-1)}{N_{ref}^2}\ \boldsymbol{h}_0\boldsymbol{h}_0^\top \\
&\quad + \Big(\frac{n_1'(n_1'-1)}{N_{ref}^2} - \frac{2n_1'}{N_{ref}} - \frac{1}{N_{test}} + 1\Big)\ \boldsymbol{h}_1\boldsymbol{h}_1^\top \\
&\quad + n_0'\ \big(\frac{n_1'}{N_{ref}^2} - \frac{1}{N_{ref}}\big)\ (\boldsymbol{h}_0\boldsymbol{h}_1^\top + \boldsymbol{h}_1\boldsymbol{h}_0^\top) \\
\boldsymbol{Z} &= \frac{1}{N_{ref}^2}\ \Big(n_0'\,\mathrm{vec}^{-1}(\boldsymbol{\Delta}_0\boldsymbol{m}_{\boldsymbol{\theta},t}) + n_1'\,\mathrm{vec}^{-1}(\boldsymbol{\Delta}_1\boldsymbol{m}_{\boldsymbol{\theta},t}) \\
&\quad + n_0'(n_0'-1)\ \boldsymbol{h}_0\boldsymbol{m}_{\boldsymbol{\theta},t}^\top\boldsymbol{H}_0 + n_0'n_1'\ \boldsymbol{h}_0\boldsymbol{m}_{\boldsymbol{\theta},t}^\top\boldsymbol{H}_1 \\
&\quad - n_0'^2\ \boldsymbol{h}_1\boldsymbol{m}_{\boldsymbol{\theta},t}^\top\boldsymbol{H}_0 - n_1'(n_0'+1)\ \boldsymbol{h}_1\boldsymbol{m}_{\boldsymbol{\theta},t}^\top\boldsymbol{H}_1\Big) \\
\boldsymbol{N} &= \frac{n_0'}{N_{ref}}\ (\boldsymbol{h}_1 - \boldsymbol{h}_0)\ \boldsymbol{m}_{\boldsymbol{\theta},t}^\top
\end{aligned}
$$

- *If* $t \geq t_0 + N_{test} + N_{ref}$

$$\begin{aligned}
\boldsymbol{T} &= \frac{1}{N_{ref}} \left(\mathrm{vec}^{-1}(\boldsymbol{\Gamma}_1 \boldsymbol{c}_{\boldsymbol{\theta},t}) + (N_{ref} - 1)\ \boldsymbol{H}_1 \boldsymbol{C}_{\boldsymbol{\theta},t} \boldsymbol{H}_1^\top \right) \\
\boldsymbol{Q} &= \frac{N_{ref} + N_{test}}{N_{ref}\ N_{test}}\ (\boldsymbol{H}_1 - \boldsymbol{h}_1 \boldsymbol{h}_1^\top) \\
\boldsymbol{Z} &= \frac{1}{N_{ref}}\ \left(\mathrm{vec}^{-1}(\boldsymbol{\Delta}_1 \boldsymbol{m}_{\boldsymbol{\theta},t}) - \boldsymbol{h}_1 \boldsymbol{m}_{\boldsymbol{\theta},t}^\top \boldsymbol{H}_1 \right) \\
\boldsymbol{N} &= \boldsymbol{0}
\end{aligned}$$

The variance of the test statistics (17) *is given by:*

$$\begin{aligned}
&N_{test}\, \mathrm{var}\{g_t\} \\
&\quad = \begin{cases}
\mathrm{tr}(\boldsymbol{H}_0 \boldsymbol{C}_{\boldsymbol{\theta},t}) - (\boldsymbol{h}_0^\top \boldsymbol{m}_{\boldsymbol{\theta},t})^2 & t < t_0 \\
\mathrm{tr}(\mathbf{H}_t^{test} \boldsymbol{C}_{\boldsymbol{\theta},t}) - (\mathbf{h}_t^{test,\top} \boldsymbol{m}_{\boldsymbol{\theta},t})^2 & t_0 \leq t < t_0 + N_{test} \\
\mathrm{tr}(\boldsymbol{H}_1 \boldsymbol{C}_{\boldsymbol{\theta},t}) - (\boldsymbol{h}_1^\top \boldsymbol{m}_{\boldsymbol{\theta},t})^2 & t \geq t_0 + N_{test}
\end{cases}
\end{aligned} \tag{54}$$

All these expressions can be further simplified by neglecting $\boldsymbol{m}_{\boldsymbol{\theta},t}$, specifically when $t < t_0$ if e.g. $\boldsymbol{\theta}_0 = \boldsymbol{0}$ and $t \gg t_0$ assuming mean stability.

4. Simulation results

The `julia` code to reproduce all these experiments will be made available at `github.com/andferrari`.

4.1. Model validation

In this subsection, we present Monte Carlo simulations to illustrate the accuracy of the models derived in Section 3. Analytical expressions of the mean and the variance of the detection statistics under the null hypothesis are first considered. The observations $\boldsymbol{y}_i$ were zero-mean two-dimensional i.i.d Gaussian vectors, with correlation coefficient equal to 0.25, and standard deviation equal to 0.5. Under these assumptions and for a Gaussian reproducing kernel, expressions of $\boldsymbol{h}$ and $\boldsymbol{H}$ are given in Appendix A. The algorithm parameters were set as follows: the bandwidth of the Gaussian kernel was $\sigma = 0.25$, the regularization parameter $\nu = 10^{-3}$, the step-size $\mu = 5.10^{-4}$. The windows lengths were set to $N_{\mathrm{ref}} = N_{\mathrm{test}} = 250$, and the $L = 16$ dictionary elements were obtained by sampling the same distribution as $\boldsymbol{y}_i$. The results were averaged over 500 Monte Carlo runs.

Figures 3 and 4 compare respectively the theoretical models of the mean given by (30), (32) and the variance given in Proposition 1 of the detection statistics, to Monte Carlo simulations. The asymptotic value of the variance computed from (41) is also reported. The initial weight vector was set to $\boldsymbol{\theta}_0 = (0.3,\, 0.3)^\top$. The simulation results clearly show a good accuracy between the models and the actual performance provided by Monte Carlo simulations.

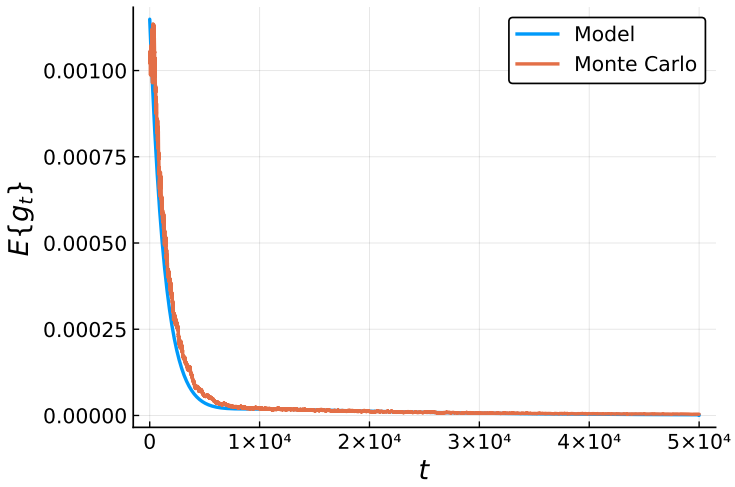

Figure 3: Mean of NOUGAT detection statistics obtained using model (32) and Monte Carlo simulations under the null hypothesis.

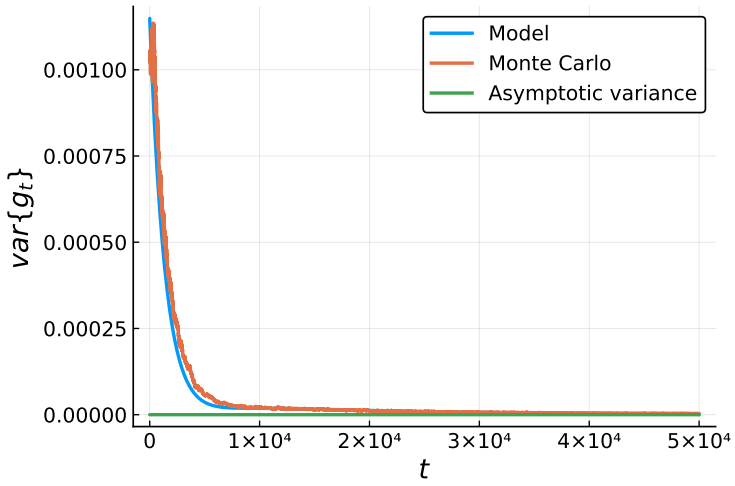

Figure 4: Variance of NOUGAT detection statistics obtained using model (38) and Monte Carlo simulations under the null hypothesis.

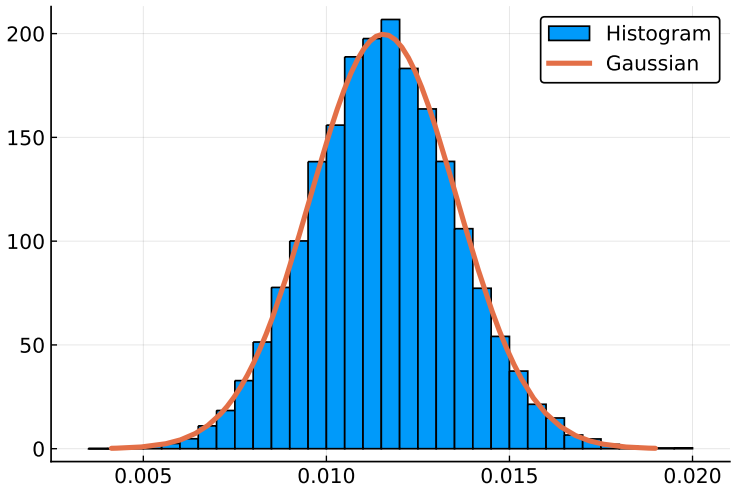

Figure 5: Histogram of NOUGAT detection statistics under the null hypothesis.

These results also confirm the asymptotic unbiasedness of the estimator: $\mathsf{E}\{g_t\}$ converges to 0 as expected, and validate the assumptions used in the derivations.

We also provide the histogram of the detection statistics in Figure 5. Contrarily to [25], the histogram is very close to its Gaussian approximation as reported in this figure. Note that, as proved above, the mean converges towards zero for larger values of t. The accuracy of g_t Gaussian approximation is a central result to set the threshold and guarantee a given false alarm rate using, e.g., the asymptotic expression of (38) computed using (41). Figures 6 compares the asymptotic variance of the test statistic computed using (40), (41) and its first order approximation (43) when the step size μ is small ($\mu = 5.10^{-4}$ in this section). Note that this expression depends on $\boldsymbol{h}$ and $\boldsymbol{H}$ which can be computed by Monte Carlo simulations in the non Gaussian case.

For the second part of the simulations, we inserted a change-point at time instant $t_0 = 25 \cdot 10^3$ by changing the input vectors correlation coefficient to 0.1 and standard deviation to 0.7. The results for the mean behavior are given in Figure 7, and for the variance in Figure 8. Both figures clearly show that the theoretical curves provided by (44)-(48) and Proposition 2 match well the actual performance provided by Monte Carlo simulations, especially in the vicinity of the change point.

4.2. Performances comparison

This section aims to compare the performances of 1) dRuLSIF, a debiased version of RuLSIF obtained solving (9) at each time instant t, 2) NOUGAT, the proposed online version of dRuLSIF, and 3) MA, as defined in (19). Note that all these algorithms share the same memory footprint.

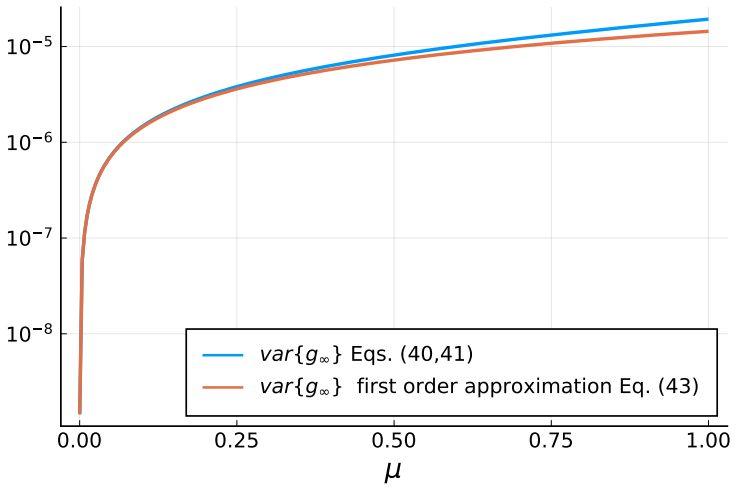

Figure 6: Asymptotic variance of the test statistic compared to its first order approximation, as a function of the step size μ.

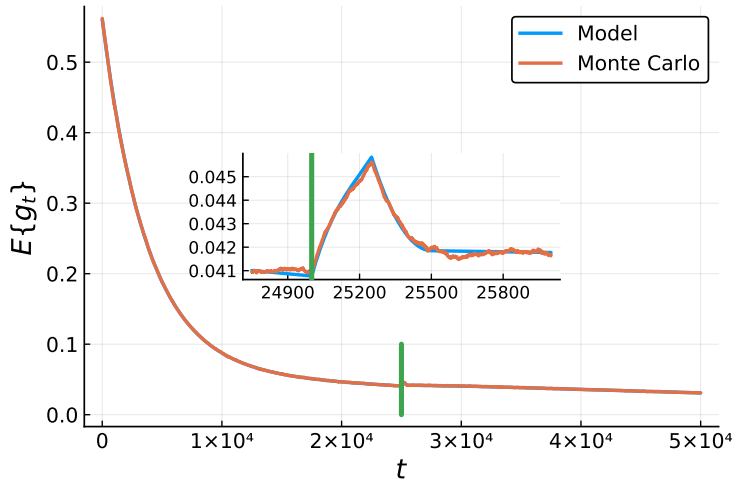

Figure 7: Mean of NOUGAT detection statistics obtained using the model (48) and Monte Carlo simulations. The change is identified by the green line.

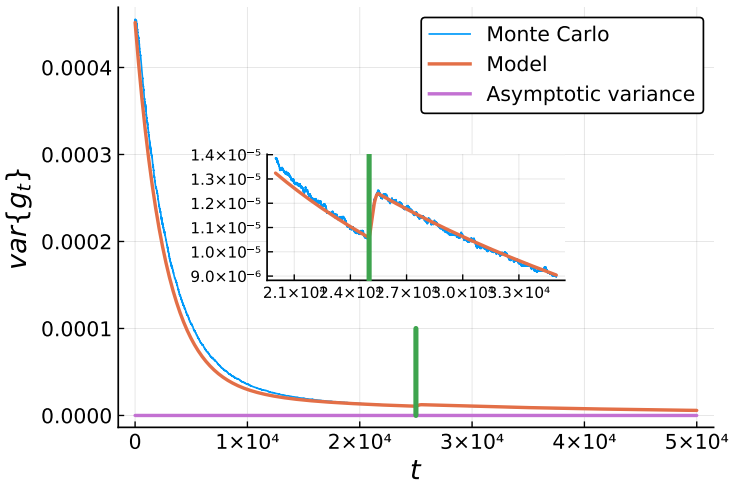

Figure 8: Variance of NOUGAT detection statistics obtained using model (54) and Monte Carlo simulations. The change is identified by the green line.

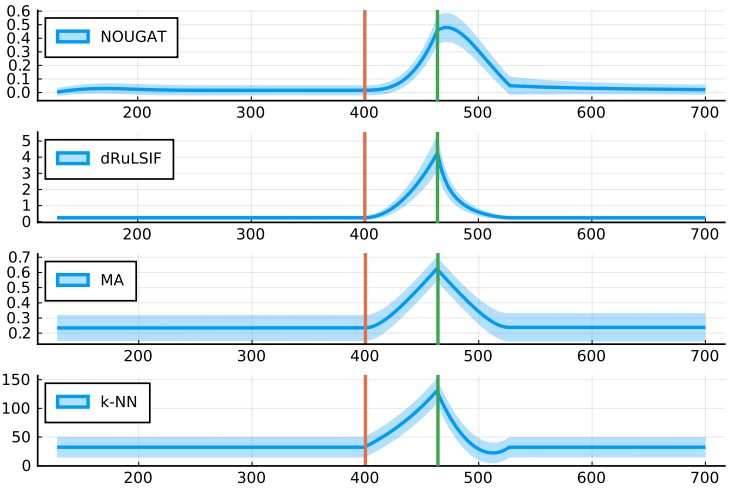

Figure 9: Mean of the test statistic (± standard deviation) for dRuLSIF, NOUGAT and MA. The change point t_0 is located at the red line and $t_0 + N_{\text{ref}}$ at the green line.

In order to assess the performance of all these algorithms compared to a non-kernel-based algorithm, we shall now report the detection performance of a nearest-neighbors based CPD algorithm. The algorithm we selected is based on the two-sample test proposed by [38, 39], and recently considered for CPD in [40]. At each time instant t, the k-nearest-neighbors algorithm is applied to the samples in the interval $(t-N_{\text{ref}}-N_{\text{test}}+1, t)$. The test statistic is related to the number of edges N_{e} of the graph that connect observations in the reference window with observations in the test window. Indeed, when these observations are driven by two different distributions, this graph tends to be clustered with respect to the reference and the test window, and N_{e} is then ideally close to zero. Correspondingly, the test statistic, denoted as k-NN, is N_e corrected by its mean value under equal distributions hypothesis, see e.g. [40].

The observations $\boldsymbol{y}_t$ were sampled from a mixture of n k-dimensional Gaussian distributions $\mathcal{N}_k(\boldsymbol{m}_q, q^{-1}\boldsymbol{C}_q)$, with $q = 1, \ldots, n$. The weights ϕ_q of the mixture model were sampled from a n-dimensional Dirichlet distribution of parameter α. The means $\boldsymbol{m}_q$ were sampled from $\mathcal{N}_k(\boldsymbol{0}, \boldsymbol{I})$ and the covariance matrices $\boldsymbol{C}_q$ from a Wishart distribution with scale matrix $\boldsymbol{I}$ and $k+2$ degrees of freedom, that is, $\mathcal{W}_k(\boldsymbol{I}, k+2)$.

The change point was set to $t_0 = 400$, the number of samples to $n_t = 700$, the dimension of measurements was fixed to $k = 6$, the number of mixture components was set to $n = 3$ and $\alpha = 5$. All the parameters $(\boldsymbol{m}_q, \phi_q, \boldsymbol{C}_q)$, with $q = 1, \ldots, n$ of the GMM were resampled at time $t = t_0$.

For all simulations, we considered a Gaussian kernel. Its bandwidth σ was set using the median trick, that is, the median of the pairwise distances between samples governed by the same distribution as $\boldsymbol{y}_t$ under the null hypothesis. A dictionary of $L = 80$ elements was designed by sampling the same distribution. For all Monte Carlo simulations, these parameters were kept fixed. For all algorithms, the window lengths were set to $N_{\text{ref}} = N_{\text{test}} = 64$. The regularization parameter for dRuLSIF and NOUGAT was set to $\nu = 10^{-2}$ and the step size for NOUGAT was set to $\mu = 47 \cdot 10^{-3}$. The number of nearest-neighbors of the k-NN was set to 10. This value was obtained experimentally in order to achieve the best performance.

4.2.1. Detection performance

Figure 9 provides the mean ± standard deviation for the four test statistics, namely, NOUGAT, dRuLSIF, MA and k-NN computed from 10^6 runs. Note that, contrarily to Figure 3, NOUGAT was initialized with $\boldsymbol{\theta}_{-1} = \boldsymbol{0}$ to guarantee unbiasedness under the null hypothesis as shown in Section II.A.

When comparing the ratio between the peak at $t_0 + N_{\text{ref}}$ (green line) and the noise level for $t < t_0$ (before the red line), Figure 9 reveals a slight drop in performance of NOUGAT compared to dRuLSIF. A larger loss of performance can be observed with MA and k-NN compared to NOUGAT and dRulSIF. As MA test statistic is the norm of the solution of (9) with $\boldsymbol{H}_t^{\text{ref}} = \boldsymbol{I}$ and $\nu = 0$, it does not take into account correlations in the feature space. In addition, MA does not take advantage of the functional approximation framework as it tests the norm of the parameters vector while NOUGAT approximates the

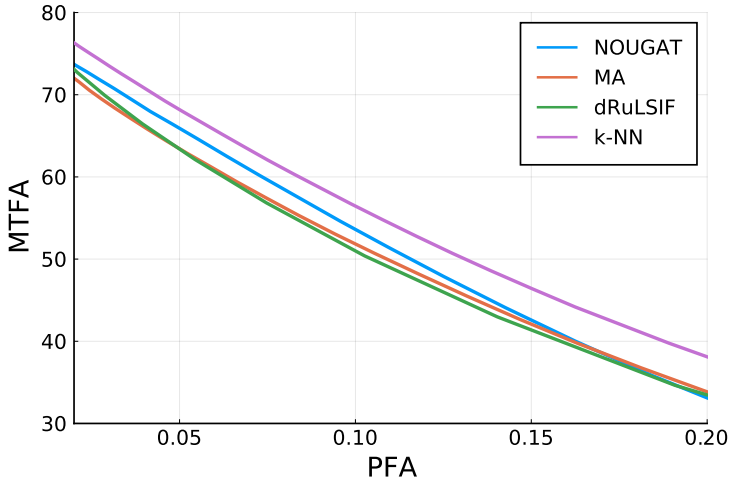

Figure 10: MTFA as a function of PFA for NOUGAT, MA and dRuLSIF.

likelihood ratio (17). Moreover, we can observe a small detection delay between NOUGAT and the other algorithms. This can be explained by the approximate resolution of (9) by a gradient descent step in (14). This loss of performance of the online NOUGAT algorithm must be put into perspective, given its much lower computational cost compared to e.g. the offline dRuLSIF algorithm, see Section 4.2.2.

To get more insight in the performance of dRuLSIF, NOUGAT and MA, we shall now analyze the Mean Time to False Alarm (MTFA) and the Mean Time to Detection (MTD). Both are usual online performance measures [41]. Let t_a be the time instant of detection and t_0 the change point. They are defined as:

$$\text{MTD} = \mathsf{E}\{t_a - t_0 \,|\, t_a \geq t_0\} \tag{55}$$

$$\text{MTFA} = \mathsf{E}\{t_a \,|\, t_a < t_0\} \tag{56}$$

Figures 10 and 11 provide the MTFA and MTD as a function of the Probability of False Alarm (PFA). The PFA was computed, for each algorithm, as the probability to detect an event at a time instant t_a with $t_a < t_0$. The Probability of Detection (PD) was estimated as the probability to detect *at least* a change at a time instant t_a with $t_0 \leq t_a \leq n_t$, i.e. the probability that the test statistics is larger than the threshold at least once. Figure 12 provides the Receiver Operating Characteristic (ROC) for the three algorithms.

Figure 10 shows that, for PFA > 0.2, the MTFA for the four algorithms is smaller than 40 samples. This means that the detection thresholds are too small and make the algorithms non-operational due to numerous false alarms.

Focusing on the case PFA < 0.2, we observe in Figure 12 that when PFA $<$ 0.01, as expected from Figure 9, MA and k-NN reach the worst performance:

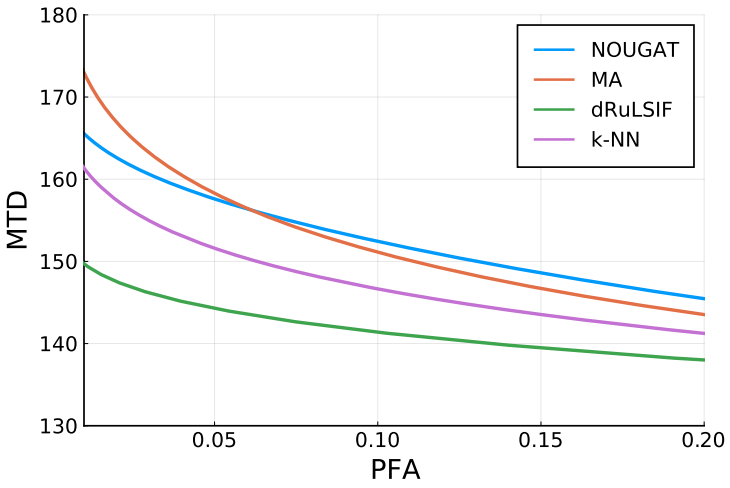

Figure 11: MTD as a function of PFA for NOUGAT, MA and dRuLSIF.

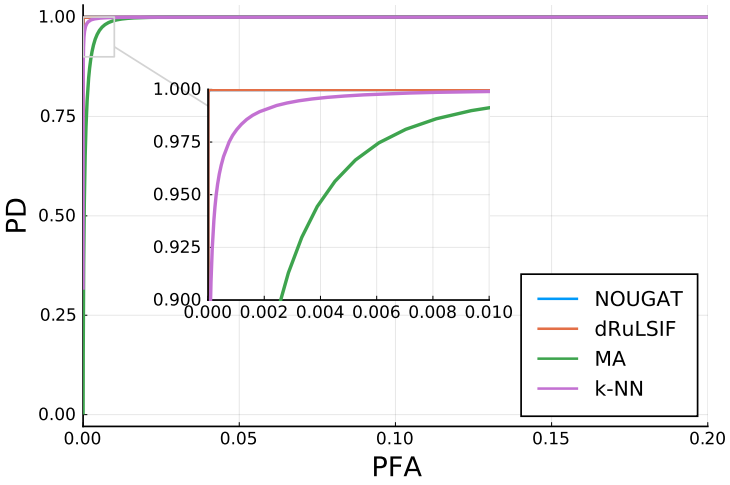

Figure 12: ROC curve for NOUGAT, MA and dRuLSIF.

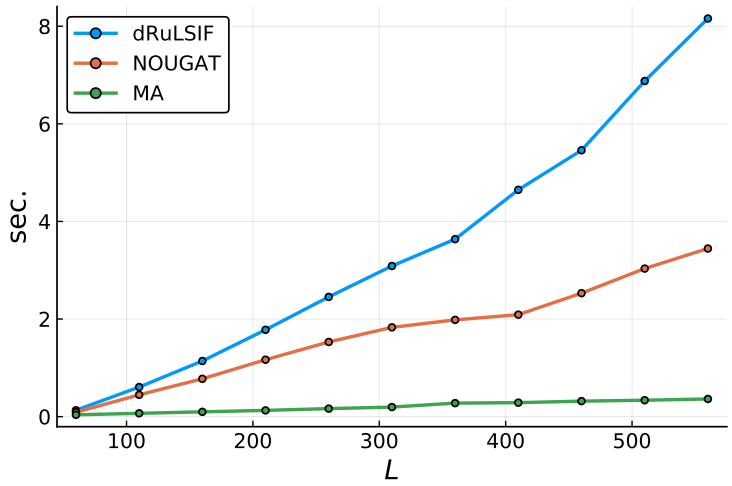

Figure 13: Run time for $n_t = 1200$ samples as a function of the dictionary size L.

for a given PFA, their PD are the smallest. We note that for $0.01 < \text{PFA} < 0.2$ the PDs of NOUGAT, dRuLSIF and k-NN are almost equal to 1.

Figure 10 shows that the MTFAs are almost the same with a larger delay of 5 samples for k-NN. Figure 11 shows that NOUGAT MTD is approximately 10 samples larger than dRuLSIF and 5 samples larger than k-NN. The delay of NOUGAT, which can be observed on Figure 9 is, as explained before, due to the online update of NOUGAT. It is worthy to note that the performance of NOUGAT depends on μ and a smaller value would result in a smaller MTD.

4.2.2. Computational cost

This experiment aims at comparing the computational cost of the four algorithms. It is worthy to note that, as long as the averages on the reference and test windows required by dRuLSIF, NOUGAT and MA are computed recursively, their computational cost does not depends on N_{ref} and N_{test}. The data dimension k only intervenes when computing $\boldsymbol{\kappa}_\omega(\cdot)$ and penalizes equally the three algorithms. On the contrary, the computational cost of k-NN strongly depends on the size of the windows, typically $O((N_{\text{ref}} + N_{\text{test}}) \log(N_{\text{ref}} + N_{\text{test}})$ using a KD tree [42].

Figures 13 depicts the run time for the three kernel-based algorithms as a function of the dictionary size L, when processing 1200 samples of dimension $k = 6$. For each value of L, the run time was calculated as the median value of 1000 runs on an Intel Core i7 3,5 GHz. Figures 13 shows that, compared to dRuLSIF, NOUGAT enjoys a considerably smaller run time while ensuring a good level of performance.

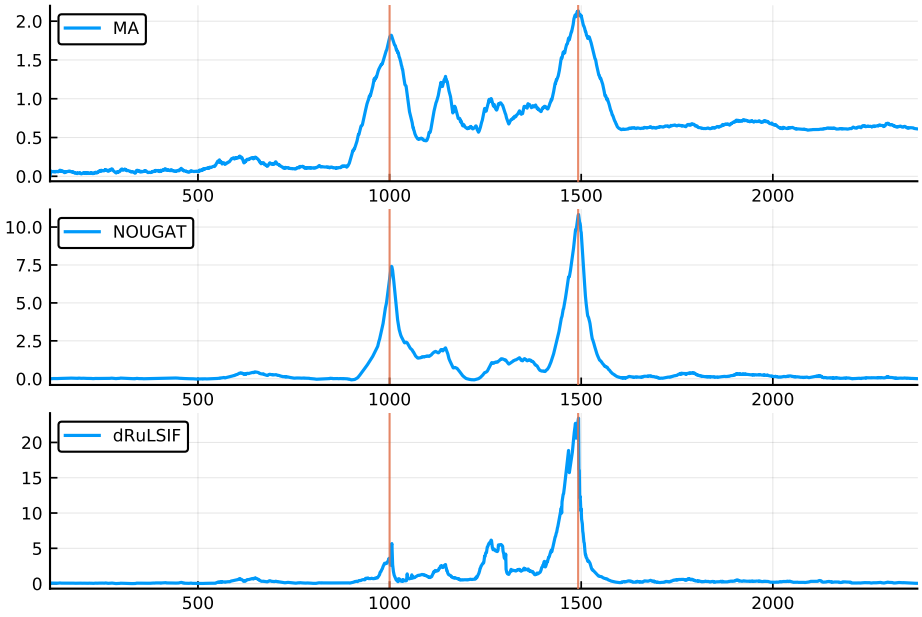

Figure 14: Credit card fraud detection. The red lines correspond to $t_0 + N_{\text{test}}$.

4.3. Experiments with real data

4.3.1. Credit card fraud detection

The data set used in this experiment, called "Credit Card Fraud Detection", contains the 28 principal components of transactions made by European cardholders in September 2013. The data set is highly unbalanced as it contains 492 frauds out of 284,807 recorded transactions; see [43] for more details. We chose to keep only 2,000 genuine transactions, and we inserted the 492 frauds in order to create two change-points at $t_0 = 1000$ and $t_0 = 1492$ in data stream $\{\boldsymbol{y}_t\}_{t\in\mathbb{N}}$. The four most significant principal components were used as inputs ($k = 4$). The Gaussian kernel with kernel bandwidth $\sigma^2 = 14$, and reference and test windows of length $N_{\text{ref}} = N_{\text{test}} = 114$, were considered for all algorithms. A regularization term with $\nu = 10^{-2}$ was used for NOUGAT and dRuLSIF, and the step size of NOUGAT was set to $\mu = 0.28$.

The online dictionary update procedure described in Section 2.3 was used for all algorithms. The coherence threshold was set to $\eta_0 = 0.7$, leading to a dictionary size of $L = 100$. Parameter vector $\boldsymbol{\theta}_{-1}$ was initially set to zero for NOUGAT.

The results provided in Figure 14 show that all the algorithms were able to detect the change-points. As expected, the detected change-points defined by the maximum value of each peak of the test statistics, were all in the vicinity of $t_0 + N_{\text{test}}$. Nevertheless, if MA was able to detect the two change-points marked by red lines in addition to some false positive detections, it suffered from a

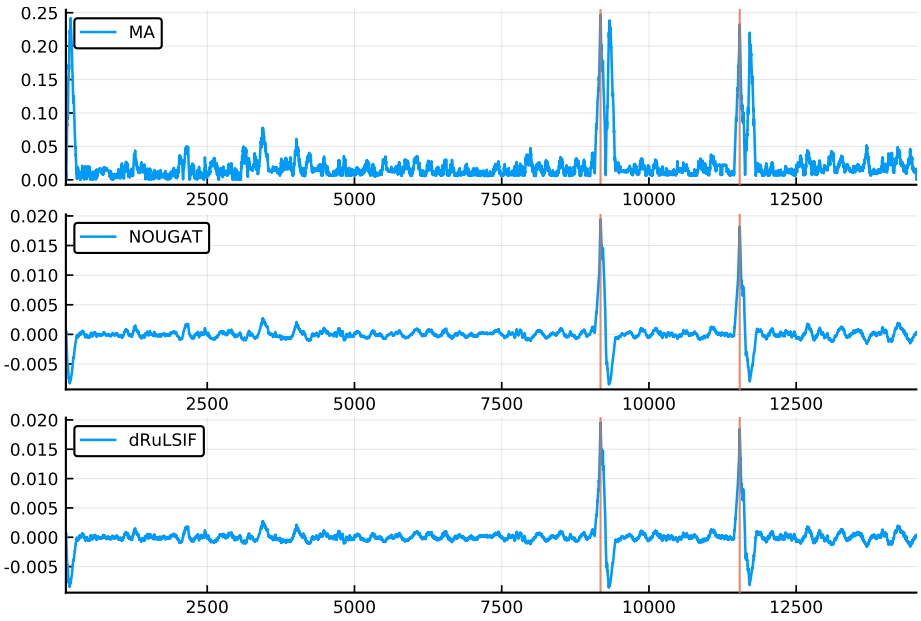

Figure 15: Sentiment change detection in Twitter data stream. The red lines correspond to $t_0 + N_{\text{Test}}$.

bias that deviated its static from zero after the first change-point. dRuLSIF hardly detected the first change-point, but successfully detected the second one. NOUGAT detected both change-points with less fluctuations of its detection statistics. Finally, NOUGAT and dRuLSIF test statistics fluctuated around 0 under the null hypothesis. These results highlight the ability of the proposed algorithm to detect consecutive change-points.

4.3.2. Sentiment change detection in Twitter data streams

The data set used in this paragraph, called "Twitter US Airline Sentiment", is available at [44]. This data set contains tweets related to US Airline in February 2015, manually tagged as positive, negative and neutral. Raw tweets were first cleaned from non-ASCII characters. Stop words from Natural Language Toolkit (NLTK) corpus were also removed. Finally, tweets were represented, using a frequency-based method, in a linear space of dimension $k = 50$. The series $\{\boldsymbol{y}_t\}_{t\in\mathbb{N}}$ was obtained by concatenating the 9178 negative-tagged tweets, the 2363 positive-tagged tweets and the 3099 neutral-tagged tweets. Parameters were set to: $\mu = 10^{-1}$, $\nu = 5.10^{-33}$, and $N_{\text{ref}} = N_{\text{test}} = 100$. A Gaussian kernel with $\sigma^2 = 1.3$ was used, along with an online dictionary learning procedure with a maximal coherence of $\eta_0 = 10^{-3}$. This resulted in a dictionary of size $L = 12$. Parameter vector $\boldsymbol{\theta}_{-1}$ was set to zero for NOUGAT.

Figure 15 provides the detection statistics of MA, NOUGAT and dRuLSIF.

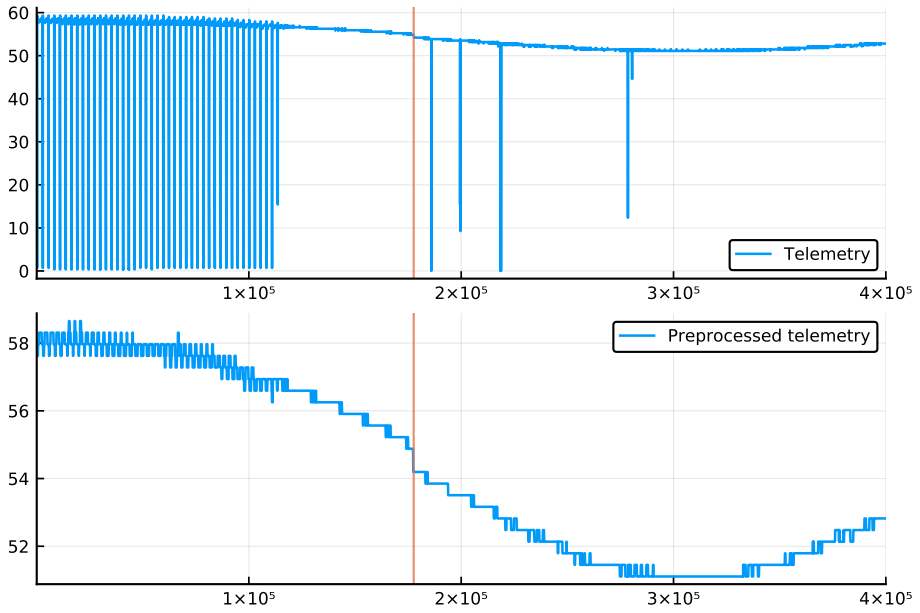

Figure 16: Top: Telemetry. Bottom: Signal pre-processed by a median filter.

MA produced 2 false alarms and the variance of its statistics was larger than the other two methods. NOUGAT and dRuLSIF led to similar results. Note that, as expected, for the three methods, the peak at the first (negative/positive) transition was slightly higher than the peak at the second (positive/neutral) transition.

4.3.3. Change detection in satellite telemetry

The data set used in this experiment was provided by Thales Alenia Space. It consists of an electrical current signal produced by a panel of a geostationary satellite. The sampling period of data points is approximately 32 seconds, and the data span a time period of six months. A change point is known to occur at time instant $t_0 = 177,630$. Marked by a red line, it represents a drop in the quantity of electrical current produced by the panel due to the loss of solar cells.

Figure 16 (top) partly shows the electrical current signal. The consecutive current drops observed at the beginning of the signal represent each a period of eclipse. These drops were removed using a median filter of length 600, which corresponds to the maximum duration of an eclipse. The filtered signal is shown in Figure 16 (bottom). Vectors $\boldsymbol{y}_t$ of dimension $k = 10$ used as inputs for the detection algorithms were extracted using a sliding window as explained in (3). Window lengths $N_{\text{ref}} = N_{\text{test}} = 3000$ were used. This value corresponds approximately to a 1-day period, which is sufficient to capture the main stationary characteristics of the signal. These characteristics depend on changes in the

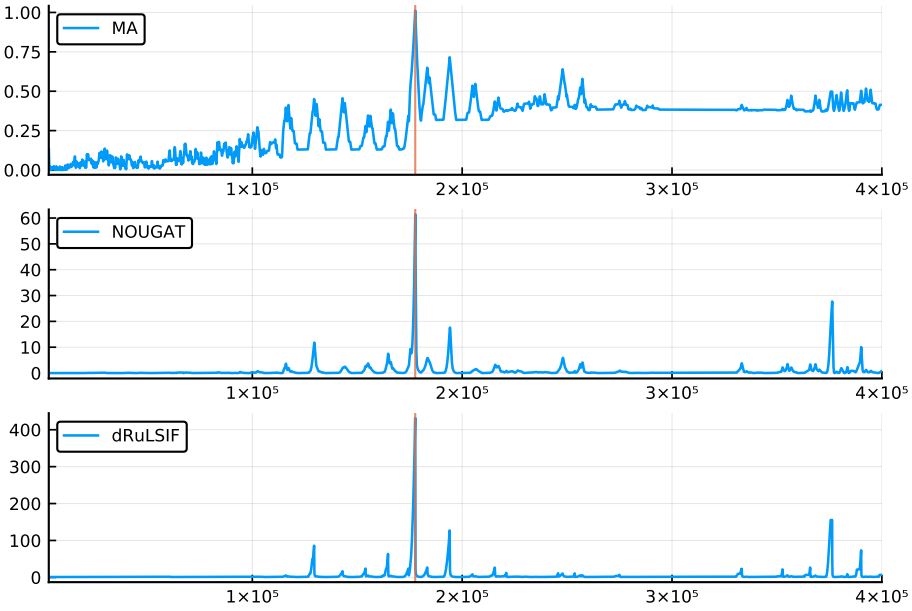

Figure 17: CPD in satellite telemetry data. The red line corresponds to $t_0 + N_{\text{Test}}$.

distance from the panels to the Sun, and the angle of incidence of the sunlight.

An online dictionary learning procedure was used with a maximal coherence value of 0.5. This resulted in a dictionary of size $L = 33$. The three algorithms produced false alarms. MA had a bias, dRuLSIF and NOUGAT showed similar results but with a much lower computational load for NOUGAT. Note that computational load is a key concern for this application.

5. Conclusion

We introduced an online kernel-based change-point detection method built upon direct estimation of the density ratio on consecutive time intervals. We analyzed its behavior in the mean and mean square sense. Finally, we evaluated its detection performance and we compared it to state-of-the-art kernel-based methodologies, MA and RuLSIF, and to another approach based on k-NN. We showed that our algorithm has a considerably lower computational complexity than dRuLSIF while ensuring comparable performance. Experiments on real-world data proved the usefulness and efficiency of our algorithm in a number of applications. These applications involved different types of data, namely, text data, raw data, and features extracted from data, showing the interest in using non-parametric techniques to perform change-point detection

We leave for future work the derivation of methods for kernel selection, and the opportunity of using a symmetric detection statistic where covariance

information on the test interval would also be considered.

Appendix A. Computation of $\boldsymbol{H}$ and $\boldsymbol{h}$

Considering the Gaussian reproducing kernel:

$$\kappa(\boldsymbol{y}, \boldsymbol{y}') = e^{-\frac{\|\boldsymbol{y}-\boldsymbol{y}'\|^2}{2\sigma^2}}$$

the entries of (23) are given by:

$$[\boldsymbol{H}]_{\ell,q} = e^{-\frac{\|\boldsymbol{y}_{\omega_\ell}\|^2+\|\boldsymbol{y}_{\omega_q}\|^2}{2\sigma^2}} \;\; \mathsf{E}_{p_{\mathrm{ref}}(\boldsymbol{y})}\left\{ e^{-\frac{\|\boldsymbol{y}\|^2-(\boldsymbol{y}_{\omega_\ell}+\boldsymbol{y}_{\omega_q})^\top \boldsymbol{y}}{\sigma^2}} \right\}$$

and those of $\boldsymbol{h}$ by:

$$[\boldsymbol{h}]_{\ell} = e^{-\frac{\|\boldsymbol{y}_{\omega_\ell}\|^2}{2\sigma^2}} \; \mathsf{E}\left\{ e^{-\frac{\|\boldsymbol{y}\|^2-2\boldsymbol{y}_{\omega_\ell}^\top \boldsymbol{y}}{2\sigma^2}} \right\}$$

with $\ell,\ q \in \{1,\ldots,L\}$. These expectations can be computed for Gaussian distributed entries $\boldsymbol{y}_i \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{R})$ using the moment generating function of a quadratic form of a Gaussian vector [45]:

$$[\boldsymbol{H}]_{\ell,q} = e^{-\frac{\|\boldsymbol{y}_{\omega_\ell}\|^2+\|\boldsymbol{y}_{\omega_q}\|^2}{2\sigma^2}} \; \Psi\big(\frac{-1}{\sigma^2}, \boldsymbol{I}, -(\boldsymbol{y}_{\omega_\ell}+\boldsymbol{y}_{\omega_q}), \boldsymbol{\mu}, \boldsymbol{R}\big)$$

$$[\boldsymbol{h}]_{\ell} = e^{-\frac{\|\boldsymbol{y}_{\omega_\ell}\|^2}{2\sigma^2}} \; \Psi\big(\frac{-1}{2\sigma^2}, \boldsymbol{I}, -2\boldsymbol{y}_{\omega_\ell}, \boldsymbol{\mu}, \boldsymbol{R}\big)$$

where:

$$\Psi(s, \boldsymbol{W}, \boldsymbol{b}, \boldsymbol{\mu}, \boldsymbol{R}) \tag{A.1}$$

$$= |\boldsymbol{I} - 2s\boldsymbol{W}\boldsymbol{R}|^{-\frac{1}{2}} \exp\Big(s\Big[(\boldsymbol{\mu}^\top \boldsymbol{W}\boldsymbol{\mu} + \boldsymbol{b}^\top \boldsymbol{\mu}) + \frac{s}{2}\|2\boldsymbol{W}\boldsymbol{\mu} + \boldsymbol{b}\|^2_{\boldsymbol{R}(\boldsymbol{I}-2s\boldsymbol{W}\boldsymbol{R})^{-1}} \Big] \Big) \tag{A.2}$$

Appendix B. Computation of $\boldsymbol{\Gamma}$ and $\boldsymbol{\Delta}$

The (r,s)-th entry of the (q,n)-th block of matrix $\boldsymbol{\Gamma}$, that is, $\mathsf{E}\{\kappa_{\omega_q}(\boldsymbol{y}_i)\kappa_{\omega_n}(\boldsymbol{y}_i)\boldsymbol{\kappa}_{\boldsymbol{\omega}}(\boldsymbol{y}_i)\boldsymbol{\kappa}_{\boldsymbol{\omega}}(\boldsymbol{y}_i)^\top\}$ is given by:

$$\boldsymbol{\Gamma}_{(q-1)L+r,(n-1)L+s} = e^{-\frac{\|\boldsymbol{y}_{\boldsymbol{\omega}_q}\|^2+\|\boldsymbol{y}_{\omega_n}\|^2+\|\boldsymbol{y}_{\omega_r}\|^2+\|\boldsymbol{y}_{\omega_s}\|^2}{2\sigma^2}}$$

$$\Psi\Big(\frac{-1}{\sigma^2},\ 2\boldsymbol{I},\ -(\boldsymbol{y}_{\omega_q}+\boldsymbol{y}_{\omega_n}+\boldsymbol{y}_{\omega_r}+\boldsymbol{y}_{\omega_s}),\ \boldsymbol{\mu},\ \boldsymbol{R}\Big) \tag{B.1}$$

Similarly, the r-th entry of the (q,n)-th block of $\boldsymbol{\Delta}$, that is, $\mathsf{E}\{\kappa_{\omega_q}(\boldsymbol{y}_i)\kappa_{\omega_n}(\boldsymbol{y}_i)\boldsymbol{\kappa}_{\boldsymbol{\omega}}(\boldsymbol{y}_i)\}$, is given by:

$$\boldsymbol{\Delta}_{(q-1)L+r,n} = e^{-\frac{\|\boldsymbol{y}_{\omega_q}\|^2+\|\boldsymbol{y}_{\omega_n}\|^2+\|\boldsymbol{y}_{\omega_r}\|^2}{2\sigma^2}}$$

$$\Psi\Big(\frac{-1}{2\sigma^2},\ 3\boldsymbol{I},\ -2(\boldsymbol{y}_{\omega_q}+\boldsymbol{y}_{\omega_n}+\boldsymbol{y}_{\omega_r}),\ \boldsymbol{\mu},\ \boldsymbol{R}\Big) \tag{B.2}$$

Acknowledgement

This work has been supported by the French government, through the 3IA Côte d'Azur Investments in the Future project managed by the National Research Agency (ANR) with the reference number ANR-19-P3IA-0002.

Ikram Bouchikhi was partly funded by Thales Alenia Space TAS.

References

[1] S. Yuan, W. Zhou, Q. Yuan, Y. Zhang, Q. Meng, Automatic seizure detection using diffusion distance and BLDA in intracranial EEG, Epilepsy & Behavior 31 (2014) 339–345.

[2] D. Gajic, Z. Djurovic, J. Gligorijevic, S. Di Gennaro, I. Savic-Gajic, Detection of epileptiform activity in EEG signals based on time-frequency and non-linear analysis, Frontiers in Computational Neuroscience 9 (2015) 38.

[3] A. L. D'Rozario, G. C. Dungan, S. Banks, P. Y. Liu, K. K. Wong, R. Killick, R. R. Grunstein, J. W. Kim, An automated algorithm to identify and reject artefacts for quantitative EEG analysis during sleep in patients with sleep-disordered breathing, Sleep and Breathing 19 (2) (2015) 607–615.

[4] R. J. Bolton, D. J. Hand, Statistical fraud detection: A review, Statistical science (2002) 235–249.

[5] A. G. Tartakovsky, B. L. Rozovskii, R. B. Blažek, H. Kim, Detection of intrusions in information systems by sequential change-point methods, Statistical Methodology 3 (3) (2006) 252–293.

[6] G. Yan, Z. Xiao, S. Eidenbenz, Catching instant messaging worms with change-point detection techniques., LEET 8 (2008) 1–10.

[7] M. A. Pimentel, D. A. Clifton, L. Clifton, L. Tarassenko, A review of novelty detection, Signal Processing 99 (2014) 215–249.

[8] C. Truong, L. Oudre, N. Vayatis, Selective review of offline change point detection methods, Signal Processing 167 (Feb. 2020).

[9] M. Basseville, I. V. Nikiforov, Detection of Abrupt Changes - Theory and Application, Prentice-Hall, 1993.

[10] C. Inclan, G. C. Tiao, Use of cumulative sums of squares for retrospective detection of changes of variance, Journal of the American Statistical Association 89 (427) (1994) 913–923.

[11] F. Gustafsson, The marginalized likelihood ratio test for detecting abrupt changes, IEEE Transactions on Automatic Control 41 (1) (1996) 66–78.

[12] Y. Kawahara, T. Yairi, K. Machida, Change-point detection in time-series data based on subspace identification, IEEE International Conference on Data Mining (2007) 559–564.

[13] N. Itoh, J. Kurths, Change-point detection of climate time series by non-parametric method, World Congress on Engineering and Computer Science 1 (2010) 445–448.

[14] V. Moskvina, A. Zhigljavsky, An algorithm based on singular spectrum analysis for change-point detection, Communications in Statistics-Simulation and Computation 32 (2) (2003) 319–352.

[15] B. Schölkopf, J. C. Platt, J. Shawe-Taylor, A. J. Smola, R. C. Williamson, Estimating the support of a high-dimensional distribution, Neural Computation 13 (7) (2001) 1443–1471.

[16] J. Ma, S. Perkins, Time-series novelty detection using one-class support vector machines, IEEE International Joint Conference on Neural Networks (2013) 1741–1745.

[17] M. Yamada, A. Kimura, F. Naya, H. Sawada, Change-point detection with feature selection in high-dimensional time-series data, International Joint Conference on Artificial Intelligence (2013) 1827–1833.

[18] H. Chen, Sequential change-point detection based on nearest neighbors, The Annals of Statistics 47 (3) (2019) 1381–1407.

[19] Y. Xie, J. Huang, R. Willett, Change-point detection for high-dimensional time series with missing data, IEEE Journal of Selected Topics in Signal Processing 7 (1) (2013) 12–27.

[20] J. Shawe-Taylor, N. Cristianini, Kernel Methods for Pattern Analysis, Cambridge University Press, New York, USA, 2004.

[21] W. Zaremba, A. Gretton, M. Blaschko, B-tests: Low variance kernel two-sample tests, Advances in Neural Information Processing Systems (NIPS) (Jan. 2013).

[22] Z. Harchaoui, E. Moulines, F. R. Bach, Kernel change-point analysis, Advances in Neural Information Processing Systems (NIPS) (2009) 609–616.

[23] N. Keriven, D. Garreau, I. Poli, NEWMA: A new method for scalable model-free online change-point detection, IEEE Transactions on Signal Processing 68 (2020) 3515–3528.

[24] S. Liu, M. Yamada, N. Collier, M. Sugiyama, Change-point detection in time series data by relative density-ratio estimation, Neural Networks 43 (2013) 72–83.

[25] I. Bouchikhi, A. Ferrari, C. Richard, A. Bourrier, M. Bernot, Non-parametric online change-point detection with kernel LMS by relative density ratio estimation, Statistical Signal Processing Workshop (SSP) (2018).

[26] W. Liu, P. P. Pokharel, J. C. Principe, The kernel least-mean-square algorithm, IEEE Transactions on Signal Processing 56 (2) (2008) 543–554.

[27] I. Bouchikhi, A. Ferrari, C. Richard, A. Bourrier, M. Bernot, Kernel based online change point detection, European Conference on Signal Processing (EUSIPCO) (2019).

[28] B. Schölkopf, R. Herbrich, A. J. Smola, R. Williamson, A generalized representer theorem, Tech. Rep. NC2-TR-2000-81, NeuroCOLT (2000).

[29] J. Rojo-Álvarez, M. Martínez-Ramón, J. Muñoz Marí, G. Camps-Valls, Digital Signal Processing with Kernel Methods, Wiley & Sons, 2017.

[30] C. Richard, J. C. M. Bermudez, P. Honeine, Online prediction of time series data with kernels, IEEE Transactions on Signal Processing 57 (3) (2009) 1058–1067.

[31] J. Chen, W. Gao, C. Richard, J. C. M. Bermudez, Convergence analysis of kernel LMS algorithm with pre-tuned dictionary, IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) (2014).

[32] W. D. Parreira, J. C. M. Bermudez, C. Richard, J. Y. Tourneret, Stochastic behavior analysis of the Gaussian kernel least-mean-square algorithm, IEEE Transactions on Signal Processing 60 (5) (2012) 2208–2222.

[33] W. Gao, J. Chen, C. Richard, J. Huang, R. Flamary, Kernel LMS algorithm with forward-backward splitting for dictionary learning, IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) (2013).

[34] Y. Engel, S. Mannor, R. Meir, The kernel recursive least-squares algorithm, IEEE Transactions on Signal Processing 52 (8) (2004) 2275–2285.

[35] S. W. Roberts, Control chart tests based on geometric moving averages, Technometrics 1 (3) (1959) 239–250.

[36] J. Minkoff, Comment on the "Unnecessary assumption of statistical independence between reference signal and filter weights in feedforward adaptive systems", IEEE Transactions on Signal Processing 49 (5) (2001) 1109.

[37] D. S. Bernstein, Matrix Mathematics: Theory, Facts, and Formulas, 2nd Edition, Princeton University Press, 2009.

[38] M. F. Schilling, Multivariate two-sample tests based on nearest neighbors, Journal of the American Statistical Association 81 (395) (1986) 799–806.

[39] N. Henze, A multivariate two-sample test based on the number of nearest neighbor type coincidences, The Annals of Statistics 16 (2) (Jun. 1988).

[40] H. Chen, Sequential change-point detection based on nearest neighbors, The Annals of Statistics 47 (3) (Jun. 2019).

[41] F. Gustafsson, Adaptive Filtering and Change Detection, Wiley, 2000.

[42] J. L. Bentley, Multidimensional binary search trees used for associative searching, Communications of the ACM 18 (1975) 509–517.

[43] Kaggle, Credit card fraud detection, `https://www.kaggle.com/mlg-ulb/creditcardfraud` (2016).

[44] Kaggle, Twitter us airline sentiment, `https://www.kaggle.com/crowdflower/twitter-airline-sentiment` (2016).

[45] J. Omura, T. Kailath, Some useful probability distributions, Tech. Rep. 7050-6, Stanford Electronics Laboratories, Stanford University (1965).