KERNEL BASED ONLINE CHANGE POINT DETECTION

Ikram Bouchikhi, André Ferrari, Cédric Richard

Université Côte d'Azur
Observatoire de la Côte d'Azur, CNRS, France

Anthony Bourrier, Marc Bernot

Thales Alenia Space
Cannes la Bocca, France

ABSTRACT

Detecting change points in time series data is a challenging problem, in particular when no prior information on the data distribution and the nature of the change is available. In a former work, we introduced an online non-parametric change-point detection framework built upon direct density ratio estimation over two consecutive time segments, rather than modeling densities separately. This algorithm based on the theory of reproducing kernels showed positive and reliable detection results for a variety of problems.To further improve the detection performance of this approach, we propose in this paper to modify the original cost function in order to achieve unbiasedness of the density ratio estimation under the null hypothesis. Theoretical analysis and numerical simulations confirm the improved behavior of this method, as well as its efficiency compared to a state of the art one. Application to sentiment change detection in Twitter data streams is also presented.

Index Terms— Non-parametric change-point detection, reproducing kernel Hilbert space, kernel least-mean-square algorithm, online learning, convergence analysis.

1. INTRODUCTION

Comparing probability distributions that underlie data in a past and present interval has been proved to be an appealing tool for change point detection (CPD). The most prominent methods derived so far from this principle are cumulative sum (CUSUM) algorithms [2]. In their simplest form, these algorithms not only assume that the parameter that undergoes the change is known, e.g., change in the mean or in the variance [7], but also require its pre- and sometimes post-change values. In cases where this information is only partially available, the generalized likelihood ratio test (GLRT) [6] can sometimes offer a practical alternative. Non-parametric CPD algorithms were introduced to handle scenarios where no prior information on the data distribution and the nature of the change is available. Recent contributions focus on modeling the density ratio over two consecutive time segments, referred to as "the importance function", rather than modeling densities separately [1, 9, 11]. Parameters of the importance function are learned from training data by minimizing a divergence such as the Kullback-Leibler or the Pearson divergence. The main characteristic of these algorithms is their non-parametric nature, which makes them suitable for real-world applications as they do not rely on strong model assumptions. In [3], the authors devise an online version of the Relative Unconstrained Least Squares Importance Fitting (RuLSIF) algorithm [11]. This algorithm operates in a reproducing kernel Hilbert space (RKHS) to deal with nonlinear models, and updates parameters in an online way with a stochastic gradient descent strategy [12, 13, 17, 18]. Convergence analysis of the online RuLSIF in [3], performed from [4, 16], gives conditions for asymptotic unbiasedeness of the model parameters, i.e., the weights of kernel expansion. Nevertheless, simulation results highlighted a model bias, i.e., approximation error, between the density ratio and its approximation in RKHS because of the usual lack of tunable static gain in kernel models. A consequence of this bias is that the density ratio estimator cannot converge toward 1 under the null hypothesis. Simulations also proved that the resulting test statistic was non Gaussian with a strong asymmetry. This makes it difficult to set a threshold to achieve a desired false alarm rate.

The aim of this communication is to introduce an alternative detection statistic to [3] which does not suffer from its drawbacks. In Section 2, we modify the original cost function in order to achieve unbiasedness of the density ratio estimation under the null hypothesis. Then we devise the new online CPD algorithm. In Section 3, we analyze its performance. In Section 4, we illustrate the improved behavior of the new CPD algorithm, as well as its efficiency compared to a state of the art one. Application to sentiment change detection in Twitter data streams is also presented.

2. PROBLEM FORMULATION

2.1. CPD algorithm

Let $\{y_t\}_{t\in\mathbb{N}}$ be a time series in which we aim at detecting whether a change occurred and, if affirmative, when it occurred. Let:

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

be a subsequence of $\{y_t\}_{t\in\mathbb{N}}$. In order to take into account any dependence that may exist between successive values of this time series, we propose to proceed as commonly reported in the literature by considering vectors $\{\boldsymbol{y}_t\}_{t\in\mathbb{N}}$ as samples. We then aim at detecting changes in the distribution of these samples by estimating a model $g(\cdot)$ for $r(\boldsymbol{y}) - 1$, where $r(\boldsymbol{y}) = p(\boldsymbol{y})/p'(\boldsymbol{y})$ is the density ratio between the probability density $p(\boldsymbol{y})$ of the data on a test interval:

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

and the probability density $p'(\boldsymbol{y})$ of the data on a reference interval:

$$\boldsymbol{Y}_t^{\text{ref}} = (\boldsymbol{y}_{t-(N_{\text{ref}}+N_{\text{test}}-1)}, , \ldots, \boldsymbol{y}_{t-N_{\text{test}}}) \in \mathbb{R}^{k\times N_{\text{ref}}} \tag{3}$$

where N_{test} and N_{ref} are the number of samples in the test and reference intervals, respectively. Note that $r(\boldsymbol{y}) - 1$ is preferred to $r(\boldsymbol{y})$ because the desired output of the detection statistic is then equal to 0 under the null hypothesis, i.e., the no change-point hypothesis.

Approximating $r(\boldsymbol{y}) - 1$ by a function $g(\cdot)$ can be performed by minimizing the mean-squared loss [21]:

$$\mathcal{C}(g) = \frac{1}{2}\mathsf{E}_{p'(\boldsymbol{y})}\{[r(\boldsymbol{y}) - 1 - g(\boldsymbol{y})]^2\} \tag{4}$$

Expanding (4) and using $r(\boldsymbol{y})p'(\boldsymbol{y}) = p(\boldsymbol{y})$ leads to:

$$\mathcal{C}(g) = \frac{1}{2}\mathsf{E}_{p'(\boldsymbol{y})}\{g^2(\boldsymbol{y})\} - \mathsf{E}_{p(\boldsymbol{y})}\{g(\boldsymbol{y})\} + \mathsf{E}_{p'(\boldsymbol{y})}\{g(\boldsymbol{y})\} + C \quad (5)$$

where C denotes a constant value. Approximating the expected values in (5) by their empirical averages over the test and reference intervals $\boldsymbol{Y}_t^{\text{test}}$ and $\boldsymbol{Y}_t^{\text{ref}}$ in (2)–(3) for any fixed t leads to the following optimization problem:

$$\min_{g\in\mathcal{G}} \left(\frac{1}{2N_{\text{ref}}} \sum_{i=t-(N_{\text{ref}}+N_{\text{test}}-1)}^{t-N_{\text{test}}} g^2(\boldsymbol{y}_i) - \frac{1}{N_{\text{test}}} \sum_{i=t-(N_{\text{test}}-1)}^{t} g(\boldsymbol{y}_i) + \frac{1}{N_{\text{ref}}} \sum_{i=t-(N_{\text{ref}}+N_{\text{test}}-1)}^{t-N_{\text{test}}} g(\boldsymbol{y}_i) + \lambda\Omega(\|g\|_{\mathcal{G}}) \right) \quad (6)$$

where $\mathcal{G}$ denotes an arbitrary reproducing kernel Hilbert space of real-valued functions on $\mathbb{R}$. Let $\kappa(\cdot\,,\cdot)$ be the reproducing kernel of $\mathcal{G}$. The term $\lambda\Omega(\|g\|_{\mathcal{G}})$ with $\lambda \geq 0$ is a regularization term added to promote smoothness of the solution. According to the Representer Theorem [20], the function $g(\cdot)$ that minimizes the regularized empirical loss in (6) can be expressed as follows:

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

where $\boldsymbol{\theta}$ is a parameter vector to learn. This model cannot be trained efficiently from streaming data as it needs to update both $\boldsymbol{\theta}$ and $\{\boldsymbol{y}_i\}$ as time t progresses. A standard strategy in the literature is to substitute $\{\boldsymbol{y}_i\}$ in (7) by a fixed dictionary $\{\boldsymbol{y}_{\omega_i}\}_{i=1}^{L}$ of size L. Several strategies were proposed to design this dictionary from input data [19]. They mainly consist of building it sequentially, by inserting selected samples $\boldsymbol{y}_i$ that improve the representation of input data according to a criterion, e.g., coherence [18], approximate linear dependence [5], or novelty [13]. To make the theoretical analysis of the algorithm tractable, this paper focuses on the *pre-tuned* dictionary case where the dictionary is fixed and assumed to be available:

$$g(\cdot\,,\boldsymbol{\theta}) = \sum_{i=1}^{L} \theta_i\, \kappa_{\omega_i}(\cdot) = \boldsymbol{\theta}^\top \boldsymbol{\kappa}_{\boldsymbol{\omega}}(\cdot) \quad (8)$$

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

2.2. Mini-batch and optimal solution

Substituting (8) into (6), assuming a ridge parameter space regularization [20], and minimizing w.r.t. $\boldsymbol{\theta}$, we find that $\hat{\boldsymbol{\theta}}_t$ is the solution of the strictly convex quadratic optimization problem:

$$\hat{\boldsymbol{\theta}}_t = \arg\min_{\boldsymbol{\theta}\in\mathbb{R}^L} J_t(\boldsymbol{\theta}) \quad \text{with} \quad J_t(\boldsymbol{\theta}) = \frac{1}{2}\boldsymbol{\theta}^\top \boldsymbol{H}_t^{\text{ref}}\,\boldsymbol{\theta} - \boldsymbol{\theta}^\top \boldsymbol{h}_t^{\text{test}} + \boldsymbol{\theta}^\top \boldsymbol{h}_t^{\text{ref}} + \frac{\lambda}{2}\|\boldsymbol{\theta}\|^2 \quad (9)$$

where:

$$\boldsymbol{h}_t^{\text{test}} = \frac{1}{N_{\text{test}}} \sum_{i=t-(N_{\text{test}}-1)}^{t} \boldsymbol{\kappa}_{\boldsymbol{\omega}}(\boldsymbol{y}_i) \quad (10)$$

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

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

Let us denote by $\boldsymbol{\theta}^*$ the optimal parameter vector that minimizes the regularized cost function (5). Following the same steps, $\boldsymbol{\theta}^*$ is the solution of $\arg\min_{\boldsymbol{\theta}\in\mathbb{R}^L} \mathsf{E}\{J_t(\boldsymbol{\theta})\}$, namely,

$$(\boldsymbol{H}^{\text{ref}} + \lambda\boldsymbol{I})\boldsymbol{\theta}^* = \boldsymbol{h}^{\text{test}} - \boldsymbol{h}^{\text{ref}} \quad (13)$$

where:

$$\boldsymbol{h}^{\text{test}} = \mathsf{E}_{p(\boldsymbol{y})}\{\boldsymbol{\kappa}_{\boldsymbol{\omega}}(\boldsymbol{y})\} \quad (14)$$

$$\boldsymbol{h}^{\text{ref}} = \mathsf{E}_{p'(\boldsymbol{y})}\{\boldsymbol{\kappa}_{\boldsymbol{\omega}}(\boldsymbol{y})\} \quad (15)$$

$$\boldsymbol{H}^{\text{ref}} = \mathsf{E}_{p'(\boldsymbol{y})}\{\boldsymbol{\kappa}_{\boldsymbol{\omega}}(\boldsymbol{y})\,\boldsymbol{\kappa}_{\boldsymbol{\omega}}^\top(\boldsymbol{y})\} \quad (16)$$

Under the null hypothesis, $p(\cdot) = p'(\cdot)$, meaning that $\boldsymbol{h}^{\text{test}} = \boldsymbol{h}^{\text{ref}}$. This leads to:

$$(\boldsymbol{H} + \lambda\boldsymbol{I})\boldsymbol{\theta}^* = \boldsymbol{0} \quad (17)$$

where we dropped the superscript of $\boldsymbol{H}^{\text{ref}}$ for simplicity. We conclude that $\boldsymbol{\theta}^* = \boldsymbol{0}$ since $\boldsymbol{H} + \lambda\boldsymbol{I}$ is a full rank matrix. This shows that, unlike [3], under the null hypothesis we have $g(\cdot\,,\boldsymbol{\theta}^*) = 0$. The estimation of the density ratio $r(\cdot)$ is unbiased, i.e., equal to 1.

2.3. Online weights update and test statistic

The KLMS was originally designed for solving kernel-based problems of the form (9) in an online manner [18]. Let $\boldsymbol{\theta}_t$ denote the estimate of the minimizer at time t. The algorithm consists of the following stochastic gradient descent step:

$$\begin{aligned}\boldsymbol{\theta}_{t+1} &= \boldsymbol{\theta}_t - \mu\hat{\nabla}J_{t+1}(\boldsymbol{\theta}_t) \\ &= \boldsymbol{\theta}_t - \mu\big[(\boldsymbol{H}_{t+1}^{\text{ref}} + \lambda\boldsymbol{I})\boldsymbol{\theta}_t - (\boldsymbol{h}_{t+1}^{\text{test}} - \boldsymbol{h}_{t+1}^{\text{ref}})\big]\end{aligned} \quad (18)$$

where $\mu > 0$ is the step-size parameter, and $\hat{\nabla}J_{t+1}(\boldsymbol{\theta}_t)$ denotes an instantaneous estimate of the gradient of $J_{t+1}(\cdot)$ evaluated at $\boldsymbol{\theta}_t$. Equation (18) is an update rule with fixed dictionary. In practice, the dictionary can be learned in an online manner. In that case, the update equation (18) has to be modified as described in [3].

We propose, as a test statistic, to consider the density ratio estimator computed at the sample just after the reference and test estimation intervals, to avoid potential bias due to correlation in the data. This means that we consider:

$$g(\boldsymbol{y}_{t+1}) = \boldsymbol{\theta}_t^\top \boldsymbol{\kappa}_{\omega}(\boldsymbol{y}_{t+1}) \quad (19)$$

where $\boldsymbol{\theta}_t$ is updated according to (18) at time t.

3. CONVERGENCE ANALYSIS

We shall now analyze the behavior of the algorithm under the null hypothesis for i.i.d Gaussian input data $\boldsymbol{y}_t$. We shall assume that the dictionary has been preset, i.e., $\{\boldsymbol{y}_{\omega_i}\}_{i=1}^{L}$ are deterministic.

To perform the analysis, we introduce the Modified Independence Assumption (MIA) [14] which suggests that $\boldsymbol{H}_{t+1}^{\text{ref}}$ is statistically independent of $\boldsymbol{\theta}_t$. Although not true in general, this assumption is commonly used for analyzing adaptive constructions as it allows to simplify the derivation without constraining the conclusions.

3.1. Mean analysis

Using the update rule (18), we obtain the following recursion for $\boldsymbol{\theta}_t$:

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

with $\boldsymbol{e}_t^{\circ} = -(\boldsymbol{h}_{t+1}^{\text{test}} - \boldsymbol{h}_{t+1}^{\text{ref}})$. Taking the expectation of both sides of (20), using $\mathsf{E}\{\boldsymbol{e}_t^{\circ}\} = 0$ and the MIA we get the mean weight model:

$$\mathsf{E}\{\boldsymbol{\theta}_{t+1}\} = \big[\boldsymbol{I} - \mu(\boldsymbol{H} + \lambda \boldsymbol{I})\big]\mathsf{E}\{\boldsymbol{\theta}_t\} \tag{21}$$

Then, for any initial condition, algorithm (18) asymptotically converges in the mean if the step-size μ is chosen to satisfy:

$$\mu < \frac{2}{\zeta_{\max}\{\boldsymbol{H} + \lambda \boldsymbol{I}\}} \tag{22}$$

where $\zeta_{\max}\{\cdot\}$ stands for the maximal eigenvalue of its matrix argument. In this case, (21) implies $\lim_{t\to\infty} \mathsf{E}\{\boldsymbol{\theta}_t\} = \mathbf{0} = \boldsymbol{\theta}^*$. This means that, unlike [3], $\boldsymbol{\theta}_t$ is asymptotically unbiased.

Taking the expectation of (19) and using the MIA, we obtain the mean of the test statistics $g(\boldsymbol{y}_{t+1})$:

$$\mathsf{E}\{g(\boldsymbol{y}_{t+1})\} = \boldsymbol{h}^{\top}\mathsf{E}\{\boldsymbol{\theta}_t\} \tag{23}$$

Assuming (22) holds, under the null hypothesis, the asymptotic unbiasedness of $\boldsymbol{\theta}_t$ implies $\lim_{t\to\infty} \mathsf{E}\{g(\boldsymbol{y}_{t+1})\} = 0 = 1 - r(\boldsymbol{y}_t)$. That is, asymptotic estimation of the density ratio $r(\boldsymbol{y}_t)$ is unbiased, which is an advantage of the proposed algorithm compared to [3]. Initializing (18) with $\boldsymbol{\theta}_0 = \mathbf{0}$, namely, $\mathsf{E}\{\boldsymbol{\theta}_0\} = \mathbf{0}$, note that (21) implies $\mathsf{E}\{\boldsymbol{\theta}_t\} = \mathbf{0}$ for all t. Consequently $\mathsf{E}\{g(\boldsymbol{y}_t)\} = 0$, meaning that the estimation of the density ratio $r(\boldsymbol{y}_t) = 1$ is unbiased under the null hypothesis for all t, without initial transient phase.

3.2. Computation of $\boldsymbol{H}$ and $\boldsymbol{h}$

To proceed further with the analysis, we now need to specify a reproducing kernel. We shall consider the Gaussian kernel in the sequel:

$$\kappa(\boldsymbol{y}, \boldsymbol{y}') = e^{-\frac{\|\boldsymbol{y}-\boldsymbol{y}'\|^2}{2\sigma^2}} \tag{24}$$

where σ denotes the kernel bandwidth. The entries of (15) and (16) can be computed for Gaussian distributed entries $\boldsymbol{y}_i \sim \mathcal{N}(\boldsymbol{m}, \boldsymbol{R})$ using the moment generating function of a quadratic form of a Gaussian vector [15]:

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

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

with $\ell, q \in \{1, \ldots, L\}$, and

$$\Psi(s, \boldsymbol{W}, \boldsymbol{b}, \boldsymbol{m}, \boldsymbol{R}) = |\boldsymbol{I} - 2s\boldsymbol{W}\boldsymbol{R}|^{-\frac{1}{2}} \exp\Big(s\Big[(\boldsymbol{m}^{\top}\boldsymbol{W}\boldsymbol{m} + \boldsymbol{b}^{\top}\boldsymbol{m}) + \frac{s}{2}\|2\boldsymbol{W}\boldsymbol{m} + \boldsymbol{b}\|^2_{\boldsymbol{R}(\boldsymbol{I}-2s\boldsymbol{W}\boldsymbol{R})^{-1}}\Big]\Big)$$

3.3. Mean squared analysis

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

$$\boldsymbol{C}_{\boldsymbol{\theta},t} = \mathsf{E}\{\boldsymbol{\theta}_t\boldsymbol{\theta}_t^{\top}\} \tag{25}$$

To estimate the variance of the test statistics, we need to calculate $\boldsymbol{C}_{\boldsymbol{\theta},t}$. Post-multiplying (20) by its transpose, taking the expectation, and using the MIA, we obtain the recursive expression:

$$\boldsymbol{C}_{\boldsymbol{\theta},t+1} = (1-\mu\lambda)^2\boldsymbol{C}_{\theta,t} - \mu(1-\mu\lambda)(\boldsymbol{H}\boldsymbol{C}_{\boldsymbol{\theta},t} + \boldsymbol{C}_{\boldsymbol{\theta},t}\boldsymbol{H}) + \mu^2(\boldsymbol{T} + \boldsymbol{Q}) + \mu^2(\boldsymbol{Z} + \boldsymbol{Z}^{\top}) - \mu(1-\mu\lambda)(\boldsymbol{N} + \boldsymbol{N}^{\top}) \tag{26}$$

where:

$$\boldsymbol{T} = \mathsf{E}\{\boldsymbol{H}_{t+1}^{\text{ref}}\boldsymbol{\theta}_t\boldsymbol{\theta}_t^{\top}\boldsymbol{H}_{t+1}^{\text{ref}}\} \tag{27}$$

$$\boldsymbol{Q} = \mathsf{E}\{\boldsymbol{e}_t^{\circ}\boldsymbol{e}_t^{\circ\top}\} \tag{28}$$

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

$$\boldsymbol{N} = \mathsf{E}\{\boldsymbol{e}_t^{\circ}\boldsymbol{\theta}_t^{\top}\} = \mathbf{0} \text{ by MIA} \tag{30}$$

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

$$\boldsymbol{T} = \frac{1}{N_{\text{ref}}}\text{vec}^{-1}(\boldsymbol{\Gamma}\boldsymbol{c}_{\boldsymbol{\theta},t}) + \frac{N_{\text{ref}}-1}{N_{\text{ref}}}\boldsymbol{H}\boldsymbol{C}_{\boldsymbol{\theta},t}\boldsymbol{H} \tag{31}$$

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

$$\boldsymbol{\Gamma} = \mathsf{E}\{\boldsymbol{\kappa}_{\boldsymbol{\omega}}(\boldsymbol{y}_i)\boldsymbol{\kappa}_{\boldsymbol{\omega}}(\boldsymbol{y}_i)^{\top} \otimes \boldsymbol{\kappa}_{\boldsymbol{\omega}}(\boldsymbol{y}_i)\boldsymbol{\kappa}_{\boldsymbol{\omega}}(\boldsymbol{y}_i)^{\top}\} \tag{32}$$

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

$$\boldsymbol{Q} = \frac{N_{\text{ref}} + N_{\text{test}}}{N_{\text{ref}}\ N_{\text{test}}}\ (\boldsymbol{H} - \boldsymbol{h}\boldsymbol{h}^{\top}) \tag{33}$$

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

$$\boldsymbol{Z} = \frac{1}{N_{\text{ref}}}\Big(\text{vec}^{-1}(\boldsymbol{\Delta}\boldsymbol{b}_{\boldsymbol{\theta},t}) - \boldsymbol{h}\boldsymbol{b}_{\boldsymbol{\theta},t}^{\top}\boldsymbol{H}\Big) \tag{34}$$

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

$$\boldsymbol{\Delta} = \mathsf{E}\{\boldsymbol{\kappa}_{\boldsymbol{\omega}}(\boldsymbol{y})\boldsymbol{\kappa}_{\boldsymbol{\omega}}(\boldsymbol{y})^{\top} \otimes \boldsymbol{\kappa}_{\boldsymbol{\omega}}(\boldsymbol{y})\} \tag{35}$$

and $\boldsymbol{b}_{\boldsymbol{\theta},t} = \mathsf{E}\{\boldsymbol{\theta}_t\}$.
The variance of the test statistics $g(\boldsymbol{y}_t)$ can now be calculated using these results. In particular:

$$\mathsf{E}\{g(\boldsymbol{y}_{t+1})^2\} = \mathsf{E}\{(\boldsymbol{\kappa}_{\omega}(\boldsymbol{y}_{t+1})^{\top}\boldsymbol{\theta}_t)^2\} \tag{36}$$

$$= \text{tr}(\boldsymbol{H}\boldsymbol{C}_{\boldsymbol{\theta},t}) \tag{37}$$

Neglecting the bias terms $\boldsymbol{b}_{\boldsymbol{\theta},t}$ in (26) and using standard results on Kronecker products, vectorizing (26) leads to:

$$\boldsymbol{c}_{\boldsymbol{\theta},t+1} = \boldsymbol{S}\boldsymbol{c}_{\boldsymbol{\theta},t} + \mu^2\text{vec}(\boldsymbol{Q}) \tag{38}$$

with:

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

where $\boldsymbol{H} \oplus \boldsymbol{I} = \boldsymbol{H} \otimes \boldsymbol{I} + \boldsymbol{I} \otimes \boldsymbol{H}$. 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:

$$\boldsymbol{c}_{\boldsymbol{\theta},\infty} = \mu^2(\boldsymbol{I} - \boldsymbol{S})^{-1}\text{vec}(\boldsymbol{Q}) \tag{39}$$

The variance of the test statistics follows from this result via (36).

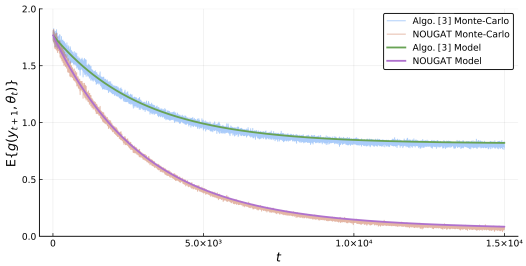

Fig. 1. Mean of the detection statistics: Monte Carlo and model.

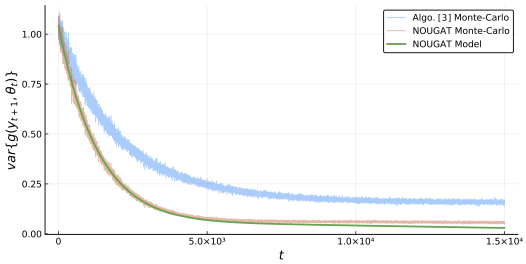

Fig. 2. Variance of the detection statistics: Monte Carlo and model.

4. EXPERIMENTS

4.1. Monte Carlo validation

This section validates the models derived in Section 3 by comparing them to Monte Carlo simulations. In these experiments, the observations $\boldsymbol{y}_n$ were two-dimensional i.i.d. zero-mean Gaussian vectors, with standard deviation and correlation equal to 0.25. The kernel bandwidth of the Gaussian kernel was set to $\sigma = 0.25$. A dictionary with $L = 16$ elements was designed randomly by sampling the same distribution as $\boldsymbol{y}_n$. The regularization parameter λ was set to 10^{-3}. The step-size μ was set to 10^{-2}. The lengths of the reference and test windows were set to $N_{\text{ref}} = N_{\text{test}} = 250$.

Figure 1 compares the behavior over time of the means of the detection statistics, $\mathsf{E}\{g(\boldsymbol{y}_t)\}$, of the proposed algorithm (18) and [3]. The new algorithm will be denoted as NOUGAT (Nonparametric Online chanGepoint AlgoriThm) in the sequel. Both algorithms were initialized with a vector $\boldsymbol{\theta}_0$ of L ones. We observe in Fig. 1 that the theoretical curves match well the actual performance of both algorithms. For NOUGAT, $\mathsf{E}\{g(\boldsymbol{y}_t)\}$ converged to 0 as expected, while Algorithm [3] was affected by a bias and converged to ≈ 0.81. This first result illustrates the difficulty in setting a threshold for CPD when using [3], unlike NOUGAT which provides an unbiased estimation of the density ratio. Figure 2 shows the variance of the detection statistics for NOUGAT, evaluated with Monte Carlo simulations and theoretically with (36), and the variance of [3] evaluated with Monte Carlo simulations. Figure 2 confirms that the theoretical curve matches well the actual performance. This result also shows that the variance of NOUGAT is lower than [3], which implies a lower false alarm rate. Finally, Fig. 3 compares the histograms of the detection statistics of the proposed algorithm and [3]. The histogram on the left exhibits a single mode centered at 1 with a strong asymmetry. On the contrary, the histogram of NOUGAT on the right shows a normal shape centered, as expected, around 0. The orange curve corresponds to a zero-mean Gaussian distribution with the variance of the samples. This simulation corroborates the assumption of a zero-mean normal distribution for $g(\boldsymbol{y}_{t+1})$. This property is important for setting a threshold that may guarantee a given false alarm rate.

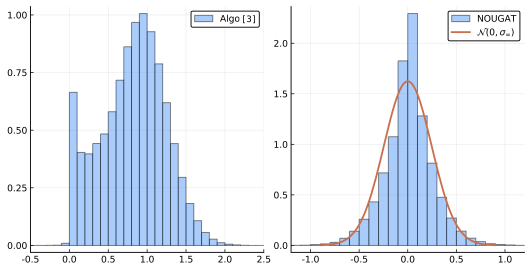

Fig. 3. Histogram of the detec. statistics. Left: [3], right: NOUGAT.

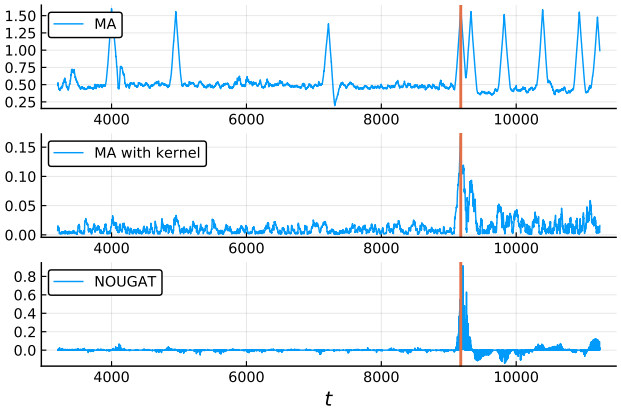

Fig. 4. Detection of sentiment change in Twitter data stream.

4.2. Application to detection of sentiment change

NOUGAT was compared to a a non-parametric CPD method of the literature, MA [10], that compares an empirical average over an interval of N_{test} samples with one computed over an interval of N_{ref} samples that comes before. Both algorithms were used to detect a change of opinion in a stream of tweets. The data set, called "Twitter US Airline Sentiment", available on Kaggle [8], contains tweets related to the US Airline in February 2015 manually tagged as positive, negative and neutral opinions. Raw tweets were first cleaned from non-ASCII characters. Stop words from Natural Language Toolkit corpus were also removed. Finally, tweets were tokenized into vectors of size $k = 11251$. The data stream $\boldsymbol{y}_t$ was obtained by concatenating the first 9178 positive tweets and the 2363 negative tweets. The dictionary consisted of the first $L = 100$ positive tweets. The algorithm parameters were set as follows: $\mu = 10^{-1}$, $\lambda = 10^{-2}$, $\sigma^2 = 1.3$, $N_{\text{ref}} = N_{\text{test}} = 100$ and $\boldsymbol{\theta}_0 = \mathbf{0}$.
Window lengths were set to $N_{\text{ref}} = N_{\text{test}} = 100$ for MA. For fair comparison, kernelized data $\boldsymbol{\kappa}_{\boldsymbol{\omega}}(\boldsymbol{y}_t)$ were fed to MA algorithm. Our own derivations showed that the CPD test performed by this algorithm, called MA with kernels, reduces to compare $\|\boldsymbol{h}_{t+1}^{\text{test}} - \boldsymbol{h}_{t+1}^{\text{ref}}\|^2$ to a threshold. This means that MA with kernels does not use covariance information $\boldsymbol{H}_{t+1}^{\text{ref}}$ to make a decision.

Figure 4 presents the detection statistics of MA, MA with kernels, and NOUGAT. These results show that all algorithms detected the change point at $t = 9178$, but MA produced 8 false alarms whereas NOUGAT did not. MA algorithm with kernels performed better than MA, but the variance of its detection statistics was significantly larger than NOUGAT. This experiment also confirms the unbiasedness of the density ratio estimation provided by NOUGAT under the null hypothesis.

5. REFERENCES

[1] S. Aminikhanghahi, T. Wang, and D. J. Cook. Real-time change point detection with application to smart home time series data. *IEEE Transactions on Knowledge and Data Engineering*, 2018 (Early Access).

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

[3] I. Bouchikhi, A. Ferrari, C. Richard, A. Bourrier, and M. Bernot. Non-parametric online change-point detection with kernel LMS by relative density ratio estimation. In *Proc. IEEE Statistical Signal Processing Workshop (IEEE SSP)*, pages 1–5, 2018.

[4] J. Chen, W. Gao, C. Richard, and J. C. M. Bermudez. Convergence analysis of kernel LMS algorithm with pre-tuned dictionary. In *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (IEEE ICASSP)*, pages 1–5, 2014.

[5] Y. Engel, S. Mannor, and R. Meier. The kernel recursive least squares algorithm. *IEEE Transactions on Signal Processing*, 52:2275–2285, 2004.

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

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

[8] Kaggle. Twitter US Airline Sentiment, 2016.

[9] Y. Kawahara and M. Sugiyama. Sequential change-point detection based on direct density-ratio estimation. *Statistical Analysis and Data Mining*, 5(2):114 – 127, 2012.

[10] D. Kifer, S. Ben-David, and J. Gehrke. Detecting change in data streams. In *Proc. International Conference on Very Large Data Bases*, pages 180–191. VLDB Endowment, 2004.

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

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

[13] W. Liu, J. Príncipe, and S. Haykin. *Kernel adaptive filtering: A comprehensive introduction*. Wiley, New Jersey, 2010.

[14] 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):1109, 2001.

[15] J. Omura and T. Kailath. Some useful probability distributions. Technical Report 7050 - 6, Stanford Electronics Laboratories, Stanford University, 1965.

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

[17] C. Richard. Filtrage adaptatif non-linéaire par méthodes de gradient stochastique court-terme à noyau. In *Actes du 20e Colloque GRETSI sur le Traitement du Signal et des Images*, pages 53–56, 2005.

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

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

[20] B. Schölkopf, R. Herbrich, A. J. Smola, and R. Williamson. A generalized representer theorem. Technical Report NC2-TR-2000-81, NeuroCOLT, 2000.

[21] M. Yamada, T. Suzuki, T. Kanamori, H. Hachiya, and M. Sugiyama. Relative density-ratio estimation for robust distribution comparison. *Neural Computation*, 25(5):1324 – 1370, 2013.