Greedy Kernel Change-Point Detection

Charles Truong, Laurent Oudre, and Nicolas Vayatis

***Abstract*—We consider the problem of detecting abrupt changes in the underlying stochastic structure of multivariate signals. A novel non-parametric and model-free off-line change-point detection method based on a kernel mapping is presented. This approach is sequential and alternates between two steps: a greedy detection to estimate a new breakpoint and a projection to remove its contribution to the signal. The resulting algorithm is able to segment time series for which no accurate model is available: it is computationally more efficient than exact kernel change-point detection and more precise than window-based approximations. The proposed method also offers some theoretical consistency properties. For the special case of a linear kernel, an even faster implementation is provided. The proposed strategy is compared to standard parametric and non-parametric procedures on a real-world data set composed of 262 accelerometer recordings.**

***Index Terms*—Change-point detection, greedy algorithms, Kernel methods.**

I. Introduction

With the explosion in sensor use and monitoring technology, numerous complex systems (for instance, industrial systems, stock exchanges, the human body) can be monitored for long periods of time. Often, those systems switch between states while being monitored. As a result, collected signals are not stationary but instead see one or several of their characteristics abruptly change at unknown instants. Such characteristics include the mean [1], the variance [2], the periodicity [3], the probability distribution of the signal samples [4], etc. Any subsequent analysis may have to rely on overly complex models, to take into account the time-varying nature of the signal characteristics. In order to avoid this pitfall, a common and efficient approach is to describe those multivariate time-series as a succession of non-overlapping segments, each one corresponding to a simple model. Therefore, a crucial pre-processing step is change-point detection (or signal segmentation) which consists in estimating the time stamps at which the characteristics of the signal change. This has enormous practical advantages in a broad range of real-world scenarios: in finance [5], [6], biomedical data [7]–[12], meteorology [13]–[16], DNA array analysis [17]–[20], etc. This work focuses on the off-line (or retrospective) setting, where change-point detection is performed a posteriori, on the complete signal. Conversely, the so-called on-line setting, in which the signal samples are assumed to be revealed progressively, was originally introduced for real-time signal analysis and is beyond our scope. Signals are assumed to be multivariate and contain multiple change-points of unknown location and amplitude.

There is a rich literature associated with change-point detection, dating back to the 50s [21]. Historically, the first type of change to be considered was a single change in the mean of a noisy univariate signal [1], [21]. This model still receives significant attention, from a theoretical standpoint, under various assumptions on the noise [3], [22]–[25] and the number of changes [26]–[28], and from an algorithmic standpoint [29], [30]. In order to detect more general types of change, likelihood-based methods were then introduced, such as the generalized likelihood ratio [31]–[35]. Under this setting, samples between two change-points are assumed to be identically distributed, following a user-defined parametric distribution. Parametric approaches also include Bayesian methods, which have state-of-the-art performances in several applications [36]–[38]. See [2], [39], [40] for reviews of parametric change-point detection procedures.

In case the signal cannot be efficiently described by a parametric model, non-parametric methods are used. Kernel change-point detection has emerged in this context and amounts to minimizing a kernel least square criterion. This approach enjoys attractive theoretical [41], [42] and computational [43]–[45] properties, which has motivated numerous developments from the machine learning community [46], [47]. In particular, changes in higher-order moments (above the first two) of probability distributions [41], [48] can be detected, with little to no calibration. First introduced by [4] in the context of change detection, kernel approaches have been applied on very different types of signals, for audio and video segmentation [49], [50], DNA arrays [43], [51], neurological signals [4], [44], [45], etc. A complete review of non-kernel and kernel based change point detection methods as well as Python implementations of the main approaches can be found in [52].

A. Problem Formulation

1) Statistical Model: We consider the segmentation of an $\mathbb{R}^d$-valued signal $\{x_t\}_{t=1}^T$ which, once mapped onto a high-dimensional space, namely a reproducing kernel Hilbert space (RKHS), is piecewise constant with additive noise. The objective is to locate changes in the mean of the mapped signal. Formally, let $k(\cdot,\cdot) : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ denote a kernel function

Manuscript received February 4, 2019; revised August 22, 2019; accepted November 12, 2019. Date of publication November 20, 2019; date of current version December 3, 2019. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Paolo Braca. This work was supported by a public grant as part of the Investissement d'avenir project, reference ANR-11-LABX-0056-LMH and LabEx LMH. *(Corresponding author: Charles Truong.)*

C. Truong and N. Vayatis are with the CMLA, Ecole Normale Superieure de Cachan, Cachan 94230, France (e-mail: charles@doffy.net; vayatis@cmla.ens-cachan.fr).

L. Oudre is with the L2TI, Universite Paris 13, 93430 Villetaneuse, France (e-mail: laurent.oudre@univ-paris13.fr).

This paper has supplementary downloadable material available at http://ieeexplore.ieee.org, provided by the authors.

Digital Object Identifier 10.1109/TSP.2019.2953670

and $\mathcal{H}$, the associated RKHS. The related mapping function $\Phi : \mathbb{R}^d \to \mathcal{H}$ is implicitly defined by $\Phi(x_t) = k(x_t, \cdot) \in \mathcal{H}$ and $\langle \Phi(x_s) | \Phi(x_t) \rangle_{\mathcal{H}} = k(x_s, x_t)$. The RKHS norm $\| \cdot \|_{\mathcal{H}}$ is also implicitly defined by $\|\Phi(x_t)\|^2_{\mathcal{H}} = k(x_t, x_t)$. We assume that the mapped signal $y \in \mathcal{H}^T$, defined by $y_t := \Phi(x_t)$, is such that

$$\forall t \in \{t^\star_k + 1, \dots, t^\star_{k+1}\}, \quad y_t = u_t + \varepsilon_t, \tag{1}$$

where u_t is a $\mathcal{H}$-valued deterministic piecewise constant signal with $K^\star$ change-points and ε_t is a $\mathcal{H}$-valued additive noise with zero-mean. (Additional technical assumptions on u_t and ε_t are described later.) Let $\mathcal{T}^\star := \{t^\star_k | k = 1, \dots, K^\star\}$ denote the set of change-point indexes ($t^\star_1 < \cdots < t^\star_{K^\star}$); define in addition the dummy indexes $t^\star_0 := 0$ and $t^\star_{K^\star+1} := T$. change-point detection consists in estimating the indexes $t^\star_k$. Change-point detection aims at recovering the unknown set of change-points $\mathcal{T}^\star$. In the context of kernel change-point detection, the estimation strategy relies on the minimization of the following kernel least square criterion $V(\cdot)$: for a given set of change-points $\mathcal{T}$, we define

$$V(\mathcal{T}) := \sum_{k=0}^{|\mathcal{T}|} \sum_{t=t_k+1}^{t_{k+1}} \left\| y_t - \bar{y}_{t_k..t_{k+1}} \right\|^2_{\mathcal{H}}, \tag{2}$$

where $\bar{y}_{a..b} := \frac{1}{b-a} \sum_{t=a+1}^{b} y_t$ ($\forall 1 \leq a < b \leq T$), and $t_0 := 0$ and $t_{K+1} := T$ are dummy variables.

B. Related Work

Generally, when the number K of changes is known, the least square criterion (2) is minimized using dynamic programming, resulting in a complexity of the order of $\mathcal{O}(KT^2)$ where T is the number of samples and K is the number of changes to estimate [44], [51], [53]. Pruned versions of the original algorithm have recently been introduced to speed-up computation [43] (worst-case complexity remains quadratic). When K is unknown, a penalized version of the criterion (2) is minimized [27], [54], [55]. In the general case, the change-point detection problem is solved (with dynamic programming) for K ranging from 1 to K_{max} (where K_{max} is a user-defined upper bound), and the optimal K is chosen a posteriori. This procedure has non-asymptotic properties on the convergence of the least square criterion and has a complexity of the order of $\mathcal{O}(K_{\max}T^2)$ [50]. In the special case of a penalty proportional to the number of changes, the Bayesian Information Criterion (BIC) [56] for instance, a faster algorithm exists [55], with linear complexity. Consistency results for the estimated change-point indexes are provided in a recent work [54]. They show that the estimated change-point indexes converge with high probability to the true segmentation, when the number of samples grows to infinity, even if the true number of changes is unknown. In this work, the observed signal is assumed to be composed of independent random variables with piecewise constant probability distribution. Certain applications, where the computational cost is an issue, require more efficient procedures. To that end, faster but sup-optimal algorithms have been introduced. Window-based methods approximately minimize $V(\cdot)$ by searching a single change-point in a window sliding over the signal [44], [49], [51], [57], [58]. Nevertheless, since detection is made locally, they are not as accurate and robust as optimal methods. Binary segmentation and bottom-up segmentation are sequential tree-based methods. Binary segmentation starts by detecting a single change-point, splits the signal around this change-point, and then repeats the operation on the two resulting sub-signals, until a stopping criterion is met [12], [55], [59]. Bottom-up segmentation starts by splitting the original signal in many small sub-signals and sequentially merges them [60]. Both are more accurate than window-based methods, as detection is made on larger sub-signals, and faster than dynamic programming, but are not optimal [61].

In the special case of a linear kernel (meaning that $\Phi = Id$ and $\mathcal{H} = \mathbb{R}^d$), the detection task consists in finding change in the mean of a noisy piecewise constant signal. In order to create efficient procedures that use the whole signal to detect change-points, relations between the change-point detection problem and sparse regression, with an appropriate design matrix, have been investigated. Methods based on basis pursuit have been applied, for instance regressions with a total variation penalty (or a fused lasso penalty) [17], [62]. Their implementations are efficient, with complexity of the order of $\mathcal{O}(KT)$ or $\mathcal{O}(T \log T)$ [17] and there are theoretical guarantees of detecting correct changes. In particular, a number of publications deal with the high-dimensional setting, where the signal dimension is as least as large as the number of samples [63]–[67].

C. Contributions

In this work, we describe a trade-off (in terms of complexity) between the optimal detection (based on dynamic programming) and the sub-optimal but fast detection methods from the literature. We propose a sequential approach, called `gkCPD` for "greedy kernel change-point detection," that sequentially generates change-point estimates and removes the associated mean-shifts from the mapped signal, until a stopping criterion is met (Section II). Arbitrary kernels can be combined with this greedy strategy. While the resulting complexity is quadratic in the number of samples, `gkCPD` is shown to be faster than optimal methods. This algorithm has desirable complexity properties and is non-parametric and model-free, thanks to the use of a kernel-based norm. A consistency result is demonstrated, which guarantees that detected change-points are asymptotically close to the true ones (Section III). In the special case of a linear kernel, a faster implementation with a linear complexity is described (Section IV).

II. Proposed Greedy Approach

We propose a greedy algorithm, which, in the following, is referred to as `gkCPD` for "greedy kernel change-point detection" and outlined in Algorithm 1. Let us consider a $\mathbb{R}^d$-valued signal $\{x_t\}_{t=1}^T$ and a mapped signal $y \in \mathcal{H}^T$, defined by $y_t := \Phi(x_t)$.

A. Heuristics for `gkCPD`

The algorithm `gkCPD` is an iterative procedure with each iteration consisting in two steps:

Algorithm 1: gkCPD.

```
Input: signal $y$, kernel $k$, stopping criterion.
Initialization: $\widehat{\mathcal{T}} \leftarrow \{\}$, $G \leftarrow [k(y_s, y_t)]_{1 \leq s,t \leq T}$,
    $I \leftarrow [\sum_{s \leq i, t \leq j} G_{st}]_{1 \leq i,j \leq T}$.
while stopping criterion is not met do
    $V \leftarrow [\,]$                ▷Empty list
    for $t = 1, \dots, T-1$ do
        $t_{\text{left}} \leftarrow \max\{s \in \widehat{\mathcal{T}} \cup \{0,T\} | s < t\}$
        $t_{\text{right}} \leftarrow \min\{s \in \widehat{\mathcal{T}} \cup \{0,T\} | s \geq t\}$
        $V[t] \leftarrow F_{t_{\text{left}}:t, t_{\text{left}}:t} + (\frac{t - t_{\text{left}}}{t_{\text{right}} - t_{\text{left}}})^2 F_{t_{\text{left}}:t_{\text{right}}, t_{\text{left}}:t_{\text{right}}} -$
        $2(\frac{t - t_{\text{left}}}{t_{\text{right}} - t_{\text{left}}}) F_{t_{\text{left}}:t, t_{\text{left}}:t_{\text{right}}}$
    end for
    $\hat{t} \leftarrow \arg\max_{t<T} [V[t]/(t(T-t))]$
    $\widehat{\mathcal{T}} \leftarrow \widehat{\mathcal{T}} \cup \{\hat{t}\}$
end while
Output: change-point estimates $\widehat{\mathcal{T}}$.
```

1) A single change-point is detected for y (greedy detection). The output of this operation is a change-point estimate $\hat{t}^{(k)}$, at the k-th iteration.
2) The contribution of the detected change is removed from the original signal with a projection (signal update). The output of this operation is a residual signal $\hat{r}^{(k)}$, at the k-th iteration. This signal is fed to the greedy detection step at the next iteration.

The algorithm continues until a stopping criterion is met (which can accommodate $K^\star$ known or unknown). This sequential greedy procedure is outlined by the following schema:

$$\begin{array}{ccccccccc} y & \longrightarrow & \widehat{r}^{(1)} & \longrightarrow & \widehat{r}^{(2)} & \longrightarrow & \widehat{r}^{(3)} & \longrightarrow & \cdots \\ & \downarrow & & \downarrow & & \downarrow & & \downarrow & \\ & \hat{t}^{(1)} & & \hat{t}^{(2)} & & \hat{t}^{(3)} & & \hat{t}^{(4)} & \end{array}$$

In the second step, gkCPD relies on an orthogonal projection which "deletes" the changes that have been detected. For any $\mathcal{T} = \{t_1, t_2, \dots\}$ and any $\mathcal{H}$-valued signal $\{v_t\}_t$, the orthogonal projection of v onto the subspace of signals that are constant over the segments delimited by $\mathcal{T}$ is denoted by $P_{\mathcal{T}} v$. As demonstrated in [50], it is given by

$$\forall t \in \{t_k + 1, \dots, t_{k+1}\} \quad (P_{\mathcal{T}} v)_t = \frac{1}{t_{k+1} - t_k} \sum_{s=t_k+1}^{t_{k+1}} v_s, \tag{3}$$

with the convention $(P_\emptyset v)_t = \frac{1}{T} \sum_{s=1}^{T} v_s$, for any index t.

2) First Iteration: At iteration $k = 1$, gkCPD starts by solving the *single* change-point detection problem. The first change estimate $\hat{t}^{(1)}$ of gkCPD is given by

$$\hat{t}^{(1)} := \underset{t<T}{\arg\min} \left[\sum_{s=1}^{t} \|y_s - \bar{y}_{0..t}\|_{\mathcal{H}}^2 + \sum_{s=t+1}^{T} \|y_s - \bar{y}_{t..T}\|_{\mathcal{H}}^2 \right]. \tag{4}$$

Then $y \in \mathcal{H}^T$ is projected on the subspace of $\mathcal{H}$-valued signals with a single mean-shift located at $\hat{t}^{(1)}$; the resulting residual is denoted $\hat{r}^{(1)} in \mathcal{H}^T$:

$$\hat{r}^{(1)} := y - P_{\widehat{\mathcal{T}}^{(1)}} y, \tag{5}$$

where $\widehat{\mathcal{T}}^{(1)} := \{\hat{t}^{(1)}\}$.

3) Iteration k: After k iterations ($k \geq 1$), the set of already estimated indexes is denoted $\widehat{\mathcal{T}}^{(k-1)} := \{\hat{t}^{(1)}, \dots, \hat{t}^{(k-1)}\}$. The k-th change-point estimate $\hat{t}^{(k)}$ is given by

$$\hat{t}^{(k)} := \underset{t<T}{\arg\min} \left[\sum_{s=1}^{t} \left\| \hat{r}_s^{(k-1)} - \bar{r}_{0..t}^{(k-1)} \right\|_{\mathcal{H}}^2 + \sum_{s=t+1}^{T} \left\| \hat{r}_s^{(k-1)} - \bar{r}_{t..T}^{(k-1)} \right\|_{\mathcal{H}}^2 \right], \tag{6}$$

where $\hat{r}^{(k-1)}$ is the residual signal from the previous iteration, and $\bar{r}_{0..t}^{(k-1)}$ and $\bar{r}_{t..T}^{(k-1)}$ are respectively the empirical means of the sub-signals $\{\hat{r}_s^{(k-1)}\}_{s \leq t}$ and $\{\hat{r}_s^{(k-1)}\}_{s>t}$. The index $\hat{t}^{(k)}$ is the solution of the *single* change-point detection problem, applied on the $(k-1)$-th residual. The k-th residual $\hat{r}^{(k)} \in \mathcal{H}^T$ is then defined as follows:

$$\hat{r}^{(k)} := y - P_{\widehat{\mathcal{T}}^{(k)}} y, \tag{7}$$

where $\widehat{\mathcal{T}}^{(k)} := \{\hat{t}^{(1)}, \dots, \hat{t}^{(k)}\}$ is the set of already estimated indexes, after k iterations. Thus defined, the residual is what remains of the signal y after the contributions of the already inferred change-points have been "projected" out.

B. The Kernel Trick

Performing the greedy detection (6) of gkCPD is not straightforward, because the mapping Φ, and therefore the residual (7), are not explicit. To overcome this issue, we express this operation using the inner-products of the signal samples, i.e. $\langle y_s | y_t \rangle_{\mathcal{H}}$. To that end, introduce the inner-product matrix (or Gram matrix) $G \in \mathbb{R}^{T \times T}$ of the implicit features: $G := [k(y_s, y_t)]_{1 \leq s,t \leq T}$ and the sub-sums of the matrix G:

$$F_{a:b,c:d} := \sum_{s=a+1}^{b} \sum_{t=c+1}^{d} G_{st} \quad (0 \leq a, b, c, d \leq T). \tag{8}$$

Assume that $k-1$ iterations have already been performed: the objective is to estimate $\hat{t}^{(k)}$ from the residual $\hat{r}^{(k-1)}$. After simple algebraic manipulations,[1] the greedy detection (6) can be rewritten as below:

$$\hat{t}^{(k)} = \underset{t}{\arg\max}\, t(T-t) \left\| \bar{r}_{0..t}^{(k-1)} - \bar{r}_{t..T}^{(k-1)} \right\|_{\mathcal{H}}^2. \tag{10}$$

[1] More precisely, we use the following relation: for any $\mathcal{H}$-valued signal $\{z_t\}_t$ with T samples and any index $t < T$, we have

$$\sum_{s=1}^{T} \|z_s - \bar{z}\|_{\mathcal{H}}^2 = \left[\sum_{s=1}^{t} \|z_s - \bar{z}_{0..t}\|_{\mathcal{H}}^2 + \sum_{s=t+1}^{T} \|z_s - \bar{z}_{t..T}\|_{\mathcal{H}}^2 \right] + \left[\frac{t(T-t)}{T} \|\bar{z}_{0..t} - \bar{z}_{t..T}\|_{\mathcal{H}}^2 \right] \tag{9}$$

where $\bar{z}, \bar{z}_{0..t}, \bar{z}_{t..T}$ are respectively the mean elements of the signals $\{z_s\}_s, \{z_s\}_{s \leq t}, \{z_s\}_{s>t}$.

The quantity to maximize is linked to an estimate of the *maximum mean discrepancy* (MMD). It is put forth in [51] in a different context to compare the distributions of two sets of samples, and has been subsequently used in several change-point detection procedures [58].

According to (10), the change-point estimate is located where the distributions between the left part of the signal and the right part are the most different. This quantity can be expressed using the inner-products from the residual, yielding

$$\hat{t}^{(k)} = \underset{t<T}{\arg\max}\, \frac{1}{t(T-t)} \sum_{s,u\leq t} \left\langle \hat{r}_s^{(k-1)} \middle| \hat{r}_u^{(k-1)} \right\rangle. \tag{11}$$

Using the fact that, by design (7), the inner-products $\left\langle r_s^{(k-1)} \middle| r_u^{(k-1)} \right\rangle$ are equal to

$$\langle y_s | y_u \rangle + \langle f_s | f_u \rangle - \langle f_s | y_u \rangle - \langle f_u | y_s \rangle, \tag{12}$$

where $f := P_{\widehat{\mathcal{T}}^{(k-1)}} y$, we are able to derive the following relation:

$$\begin{aligned}\sum_{s,u=1}^{t} \left\langle \hat{r}_s^{(k-1)} \middle| \hat{r}_u^{(k-1)} \right\rangle &= F_{\hat{t}_j:t,\hat{t}_j:t} \\ &+ \left(\frac{t-\hat{t}_j}{\hat{t}_{j+1}-\hat{t}_j} \right)^2 F_{\hat{t}_j:\hat{t}_{j+1},\hat{t}_j:\hat{t}_{j+1}} \\ &- 2\left(\frac{t-\hat{t}_j}{\hat{t}_{j+1}-\hat{t}_j} \right) F_{\hat{t}_j:t,\hat{t}_j:\hat{t}_{j+1}},\end{aligned} \tag{13}$$

where $\hat{t}_j$ $(j = 0, \ldots, k)$ is the unique element of $\widehat{\mathcal{T}}^{(k-1)}$ (with $\hat{t}_0 = 0$) such that $\hat{t}_j < t \leq \hat{t}_{j+1}$ and $F_{a:b,c:d}$ is defined in (8). By combining (11) and (13), the greedy estimated $\hat{t}^{(k)}$ can be computed without explicitly calculating the residual signal. The complete algorithm is described in Algorithm 1.

C. Complexity Analysis

To analyze the computational complexity of gkCPD, the algorithm is split into two phases: initialization and estimation. Initialization consists in computing the Gram matrix G of the samples, as well as the image integral matrix I, defined by

$$I := \left[\sum_{s\leq i,t\leq j} G_{st} \right]_{1\leq i,j\leq T} \in \mathbb{R}^{T\times T}. \tag{14}$$

Computing G and (recursively) filling I is performed in quadratic time. Estimation consists in performing the successive iterations. Observe that the cumulative sums $F_{a:b,c:d}$, which are used in the greedy detection (see Equations 11 and 13), are computed in constant time using the matrix I:

$$F_{a:b,c:d} = I_{bd} + I_{ac} - I_{bc} - I_{ad}. \tag{15}$$

Therefore, at each iteration, greedy detection (11) is performed in linear time. Consequently, the complexity of gkCPD is $\mathcal{O}(T^2 + KT)$ where the quadratic term comes from the initialization and the linear term, from K iterations. As a comparison, the exact change-point detection method [4], [50], [54] shares the exact same initialization phase, and then performs a dynamic programming procedure (estimation phase) which has quadratic complexity. The resulting complexity is $\mathcal{O}(T^2 + KT^2)$. To conclude, since estimation is performed in linear time by gkCPD, our greedy approach runs faster than dynamic programming, even though both have quadratic complexities overall.

D. Stopping Criterion

A crucial element of the gkCPD algorithm is the stopping criterion. The choice and calibration of a stopping rule is closely related to the issue of finding the number of change-points in a signal. If the number of change-points $K^\star$ is known, one simply stops the algorithm after $K^\star$ iterations. If it is unknown, a linear penalty [18], [26] can be added to yield a the following optimization problem:

$$\min_{\mathcal{T}} V(\mathcal{T}) + \beta|\mathcal{T}|, \tag{16}$$

where $\beta > 0$ is the smoothing parameter and $|\mathcal{T}|$ is the cardinal of $\mathcal{T}$. An adapted stopping rule to approximate the linearly penalized change-point detection is to stop at the k-th iteration if

$$\left\| \hat{r}^{(k-1)} \right\|_{\mathcal{H}}^2 - \left\| \hat{r}^{(k)} \right\|_{\mathcal{H}}^2 < \beta. \tag{17}$$

III. Theoretical Analysis

A theoretical study of gkCPD is now presented. The objective is to show that the true change-points are estimated more and more precisely by gkCPD, as the number of samples T grows to infinity. This result is obtained under two assumptions on the noise and the underlying piecewise constant signal.

1) Technical Assumptions: In order to establish our main result, assume that the Hilbert space $\mathcal{H}$ is separable. Moreover, two assumptions on the signal $(u_t)_t$ and the noise $(\varepsilon_t)_t$ are made.

Assumption 1: There exists a piecewise constant function $f : [0,1] \to \mathcal{H}$ given by

$$\forall \tau \in [0,1], \quad f(\tau) = \delta_0^\star + \sum_{k=1}^{K^\star} \delta_k^\star \mathbb{1}(\tau > \tau_k^\star), \tag{18}$$

where $0 < \tau_1^\star < \cdots < \tau_{K^\star}^\star < 1$ and $\delta_k^\star \in \mathcal{H}$, such that

$$\forall t = 1, \ldots, T, \quad u_t = f(t/T). \tag{19}$$

Assumption 2: The $\mathcal{H}$-valued random variables ε_t $(t = 1, \ldots, T)$ are independent, centered and such that

$$\|\varepsilon_t\|_{\mathcal{H}} \leq M_\varepsilon \quad \text{a.s.} \tag{20}$$

for a certain positive constant $M_\varepsilon < \infty$.

According to Assumption 1, the signal u is sampled from a piecewise constant function f. The change-points $t_k^\star$ are linked to the change-point *fractions* $\tau_k^\star$ through the simple relationship $t_k^\star = \lfloor T\tau_k^\star \rfloor$, where, for any $x \in \mathbb{R}$, $\lfloor x \rfloor$ is the largest integer smaller than or equal to x. In the literature, this assumption is often broken down in three weaker hypotheses, which are: u is bounded independently of T, the change amplitudes are bounded away from zero, and the minimum spacing between two consecutive change-point fractions is bounded away from zero. It is easy to see that those are three consequences of

Assumption 1, however assuming the existence of f greatly facilitates algebraic manipulations.

In the second assumption, the boundedness of the noise signal ε (Inequality 20) is satisfied for several change-point detection settings from the literature. In particular, if $k(\cdot,\cdot)$ is translation invariant, i.e. $k(x_s, x_t) = g(x_s - x_t)$ for every samples x_s, x_t and a certain measurable function $g : \mathbb{R}^d \longrightarrow \mathcal{H}$, then $\|\varepsilon_t\| = g(0)$ and Inequality 20 holds true. Note that the well-known Gaussian and Laplace kernels are translation invariant.

2) Main Result: Under this setting, the asymptotic consistency of gkCPD can be demonstrated, as formally stated in Theorem 1. In the following, the quantity $d(A|B)$ between two sets A and B is defined by

$$d(A|B) := \sup_{b\in B} \inf_{a\in A} |a - b|. \tag{21}$$

This quantity serves as a measure of dissimilarity between sets of change-points.

Theorem 1: Let y be as in Equation (1), and suppose that Assumption 1 and Assumption 2 hold. Let $\widehat{\mathcal{T}}$ denote the set of estimated change-points after $k \leq K^\star$ steps of gkCPD. Then some numerical constant $C > 0$ exists, such that for any $\alpha > 0$, an event of probability at least $1 - \alpha$ exists, on which the following holds true:

$$\frac{1}{T} d(\mathcal{T}^\star|\widehat{\mathcal{T}}) \leq C\sqrt{\frac{2\ln(T) + \ln(1/\alpha)}{T}}. \tag{22}$$

Intuitively, Theorem 1 means that the true change-point fractions $\tau_k^\star$ are estimated more and more precisely by $\hat{t}_k/T$ as the number of samples increases. The numerical constant C in (22) only depends on the change locations and amplitudes (the $\tau_k^\star$ and $\delta_k^\star$) and the noise characteristics. Note that gkCPD produces consistent estimates of the change-point fractions of the signal u, as long as the iterations stop before all true breakpoints are detected. In other words, only the first $K^\star$ steps are meaningful. In the situation where $k > K^\star$, meaning that all change-points have been selected, the orthogonal projection "deletes" all breaks and the detection is only driven by noise. Compared to optimal algorithms, the convergence rate $\sqrt{\ln T/T}$ is slower (typically optimal algorithms are of the order of in $1/T$, up to a logarithmic factor [62], [68]).

IV. Special Case of the Linear Kernel

A large part of the signal segmentation literature is dedicated to the detection of mean-shifts. This section describes a significant computational speed-up under this classical setting. In the following, the kernel k is set to the linear kernel, i.e. $k(\cdot,\cdot) = \langle\cdot|\cdot\rangle_{\mathbb{R}^d}$, where $\langle\cdot|\cdot\rangle_{\mathbb{R}^d}$ is the Euclidean scalar product of $\mathbb{R}^d$. As a consequence, the mapping Φ is equal to identity function ($\Phi = Id$). Also, $\mathcal{H} = \mathbb{R}^d$ and the original signal x and the mapped signal y are equal. The signal model is now

$$\forall t \in \{t_k^\star + 1, \ldots, t_{k+1}^\star\}, \quad y_t = u_t + \varepsilon_t, \tag{23}$$

where u_t is a $\mathbb{R}^d$-valued deterministic piecewise constant signal with $K^\star$ change-points and ε_t is a $\mathbb{R}^d$-valued additive noise with zero-mean. This is the classical mean-shift model, which was the first to be introduced in the change-point detection literature [1], [21], and is continuously the subject of active research [29], [30]. For clarity, the linear version of gkCPD is denoted gCPD, for "greedy change-point detection".

Algorithm 2: gCPD.

```
Input: centered data $y$, stopping criterion.
Initialize $\widehat{r} \leftarrow y$, $\widehat{\mathcal{T}} \leftarrow \{\}$
while stopping criterion is not met do
    Set $\hat{t} \leftarrow 1$, $r \leftarrow \widehat{r}_1$, $m^2 \leftarrow \frac{T}{T-1}\|r\|^2$.
    for $t = 2, \ldots, T-1$ do
        if $\frac{T}{t(T-t)}\|r + \widehat{r}_t\|^2 > m^2$ then
            $\hat{t} \leftarrow t$
            $m^2 \leftarrow \frac{T}{t(T-t)}\|r + \widehat{r}_t\|^2$
        end if
        $r \leftarrow r + \widehat{r}_t$
    end for
    Add the change-point estimate $\hat{t}$ to the set of selected breakpoints:
        $$\widehat{\mathcal{T}} \leftarrow \widehat{\mathcal{T}} \cup \{\hat{t}\}. \quad (24)$$
    Update the residual
        $$\widehat{r} \leftarrow y - P_{\widehat{\mathcal{T}}}y. \quad (25)$$
 where the orthogonal projection $P_{\widehat{\mathcal{T}}}$ is defined in (3).
end while
Output: set $\widehat{\mathcal{T}}$ of change-point estimates.
```

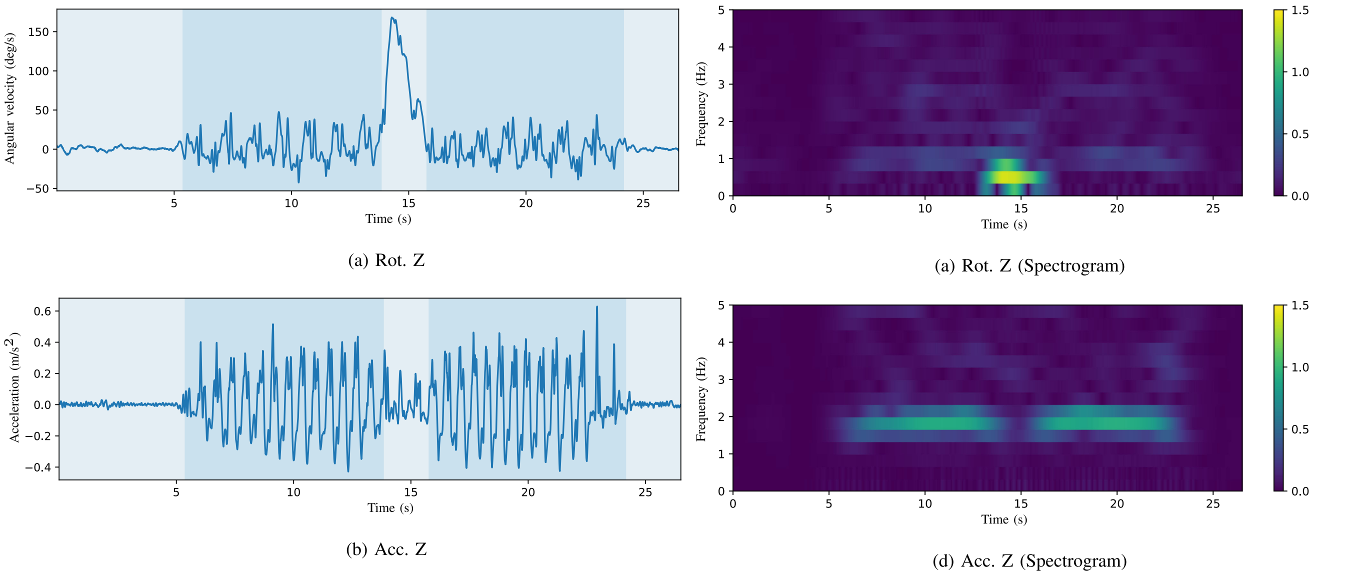
Since the mapping Φ is explicit for the linear kernel, there is no need for the computation of the Gram matrix, as is the case for arbitrary kernels. If again, the algorithm is split into an initialization phase and an estimation phase, this means that initialization is removed (recall that initialization of gkCPD has quadratic complexity). To see why estimation can be performed without initialization, observe that, for any signal v with zero mean, the following equality holds:

$$\overline{v}_{t..T} - \overline{v}_{0..t} = -\frac{T}{t(T-t)} \sum_{s=1}^{t} v_s. \tag{26}$$

where t is any index between 1 and $T - 1$. By replacing the signal v with any residual signal $\widehat{r}^{(k)}$, we see that the largest difference in the empirical means can be found incrementally using a cumulative sum and keeping track of the current maximum, which yields a $\mathcal{O}(dT)$ complexity. The k-th change-point $\hat{t}^{(k)}$ is estimated in linear time, without resorting to the Gram matrix. Updating the residual remains a linear operation, therefore the complexity of one iteration of the algorithm is $\mathcal{O}(dT)$. Overall, the complexity of gCPD is $\mathcal{O}(KdT)$ (if K iterations are performed). The implementation of gCPD is outlined in Algorithm 2.

V. Experimental Setting

This section describes the data sets and evaluation metrics that are used to evaluate the detection performances.



(a) Rot. Z

(a) Rot. Z (Spectrogram)

(b) Acc. Z

(d) Acc. Z (Spectrogram)

Fig. 1. Signal example from *Gait*. The acceleration and rotation on axis (Oz) (time and time-frequency representation) are shown. Alternating colours mark the consecutive phases: "Stand," "Walk," "Turnaround," "Walk" and "Stop".

A. *Presentation of the Data Sets*

The corpus is composed of a synthetic data set, *MeanShift*, and a real-world data set, *Gait*.

1) The MeanShift *Data Set:* The *MeanShift* data set contains $\mathbb{R}^d$-valued ($d = 20$) piecewise constant signals with $K = 4$ change-points, $T \in \{500, 2000\}$ samples and a noise level $\sigma \in \{1, 3\}$. We consider four scenarios for different values of (T, σ): Scenario 1, 2, 3 and 4 respectively correspond to (T, σ) equal to $(500, 1)$, $(500, 3)$, $(2000, 1)$ and $(2000, 3)$. For a given scenario, 100 signals are generated according to the following model:

$$y_t = \sum_{k=1}^{K} \delta_k \mathbb{1}(t_k < t) + \sigma\varepsilon_t \quad (t = 1, \ldots, T), \tag{27}$$

where $\mathcal{T} = \{t_k\}_{k=1}^K$ is a random set change indexes, and the $\delta_k \in \mathbb{R}^d$ are such that $\delta_k = [\pm 1, \ldots, \pm 1] \in \mathbb{R}^d$ with random coefficients equal to ± 1. Noise is Gaussian and white, with variance equal to σ^2. The complete *MeanShift* data set contains 400 signals. The change-point indexes are randomly drawn from $\{1, \ldots, T\}$ using a Dirichlet distribution.[2]

2) The Gait *Data Set:* The *Gait* data set consists of 262 time-series (sampling frequency: 100 Hz) from an inertial sensor placed at the lower back of 54 subjects [69], [70]. Each subject's movements are recorded while performing a succession of simple activities: standing for a few seconds, walking 10 meters, turning around, walking back and stopping. The successive regimes are straightforwardly denoted "Stand," "Walk," "Turnaround," "Walk," "Stop". The task is to detect the time indexes at which subject's activity changes. For this study, $d = 2$ dimensions are used: the angular velocity around the vertical axis ("Rot. Z") and the vertical acceleration ("Acc. Z"). Both dimensions are scaled to have zero mean and unit variance. In the following, the time-frequency representation of signals from *Gait* is defined as the short-term Fourier transform (STFT), computed with 300 samples per segment and an overlap of 299 samples. Only the 0–5 Hz frequency band, where phenomena of interest are contained, is kept. In this representation, the signals have $d = 32$ dimensions. An example is displayed on Figure 1.

B. *Evaluation Metrics*

To compare segmentations, two metrics are introduced: HAUSDORFF and F1 SCORE. In the following, the set of true change-points is denoted by $\mathcal{T}^\star = \{t_1^\star, \ldots, t_{K^\star}^\star\}$, and the set of estimated change-points is denoted by $\widehat{\mathcal{T}} = \{\hat{t}_1, \ldots, \hat{t}_{\widehat{K}}\}$.

1) HAUSDORFF: The HAUSDORFF metric measures the worst prediction error [71] between a set of change-point indexes $\{t_1, t_2, \ldots\}$ and their estimates $\{\hat{t}_1, \hat{t}_2, \ldots\}$. Formally, HAUSDORFF is expressed in number of samples or in second and is equal to

$$\text{HAUSDORFF}(\mathcal{T}^\star, \widehat{\mathcal{T}}) = \max[d(\mathcal{T}^\star|\widehat{\mathcal{T}}), d(\widehat{\mathcal{T}}|\mathcal{T}^\star)], \tag{28}$$

where $d(\cdot|\cdot)$ is defined by (21).

2) F1 SCORE: The F1 SCORE is the geometric mean of precision $\text{PR} := \#\text{TP}/\#\widetilde{\mathcal{T}}$ and recall $\text{RE} := \#\text{TP}/\#\mathcal{T}^\star$ where the true positive set $\text{TP} := \{t_k^\star \,|\, \exists\, \hat{t}_l \text{ s.t. } |\hat{t}_l - t_k^\star| < M\}$ contains detected change-points, up to a user-defined margin M (expressed in number of samples or in second).

VI. RESULTS

This section compares our greedy approach to standard segmentation methods. The following algorithms minimize the least square criterion $V(\cdot)$ (2) with the linear kernel ($\mathcal{H} = \mathbb{R}^d$): `gCPD`, `BinSegLin` (binary segmentation), `BotUpLin` (bottom-up segmentation), `OptLin` (dynamic programming) and `WinLin` (window-sliding). They are devoted to the detection of

[2] The Dirichlet distribution is parametrized by a vector $\alpha \in \mathbb{R}^{K+1}$ which is arbitrarily set to $(5, 5, 3, 5, 1) \times 2000$ to match the segmentations found in the *Gait* data set.

TABLE I

MEANS AND STANDARD DEVIATIONS ON THE *MeanShift* DATA SET ARE SHOWN. HAUSDORFF IS EXPRESSED IN NUMBER OF SAMPLES. THE MARGIN OF F1 SCORE IS $M = 10$ SAMPLES FOR SCENARIO 1 AND 2, AND $M = 20$ SAMPLES FOR SCENARIO 3 AND 4

MeanShift	Metric	Approximate						Optimal	
		gCPD	BinSegLin	BotUpLin	WinLin	gkCPD	WinGau	OptLin	OptGau
Scenario 1	HAUSDORFF	0.32 (±0.58)	**0.23 (±0.51)**	2.13 (±0.80)	0.43 (±0.67)	0.28 (±0.58)	0.30 (±0.56)	0.08 (±0.27)	0.08 (±0.27)
	F1 SCORE	1.00 (±0.00)	1.00 (±0.00)	1.00 (±0.00)	1.00 (±0.00)	1.00 (±0.00)	1.00 (±0.00)	1.00 (±0.00)	1.00 (±0.00)
Scenario 2	HAUSDORFF	**5.55 (±5.06)**	7.18 (±10.48)	7.96 (±4.74)	29.62 (±35.95)	15.97 (±26.68)	41.06 (±40.68)	4.29 (±3.61)	4.51 (±4.16)
	F1 SCORE	**0.95 (±0.12)**	0.94 (±0.13)	0.91 (±0.15)	0.85 (±0.16)	0.91 (±0.16)	0.82 (±0.17)	0.97 (±0.10)	0.96 (±0.10)
Scenario 3	HAUSDORFF	**0.28 (±0.53)**	0.36 (±0.67)	2.17 (±0.63)	1.42 (±0.49)	0.31 (±0.54)	1.38 (±0.49)	0.13 (±0.34)	1.69 (±0.48)
	F1 SCORE	1.00 (±0.00)	1.00 (±0.00)	1.00 (±0.00)	1.00 (±0.00)	1.00 (±0.00)	1.00 (±0.00)	1.00 (±0.00)	1.00 (±0.00)
Scenario 4	HAUSDORFF	**4.63 (±5.95)**	5.35 (±6.71)	7.68 (±4.86)	10.34 (±27.14)	5.80 (±7.14)	16.28 (±55.17)	3.14 (±2.60)	4.11 (±2.77)
	F1 SCORE	0.99 (±0.03)	0.99 (±0.05)	**1.00 (±0.02)**	0.99 (±0.05)	0.99 (±0.05)	0.99 (±0.05)	1.00 (±0.00)	1.00 (±0.00)

mean-shifts. In addition, the following algorithms minimize the least square criterion $V(\cdot)$ (2) with the Gaussian kernel $k(x, y) = \exp(-\gamma\|x - y\|^2)$: gkCPD, OptGau and WinGau. They are able to detect general distribution changes. Optimal methods OptLin and OptGau perform an exhaustive search over the set of signal partitions and return the exact minimum of the sum of costs. Conversely, other methods are sub-optimal and only approximate.

The parameters of the different algorithms are calibrated as follows. In all experiments and for all methods, the number K of change-points to detect is assumed to be known. For BotUpLin, the input signal is first divided in 5-sample long sub-signals. For Win{Lin,Gau}, the window size is set to 50 samples for the 500-point time series, to 100 for the 2000-point time series and to 100 samples for all time series from the *Gait* data set. The so-called bandwidth parameter γ of the Gaussian kernel is heuristically chosen as the inverse of the empirical median of the pairwise distances, as in [48].

A. *Results on the* MeanShift *Data Set*

In this section, detection methods are compared on the *MeanShift* data set. Several observations can be made from the results that are reported in Table I.

- According to HAUSDORFF, gCPD performs slightly better than all other approximate methods, except on Scenario 1, where BinSegLin is a little more accurate. On Scenario 2, which is the most difficult (least number of samples and highest noise level), gCPD has also the best F1 SCORE. Overall, BinSegLin and BotUpLin are the closest methods to gCPD.
- On *MeanShift*, using the Gaussian kernel with our greedy approach does not improve segmentation performance. For instance, on Scenario 2, gkCPD is less accurate than gCPD. This can be explained by the fact that the Gaussian kernel considers a general class of change-points while the linear kernel focuses on mean-shifts, exactly the type of changes present in *MeanShift*. Interestingly, when using the optimal search method Opt, both are equivalently precise.

B. *Results on the* Gait *Data Set*

On the *Gait* data set, detection is performed on the STFT representation of the signals. Global results are reported in Table IIa and the accuracy by change-point type is reported in Table IIb.

1) Global Results: Several observations can be made from the results that are reported in Table IIa.

- According to both metrics, gkCPD outperforms approximate as well as optimal methods. Here, optimal segmentation is not necessarily best when the signals do not follow exactly the assumed model.
- On the *Gait* data set, combining a non-linear kernel with our greedy approach improves segmentation performance, as gkCPD is more accurate than gCPD. This can be explained by the fact that the STFT representations of the signals contain changes more complex than mean-shifts, and that the Gaussian is flexible enough to detect them.
- Compared to WinLin, BinSegLin and BotUpLin, gCPD's worst error is lower by more than one second, but its F1 SCORE is inferior. This means that gCPD does not make large mistakes, and this smaller error is shared by more than one change-point. On the contrary, WinLin, BinSegLin and BotUpLin widely misestimate one change-point (by about three seconds or more), resulting in a high HAUSDORFF, but are accurate on the other changes.

These observations are illustrated on a segmentation example displayed on Figures 2 and 3. The best method on this signal is gkCPD, followed by OptLin and OptGau. A common behaviour, (see for instance gCPD) is to include in the "Stand" phase the first footstep of the "Walk" phase, and to include in the "Stop" phase the last step of the "Walk" phase. This can be explained by the fact that either the first or the last footstep has a smaller amplitude than the others.

To sum up, gkCPD is the most accurate method on the *Gait* data set, with a worst error which is lower than OptLin and OptGau.

2) Results by Change-Point Type: change-points in the *Gait* data set are not equivalent: they limit phases of different natures. To understand how segmentation methods operate, the mean values and standard deviations of the temoral distances $|\hat{t}_k - t^\star_k|$ $(k = 1, 2, 3, 4)$ are provided in Table IIb. Recall that $t^\star_1$, $t^\star_2$, $t^\star_3$ and $t^\star_4$ respectively refer to Stand/Walk, Walk/Turnaround, Turnaround/Walk and Walk/Stop.

- On average, window-based methods make error above 1 second on several change-points. BinSegLin and BotUpLin have sometimes the best score but an error well-above two seconds on the last change. Our greedy

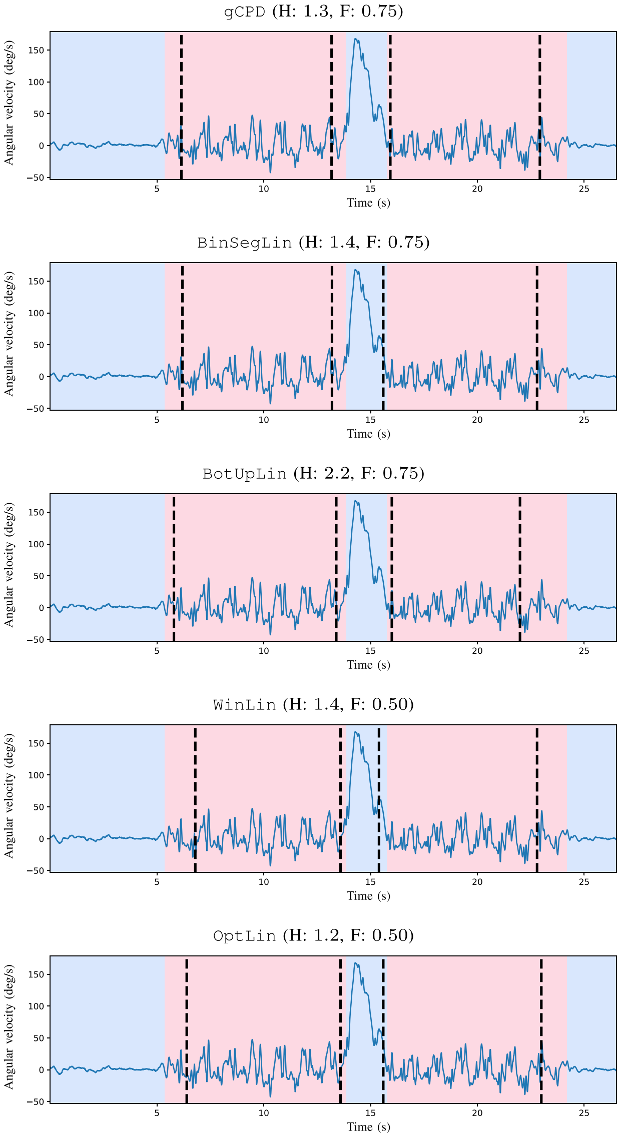
TABLE II
PERFORMANCE (MEANS AND STANDARD DEVIATIONS) ON THE *Gait* DATA SET

Metric	Approximate						Optimal	
	gCPD	BinSegLin	BotUpLin	WinLin	gkCPD	WinGau	OptLin	OptGau
HAUSDORFF	1.34 (±0.58)	4.44 (±3.38)	3.07 (±3.18)	2.92 (±3.21)	**1.13 (±0.67)**	3.31 (±3.44)	1.80 (±2.35)	1.29 (±1.06)
F1 SCORE	0.73 (±0.20)	0.79 (±0.13)	0.78 (±0.16)	0.81 (±0.17)	**0.85 (±0.15)**	0.80 (±0.18)	0.85 (±0.13)	0.83 (±0.15)

(a) Global results. (Margin for F1 SCORE is one second. HAUSDORFF is in second.)

	Approximate						Optimal	
	gCPD	BinSegLin	BotUpLin	WinLin	gkCPD	WinGau	OptLin	OptGau
Stand/Walk	0.37 (±0.32)	0.53 (±0.91)	0.60 (±1.00)	2.00 (±2.81)	0.48 (±0.33)	2.20 (±2.95)	0.60 (±0.92)	0.52 (±0.32)
Walk/Turnaround	0.88 (±0.69)	0.57 (±0.64)	0.41 (±0.74)	0.94 (±1.58)	0.65 (±0.64)	0.93 (±1.57)	0.38 (±0.62)	0.62 (±0.81)
Turnaround/Walk	0.99 (±0.95)	0.69 (±1.00)	0.51 (±0.59)	1.28 (±2.11)	0.63 (±0.78)	1.29 (±2.07)	0.38 (±0.43)	0.49 (±0.43)
Walk/Stop	1.01 (±0.52)	4.32 (±3.41)	2.89 (±3.19)	1.42 (±2.39)	0.98 (±0.72)	1.69 (±2.76)	1.69 (±2.24)	1.22 (±1.09)

(b) Results by change-point type. (Average temporal distance of a predicted change-point to an annotated change-point, in seconds.)


gCPD (H: 1.3, F: 0.75)

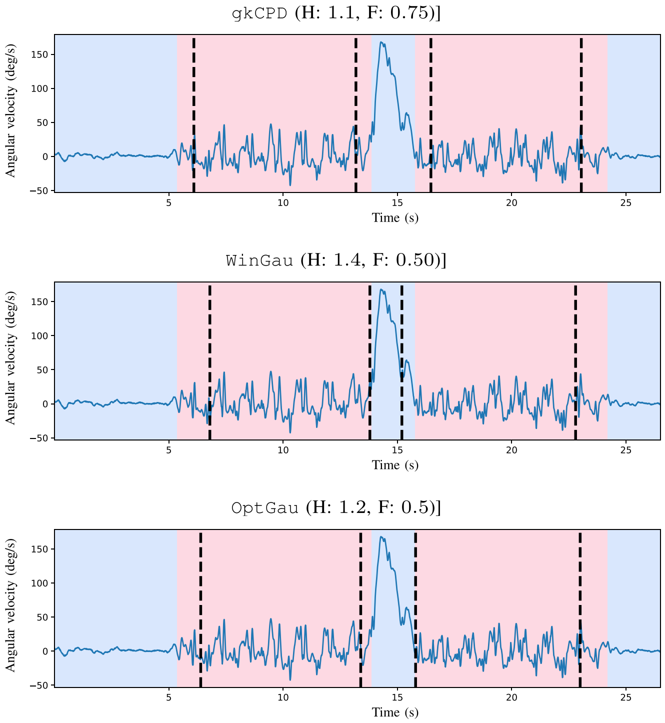
BinSegLin (H: 1.4, F: 0.75)

BotUpLin (H: 2.2, F: 0.75)

WinLin (H: 1.4, F: 0.50)

OptLin (H: 1.2, F: 0.50)

Fig. 2. Segmentation example (linear kernel) on a signal from *Gait*. H and F respectively denote HAUSDORFF (in seconds) and F1 SCORE. The alternating coloured areas denote the true segmentation. Dashed lines mark estimated change-points. The acceleration on the (0z) axis is shown.


gkCPD (H: 1.1, F: 0.75)]

WinGau (H: 1.4, F: 0.50)]

OptGau (H: 1.2, F: 0.5)]

Fig. 3. Segmentation example (Gaussian kernel) on a signal from *Gait*. H and F respectively denote HAUSDORFF (in seconds) and F1 SCORE. The alternating coloured areas denote the true segmentation. Dashed lines mark estimated change-points. The acceleration on the (0z) axis is shown.

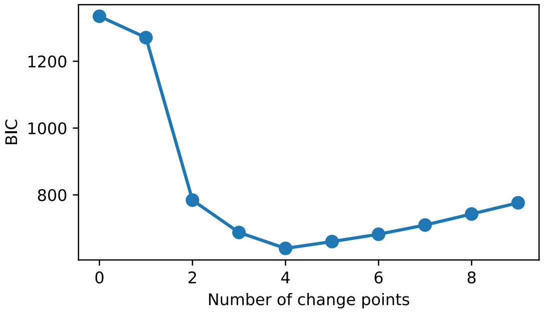


Fig. 4. Constrained costs (BIC) of the sequential estimates of gCPD. Mean values on the *Gait* data set are shown.

approaches gCPD and gkCPD are always close to the best score, and their errors are always below 1.01 second.

- The first change-point is the best detected by gCPD, gkCPD and BinSegLin. The average temporal error for those algorithms is 0.53 second or less. This can be explained by the fact that this change-point separates two long regimes which are visibly very different ("Stand" is somewhat flat, and "Walk" has a large amplitude). Interestingly, for window-based methods and BotUpLin, the best detected change is not this particular change-point. Since the search for a change is limited to a small region (the window for WinLin and WinGau and the small initial sub-segments for BotUpLin), they cannot take advantage of the length of the regimes.
- Conversely, estimation of the last change-point is generally less accurate, even though it is of the same type as the first one, as both change-points separate a "Walk" phase and a rest phase ("Stand" or "Stop"). This error is what drives the HAUSDORFF score (see Table IIa): it is the worst estimation error. It is interesting to note that BinSegLin has errors comparable to gCPD and gkCPD on the first three changes but is four times less precise on the last change-point. Two reasons can justify this observation. The last change-point corresponds to a smaller mean-shift amplitude (in the STFT representation). Also, it is located at the edge of the signal, which adversely influences certain methods, BinSegLin being a prime example.
- Detection error on the "Turnaround" is below one second for gCPD, BinSegLin, BotUpLin, and gkCPD. The best one for this phase is BotUpLin, with an error close to the one of OptLin. After a closer look on the segmentations, we observe that gkCPD and gCPD tend to include in the "Turnaround" phase the last footstep of the previous "Walk" phase and the first footstep of the previous "Walk" phase. This behaviour is displayed on the segmentation example on Figures 2 and 3. This is understandable because the footsteps at the beginning or end of each regime can be different from the footsteps in the middle of the regime (for instance, of smaller amplitude because the subject is accelerating or slowing down).
- The two algorithms WinLin and WinGau have F1 SCORE around 0.80 (Table IIa) but on three out of four change-points, the average temporal error is well above the margin of one second. This indicates that on average, three change-points are correctly estimated (thus the F1 SCORE above 0.75), but the last one is greatly misplaced. Also, those algorithms make an error indiscriminately on either the first ("Stand/Walk"), the third ("Turnaround/Walk") or the fourth ("Walk/Stop"), as evidenced by the high temporal distances. Conversely, BotUpLin is more likely to make an error on the last one only.

To sum up, observing the error by change-point type allows us to better understand the behaviour of segmentation methods. Window-based methods are confirmed not to be as precise as the other methods on this data set. Tree-based methods algorithms, BotUpLin and BinSegLin are precise on the first three change-points but misplace the last one by a large margin. The greedy procedures gCPD and gkCPD are able to locate all change-points with an error of less than one second. In particular, gkCPD is relatively more accurate on average.

TABLE III
PERFORMANCE (MEANS AND STANDARD DEVIATIONS) ON THE *Gait* DATA SET (UNTRANSFORMED SIGNALS)

Metric	Approximate		Optimal
	gkCPD	WinGau	OptGau
HAUSDORFF	1.35 ($\pm$1.35)	5.79 ($\pm$2.86)	1.44 ($\pm$2.12)
F1 SCORE	0.86 ($\pm$0.18)	0.64 ($\pm$0.17)	0.90 ($\pm$0.15)

C. *Results on the* Gait *Data Set With Untransformed Signals*

Change-point detection is greatly improved thanks to the use of a kernel. To intuitively measure the usefulness of the kernel, gkCPD, WinGau and OptGau are applied on the *untransformed* signals of the *Gait* data set. Contrary to previous experiments (Section VI-B), segmentation is performed on *the time domain representation* of the signals and not the the time-frequency representation. The only preprocessing consists in centering and scaling all dimensions of the signal to unit variance. Only the Gaussian kernel is considered here because the linear kernel can only detect mean-shifts. Results are reported in Table III.

This experiment confirms a few of the remarks from the previous section, but an interesting observation can be made when comparing gkCPD on the untransformed signals and gCPD on the time-frequency representation of the same signals (see Table IIa): while both have a comparable HAUSDORFF measure (around 1.35 s), the kernel-based method has a better F1 SCORE (0.86 against 0.73). This indicates that on the *Gait* data set, using a kernel can replace the careful design of a relevant signal representation. Combining both the time-frequency representation and a kernel performs even better, as gkCPD on the time-frequency representation of the signals is more accurate than gkCPD on the untransformed signals. This observation is encouraging as it indicates that the Gaussian kernel can cope with signals in their original representation space as well as finely calibrated transformed time-series. In particular, gkCPD could be combined with automatic calibration procedures [72], [73].

VII. DISCUSSION

A. *Execution Time Comparison*

Table IV presents average execution times (for 100 signals) of the different methods. Visibly, differences in execution time increase as the number of samples and the number of dimensions grows. Even if gCPD, BinSegLin and BotUpLin all have a linear computational complexity (up to a logarithmic factor), gCPD's implementation ease allows for an efficient execution. Specifically, operations described in Algorithm 2 are naturally "vectorized". In languages like Python and Matlab, such operations are more cost-effective than the explicit looping instructions needed in tree-based methods. As a comparison, the signal acquisition system takes around 50 seconds to record one signal, while gCPD takes 1 second to process 100 signals,

TABLE IV
AVERAGE RUNTIME FOR EACH ALGORITHM TO PROCESS 100 SIGNALS. ALL TIMES REFER TO A PYTHON IMPLEMENTATION ON A LINUX COMPUTER WITH 4 PROCESSORS RUNNING AT 2.80 GHZ

Data set	Approximate						Optimal	
	`gCPD`	`BinSegLin`	`BotUpLin`	`WinLin`	`gkCPD`	`WinGau`	`OptLin`	`OptGau`
MeanShift ($T = 500$)	< 1 s	0 m 17 s	0 m 03 s	0 m 06 s	0 m 06 s	0 m 02 s	7 m 23 s	6 m 45 s
MeanShift ($T = 2000$)	< 1 s	2 m 09 s	0 m 23 s	0 m 23 s	1 m 29 s	0 m 28 s	42 m 02 s	11 h
Gait data set	< 1 s	3 m 55 s	0 m 33 s	0 m 26 s	1 m 34 s	0 m 22 s	4 h	> 2 days

and `gkCPD` takes less than 100 seconds. `OptLin` is far slower, as expected. One advantage window-based methods have over `gCPD` is the fact that they only process a portion of the signal at a time. When memory is an issue or when faced with a continuous stream, `WinLin` and `WinGau` are the only appropriate methods, as other methods are applied on the whole signal. As for `gkCPD`, it is more computationally intensive than its counterpart `gCPD`, but remains faster than `OptGau`.

B. Estimation of the Number of Change-Points With `gCPD`

In our experiments, the number of change-point is known beforehand. However, in certain applications, such information might not be known. We show in the following that our greedy strategy can be combined with a simple model selection procedure to accommodate situations where the number of changes is unknown. In practice, the Bayesian information criterion (BIC) is commonly used in change-point detection to determine the number of change-points [56], [74]. It is a model selection procedure that consists in minimizing a constrained likelihood function. In the context of piecewise constant signals with white Gaussian noise, the BIC of the sequential `gCPD` estimates is

$$\mathrm{BIC}(k) = \left\| \hat{r}^{(k)} \right\|^2 + k\,\sigma^2 d \log T, \qquad (29)$$

where k is the step number (as well as the number of change-points). The model with lowest BIC is preferred. On the *Gait* data set, the average BIC values are displayed on Figure VII-B. The minimum value is 639.95 and is reached for $k = 4$, which indeed the true number of changes. This substantiates the fact that `gCPD` can accommodate a standard model selection procedure even if it is only an approximation of the optimal signal segmentation.

VIII. CONCLUSION

In this work, we described `gkCPD`, a greedy strategy for the kernel change-point detection task. Thanks to the properties of reproducing Hilbert spaces, `gkCPD` detects changes in higher-order moments of probability distributions. As a result of an efficient implementation, this algorithm is faster than its optimal counterpart.

A consistency result is provided which guarantees that, for an arbitrary kernel, the, detected change-points are asymptotically close to the true change-points.

The special case of a linear kernel is also tackled, which yields `gCPD`, the linear version of `gkCPD`. A faster implementation, with complexity of the order of $\mathcal{O}(T)$ is described. Numerical experiments show that our greedy approach is more accurate than standard sub-optimal methods and faster than optimal methods.

REFERENCES

[1] E. S. Page, "A test for a change in a parameter occurring at an unknown point," *Biometrika*, vol. 42, pp. 523–527, 1955.

[2] M. Basseville and I. Nikiforov, *Detection of Abrupt Changes: Theory and Application*, vol. 104. Englewood Cliffs, NJ, USA: Prentice-Hall, 1993.

[3] M. Lavielle and É. Moulines, "Least-squares estimation of an unknown number of shifts in a time series," *J. Time Series Anal.*, vol. 21, no. 1, pp. 33–59, 2000.

[4] Z. Harchaoui and O. Cappé, "Retrospective multiple change-point estimation with Kernels," in *Proc. IEEE/SP Workshop Statist. Signal Process.*, Madison, WI, USA, 2007, pp. 768–772.

[5] J. Chen and A. K. Gupta, "Testing and locating variance changepoints with application to stock prices," *J. Amer. Statist. Assoc.*, vol. 92, no. 438, pp. 739–747, 1997.

[6] J. Bai, "Vector autoregressive models with structural changes in regression coefficients and in variance covariance matrices," *Ann. Econ. Finance*, vol. 1, no. 2, pp. 301–336, 2000.

[7] M. Lavielle, "Optimal segmentation of random processes," *IEEE Trans. Signal Process.*, vol. 46, no. 5, pp. 1365–1373, May 1998.

[8] L. Oudre, A. Lung-Yut-Fong, and P. Bianchi, "Segmentation of accelerometer signals recorded during continuous treadmill walking," in *Proc. 19th Eur. Signal Process. Conf.*, 2011, pp. 1564–1568.

[9] C. Truong, L. Oudre, and N. Vayatis, "Penalty learning for change-point detection," in *Proc. Eur. Signal Process. Conf.*, Kos, Greece, 2017, pp. 1569–1573.

[10] S. Liu, A. Wright, and M. Hauskrecht, "Change-point detection method for clinical decision support system rule monitoring," *Artif. Intell. Med.*, vol. 91, pp. 49–56, 2018.

[11] N. Omranian, B. Mueller-Roeber, and Z. Nikoloski, "Segmentation of biological multivariate time-series data," *Sci. Rep.*, vol. 5, pp. 1–6, 2015.

[12] A. L. Schröder and H. Ombao, "FreSpeD: Frequency-Specific change-point detection in epileptic seizure multi-channel EEG data," *J. Amer. Statist. Assoc.*, vol. 114, pp. 1–14, 2019.

[13] J. Reeves, J. Chen, X. L. Wang, R. Lund, and Q. Q. Lu, "A review and comparison of changepoint detection techniques for climate data," *J. Appl. Meteorol. Climatol.*, vol. 46, no. 6, pp. 900–915, 2007.

[14] J.-J. Jeon, J. Hyun Sung, and E.-S. Chung, "Abrupt change point detection of annual maximum precipitation using fused lasso," *J. Hydrol.*, vol. 538, pp. 831–841, 2016.

[15] J. Verbesselt, R. Hyndman, G. Newnham, and D. Culvenor, "Detecting trend and seasonal changes in satellite images time series," *Remote Sens. Environ.*, no. 114, pp. 106–115, 2010.

[16] J. Ding, Y. Xiang, L. Shen, and V. Tarokh, "Multiple change point analysis: Fast implementation and strong consistency," *IEEE Trans. Signal Process.*, vol. 65, no. 17, pp. 4495–4510, Sep. 2017.

[17] J.-P. Vert and K. Bleakley, "Fast detection of multiple change-points shared by many signals using group LARS," in *Proc. Adv. Neural Inf. Process. Syst.*, Vancouver, BC, Canada, 2010, vol. 1, pp. 2343–2351.

[18] T. Hocking, G. Rigaill, J.-P. Vert, and F. Bach, "Learning sparse penalties for change-point detection using max margin interval regression," in *Proc. Int. Conf. Mach. Learn.*, Atlanta, GA, USA, 2013, pp. 172–180.

[19] F. Harlé, F. Chatelain, C. Gouy-Pailler, and S. Achard, "Bayesian model for multiple change-points detection in multivariate time series," *IEEE Trans. Signal Process.*, vol. 64, no. 16, pp. 4351–4362, Aug. 2016.

[20] F. Picard, S. Robin, M. Lavielle, C. Vaisse, and J.-J. Daudin, “A statistical approach for array CGH data analysis,” *BMC Bioinformat.*, vol. 6, no. 27, 2005.

[21] E. S. Page, “Continuous inspection schemes,” *Biometrika*, vol. 41, pp. 100–105, 1954.

[22] F. Pein, H. Sieling, and A. Munk, “Heterogeneous change point inference,” *J. Roy. Statist. Soc.. B (Statist. Methodol.)*, vol. 79, no. 4, pp. 1207–1227, 2017.

[23] H. Keshavarz, C. Scott, and X. Nguyen, “Optimal change point detection in Gaussian processes,” *J. Statist. Planning Inference*, vol. 193, pp. 151–178, 2018.

[24] P. Fearnhead and G. Rigaill, “Changepoint detection in the presence of outliers,” *J. Am. Stat. Assoc.*, vol. 114, no. 525, pp. 169–183, 2019.

[25] T. Górecki, L. Horváth, and P. Kokoszka, “Change point detection in heteroscedastic time series,” *Econometrics Statist.*, vol. 7, pp. 63–88, 2018.

[26] É. Lebarbier, “Detecting multiple change-points in the mean of Gaussian process by model selection,” *Signal Process.*, vol. 85, no. 4, pp. 717–736, 2005.

[27] M. Lavielle, “Using penalized contrasts for the change-point problem,” *Signal Process.*, vol. 85, no. 8, pp. 1501–1510, 2005.

[28] S. Ma and L. Su, “Estimation of large dimensional factor models with an unknown number of breaks,” *J. Econometrics*, vol. 207, no. 1, pp. 1–29, 2018.

[29] P. Fryzlewicz, “Wild binary segmentation for multiple change-point detection,” *Ann. Statist.*, vol. 42, no. 6, pp. 2243–2281, 2014.

[30] P. Fryzlewicz, “Unbalanced Haar technique for nonparametric function estimation,” *J. Amer. Statist. Assoc.*, vol. 102, no. 480, pp. 1318–1327, 2007.

[31] D. Siegmund and E. S. Venkatraman, “Using the generalized likelihood ratio statistic for sequential detection of a change-point,” *Ann. Statist.*, vol. 23, no. 1, pp. 255–271, 1995.

[32] M. Lavielle, “Detection of multiples changes in a sequence of dependant variables,” *Stochastic Processes Their Appl.*, vol. 83, no. 1, pp. 79–102, 1999.

[33] H. Cho and P. Fryzlewicz, “Multiple change-point detection for high dimensional time series via sparsified binary segmentation,” *J. Roy. Statist. Soc.: B (Statist. Methodol.)*, vol. 77, no. 2, pp. 475–507, 2014.

[34] Z. Qu and P. Perron, “Estimating and testing structural changes in multivariate regressions,” *Econometrica*, vol. 75, no. 2, pp. 459–502, 2007.

[35] A. Cleynen and É. Lebarbier, “Model selection for the segmentation of multiparameter exponential family distributions,” *Electron. J. Statist.*, vol. 11, no. 1, pp. 800–842, 2017.

[36] L. Rabiner, “A tutorial on hidden Markov models and selected applications in speech recognition,” *Proc. IEEE*, vol. 77, no. 2, pp. 257–286, Feb. 1989.

[37] S. I. M. Ko, T. T. L. Chong, and P. Ghosh, “Dirichlet process hidden Markov multiple change-point model,” *Bayesian Anal.*, vol. 10, no. 2, pp. 275–296, 2015.

[38] D. Barry and J. A. Hartigan, “A Bayesian analysis for change point problems,” *J. Amer. Statist. Assoc.*, vol. 88, no. 421, pp. 309–319, 1993.

[39] V. Jandhyala, S. Fotopoulos, I. Macneill, and P. Liu, “Inference for single and multiple change-points in time series,” *J. Time Series Anal.*, vol. 34, no. 4, pp. 423–446, 2013.

[40] J. Chen and A. K. Gupta, *Parametric Statistical Change Point Analysis: With Applications to Genetics, Medicine, and Finance*. Berlin, Germany: Springer, 2011.

[41] B. K. Sriperumbudur, A. Gretton, K. Fukumizu, G. Lanckriet, and B. Schölkopf, “Injective Hilbert space embeddings of probability measures,” in *Proc. 21st Conf. Learn. Theory*, Helsinki, Finland, 2008, pp. 9–12.

[42] Z. Harchaoui, F. Bach, O. Cappé, and É. Moulines, “Kernel-based methods for hypothesis testing: A unified view,” *IEEE Signal Process. Mag.*, vol. 30, no. 4, pp. 87–97, Jul. 2013.

[43] A. Celisse, G. Marot, M. Pierre-Jean, and G. Rigaill, “New efficient algorithms for multiple change-point detection with reproducing Kernels,” *Comput. Statist. Data Anal.*, vol. 128, pp. 200–220, 2018.

[44] Z. Harchaoui, F. Bach, and É. Moulines, “Kernel change-point analysis,” in *Proc. Adv. Neural Inf. Process. Syst.*, Vancouver, BC, Canada, 2008, pp. 609–616.

[45] J. Cabrieto, F. Tuerlinckx, P. Kuppens, F. H. Wilhelm, M. Liedlgruber, and E. Ceulemans, “Capturing correlation changes by applying Kernel change point detection on the running correlations,” *Inf. Sci.*, vol. 447, pp. 117–139, 2018.

[46] S. Aminikhanghahi and D. J. Cook, “A survey of methods for time series change point detection,” *Knowl. Inf. Syst.*, vol. 51, no. 2, pp. 339–367, 2017.

[47] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning*, 2nd ed. Berlin, Germany: Springer, 2009.

[48] B. Schölkopf and A. Smola, *Learning With Kernels*. Cambridge, Cambridge, MA, USA: MIT Press, 2002.

[49] Z. Harchaoui, F. Vallet, A. Lung-Yut-Fong, and O. Cappé, “A regularized Kernel-based approach to unsupervised audio segmentation,” in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, Taipei, Taiwan, 2009, pp. 1665–1668.

[50] S. Arlot, A. Celisse, and Z. Harchaoui, “A Kernel multiple change-point algorithm via model selection,” 2012, *arXiv:1202.3878v3*.

[51] A. Gretton, K. M. Borgwardt, M. J. Rasch, B. Schölkopf, and A. Smola, “A Kernel two-sample test,” *J. Mach. Learn. Res.*, vol. 13, pp. 723–773, 2012.

[52] C. Truong, L. Oudre, and N. Vayatis, “Selective review of offline change point detection methods,” *Signal Process.*, vol. 167, 2020, Art. no. 107299.

[53] R. Maidstone, T. Hocking, G. Rigaill, and P. Fearnhead, “On optimal multiple changepoint algorithms for large data,” *Statist. Comput.*, vol. 27, no. 2, pp. 519–533, 2017.

[54] D. Garreau and S. Arlot, “Consistent change-point detection with Kernels,” *Electron. J. Statist.*, vol. 12, no. 2, pp. 4440–4486, 2018.

[55] R. Killick, P. Fearnhead, and I. Eckley, “Optimal detection of changepoints with a linear computational cost,” *J. Amer. Statist. Assoc.*, vol. 107, no. 500, pp. 1590–1598, 2012.

[56] Y.-C. Yao, “Estimating the number of change-points via Schwarz’ criterion,” *Statist. Probability Lett.*, vol. 6, no. 3, pp. 181–189, 1988.

[57] B. E. Brodsky, B. S. Darkhovsky, A. Y. Kaplan, and S. L. Shishkin, “A nonparametric method for the segmentation of the EEG,” *Comput. Methods Programs Biomed.*, vol. 60, no. 2, pp. 93–106, 1999.

[58] S. Li, Y. Xie, H. Dai, and L. Song, “M-statistic for Kernel change-point detection,” in *Proc. Adv. Neural Inf. Process. Syst.*, Montréal, QC, Canada, 2015, pp. 3366–3374.

[59] A. Sen and M. S. Srivastava, “On tests for detecting change in mean,” *Ann. Statist.*, vol. 3, no. 1, pp. 98–108, 1975.

[60] E. Keogh, S. Chu, D. Hart, and M. Pazzani, “An online algorithm for segmenting time series,” in *Proc. IEEE Int. Conf. Data Mining*, San Jose, CA, USA, 2001, pp. 289–296.

[61] M. Lavielle and G. Teyssière, “Adaptive detection of multiple change-points in asset price volatility,” in *Long-Memory in Economics*. Berlin, Germany: Springer-Verlag, 2007, pp. 129–156.

[62] Z. Harchaoui and C. Lévy-Leduc, “Multiple change-point estimation with a total variation penalty,” *J. Amer. Statist. Assoc.*, vol. 105, no. 492, pp. 1480–1493, 2010.

[63] F. Enikeeva and Z. Harchaoui, “High-dimensional change-point detection with sparse alternatives,” 2014, *arXiv:1312.1900*.

[64] M. Barigozzi, H. Cho, and P. Fryzlewicz, “Simultaneous multiple change-point and factor analysis for high-dimensional time series,” *J. Econometrics*, vol. 206, no. 1, pp. 187–225, 2018.

[65] T. Wang and R. J. Samworth, “High dimensional change point estimation via sparse projection,” *J. Roy. Statist. Soc. B (Statist. Methodol.)*, vol. 80, no. 1, pp. 57–83, 2018.

[66] M. Jirak, “Uniform change point tests in high dimension,” *Ann. Statist.*, vol. 43, no. 6, pp. 2451–2483, 2015.

[67] Y. S. Sog and V. Chandrasekaran, “High-dimensional change-point estimation: combining filtering with convex optimization,” *Appl. Comput. Harmon. Anal.*, vol. 43, no. 1, pp. 122–147, 2017.

[68] Y.-C. Yao and S. T. Au, “Least-squares estimation of a step function,” *Sankhy: Indian J. Statist., A*, vol. 51, no. 3, pp. 370–381, 1989.

[69] R. Barrois-Müller *et al.*, “Quantify osteoarthritis gait at the doctor’s office: A simple pelvis accelerometer based method independent from footwear and aging,” *Comput. Methods Biomechanics Biomed. Eng.*, vol. 18, no. Suppl 1, pp. 1880–1881, 2015.

[70] R. Barrois-Müller *et al.*, “Étude observationnelle du demi-tour à l’aide de capteurs inertiels chez les sujets victimes d’AVC et relation avec le risque de chute,” *Neurophysiologie Clinique/Clinical Neurophysiol.*, vol. 46, no. 4, p. 244, 2016.

[71] L. Boysen, A. Kempe, V. Liebscher, A. Munk, and O. Wittich, “Consistencies and rates of convergence of jump-penalized least squares estimators,” *Ann. Statist.*, vol. 37, no. 1, pp. 157–183, 2009.

[72] C. Truong, L. Oudre, and N. Vayatis, “Supervised Kernel change point detection with partial annotations,” in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, Brighton, U.K., 2019, pp. 1–5.

[73] R. Lajugie, F. Bach, and S. Arlot, “Large-margin metric learning for constrained partitioning problems,” in *Proc. 31st Int. Conf. Mach. Learn.*, Beijing, China, 2014, pp. 297–395.

[74] N. R. Zhang and D. O. Siegmund, “A modified Bayes information criterion with applications to the analysis of comparative genomic hybridization data.” *Biometrics*, vol. 63, no. 1, pp. 22–32, 2007.