Signal Processing 167 (2020) 107299

Contents lists available at ScienceDirect

Signal Processing

journal homepage: www.elsevier.com/locate/sigpro

Review

Selective review of offline change point detection methods

Charles Truong[a], Laurent Oudre[b,*], Nicolas Vayatis[a]

[a] CMLA, CNRS, ENS Paris Saclay France
[b] L2TI, University Paris 13 France

ARTICLE INFO

Article history:
Received 14 December 2018
Revised 19 August 2019
Accepted 16 September 2019
Available online 19 September 2019

Keywords:
Change point detection
Segmentation
Statistical signal processing

ABSTRACT

This article presents a selective survey of algorithms for the offline detection of multiple change points in multivariate time series. A general yet structuring methodological strategy is adopted to organize this vast body of work. More precisely, detection algorithms considered in this review are characterized by three elements: a cost function, a search method and a constraint on the number of changes. Each of those elements is described, reviewed and discussed separately. Implementations of the main algorithms described in this article are provided within a Python package called ruptures.

1. Introduction

A common task in signal processing is the identification and analysis of complex systems whose underlying state changes, possibly several times. This setting arises when industrial systems, physical phenomena or human activity are continuously monitored with sensors. The objective of practitioners is to extract from the recorded signals a posteriori meaningful information about the different states and transitions of the monitored object for analysis purposes. This setting encompasses a broad range of real-world scenarios and a wide variety of signals.

Change point detection is the task of finding changes in the underlying model of a signal or time series. The first works on change point detection go back to the 50s [1,2]: the goal was to locate a shift in the mean of independent and identically distributed (iid) Gaussian variables for industrial quality control purposes. Since then, this problem has been actively investigated, and is periodically the subject of in-depth monographs [3–6]. This subject has generated important activity in statistics and signal processing [7–9] but also in various application settings such as speech processing [10–13], financial analysis [7,14,15], bio-informatics [16–24], climatology [25–27], network traffic data analysis [28,29]. Modern applications in bioinformatics, finance, monitoring of complex systems have also motivated recent developments from the machine learning community [18,30,31].

Let us take the example of gait analysis, illustrated on the flowchart displayed on Fig. 1. In this context, a patient's movements are monitored with accelerometers and gyroscopes while performing simple activities, for instance walking at preferred speed, running or standing still. The objective is to objectively quantify gait characteristics [32–36]. The resulting signal is described as a succession of non-overlapping segments, each one corresponding to an activity and having its own gait characteristics. Insightful features from homogeneous phases can be extracted if the temporal boundaries of those segments are identified. This analysis therefore needs a preliminary processing of the signals: change point detection.

Change point detection methods are divided into two main branches: *online* methods, that aim to detect changes as soon as they occur in a real-time setting, and *offline* methods that retrospectively detect changes when all samples are received. The former task is often referred to as *event or anomaly detection*, while the latter is sometimes called *signal segmentation*.

In this article, we propose a survey of algorithms for the detection of multiple change points in multivariate time series. All reviewed methods presented in this paper address the problem of *offline* (also referred to as *retrospective* or *a posteriori*) change point detection, in which segmentation is performed after the signal has been collected. The objective of this article is to facilitate the search of a suitable detection method for a given application. In particular, focus is made on practical considerations such as implementations and procedures to calibrate the algorithms. This review also presents the mathematical properties of the main approaches, as well as the metrics to evaluate and compare their results. This article is linked with a Python scientific library called `ruptures` [37], that includes a modular and easy-to-use implementation of a subset of the methods presented in this paper.

[*] Corresponding author.
E-mail address: laurent.oudre@univ-paris13.fr (L. Oudre).

https://doi.org/10.1016/j.sigpro.2019.107299
0165-1684/© 2019 Elsevier B.V. All rights reserved.

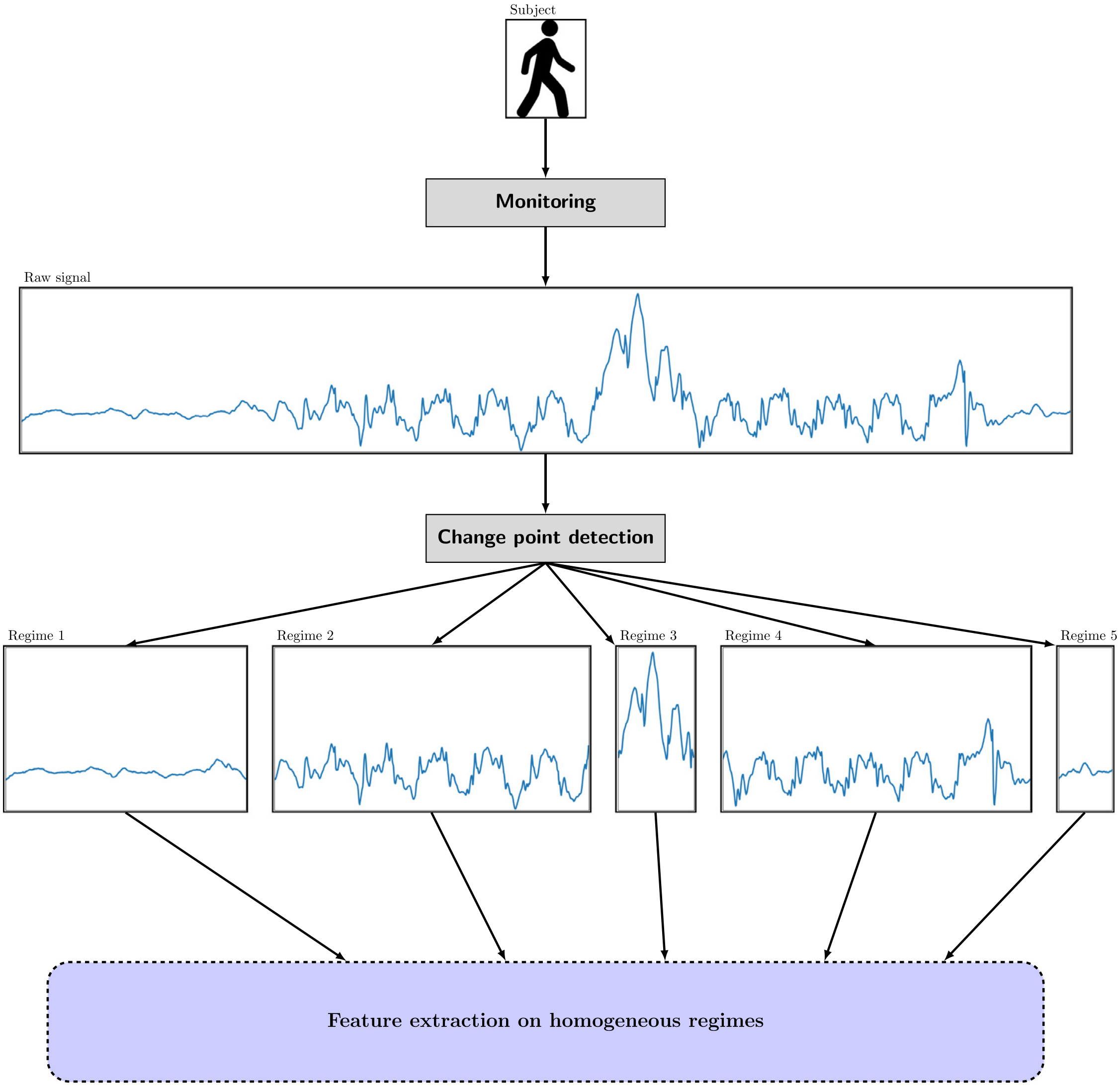

Fig. 1. Flowchart of a study scheme, for gait analysis.

2. Background

This section introduces the main concepts for change point detection, as well as the selection criteria and the outline of this review.

2.1. Notations

In the remainder of this article, we use the following notations. For a given signal $y = \{y_t\}_{t=1}^{T}$, the $(b-a)$-sample long sub-signal $\{y_t\}_{t=a+1}^{b}$ $(1 \leq a < b \leq T)$ is simply denoted $y_{a..b}$; the complete signal is therefore $y = y_{0..T}$. A set of indexes is denoted by a calligraphic letter: $\mathcal{T} = \{t_1, t_2, \dots\} \subset \{1, \dots, T\}$, and its cardinal is $|\mathcal{T}|$. For a set of indexes $\mathcal{T} = \{t_1, \dots, t_K\}$, the dummy indexes $t_0 := 0$ and $t_{K+1} := T$ are implicitly available.

2.2. Problem formulation

Let us consider a multivariate non-stationary random process $y = \{y_1, \dots, y_T\}$ that takes value in $\mathbb{R}^d$ $(d \geq 1)$ and has T samples. The signal y is assumed to be piecewise stationary, meaning that some characteristics of the process change abruptly at some unknown instants $t_1^* < t_2^* < \dots < t_{K^*}^*$. Change point detection consists in estimating the indexes t_k^*. Depending on the context, the number K^* of changes may or may not be known, in which case it has to be estimated too.

Formally, change point detection is cast as a model selection problem, which consists in choosing the best possible segmentation $\mathcal{T}$ according to a quantitative criterion $V(\mathcal{T}, y)$ that must be minimized. (The function $V(\mathcal{T}, y)$ is simply denoted $V(\mathcal{T})$ when it is obvious from the context that it refers to the signal y.) The choice of the criterion function $V(\cdot)$ depends on preliminary knowledge on the task at hand.

In this work, we make the assumption that the criterion function $V(\mathcal{T})$ for a particular segmentation is a sum of costs of all the segments that define the segmentation:

$$V(\mathcal{T}, y) := \sum_{k=0}^{K} c(y_{t_k..t_{k+1}}) \tag{1}$$

where $c(\cdot)$ is a cost function which measures goodness-of-fit of the sub-signal $y_{t_k..t_{k+1}} = \{y_t\}_{t_k+1}^{t_{k+1}}$ to a specific model. The “best segmentation” $\widehat{\mathcal{T}}$ is the minimizer of the criterion $V(\mathcal{T})$. In practice, depending on whether the number K^* of change points is known beforehand, change point detection methods fall into two categories.

- **Problem 1 : known number of changes** K**.** The change point detection problem with a fixed number K of change points consists in solving the following discrete optimization problem

$$\min_{|\mathcal{T}|=K} V(\mathcal{T}). \tag{P1}$$

- **Problem 2 : unknown number of changes.** The change point detection problem with an unknown number of change points consists in solving the following discrete optimization problem

$$\min_{\mathcal{T}} V(\mathcal{T}) + \text{pen}(\mathcal{T}) \tag{P2}$$

where $\text{pen}(\mathcal{T})$ is an appropriate measure of the complexity of a segmentation $\mathcal{T}$.

All change point detection methods considered in this work yield an exact or an approximate solution to either Problem 1 (P1) or Problem 2 (P2), with the function $V(\mathcal{T}, y)$ adhering to the format (1).

2.3. Selection criteria for the review

To better understand the strengths and weaknesses of change point detection methods, we propose to classify algorithms according to a comprehensive typology. Precisely, detection methods are expressed as the combination of the following three elements.

- **Cost function.** The cost function $c(\cdot)$ is a measure of "homogeneity". Its choice encodes the type of changes that can be detected. Intuitively, $c(y_{a.b})$ is expected to be low if the sub-signal $y_{a.b}$ is "homogeneous" (meaning that it does not contain any change point), and large if the sub-signal $y_{a.b}$ is "heterogeneous" (meaning that it contains one or several change points).
- **Search method.** The search method is the resolution procedure for the discrete optimization problems associated with Problem 1 (P1) and Problem 2 (P2). The literature contains several methods to efficiently solve those problems, in an exact fashion or in an approximate fashion. Each method strikes a balance between computational complexity and accuracy.
- **Constraint (on the number of change points).** When the number of changes is unknown (P2), a constraint is added, in the form of a complexity penalty $\text{pen}(\cdot)$ (P2), to balance out the goodness-of-fit term $V(\mathcal{T}, y)$. The choice of the complexity penalty is related to the amplitude of the changes to detect: with too "small" a penalty (compared to the goodness-of-fit) in (P2), many change points are detected, even those that are the result of noise. Conversely, too much penalization only detects the most significant changes, or even none.

This typology of change point detection methods is schematically shown on Fig. 2.

2.4. Limitations

The described framework, however general, does not encompass all published change point detection methods. In particular, Bayesian approaches are not considered in the remainder of this article, even though they provide state-of-the-art results in several domains, such as speech and sound processing. The most well-known Bayesian algorithm is the Hidden Markov Model (HMM) [38]. This model was later extended, for instance with Dirichlet processes [39,40] or product partition models [41,42]. The interested reader can find reviews of Bayesian approaches in [4,6].

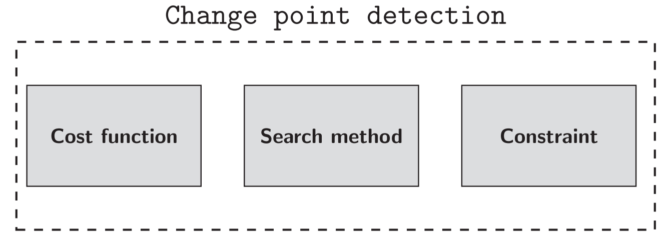

Fig. 2. Typology of change point detection methods described in this article. Reviewed algorithms are defined by three elements: a cost function, a search method and a constraint (on the number of change points).

Also, several literature reviews with different selection criteria can be found. Recent and important works include [43] which focuses on window-based detection algorithms. In particular, the authors use the quantity of samples needed to detect a change as a basis for comparison. Maximum likelihood and Bayes-type detection are reviewed, from a theoretical standpoint, in [8]. Existing asymptotic distributions for change point estimates are described for several statistical models. In [44], detection is formulated as a statistical hypothesis testing problem, and emphasis is put on the algorithmic and theoretical properties of several sequential mean-shift detection procedures.

2.5. Outline of the article

Before starting this review, we propose in Section 3 a detailed overview of the main mathematical tools that can be used for evaluating and comparing the change point detection methods. The organization of the remaining of this review article reflects the typology of change point detection methods, which is schematically shown on Fig. 2. Precisely, the three defining elements of a detection algorithm are reviewed separately. In Section 4, cost functions from the literature are presented, along with the associated signal model and the type of change that can be detected. Whenever possible, theoretical results on asymptotic consistency are also given. Section 5 lists search methods that efficiently solve the discrete optimizations associated with Problem 1 (P1) and Problem 2 (P2). Both exact and approximate methods are described. Constraints on the number of change points are reviewed in Section 6. A summary table of the literature review can be found in Section 8. Additional considerations to apply change point detection in real-world scenarios are given in Section 7. The last Section 9 is dedicated to the presentation of the Python package that goes with this article and propose a modular implementation of a subset of the approaches described in this article.

3. Evaluation

Change point detection methods can be evaluated either by proving some mathematical properties of the algorithms (such as consistency) in general case, or empirically by computing several metrics to assess the performances on a given dataset.

3.1. Consistency

A natural question when designing detection algorithms is the consistency of estimated change point indexes, as the number of samples T goes to infinity. In the literature, the "asymptotic setting" is intuitively described as follows: the observed signal y is regarded as a realization of a continuous-time process on an equispaced grid of size $1/T$, and "T goes to infinity" means that the spacing of the sampling grid converges to 0. Precisely, for all $\tau \in [0, 1]$, let $Y(\tau)$ denote an $\mathbb{R}^d$-valued random variable such that

$$y_t = Y(t/T) \quad \forall t = 1, \ldots, T. \tag{2}$$

The continuous-time process undergoes K^* changes in the probability distribution at the time instants $\tau_k^* \in (0,1)$. Those τ_k^* are related to the change point indexes t_k^* through the following relationship:

$$t_k^* = \lfloor T\tau_k^* \rfloor. \tag{3}$$

Generally, for a given change point index t_k, the associated quantity $\tau_k = t_k/T \in (0,1)$ is referred to as a change point *fraction*. In particular, the change point fractions τ_k^* $(k = 1, \ldots, K^*)$ of the time-continuous process Y are change point indexes of the discrete-time signal y. Note that in this asymptotic setting, the lengths of each regime of y increase linearly with T. The notion of asymptotic consistency of a change point detection method is formally introduced as follows.

Definition 1 (Asymptotic consistency)**.** A change point detection algorithm is said to be asymptotically consistent if the estimated segmentation $\widehat{\mathcal{T}} = \{\hat{t}_1, \hat{t}_2, \ldots\}$ satisfies the following conditions, when $T \longrightarrow +\infty$:

(i) $P(|\widehat{\mathcal{T}}| = K^*) \longrightarrow 1,$

(ii) $\frac{1}{T} \max\left\{ \max\limits_{\hat{t}\in\widehat{\mathcal{T}}} \min\limits_{t^*\in\mathcal{T}^\star} |\hat{t} - t^*|, \ \max\limits_{t^*\in\mathcal{T}^\star} \min\limits_{\hat{t}\in\widehat{\mathcal{T}}} |\hat{t} - t^*| \right\} \xrightarrow{p} 0,$

In Definition 1, the first condition is trivially verified when the number K^* of change points is known beforehand.

As for the second condition, it means that the estimated change point set $\widehat{\mathcal{T}}$ converges to $\mathcal{T}^\star$, for a certain measure, which is often referred to as the Hausdorff distance, and is formally introduced later in the article.

More precisely, it is the estimated change point fractions that are consistent, and not the indexes themselves. In general, distances $|\hat{t} - t^*|$ between true change point indexes and their estimated counterparts do not converge to 0, even for simple models [18,45–47]. As a result, consistency results in the literature only deal with change point fractions.

3.2. Evaluation metrics

Several metrics from the literature are presented below. Each metric correspond to one of the previously listed criteria by which segmentation performances are assessed. 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 $\widehat{\mathcal{T}} = \{\hat{t}_1, \ldots, \hat{t}_{\widehat{K}}\}$. Note that that the cardinals of each set, $K^\star$ and $\widehat{K}$, are not necessarily equal.

3.2.1. Annotation error

The annotation error, denoted Δ_{AE}, is simply the difference between the predicted number of change points $|\widehat{\mathcal{T}}|$ $(=\widehat{K})$ and the true number of change points $|\mathcal{T}^\star|$ $(=K^\star)$:

$$\Delta_{\mathrm{AE}}(\mathcal{T}^\star, \widehat{\mathcal{T}}) := |\widehat{K} - K^\star|. \tag{4}$$

This metric can be used to discriminate detection methods when the number of changes is unknown.

3.2.2. Hausdorff

The Hausdorff metric, denoted Δ_{HA}, measures the robustness of detection methods [47,48]. Formally, it is equal to the greatest temporal distance between a change point and its prediction:

$$\Delta_{\mathrm{HA}}\left(\mathcal{T}^\star, \widehat{\mathcal{T}}\right) := \max\left\{ \max_{\hat{t}\in\widehat{\mathcal{T}}} \min_{t^\star\in\mathcal{T}^\star} |\hat{t} - t^\star|, \ \max_{t^\star\in\mathcal{T}^\star} \min_{\hat{t}\in\widehat{\mathcal{T}}} |\hat{t} - t^\star| \right\}. \tag{5}$$

It is the worst error made by the algorithm that produced $\widehat{\mathcal{T}}$ and is expressed in number of samples. If this metric is equal to zero, both breakpoint sets are equal; it is large when a change point from either $\mathcal{T}^\star$ or $\widehat{\mathcal{T}}$ is far from every change point of $\widehat{\mathcal{T}}$ or $\mathcal{T}^\star$ respectively. Over-segmentation as well as under-segmentation is penalized. An illustrative example is displayed on Fig. 3.

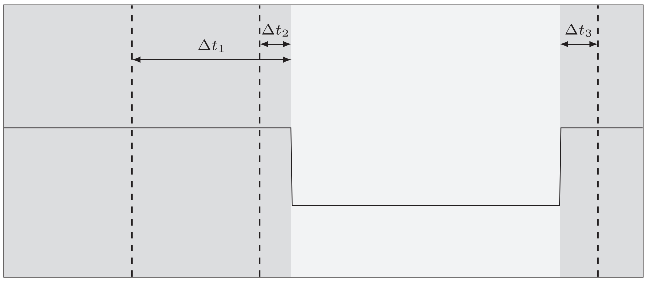

Fig. 3. Hausdorff metric: Δ_{HA}. Alternating gray areas mark the segmentation $\mathcal{T}^\star$; dashed lines mark the segmentation $\widehat{\mathcal{T}}$. Here, Δ_{HA} is equal to $\Delta t_1 = \max(\Delta t_1, \Delta t_2, \Delta t_3)$.

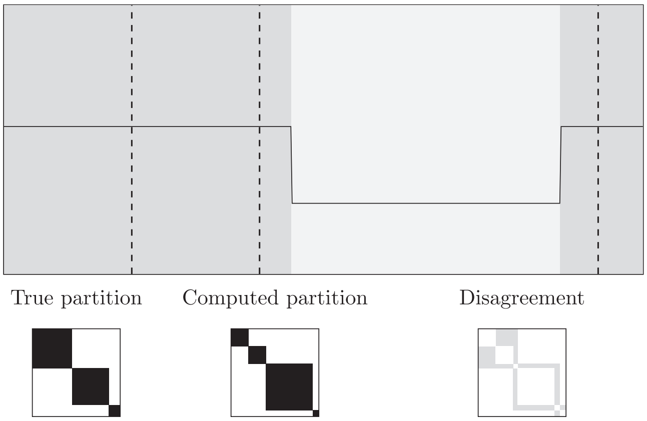

Fig. 4. Rand index: Δ_{RI}. Top: alternating gray areas mark the segmentation $\mathcal{T}^\star$; dashed lines mark the segmentation $\widehat{\mathcal{T}}$. Below: representations of associated adjacency matrices and disagreement matrix. The adjacency matrix of a segmentation is the $T \times T$ binary matrix with coefficient (s, t) equal to 1 if s and t belong to the same segment, 0 otherwise. The disagreement matrix is the $T \times T$ binary matrix with coefficient (s, t) equal to 1 where the two adjacency matrices disagree, and 0 otherwise. The value of Δ_{RI} is equal to the white area (where coefficients are 0) of the disagreement matrix.

3.2.3. Rand index

Accuracy can be measured by the Rand index, denoted Δ_{RI}, which is the average similarity between the predicted breakpoint set $\widehat{\mathcal{T}}$ and the ground truth $\mathcal{T}^\star$ [30]. Intuitively, it is equal to the number of agreements between two segmentations. An agreement is a pair of indexes which are either in the same segment according to both $\widehat{\mathcal{T}}$ and $\mathcal{T}^\star$ or in different segments according to both $\widehat{\mathcal{T}}$ and $\mathcal{T}^\star$. Formally, for a breakpoint set $\mathcal{T}$, the set of grouped indexes and the set of non-grouped indexes are respectively $\mathrm{gr}(\mathcal{T})$ and $\mathrm{ngr}(\mathcal{T})$:

$$\begin{aligned}\mathrm{gr}(\mathcal{T}) &:= \{(s,t), 1 \le s < t \le T \text{ s.t. } s \text{ and } t \text{ belong to the same} \\ &\qquad \text{segment according to } \mathcal{T}\ \}, \\ \mathrm{ngr}(\mathcal{T}) &:= \{(s,t), 1 \le s < t \le T \text{ s.t. } s \text{ and } t \text{ belong to different} \\ &\qquad \text{segments according to } \mathcal{T}\ \}.\end{aligned}$$

The Rand index is then defined as follows:

$$\Delta_{\mathrm{RI}}(\mathcal{T}^\star, \widehat{\mathcal{T}}) := \frac{|\mathrm{gr}(\widehat{\mathcal{T}}) \cap \mathrm{gr}(\mathcal{T}^\star)| + |\mathrm{ngr}(\widehat{\mathcal{T}}) \cap \mathrm{ngr}(\mathcal{T}^\star)|)}{T(T-1)}. \tag{6}$$

It is normalized between 0 (total disagreement) and 1 (total agreement). Originally, the Rand index has been introduced to evaluate clustering methods [30,47]. An illustrative example is displayed on Fig. 4.

3.2.4. F1-Score

Another measure of accuracy is the F1-score, denoted Δ_{F1}. Precision is the proportion of predicted change points that are true change points. Recall is the proportion of true change points that

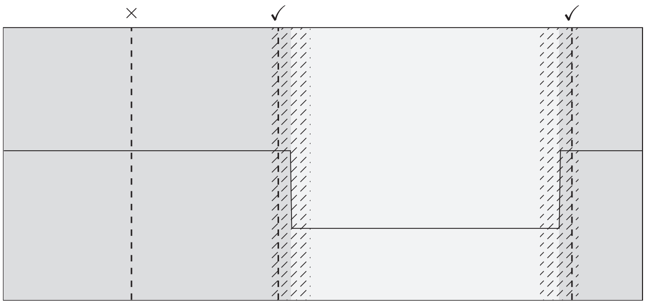

Fig. 5. F1-score: Δ_{F1}. Alternating gray areas mark the segmentation $\mathcal{T}^\star$; dashed lines mark the segmentation $\widehat{\mathcal{T}}$; dashed areas mark the allowed margin of error around true change points. Here, precision and recall are equal to 2/3 and 2/2 respectively, and Δ_{F1} is equal to 4/5.

are well predicted. A breakpoint is considered detected up to a user-defined margin of error $M > 0$; true positives Tp are true change points for which there is an estimated one at less than M samples, *i.e.*

$$Tp(\mathcal{T}^\star, \widehat{\mathcal{T}}) := \{t^\star \in \mathcal{T}^\star \mid \exists \hat{t} \in \widehat{\mathcal{T}} \ \text{ s.t. } |\hat{t} - t^\star| < M\}. \tag{7}$$

Precision Prec and recall Rec are then given by

$$Prec(\mathcal{T}^\star, \widehat{\mathcal{T}}) := |Tp(\mathcal{T}^\star, \widehat{\mathcal{T}})|/\widehat{K} \quad \text{and} \quad Rec(\mathcal{T}^\star, \widehat{\mathcal{T}}) := |Tp(\mathcal{T}^\star, \widehat{\mathcal{T}})|/K^\star. \tag{8}$$

Precision and Recall are well-defined (ie. between 0 and 1) if the margin M is smaller than the minimum spacing between two true change point indexes $t_k^\star$ and $t_{k+1}^\star$. Over-segmentation of a signal causes the precision to be close to zero and the recall close to one. Under-segmentation has the opposite effect. The F1 score is the harmonic mean of precision Prec and recall Rec:

$$\Delta_{\text{F1}}(\mathcal{T}^\star, \widehat{\mathcal{T}}) := 2 \times \frac{\text{Prec}(\mathcal{T}^\star, \widehat{\mathcal{T}}) \times \text{Rec}(\mathcal{T}^\star, \widehat{\mathcal{T}})}{\text{Prec}(\mathcal{T}^\star, \widehat{\mathcal{T}}) + \text{Rec}(\mathcal{T}^\star, \widehat{\mathcal{T}})}. \tag{9}$$

Its best value is 1 and its worse value is 0. An illustrative example is displayed on Fig. 5.

4. Models and cost functions

This section presents the first defining element of change detection methods, namely the cost function. In most cases, cost functions are derived from a signal model. In the following, models and their associated cost function are organized in two categories: parametric and non-parametric, as schematically shown in Fig. 6. For each model, the most general formulation is first given, then special cases, if any, are described. A summary table of all reviewed costs can be found at the end of this section.

4.1. Parametric models

Parametric detection methods focus on changes in a finite-dimensional parameter vector. Historically, they were the first to be introduced, and remain extensively studied in the literature.

4.1.1. Maximum likelihood estimation

Maximum likelihood procedures are ubiquitous in the change point detection literature. They generalize a large number of models and cost functions, such as mean-shifts and scale shifts in normally distributed data [2,49–51], changes in the rate parameter of Poisson distributed data [39], etc. In the general setting of maximum likelihood estimation for change detection, the observed signal $y = \{y_1, \dots, y_T\}$ is composed of independent random variables, such that

$$y_t \sim \sum_{k=0}^{K^*} f(\cdot|\theta_k) \mathbb{1}\left(t_k^* < t \leq t_{k+1}^*\right) \tag{M1}$$

where the t_k^* are change point indexes, the $f(\cdot|\theta)$ are probability density functions parametrized by the vector-valued parameter θ, and the θ_k are parameter values. In other words, the signal y is modelled by iid variables with piecewise constant distribution. The parameter θ represents a quantity of interest whose value changes abruptly at the unknown instants t_k^*, which are to be estimated. Under this setting, change point detection is equivalent to maximum likelihood estimation if the sum of cost $V(\mathcal{T}, y)$ is equal to the negative log-likelihood. The corresponding cost function, denoted $c_{\text{i.i.d.}}$, is defined as follows.

Cost function 1 ($c_{\text{i.i.d.}}$)**.** For a given parametric family of distribution densities $\{f(\cdot|\theta)|\theta \in \Theta\}$ where Θ is a compact subset of $\mathbb{R}^p$ (for a certain p), the cost function $c_{\text{i.i.d.}}$ is defined by

$$c_{\text{i.i.d.}}(y_{a.b}) := -\sup_{\theta} \sum_{t=a+1}^{b} \log f(y_t|\theta). \tag{C1}$$

Model M1 and the related cost function $c_{\text{i.i.d.}}$ encompasses a large number of change point methods. Note that, in this context, the family of distributions must be known before performing the detection, usually thanks to prior knowledge on the data. Historically, the Gaussian distribution was first used, to model mean-shifts [52–54] and scale shifts [39,50,55]. A large part of the literature then evolved towards other parametric distributions, most notably resorting to distributions from the general exponential family [15,25,56].

From a theoretical point of view, asymptotic consistency, as described in Definition 1, has been demonstrated, in the case of *a single change point*, first with Gaussian distribution (fixed variance), then for several specific distributions, e.g. Gaussian with mean and scale shifts [3,6,51,57], discrete distributions [49], etc. The case with *multiple change points* has been tackled later. For certain distributions (e.g. Gaussian), the solutions of the change point detection problems (P1) (known number of change points) and (P2) (unknown number of change points) have been proven to be asymptotically consistent [58]. The general case of multiple change points and a generic distribution family has been addressed decades after the change detection problem has been introduced: the solution of the change point detection problem with a known number of changes and a cost function set to $c_{\text{i.i.d.}}$ is asymptotically consistent [59]. This is true if certain assumptions are satisfied: (i) the signal follows the model (M1) for a distribution family that verifies some regularity assumptions (which are no different from the assumptions needed for generic maximum likelihood estimation, without any change point) and (ii) technical assumptions on the value of the cost function on homogeneous and heterogeneous sub-signals. As an example, distributions from the exponential family satisfy those assumptions.

Related cost functions. The general model (M1) has been applied with different families of distributions. We list below three notable examples and the associated cost functions: change in mean, change in mean and scale, and change in the rate parameter of count data.

- The mean-shift model is the earliest and one of the most studied model in the change point detection literature [2,53,60–62]. Here, the distribution is Gaussian, with fixed variance. In other words, the signal y is simply a sequence of independent normal random variables with piecewise constant mean and same variance. In this context, the cost function $c_{\text{i.i.d.}}$ becomes c_{L_2}, defined below. This cost

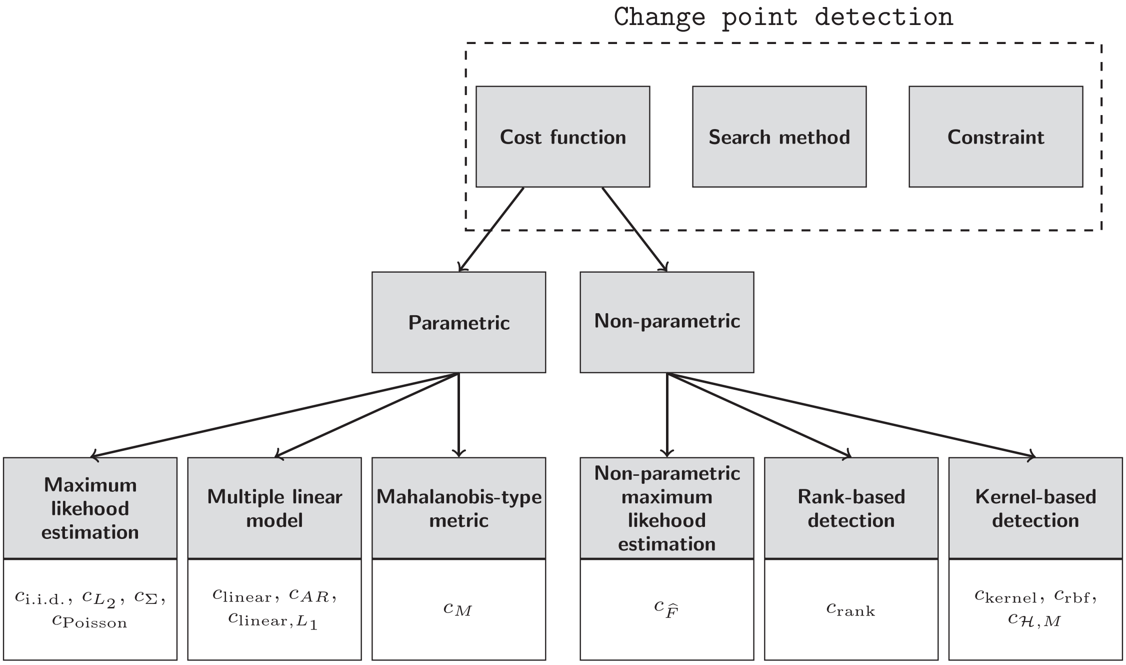
Change point detection
Cost function
Search method
Constraint
Parametric
Non-parametric
Maximum likehood estimation
Multiple linear model
Mahalanobis-type metric
Non-parametric maximum likehood estimation
Rank-based detection
Kernel-based detection
$c_{\text{i.i.d.}}$, c_{L_2}, c_{Σ}, c_{Poisson}
c_{linear}, c_{AR}, c_{linear,L_1}
c_M
$c_{\widehat{F}}$
c_{rank}
c_{kernel}, c_{rbf}, $c_{\mathcal{H},M}$

Fig. 6. Typology of the cost functions described in Section 4.

function is also referred to as the quadratic error loss and has been applied for instance on DNA array data [15] and geology signals [6].

Cost function 2 (c_{L_2})**.** The cost function c_{L_2} is given by

$$c_{L_2}(y_{a..b}) := \sum_{t=a+1}^{b} \|y_t - \bar{y}_{a..b}\|_2^2 \tag{C2}$$

where $\bar{y}_{a..b}$ is the empirical mean of the sub-signal $y_{a..b}$.

- A natural extension to the mean-shift model consists in letting the variance abruptly change as well. In this context, the cost function $c_{\text{i.i.d.}}$ becomes c_{Σ}, defined below. This cost function can be used to detect changes in the first two moments of random (not necessarily Gaussian) variables, even though it is the Gaussian likelihood that is plugged in $c_{\text{i.i.d.}}$ [8,49]. It has been applied for instance on stock market time series [49], biomedical data [6], and electric power consumption monitoring [63].

Cost function 3 (c_{Σ})**.** The cost function c_{Σ} is given by

$$c_{\Sigma}(y_{a..b}) : = (b-a)\log\det\widehat{\Sigma}_{a..b} + \sum_{t=a+1}^{b} (y_t - \bar{y}_{a..b})'\widehat{\Sigma}_{a..b}^{-1}(y_t - \bar{y}_{a..b}) \tag{C3}$$

where $\bar{y}_{a..b}$ and $\widehat{\Sigma}_{a..b}$ are respectively the empirical mean and the empirical covariance matrix of the sub-signal $y_{a..b}$.

- Change point detection has also be applied on count data modelled by a Poisson distribution [39,64]. More precisely, the signal y is a sequence of independent Poisson distributed random variables with piecewise constant rate parameter. In this context, the cost function $c_{\text{i.i.d.}}$ becomes c_{Poisson}, defined below.

Cost function 4 (c_{Poisson})**.** The cost function c_{Poisson} is given by

$$c_{\text{Poisson}}(y_{a..b}) := -(b-a)\bar{y}_{a..b}\log\bar{y}_{a..b} \tag{C4}$$

where $\bar{y}_{a..b}$ is the empirical mean of the sub-signal $y_{a..b}$.

Remark 1. A model slightly more general than (M1) can be formulated by letting the signal samples to be dependant and the distribution function $f(\cdot|\theta)$ to change over time. This can in particular model the presence of unwanted changes in the statistical properties of the signal (for instance in the statistical structure of the noise [49]). The function $f(\cdot|\theta)$ is replaced in (M1) by a sequence of distribution functions $f_t(\cdot|\theta)$ which are not assumed to be identical for all indexes t. Changes in the functions f_t are considered nuisance parameters and only the variations of the parameter θ must be detected. Properties on the asymptotic consistency of change point estimates can be obtained in this context. We refer the reader to [49,65] for theoretical results.

4.1.2. Piecewise linear regression

Piecewise linear models are often found, most notably in the econometrics literature, to detect so-called “structural changes” [66–68]. In this context, a linear relationship between a response variable and covariates exists, and this relationship changes abruptly at some unknown instants. Formally, the observed signal y follows a piecewise linear model with change points located at the t_k^*:

$$\forall\ t,\ t_k^* < t \le t_{k+1}^*,\quad y_t = x_t' u_k + z_t' v + \varepsilon_t \quad (k = 0, \dots, K^*) \tag{M2}$$

where the $u_k \in \mathbb{R}^p$ and $v \in \mathbb{R}^q$ are unknown regression parameters and ε_t is noise. Under this setting, the observed signal y is regarded as a univariate response variable (i.e. $d = 1$) and the signals $x = \{x_t\}_{t=1}^T$ and $z = \{z_t\}_{t=1}^T$ are observed covariates, respectively $\mathbb{R}^p$-valued and $\mathbb{R}^q$-valued.

In the literature, Model (M2) is also known as a *partial* structural change model because the linear relationship between y and x changes abruptly, while the linear relationship between y and z remains constant. The *pure* structural change model is obtained by removing the term $z_t' v$ from (M2). The formulation (M2) generalizes several well-known models such as the autoregressive (AR) model [12,69], multiple regressions [68,70], etc. A more general formulation of (M2) that can accommodate a multivariate response variable y exists [71], but is more involved, from a notational standpoint.

The procedure to detect *pure* structural changes amounts to fitting a linear regression on each segment of the signal. To that end, the sum of costs is made equal to the sum of squared residuals. The corresponding cost function, denoted c_{linear}, is defined as follows.

Cost function 5 (c_{linear})**.** For a signal y (response variable) and covariates x and z, the cost function c_{linear} is defined by

$$c_{\text{linear}}(y_{a..b}) := \min_{u \in \mathbb{R}^p} \sum_{t=a+1}^{b} \left(y_t - x_t' u\right)^2. \tag{C5}$$

Detecting *partial* structural changes is more complex because the regression parameters contained in v are shared by all segments, and therefore, the associated sum of squared residuals is no longer a sum of segment costs, as in (1). Approximate procedures have been proposed to minimize the sum of residuals. For instance, one can iteratively alternate between the two following steps: keep v fixed and detect pure structural changes in $y_t - z_t' v$ (using c_{linear}), then keep the segmentation fixed and minimize over v and the u_k of each segment [72]. While finding the global minimum is not guaranteed, the authors argued, using numerical simulations, that their method "has very rapid convergence" (no more than two iterations were enough to find the minimum).

From a theoretical point of view, piecewise linear models are extensively studied in the context of change point detection by a series of important contributions [14,66–70,73–77]. When the number of changes is known, the most general consistency result can be found in [14]. A multivariate extension of this result has been demonstrated in [71]. As for the more difficult situation of an unknown number of changes, statistical tests have been proposed for a single change [78] and multiple changes [74]. All of those results are obtained under various sets of general assumptions on the distributions of the covariates and the noise. The most general of those sets can be found in [79]. Roughly, in addition to some technical assumptions, it imposes the processes x and z to be weakly stationary within each regime, and precludes the noise process to have a unit root.

A few articles from the literature adopt a more complex view of partial changes, in which all covariates are pooled and it is not known which ones, in (M2), belong to x (i.e. their linear relationships with the response variable y abruptly change over time) or to z (i.e. their linear relationships with the response variable y remain constant over time). Since treating partial structural changes as pure structural changes can lead to underestimation of the number of changes [80], better methods are needed. Often, segmentations are computed for all combinations of regressors in z or in x, then the optimal segmentation and combination of covariates are chosen according a statistical criterion [79]. A more recent work [80] compares a number of procedures and propose extensions, based on a finely tuned Bayesian Information Criterion (BIC).

Related cost functions. In the rich literature related to piecewise linear models, the cost function c_{linear} has been applied and extended in several different settings. Two related cost functions are listed below.

- The first one is c_{linear,L_1}, which was introduced in order to accommodate certain noise distributions with heavy tails [67,73] and is defined as follows. Again, it is designed to detect pure structural changes.

  Cost function 6 (c_{linear,L_1})**.** For a signal y (response variable) and covariates x and z, the cost function c_{linear,L_1} is defined by

  $$c_{\text{linear},L_1}(y_{a..b}) := \min_{u \in \mathbb{R}^p} \sum_{t=a+1}^{b} |y_t - x_t' u - z_t' v|. \tag{C6}$$

  The difference between c_{linear,L_1} and c_{linear} lies in the norm used to measure errors: c_{linear,L_1} is based on a least absolute deviations criterion, while c_{linear} is based on a least squares criterion. As a result, c_{linear,L_1} is often applied on data with noise distributions with heavy tails [8,25]. In practice, the cost function c_{linear,L_1} is computationally less efficient than the cost function c_{linear}, because the associated minimization problem (C6) has no analytical solution. Nevertheless, the cost function c_{linear,L_1} is often applied on economic and financial data [66–68]. For instance, changes in several economic parameters of the G-7 growth have been investigated using a piecewise linear model and c_{linear,L_1} [81].
- The second cost function related to c_{linear} has been introduced to deal with piecewise autoregressive signals. The autoregressive model is a popular representation of random processes, where each variable depends linearly on the previous variables. The associated cost function, denoted c_{AR}, is defined as follows.

  Cost function 7 (c_{AR})**.** For a signal y and an order $p \geq 1$, the cost function c_{AR} is defined by

  $$c_{\text{AR}}(y_{a..b}) := \min_{u \in \mathbb{R}^p} \sum_{t=a+1}^{b} \left\| y_t - x_t' u \right\|^2 \tag{C7}$$

  where $x_t := [y_{t-1}, y_{t-2}, \ldots, y_{t-p}]$ is the vector of lagged samples.

  The piecewise autoregressive model is a special case of the generic piecewise linear model, where the term $z_t' v$ is removed (yielding a pure structural change model) and the covariate signal x is equal to the signal of lagged samples. The resulting cost function c_{AR} is able to detect shifts in the autoregressive coefficients of a non-stationary process [21,69]. This model has been applied on EEG/ECG time series [71], functional magnetic resonance imaging (fMRI) time series [82] and speech recognition tasks [12].

4.1.3. Mahalanobis-type metric

The cost function c_{L_2} (C2), adapted for mean-shift detection, can be extended through the use of Mahalanobis-type seminorm. Formally, for any symmetric positive semi-definite matrix $M \in \mathbb{R}^{d \times d}$, the associated seminorm $\|\cdot\|_M$ is given by:

$$\|y_t\|_M^2 := y_t' M y_t \tag{10}$$

for any sample y_t. The resulting cost function c_M is defined as follows.

Cost function 8 (c_M)**.** The cost function c_M, parametrized by a symmetric positive semi-definite matrix $M \in \mathbb{R}^{d \times d}$, is given by

$$c_M(y_{a..b}) := \sum_{t=a+1}^{b} \|y_t - \bar{y}_{a..b}\|_M^2 \tag{C8}$$

where $\bar{y}_{a..b}$ is the empirical mean of the sub-signal $y_{a..b}$.

Intuitively, measuring distances with the seminorm $\|\cdot\|_M$ is equivalent to applying a linear transformation on the data and using the regular (Euclidean) norm $\|\cdot\|$. Indeed, decomposing the matrix $M = U'U$ yields:

$$\|y_t - y_s\|_M^2 = \|Uy_t - Uy_s\|^2. \tag{11}$$

Originally, the metric matrix M was set equal to the inverse of the covariance matrix, yielding the Mahalanobis metric [83], ie

$$M = \widehat{\Sigma}^{-1} \tag{12}$$

where $\widehat{\Sigma}$ is the empirical covariance matrix of the signal y. By using c_M, shifts in the mean of the transformed signal can be detected. In practice, the transformation U (or equivalently, the matrix M) is chosen to highlight relevant changes. This cost function generalizes all linear transformations of the data samples. In the context of change point detection, most of the transformations are unsupervised, for instance principal component analysis or linear discriminant analysis [84]. Supervised strategies are more rarely found, even though there exist numerous methods to learn a task-specific matrix M in the context of supervised classification [84–86]. Those strategies fall under the umbrella of metric learning algorithms. In the change point detection literature, there is only one work that proposes a supervised procedure to calibrate a metric matrix M [30]. In this contribution, the authors use a training set of annotated signals (meaning that an expert has provided the change point locations) to learn M iteratively. Roughly, at each step, a new matrix M is generated in order to improve change point detection accuracy on the training signals. However, using the cost function c_M is not adapted to certain applications, where a linear treatment of the data is insufficient. In that situation, a well-chosen non-linear transformation of the data samples must be applied beforehand [30].

4.2. Non-parametric models

When the assumptions of parametric models are not adapted to the data at hand, non-parametric change point detection methods can be more robust. Three major approaches are presented here, each based on different non-parametric statistics, such as the empirical cumulative distribution function, rank statistics and kernel estimation.

Signal model. Assume that the observed signal $y = \{y_1, \dots, y_T\}$ is composed of independent random variables, such that

$$y_t \sim \sum_{k=0}^{K^*} F_k \ \mathbb{1}\left(t_k^* < t \le t_{k+1}^*\right) \tag{M3}$$

where the t_k^* are change point indexes and the F_k are cumulative distribution functions (c.d.f.), not necessarily parametric as in (M1). Under this setting, the sub-signal $y_{t_k^*..t_{k+1}^*}$, bounded by two change points, is composed of iid variables with c.d.f. F_k. When the F_k belong to a known parametric distribution family, change point detection is performed with the MLE approach described in Section 4.1.1, which consists in applying the cost function $c_{\text{i.i.d.}}$. However, this approach is not possible when the distribution family is either non-parametric or not known beforehand.

4.2.1. Non-parametric maximum likelihood

The first non-parametric cost function example, denoted $c_{\widehat{F}}$, has been introduced for the *single* change point detection problem in [87] and extended for *multiple* change points in [88]. It relies on the empirical cumulative distribution function, estimated on sub-signals. Formally, the signal is assumed to be univariate (ie $d = 1$) and the empirical cdf on the sub-signal $y_{a.b}$ is given by [89]

$$\forall u \in \mathbb{R}, \quad \widehat{F}_{a.b}(u) := \frac{1}{b-a}\left[\sum_{t=a+1}^{b} \mathbb{1}(y_t < u) + 0.5 \times \mathbb{1}(y_t = u)\right]. \tag{13}$$

In order to derive a log-likelihood function that does not depend on the probability distribution of the data, ie the $f(\cdot|\theta_k)$, the authors use the following fact: for a fixed $u \in \mathbb{R}$, the empirical cdf $\widehat{F}$ of n iid random variables, distributed from a certain cdf F is such that $n\widehat{F}(u) \sim \text{Binomial}(n, F(u))$ [88]. This observation, combined with careful summation over u, allows a distribution-free maximum likelihood estimation. The resulting cost function $c_{\widehat{F}}$ is defined as follows. Interestingly, this strategy was first introduced to design non-parametric two-sample statistical tests, which were experimentally shown to be more powerful than classical tests such as Kolmogorov-Smirnov and CramÃ©r-von Mises [87,90].

Cost function 9 ($c_{\widehat{F}}$). The cost function $c_{\widehat{F}}$ is given by

$$c_{\widehat{F}}(y_{a.b}) := -(b-a) \sum_{u=1}^{T} \frac{\widehat{F}_{a.b}(u)\log \widehat{F}_{a.b}(u) + \left(1-\widehat{F}_{a.b}(u)\right)\log\left(1-\widehat{F}_{a.b}(u)\right)}{(u-0.5)(T-u+0.5)} \tag{C9}$$

where the empirical cdf $\widehat{F}_{a.b}$ is defined by (13).

From a theoretical point of view, asymptotic consistency of change point estimates is verified, when the number of change points is either known or unknown [88]. However, solving either one of the detection problems can be computationally intensive, because calculating the value of the cost function $c_{\widehat{F}}$ on one sub-signal requires to sum T terms, where T is the signal length. As a result, the total complexity of change point detection is of the order of $\mathcal{O}(T^3)$ [88]. To cope with this computational burden, several preliminary steps are proposed. For instance, irrelevant change point indexes can be removed before performing the detection, thanks to a screening step [88]. Also, the cost function $c_{\widehat{F}}$ can be approximated, by summing, in (C9), over a few (carefully chosen) terms, instead of T terms originally [89]. Thanks to those implementation techniques, the cost function $c_{\widehat{F}}$ has been applied on DNA sequences [88] and heart-rate monitoring signals [89].

4.2.2. Rank-based detection

In statistical inference, a popular strategy to derive distribution-free statistics is to replaced the data samples by their ranks within the set of pooled observations [28,91,92]. In the context of change point detection, this strategy has first been applied to detect a *single* change point [28,29], and then has been extended by [93] to find *multiple* change points. The associated cost function, denoted c_{rank}, is defined as follows. Formally, it relies on the centered $\mathbb{R}^d$-valued "rank signal" $r = \{r_t\}_{t=1}^T$, given by

$$r_{t,j} := \sum_{s=1}^{T} \mathbb{1}(y_{s,j} \le y_{t,j}) \ - \ \frac{T+1}{2}, \quad \forall 1 \le t \le T, \ \forall 1 \le j \le d. \tag{14}$$

In other words, $r_{t,j}$ is the (centered) rank of the j^{th} coordinate of the t^{th} sample, ie $y_{t,j}$, among the $\{y_{1,j}, y_{2,j}, \dots, y_{T,j}\}$.

Cost function 10 (c_{rank}). The cost function c_{rank} is given by

$$c_{\text{rank}}(y_{a.b}) := -(b-a) \ \bar{r}_{a.b}' \ \widehat{\Sigma}_r^{-1} \ \bar{r}_{a.b} \tag{C10}$$

where the signal r is defined in (14), $\bar{r}_{a.b}$ is the empirical mean of the sub-signal $\{r_t\}_{t=a+1}^b$ and $\widehat{\Sigma}_r \in \mathbb{R}^{d\times d}$ is the following matrix

$$\widehat{\Sigma}_r := \frac{1}{T}\sum_{t=1}^{T}(r_t + \mathbf{1_d}/2)(r_t + \mathbf{1_d}/2)' \tag{15}$$

with $\mathbf{1}_d \in \mathbb{R}^d$ the all-ones vector. Intuitively, c_{rank} measures changes in the joint behaviour of the marginal rank statistics of each coordinate, which are contained in r. One of the advantages of this cost function is that it is invariant under any monotonic transformation of the data. Several well-known statistical hypothesis testing procedures are based on this scheme, for instance the Wilcoxon-Mann-Whitney test [94], the Friedman test [95], the Kruskal-Wallis test [96], and several others [91,92]. From a computational point of view, two steps must be performed before the change point detection: the calculation of the rank statistics, in $\mathcal{O}(dT\log T)$ operations, and the calculation of the matrix $\widehat{\Sigma}_r$, in $\mathcal{O}(d^2T + d^3)$ operations. The resulting algorithm has been applied on DNA sequences [93] and network traffic data [28,29].

4.2.3. Kernel-based detection

A kernel-based method has been proposed by [97] to perform change point detection in a non-parametric setting. To that end, the original signal y is mapped onto a reproducing Hilbert space (rkhs) $\mathcal{H}$ associated with a user-defined kernel function $k(\cdot,\cdot):\mathbb{R}^d\times\mathbb{R}^d\to\mathbb{R}$. The mapping function $\phi:\mathbb{R}^d\to\mathcal{H}$ onto this rkhs is implicitly defined by $\phi(y_t)=k(y_t,\cdot)\in\mathcal{H}$, resulting in the following inner-product and norm:

$$\langle\phi(y_s)|\phi(y_t)\rangle_{\mathcal{H}}=k(y_s,y_t)\quad\text{and}\quad\|\phi(y_t)\|^2_{\mathcal{H}}=k(y_t,y_t)\tag{16}$$

for any samples $y_s,y_t\in\mathbb{R}^d$. The associated cost function, denoted c_{kernel}, is defined as follows. This kernel-based mapping is central to many machine learning developments such as support vector machine or clustering [98,99].

Cost function 11 (c_{kernel})**.** For a given kernel function $k(\cdot,\cdot):\mathbb{R}^d\times\mathbb{R}^d\to\mathbb{R}$, the cost function c_{kernel} is given by

$$c_{\text{kernel}}(y_{a.b}):=\sum_{t=a+1}^{b}\|\phi(y_t)-\bar{\mu}_{a.b}\|^2_{\mathcal{H}}\tag{C11}$$

where $\bar{\mu}_{a.b}\in\mathcal{H}$ is the empirical mean of the embedded signal $\{\phi(y_t)\}_{t=a+1}^{b}$ and $\|\cdot\|_{\mathcal{H}}$ is defined in (16).

Remark 2 (Computing the cost function)**.** Thanks to the well-known "kernel trick", the explicit computation of the mapped data samples $\phi(y_t)$ is not required to calculate the cost function value [100]. Indeed, after simple algebraic manipulations, $c_{\text{kernel}}(y_{a.b})$ can be rewritten as follows:

$$c_{\text{kernel}}(y_{a.b})=\sum_{t=a+1}^{b}k(y_t,y_t)\;-\;\frac{1}{b-a}\sum_{s,t=a+1}^{b}k(y_s,y_t).\tag{17}$$

Remark 3 (Intuition behind the cost function)**.** Intuitively, the cost function c_{kernel} is able to detect mean-shifts in the transformed signal $\{\phi(y_t)\}_t$. Its use is motivated in the context of Model M3 by the fact that, under certain conditions on the kernel function, changes in the probability distribution coincide with mean-shifts in the transformed signal. This connection has been investigated in several works on kernel methods [98,99,101,102].

Formally, let $\mathbb{P}$ denote a Borel probability measure defined over $\mathbb{R}^d$ and assume that $k(\cdot,\cdot)$ is measurable and $\mathbb{E}_{X\sim\mathbb{P}}[k(X,X)]<+\infty$. Then there exists a unique element $\mu_{\mathbb{P}}\in\mathcal{H}$ [99], called the mean embedding (of $\mathbb{P}$), such that

$$\mu_{\mathbb{P}}=\mathbb{E}_{X\sim\mathbb{P}}\,[\phi(X)].\tag{18}$$

In addition, under certain conditions on the kernel $k(\cdot,\cdot)$, the mapping $\mathbb{P}\mapsto\mu_{\mathbb{P}}$ is injective (in which case the kernel is said to be *characteristic*), meaning that

$$\forall\mathbb{P},\mathbb{Q},\quad\mu_{\mathbb{P}}=\mu_{\mathbb{Q}}\Rightarrow\mathbb{P}=\mathbb{Q},\tag{19}$$

where $\mathbb{Q}$ denotes a Borel probability measure defined over $\mathbb{R}^d$.

In order to determine if a kernel is characteristic (and therefore, useful for change point detection), several conditions can be found in the literature [98,99,101]. For instance, if a kernel $k(\cdot,\cdot)$ is translation invariant (meaning that $k(y_s,y_t)=\psi(y_s-y_t)\ \forall s,t$, where ψ is a bounded continuous positive definite function on $\mathbb{R}^d$) and verifies a condition on its Fourier transform, then it is characteristic [101]. This condition is verified by the commonly used Gaussian kernel. As a consequence, two transformed samples $\phi(y_s)$ and $\phi(y_t)$ are distributed around the *same* mean value if they belong to the same regime, and around *different* mean-values if they each belong to two consecutive regimes. To put it another way, a signal that follows (M3) is mapped by $\phi(\cdot)$ to a random signal with piecewise constant mean.

From a theoretical point of view, asymptotic consistency of the change point estimates has been demonstrated for both a known and unknown number of change points in the recent work of [103]. This result, as well as an important oracle inequality on the sum of cost $V(\mathcal{T})$ [104], also holds in a non-asymptotic setting. In addition, kernel change point detection was experimentally shown to be competitive in many different settings, in an unsupervised manner and with very few parameters to manually calibrate. For instance, the cost function c_{kernel} was applied on the Brain-Computer Interface (BCI) data set [97], on a video time series segmentation task [104], DNA sequences [100] and emotion recognition [105].

Related cost functions. The cost function c_{kernel} can be combined with any kernel to accommodate various types of data (not just $\mathbb{R}^d$-valued signals). Notable examples of kernel functions include [102]:

- The linear kernel $k(x,y)=\langle x|y\rangle$ with $x,y\in\mathbb{R}^d$.
- The polynomial kernel $k(x,y)=(\langle x|y\rangle+C)^{deg}$ with $x,y\in\mathbb{R}^d$, and C and deg are parameters.
- The Gaussian kernel $k(x,y)=\exp(-\gamma\|x-y\|^2)$ with $x,y\in\mathbb{R}^d$ and $\gamma>0$ is the so-called bandwidth parameter.
- The χ^2-kernel $k(x,y)=\exp(-\gamma\sum_i[(x_i-y_i)^2/(x_i+y_i)])$ with $\gamma\in\mathbb{R}$ a parameter. It is often used for histogram data.

Arguably, the most commonly used kernels for numerical data are the linear kernel and the Gaussian kernel. When combined with the linear kernel, the cost function c_{kernel} is formally equivalent to c_{L_2}. As for the Gaussian kernel, the associated cost function, denoted c_{rbf}, is defined as follows.

Cost function 12 (c_{rbf})**.** The cost function c_{rbf} is given by

$$c_{\text{rbf}}(y_{a.b}):=(b-a)\;-\;\frac{1}{b-a}\sum_{s,t=a+1}^{b}\exp\left(-\gamma\|y_s-y_t\|^2\right)\tag{C12}$$

where $\gamma>0$ is the so-called bandwidth parameter.

The parametric cost function c_M (based on a Mahalanobis-type norm) can be extended to the non-parametric setting through the use of a kernel. Formally, the Mahalanobis-type norm $\|\cdot\|_{\mathcal{H},M}$ in the feature space $\mathcal{H}$ is defined by

$$\|\phi(y_s)-\phi(y_t)\|^2_{\mathcal{H},M}=(\phi(y_s)-\phi(y_t))'M(\phi(y_s)-\phi(y_t))\tag{20}$$

where M is a (possibly infinite dimensional) symmetric positive semi-definite matrix defined on $\mathcal{H}$. The so-called metric matrix M is often chosen to enforce distance constraints, for instance, have (relatively) small distances between samples labelled as similar and (relatively) large distances between samples labelled as dissimilar [85,86,106]. Thus defined (20), the seminorm $\|\cdot\|_{\mathcal{H},M}$ is unpractical to compute because of the infinite dimension of arbitrary M. However, there exist procedures (particularly in the metric learning literature) [106,107] that learn an appropriate metric matrix, which can be fully represented by the constrained training samples. In such a setting, computing $\|\cdot\|_{\mathcal{H},M}$ amounts to evaluating $k(\cdot,\cdot)$ on the set of training samples.

The associated cost function, denoted $c_{\mathcal{H},M}$, is defined below. Intuitively, using $c_{\mathcal{H},M}$ instead of c_M introduces a non-linear treatment of the data samples.

Cost function 13 ($c_{\mathcal{H},M}$)**.** For a given rkhs $\mathcal{H}$, implicitly determined by a kernel function $k(\cdot,\cdot):\mathbb{R}^d\times\mathbb{R}^d\to\mathbb{R}$, and M a symmetric positive semi-definite matrix defined on $\mathcal{H}$, the cost function $c_{\mathcal{H},M}$ is given by

$$c_{\mathcal{H},M}(y_{a.b}):=\sum_{t=a+1}^{b}\|\phi(y_t)-\bar{\mu}_{a.b}\|^2_{\mathcal{H},M}\tag{C13}$$

where $\mu_{a.b}$ is the empirical mean of the transformed sub-signal $\{\phi(y_t)\}_{t=a+1}^{b}$ and $\|\cdot\|_{\mathcal{H},M}$ is defined in (20).

4.3. Summary table

Reviewed cost functions (parametric and non-parametric) are summarized in Table 1. For each cost, the name, expression and parameters of interest are given.

Table 1
Summary of reviewed cost functions.

Name	$c(y_{a.b})$	Parameters
$c_{\text{i.i.d.}}$ (C1)	$-\sup_{\theta} \sum_{t=a+1}^{b} \log f(y_t\|\theta)$	θ: changing parameter; density function: $f(\cdot\|\theta)$
c_{L_2} (C2)	$\sum_{t=a+1}^{b} \\|y_t - \bar{y}_{a.b}\\|_2^2$	$\bar{y}_{a.b}$: empirical mean of $y_{a.b}$
c_{Σ} (C3)	$(b-a) \log\det \widehat{\Sigma}_{a.b} + \sum_{t=a+1}^{b} (y_t - \bar{y}_{a.b})' \widehat{\Sigma}_{a.b}^{-1} (y_t - \bar{y}_{a.b})$	$\widehat{\Sigma}_{a.b}$: empirical covariance of $y_{a.b}$
c_{Poisson} (C4)	$-(b-a)\bar{y}_{a.b} \log \bar{y}_{a.b}$	$\bar{y}_{a.b}$: empirical mean of $y_{a.b}$
c_{linear} (C5)	$\min_{u\in\mathbb{R}^p} \sum_{t=a+1}^{b} (y_t - x_t' u)^2$	$x_t \in \mathbb{R}^p$: covariates
c_{linear,L_1} (C6)	$\min_{u\in\mathbb{R}^p} \sum_{t=a+1}^{b} \|y_t - x_t' u\|$	$x_t \in \mathbb{R}^p$: covariates
c_{AR} (C7)	$\min_{u\in\mathbb{R}^p} \sum_{t=a+1}^{b} (y_t - x_t' u)^2$	$x_t = [y_{t-1}, y_{t-2}, \dots, y_{t-p}]$: lagged samples
c_M (C8)	$\sum_{t=a+1}^{b} \\|y_t - \bar{y}_{a.b}\\|_M^2$	$M \in \mathbb{R}^{d\times d}$: positive semi-definite matrix
$c_{\widehat{F}}$ (C9)	$-(b-a) \sum_{u=1}^{T} \frac{\widehat{F}_{a.b}(u) \log \widehat{F}_{a.b}(u) + (1-\widehat{F}_{a.b}(u)) \log(1-\widehat{F}_{a.b}(u))}{(u-0.5)(T-u+0.5)}$	$\widehat{F}$: empirical c.d.f. (13)
c_{rank} (C10)	$-(b-a)\ \bar{r}_{a.b}'\ \widehat{\Sigma}_r^{-1}\ \bar{r}_{a.b}$	r: rank signal (14); $\widehat{\Sigma}_r$: empirical covariance of r (15)
c_{kernel} (C11)	$\sum_{t=a+1}^{b} k(y_t, y_t) - \frac{1}{b-a} \sum_{s,t=a+1}^{b} k(y_s, y_t)$	$k(\cdot,\cdot) : \mathbb{R}^d \times \mathbb{R}^d \mapsto \mathbb{R}$: kernel function
c_{rbf} (C12)	$(b-a) - \frac{1}{b-a} \sum_{s,t=a+1}^{b} \exp(-\gamma \\|y_s - y_t\\|^2)$	$\gamma > 0$: bandwidth parameter
$c_{\mathcal{H},M}$ (C13)	$\sum_{t=a+1}^{b} \\|y_t - \bar{y}_{a.b}\\|_{\mathcal{H},M}^2$	M: positive semi-definite matrix (in the feature space $\mathcal{H}$)

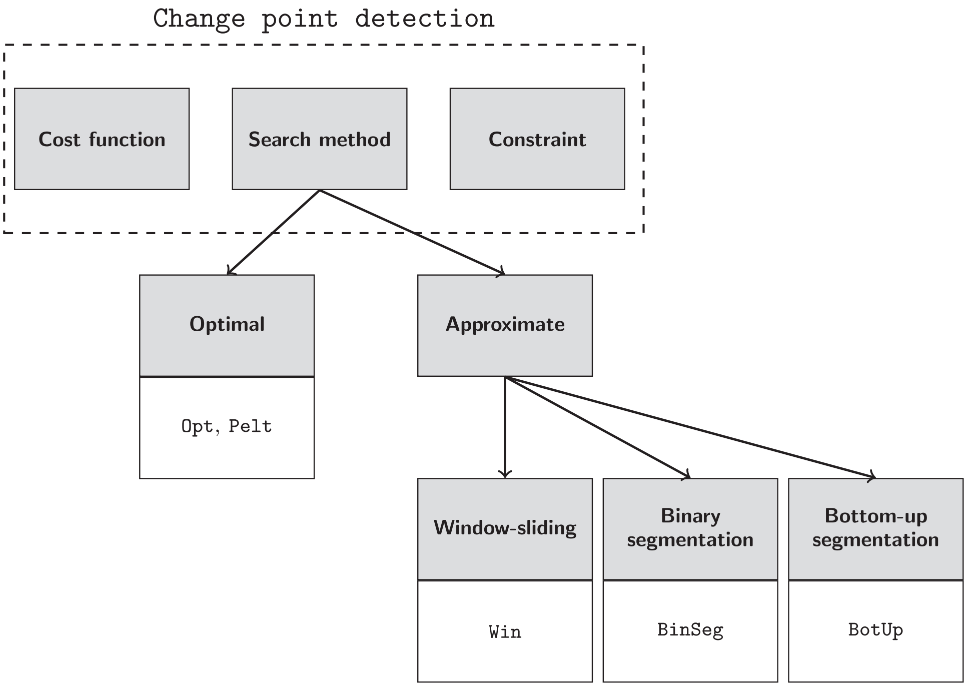

Fig. 7. Typology of the search methods described in Section 5.

5. Search methods

This section presents the second defining element of change detection methods, namely the search method. Reviewed search methods are organized in two general categories, as shown on Fig. 7: optimal methods, that yield the exact solution to the discrete optimization of (P1) and (P2), and the approximate methods, that yield an approximate solution. Described algorithms can be combined with cost functions from Section 4. Note that, depending on the chosen cost function, the computational complexity of the complete algorithm changes. As a consequence, in the following, complexity analysis is done with the assumption that applying the cost function on a sub-signal requires $\mathcal{O}(1)$ operations. Also, the practical implementations of the most important algorithms are given in pseudocode.

5.1. Optimal detection

Optimal detection methods find the exact solutions of Problem 1 (P1) and Problem 2 (P2). A naive approach consists in enumerating all possible segmentations of a signal, and returning the one that minimizes the objective function. However, for (P1), minimization is carried out over the set $\{\mathcal{T} \text{ s.t. } |\mathcal{T}| = K\}$ (which contains $\binom{T-1}{K-1}$ elements), and for (P2), over the set $\{\mathcal{T} \text{ s.t. } 1 \le |\mathcal{T}| < T\}$ (which contains $\sum_{K=1}^{T-1} \binom{T-1}{K-1}$ elements). This makes exhaustive enumeration impractical, in both situations. We describe in this section two major approaches to efficiently find the exact solutions of (P1) and (P2).

5.1.1. Solution to problem 1 (P1): Opt

In (P1), the number of change points to detect is fixed to a certain $K \geq 1$. The optimal solution to this problem can be computed efficiently, thanks to a method based on dynamic programming. The algorithm, denoted Opt, relies on the additive nature of the objective function $V(\cdot)$ to recursively solve sub-problems. Precisely, Opt is based on the following observation:

$$\begin{aligned}
&\min_{|\mathcal{T}|=K} V(\mathcal{T}, y = y_{0.T}) \\
&= \min_{0=t_0<t_1<\ldots<t_K<t_{K+1}=T} \sum_{k=0}^{K} c(y_{t_k.t_{k+1}}) \\
&= \min_{t \leq T-K} \left[c(y_{0.t}) + \min_{t=t_0<t_1<\ldots<t_{K-1}<t_K=T} \sum_{k=0}^{K-1} c(y_{t_k.t_{k+1}}) \right] \\
&= \min_{t \leq T-K} \left[c(y_{0.t}) + \min_{|\mathcal{T}|=K-1} V(\mathcal{T}, y_{t.T}) \right] \qquad (21)
\end{aligned}$$

Intuitively, Eq. (21) means that the first change point of the optimal segmentation is easily computed if the optimal partitions with $K-1$ elements of all sub-signals $y_{t.T}$ are known. The complete segmentation is then computed by recursively applying this observation. This strategy, described in detail in Algorithm 1, has a complexity of the order of $\mathcal{O}(KT^2)$ [72,108]. Historically, Opt was introduced for a non-related problem [109] and later applied to change point detection, in many different contexts, such as EEG recordings [65,110], DNA sequences [100,111], tree growth monitoring [20], financial time-series [49,76], radar waveforms [112], etc.

Algorithm 1 Algorithm Opt.

```
Input: signal ${y_t}_{t=1}^T$, cost function $c(\cdot)$, number of regimes $K \geq 2$.
for all $(u, v)$, $1 \leq u < v \leq T$ do
    Initialize $C_1(u, v) \leftarrow c(\{y_t\}_{t=u}^{v})$.
end for
for $k = 2, \ldots, K-1$ do
    for all $u, v \in \{1, \ldots, T\}, v - u \geq k$ do
        $C_k(u, v) \leftarrow \min_{u+k-1 \leq t < v} C_{k-1}(u, t) + C_1(t+1, v)$
    end for
end for
Initialize $L$, a list with $K$ elements.
Initialize the last element: $L[K] \leftarrow T$.
Initialize $k \leftarrow K$.
while $k > 1$ do
    $s \leftarrow L(k)$
    $t^* \leftarrow \mathrm{argmin}_{k-1 \leq t < s} C_{k-1}(1, t) + C_1(t+1, s)$
    $L(k-1) \leftarrow t^*$
    $k \leftarrow k - 1$
end while
Remove $T$ from $L$
Output: set $L$ of estimated breakpoint indexes.
```

Related search methods. Several extensions of Opt have been proposed in the literature. The proposed methods still find the exact solution to (P1).

- The first extension is the "forward dynamic programming" algorithm [20]. Contrary to Opt, which returns a single partition, the "forward dynamic programming" algorithm computes the top L ($L \geq 1$) most probable partitions (ie with lowest sum of costs). The resulting computational complexity is $\mathcal{O}(LKT^2)$ where L is the number of computed partitions. This method is designed as a diagnostic tool: change points present in many of the top partitions are considered very likely, while change points present in only a few of the top partitions might not be as relevant. Thanks to "forward dynamic programming", insignificant change points are trimmed and overestimation of the number of change point is corrected [20], at the expense of a higher computational burden. It is applied on tree growth monitoring time series [20] that are relatively short with around a hundred samples.
- The "pruned optimal dynamic programming" procedure [111] is an extension of Opt that relies on a pruning rule to discard indexes that can never be change points. Thanks to this trick, the set of potential change point indexes is reduced. All described cost functions can be plugged into this method. As a result, longer signals can be handled, for instance long array-based DNA copy number data (up to 10^6 samples, with the quadratic error cost function) [111]. However, worst case complexity remains of the order of $\mathcal{O}(KT^2)$.

5.1.2. Solution to problem 2 (P2): Pelt

In (P2), the number of changes point is unknown, and the objective function to minimize is the penalized sum of costs. A naive approach consists in applying Opt for $K = 1, \ldots, K_{\max}$ for a sufficiently large $K_{\max}$, then choosing among the computed segmentations the one that minimizes the penalized problem. This would prove computational cumbersome because of the quadratic complexity of the resolution method Opt. Fortunately a faster method exists for a general class of penalty functions, namely linear penalties. Formally, linear penalties are linear functions of the number of change points, meaning that

$$\mathrm{pen}(\mathcal{T}) = \beta |\mathcal{T}| \qquad (22)$$

where $\beta > 0$ is a smoothing parameter. (More details on such penalties can be found in Section 6.1.) The algorithm Pelt (for "Pruned Exact Linear Time") [113] was introduced to find the exact solution of (P2), when the penalty is linear (22). This approach considers each sample sequentially and, thanks to an explicit pruning rule, may or may not discard it from the set of potential change points. Precisely, for two indexes t and s ($t < s < T$), the pruning rule is given by:

$$\begin{aligned}
\text{if } & \left[\min_{\mathcal{T}} V(\mathcal{T}, y_{0.t}) + \beta |\mathcal{T}| \right] + c(y_{t.s}) \\
& \geq \left[\min_{\mathcal{T}} V(\mathcal{T}, y_{0.s}) + \beta |\mathcal{T}| \right] \quad \text{holds,} \\
& \text{then } t \text{ cannot be the last change point prior to } T. \qquad (23)
\end{aligned}$$

This results in a considerable speed-up: under the assumption that regime lengths are randomly drawn from a uniform distribution, the complexity of Pelt is of the order $\mathcal{O}(T)$. The detailed algorithm can be found in Algorithm 2. An extension of Pelt is described in [9] to solve the linearly penalized change point detection for a range of smoothing parameter values $[\beta_{\min}, \beta_{\max}]$. Pelt has been applied on DNA sequences [16,17], physiological signals [89], and oceanographic data [113].

5.2. Approximate detection

When the computational complexity of optimal methods is too great for the application at hand, one can resort to approximate methods. In this section, we describe three major types of approximate segmentation algorithms, namely window-based methods, binary segmentation and bottom-up segmentation. All described procedures fall into the category of sequential detection

Algorithm 2 Algorithm Pelt.

```
Input: signal $\{y_t\}_{t=1}^T$, cost function $c(\cdot)$, penalty value $\beta$.
Initialize $Z$ a $(T+1)$-long array; $Z[0] \leftarrow -\beta$.
Initialize $L[0] \leftarrow \emptyset$.
Initialize $\chi \leftarrow \{0\}$.                    ▷ Admissible indexes.
for $t = 1, \ldots, T$ do
    $\hat{t} \leftarrow \text{argmin}_{s \in \chi} \left[Z[s] + c(y_{s.t}) + \beta\right]$.
    $Z[t] \leftarrow \left[Z[\hat{t}] + c(y_{\hat{t}.t}) + \beta\right]$
    $L[t] \leftarrow L[\hat{t}] \cup \{\hat{t}\}$.
    $\chi \leftarrow \{s \in \chi \ : \ Z[s] + c(y_{s.t}) \le Z[t]\} \cup \{t\}$
end for
Output: set $L[T]$ of estimated breakpoint indexes.
```

approaches, meaning that they return a single change point estimate $\hat{t}^{(k)}$ $(1 \le \hat{t}^{(k)} < T)$ at the k-th iteration. (In the following, the subscript $\cdot^{(k)}$ refers to the k-th iteration of a sequential algorithm.) Such methods can be used to solve (approximately) either (P1) or (P2). Indeed, if the number K^* of changes is known, K^* iterations of a sequential algorithm are enough to retrieve a segmentation with the correct number of changes. If K^* is unknown, the sequential algorithm is run until an appropriate stopping criterion is met.

5.2.1. Window sliding

The window-sliding algorithm, denoted Win, is a fast approximate alternative to optimal methods. It consists in computing the discrepancy between two adjacent windows that slide along the signal y. For a given cost function $c(\cdot)$, this discrepancy between two sub-signals is given by

$$d(y_{a.t}, y_{t.b}) = c(y_{a.b}) - c(y_{a.t}) - c(y_{t.b}) \quad (1 \le a < t < b \le T). \tag{24}$$

When the two windows cover dissimilar segments, the discrepancy reaches large values, resulting in a peak. In other other words, for each index t, Win measures the discrepancy between the immediate past ("left window") and the immediate future ("right window"). Once the complete discrepancy curve has been computed, a peak search procedure is performed to find change point indexes. The complete Win algorithm is given in Algorithm 3 and a schematic view is displayed on Fig. 8. The main benefits of Win are its low complexity (linear in the number of samples) and ease of implementation.

Algorithm 3 Algorithm Win.

```
Input: signal $\{y_t\}_{t=1}^T$, cost function $c(\cdot)$, half-window width $w$,
peak search procedure PKSearch.
Initialize $Z \leftarrow [0, 0, \ldots]$ a $T$-long array filled with 0.   ▷ Score list.
for $t = w, \ldots, T - w$ do
    $p \leftarrow (t-w).t$.
    $q \leftarrow t.(t+w)$.
    $r \leftarrow (t-w).(t+w)$.
    $Z[t] \leftarrow c(y_r) - [c(y_p) + c(y_q)]$.
end for
$L \leftarrow$ PKSearch($Z$)                    ▷ Peak search procedure.
Output: set $L$ of estimated breakpoint indexes.
```

In the literature, the discrepancy measure $d(\cdot, \cdot)$ is often derived from a two-sample statistical test (see Remark 4), and not from a cost function, as in (24). However, the two standpoints are generally equivalent: for instance, using c_{L_2}, $c_{\text{i.i.d.}}$ or c_{kernel} is respectively equivalent to applying a Student t-test [3], a generalized likelihood ratio (GLR) [6] test and a kernel Maximum Mean Discrepancy (MMD) test [99]. As a consequence, practitioners can capitalize on the vast body of work in the field of statistical tests to obtain asymptotic distributions for the discrepancy measure [28,29,99,114], and sensible calibration strategies for important parameters of Win (such as the window size or the peak search procedure). Win has been applied in numerous contexts: for instance, on biological signals [11,115–118], on network data [28,29], on speech time series [10,11,119] and on financial time series [3,120,121]. It should be noted that certain window-based detection methods in the literature rely on a discrepancy measure which is not related to a cost function, as in (24) [11,122–124]. As a result, those methods, initially introduced in the online detection setting, cannot be extended to work with optimal algorithms (Opt, Pelt).

Remark 4 (Two-sample test)**.** A two-sample test (or homogeneity test) is a statistical hypothesis testing procedure designed to assess whether two populations of samples are identical in distribution. Formally, consider two sets of iid $\mathbb{R}^d$-valued random samples $\{x_t\}_t$ and $\{z_t\}_t$. Denote by $\mathbb{P}_x$ the distribution function of the x_t and by $\mathbb{P}_z$ the distribution function of the z_t. A two-sample test procedure compares the two following hypotheses:

$$\begin{aligned} H_0 &\quad : \quad \mathbb{P}_x = \mathbb{P}_z \\ H_1 &\quad : \quad \mathbb{P}_x \neq \mathbb{P}_z. \end{aligned} \tag{25}$$

A general approach is to consider a probability (pseudo)-metric $d(\cdot, \cdot)$ on the space of probability distributions on $\mathbb{R}^d$. Well-known examples of such a metric include the Kullback-Leibler divergence, the Kolmogorov-Smirnov distance, the Maximum Mean Discrepancy (MMD), etc. Observe that, under the null hypothesis, $d(\mathbb{P}_x, \mathbb{P}_z) = 0$. The testing procedure consists in computing the empirical estimates $\widehat{\mathbb{P}}_x$ and $\widehat{\mathbb{P}}_z$ and rejecting H_0 for "large" values of the statistics $d(\widehat{\mathbb{P}}_x, \widehat{\mathbb{P}}_z)$. This general formulation relies on a consistent estimation of arbitrary distributions from a finite number of samples. In the parametric setting, additional assumptions are made on the distribution functions: for instance, Gaussian assumption [3,6,114], exponential family assumption [15,125], etc. In the non-parametric setting, the distributions are only assumed to be continuous. They are not directly estimated; instead, the statistics $d(\widehat{\mathbb{P}}_x, \widehat{\mathbb{P}}_z)$ are computed [11,91,99,123].

In the context of *single* change point detection, the two-sample test setting is adapted to assess whether a distribution change has occurred at some instant in the input signal. Practically, for a given index t, the homogeneity test is performed on the two populations $\{y_s\}_{s \le t}$ and $\{y_s\}_{s > t}$. The estimated change point location is given by

$$\hat{t} = \text{argmax}_t \ d\left(\widehat{\mathbb{P}}_{\bullet \le t}, \widehat{\mathbb{P}}_{\bullet > t}\right) \tag{26}$$

where $\widehat{\mathbb{P}}_{\bullet \le t}$ and $\widehat{\mathbb{P}}_{\bullet > t}$ are the empirical distributions of respectively $\{y_s\}_{s \le t}$ and $\{y_s\}_{s > t}$.

5.2.2. Binary segmentation

Binary segmentation, denoted BinSeg, is a well-known alternative to optimal methods [53], because it is conceptually simple and easy to implement [6,113,126]. BinSeg is a greedy sequential algorithm, outlined as follows. The first change point estimate $\hat{t}^{(1)}$ is given by

$$\hat{t}^{(1)} := \text{argmin}_{1 \le t < T-1} \underbrace{c(y_{0..t}) + c(y_{t..T})}_{V(\mathcal{T}=\{t\})}. \tag{27}$$

This operation is "greedy", in the sense that it searches the change point that lowers the most the sum of costs. The signal is then split in two at the position of $\hat{t}^{(1)}$; the same operation is repeated on the resulting sub-signals until a stopping criterion is met. A schematic view of the algorithm is displayed on Fig. 9 and an implementation is given in Algorithm 4. The complexity of BinSeg

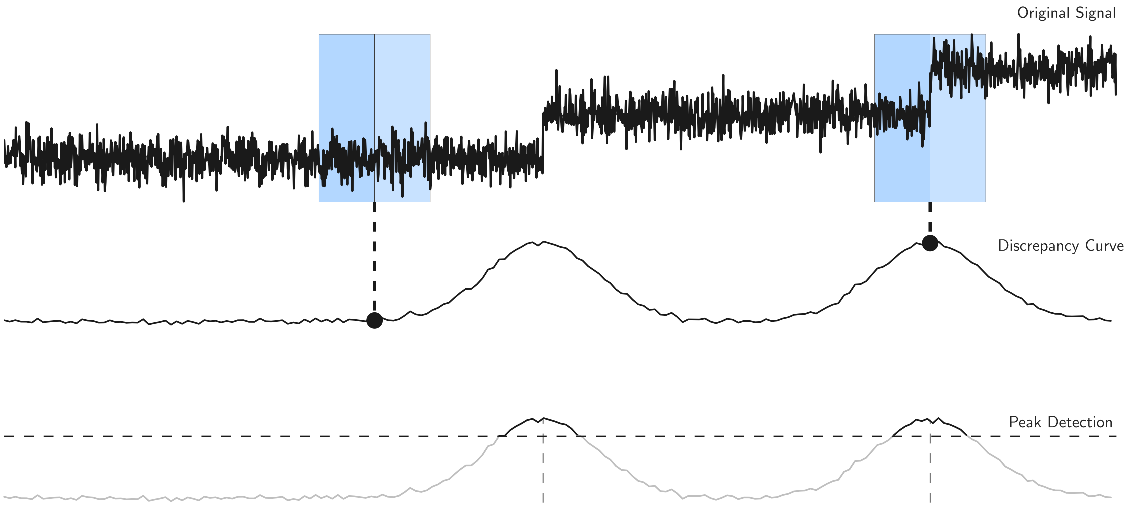

Fig. 8. Schematic view of Win.

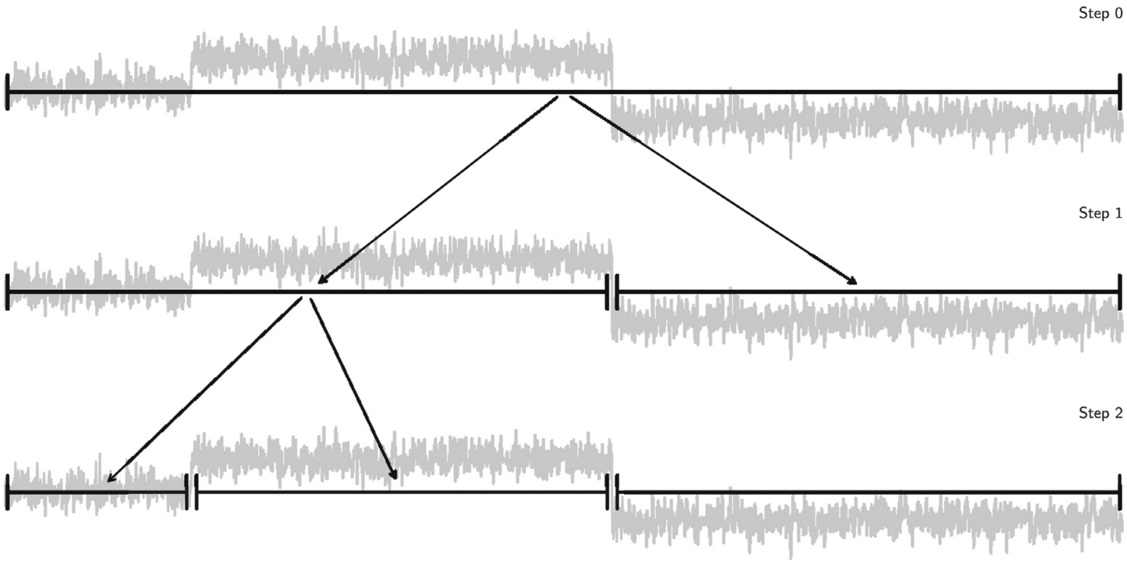

Fig. 9. Schematic example of BinSeg.

Algorithm 4 Algorithm BinSeg.

```
Input: signal $\{y_t\}_{t=1}^{T}$, cost function $c(\cdot)$, stopping criterion.
Initialize $L \leftarrow \{\ \}$.                              ▷ Estimated breakpoints.
repeat
    $k \leftarrow |L|$.                                        ▷ Number of breakpoints
    $t_0 \leftarrow 0$ and $t_{k+1} \leftarrow T$              ▷ Dummy variables.
    if $k > 0$ then
        Denote by $t_i$ $(i = 1, \dots, k)$ the elements (in ascending order) of $L$, ie $L = \{t_1, \dots, t_k\}$.
    end if
    Initialize $G$ a $(k+1)$-long array.                      ▷ list of gains
    for $i = 0, \dots, k$ do
        $G[i] \leftarrow c(y_{t_i.t_{i+1}}) - \min_{t_i<t<t_{i+1}} [c(y_{t_i.t}) + c(y_{t.t_{i+1}})]$.
    end for
    $\widehat{i} \leftarrow \operatorname{argmax}_i G[i]$
    $\hat{t} \leftarrow \operatorname{argmin}_{t_{\widehat{i}}<t<t_{\widehat{i}+1}} [c(y_{t_{\widehat{i}}.t}) + c(y_{t.t_{\widehat{i}+1}})]$.
    $L \leftarrow L \cup \{\hat{t}\}$
until stopping criterion is met.
Output: set $L$ of estimated breakpoint indexes.
```

is of the order of $\mathcal{O}(T \log T)$. This low complexity comes at the expense of optimality: in general, BinSeg's output is only an approximation of the optimal solution. As argued in [113,127], the issue is that the estimated change points $\hat{t}^{(k)}$ are not estimated from homogeneous segments and each estimate depends on the previous ones. Change points that are close are imprecisely detected especially [8]. Applications of BinSeg range from financial time series [6,7,114,127,128] to context recognition for mobile devices [129] and array-based DNA copy number data [19,126,130].

Related search methods. Several extensions of BinSeg have been proposed to improve detection accuracy.

- Circular binary segmentation [126] is a well-known extension of BinSeg. This method is also a sequential detection algorithm that splits the original at each step. Instead of searching for a single change point in each sub-signal, circular binary segmentation searches two change points. Within each treated sub-segment, it assumes a so-called "epidemic change model": the parameter of interest shifts from one value to another at the first change point and returns to the original value at the second change point. The algorithm is dubbed "circular" because, under this model, the

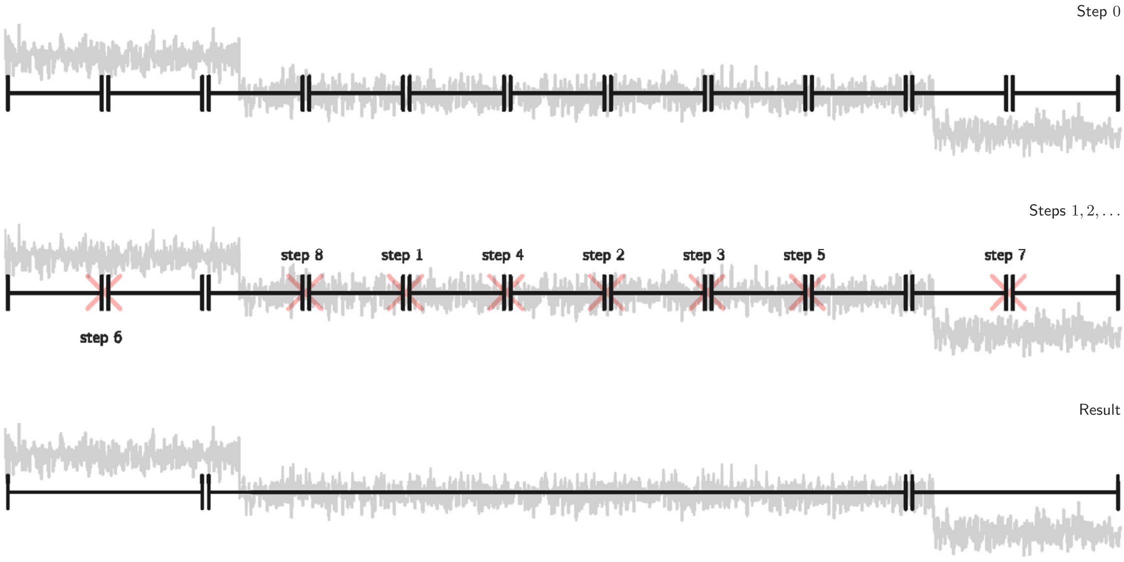

Fig. 10. Schematic view of BotUp.

sub-segment has its two ends (figuratively) joining to form a circle. Practically, this method has been combined with c_{L_2} C2, to detect changes in the mean of array-based DNA copy number data [126,131,132]. A faster version of the original algorithm is described in [133].

- Another extension of BinSeg is the wild binary segmentation algorithm [128]. In a nutshell, a single point detection is performed on multiple intervals with start and end points that are drawn uniformly. Small segments are likely to contain at most one change but have lower statistical power, while the opposite is true for long segments. After a proper weighting of the change score to account for the differences on sub-signals' length, the algorithm returns the most "pronounced" ones, ie those that lower the most the sum of costs. An important parameter of this method is the number of random sub-segments to draw. Wild binary search is combined with c_{L_2} C2 to detect mean-shifts of univariate piecewise constant signals (up to 2000 samples) [128].

5.2.3. Bottom-up segmentation

Bottom-up segmentation, denoted BotUp, is the natural counterpart of BinSeg. Contrary to BinSeg, BotUp starts by splitting the original signal in many small sub-signals and sequentially merges them until there remain only K change points. At every step, all potential change points (indexes separating adjacent sub-segments) are ranked by the discrepancy measure $d(\cdot,\cdot)$, defined in 24, between the segments they separate. Change points with the lowest discrepancy are then deleted, meaning that the segments they separate are merged. BotUp is often dubbed a "generous" method, by opposition to BinSeg, which is "greedy" [134]. A schematic view of the algorithm is displayed on Fig. 10 and an implementation is provided in Algorithm 5. Its benefits are its linear computational complexity and conceptual simplicity. However, if a true change point does not belong to the original set of indexes, BotUp never considers it. Moreover, in the first iterations, the merging procedure can be unstable because it is performed on small segments, for which statistical significance is smaller. In the literature, BotUp is somewhat less studied than its counterpart, BinSeg: no theoretical convergence study is available. It has been applied on speech time series to detect mean and scale shifts [120]. Besides, the authors of [134] have found that BotUp outperforms BinSeg on ten different data sets such as physiological signals (ECG), financial time-series (exchange rate), industrial monitoring (water levels), etc.

Algorithm 5 Algorithm BotUp.

```
Input: signal ${y_t}_{t=1}^T$, cost function $c(\cdot)$, stopping criterion, grid size $\delta > 2$.
Initialize $L \leftarrow \{\delta, 2\delta, \dots, (\lfloor T/\delta \rfloor - 1)\,\delta\}$.   ▷ Estimated breakpoints.
repeat
    $k \leftarrow |L|$.   ▷ Number of breakpoints
    $t_0 \leftarrow 0$ and $t_{k+1} \leftarrow T$   ▷ Dummy variables.
    Denote by $t_i$ ($i = 1, \dots, k$) the elements (in ascending order) of $L$, ie $L = \{t_1, \dots, t_k\}$.
    Initialize $G$ a $(k-1)$-long array.   ▷ list of gains
    for $i = 1, \dots, k-1$ do
        $G[i-1] \leftarrow c(y_{t_{i-1}.t_{i+1}}) - [c(y_{t_{i-1}.t_i}) + c(y_{t_i.t_{i+1}})]$.
    end for
    $\widehat{i} \leftarrow \mathrm{argmin}_i\, G[i]$
    Remove $t_{\widehat{i}+1}$ from $L$.
until stopping criterion is met.
Output: set $L$ of estimated breakpoint indexes.
```

6. Estimating the number of changes

This section presents the third defining element of change detection methods, namely the constraint on the number of change points. Here, the number of change points is assumed to be unknown (P2). Existing procedures are organized by the penalty function that they are based on. Common heuristics are also described. The organization of this section is schematically shown in Fig. 11.

6.1. Linear penalty

Arguably the most popular choice of penalty [113], the linear penalty (also known as l_0 penalty) generalizes several well-known criteria from the literature such as the Bayesian Information Criterion (BIC) and the Akaike Information Criterion (AIC) [135,136]. The linear penalty, denoted pen_{l_0}, is formally defined as follows.

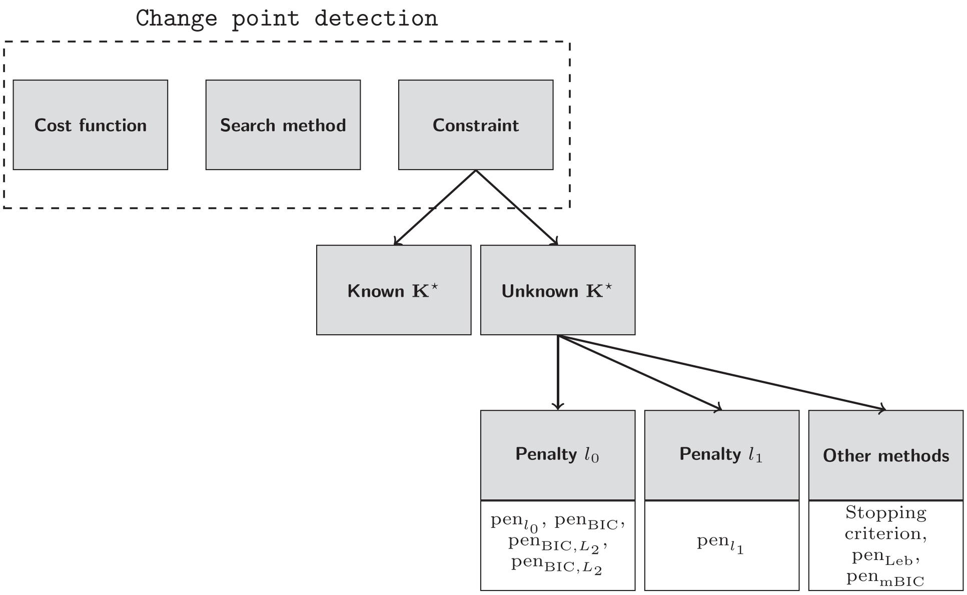
Change point detection
Cost function
Search method
Constraint
Known $\mathbf{K}^\star$
Unknown $\mathbf{K}^\star$
Penalty l_0
Penalty l_1
Other methods
pen_{l_0}, $\mathrm{pen}_{\mathrm{BIC}}$, $\mathrm{pen}_{\mathrm{BIC},L_2}$, $\mathrm{pen}_{\mathrm{BIC},L_2}$
pen_{l_1}
Stopping criterion, $\mathrm{pen}_{\mathrm{Leb}}$, $\mathrm{pen}_{\mathrm{mBIC}}$

Fig. 11. Typology of the constraints (on the number of change points) described in Section 6.

Penalty 1 (pen_{l_0}). The penalty function pen_{l_0} is given by

$$\mathrm{pen}_{l_0}(\mathcal{T}) := \beta|\mathcal{T}| \tag{28}$$

where $\beta > 0$ is the smoothing parameter.

Intuitively, the smoothing parameter controls the trade-off between complexity and goodness-of-fit (measured by the sum of costs): low values of β favour segmentations with many regimes and high values of β discard most change points.

Calibration. From a practical standpoint, once the cost function has been chosen, the only parameter to calibrate is the smoothing parameter. Several approaches, based on model selection, can be found in the literature: they assume a model on the data, for instance (M1), (M2), (M3), and choose a value of β that optimizes a certain statistical criterion. The best-known example of such an approach is BIC, which aims at maximizing the constrained log-likelihood of the model. The exact formulas of several linear penalties, derived from model selection procedures, are given the following paragraph. Conversely, when no model is assumed, different heuristics are applied to tune the smoothing parameter. For instance, one can use a procedure based on cross-validation [137] or the slope heuristics [138]. In [139,140], supervised algorithms are proposed: the chosen β is the one that minimizes an approximation of the segmentation error on an annotated set of signals.

Related penalties A number of model selection criteria are special cases of the linear penalty pen_{l_0}. For instance, under Model (M1) (iid with piecewise constant distribution), the constrained likelihood that is derived from the BIC and the penalized sum of costs are formally equivalent, upon setting $c = c_{\mathrm{i.i.d.}}$ and $\mathrm{pen} = \mathrm{pen}_{\mathrm{BIC}}$, where $\mathrm{pen}_{\mathrm{BIC}}$ is defined as follows.

Penalty 2 ($\mathrm{pen}_{\mathrm{BIC}}$). The penalty function $\mathrm{pen}_{\mathrm{BIC}}$ is given by

$$\mathrm{pen}_{\mathrm{BIC}}(\mathcal{T}) := \frac{p}{2}\log T\ |\mathcal{T}| \tag{29}$$

where $p \geq 1$ is the dimension of the parameter space in (M1).

In the extensively studied model of an univariate Gaussian signal, with fixed variance σ^2 and piecewise constant mean, the penalty $\mathrm{pen}_{\mathrm{BIC}}$ becomes pen_{L_2}, defined below. Historically, it was one of the first penalties introduced for change point detection [135,141].

Penalty 3 ($\mathrm{pen}_{\mathrm{BIC},L_2}$). The penalty function $\mathrm{pen}_{\mathrm{BIC},L_2}$ is given by

$$\mathrm{pen}_{\mathrm{BIC},L_2}(\mathcal{T}) := \sigma^2 \log T\ |\mathcal{T}|. \tag{30}$$

where σ is the standard deviation and T is the number of samples.

In the same setting, AIC, which is a generalization of Mallows' C_p [62], also yields a linear penalty, namely $\mathrm{pen}_{\mathrm{AIC},L_2}$, defined as follows.

Penalty 4 ($\mathrm{pen}_{\mathrm{AIC},L_2}$). The penalty function $\mathrm{pen}_{\mathrm{AIC},L_2}$ is given by

$$\mathrm{pen}_{\mathrm{AIC},L_2}(\mathcal{T}) := \sigma^2\ |\mathcal{T}|. \tag{31}$$

where σ is the standard deviation.

6.2. Fused lasso

For the special case where the cost function is c_{L_2}, a faster alternative to pen_{l_0} can be used. To that end, the l_0 penalty is relaxed to a l_1 penalty [18,48]. The resulting penalty function, denoted pen_{l_1}, is defined as follows.

Penalty 5 (pen_{l_1}). The penalty function pen_{l_1} is given by

$$\mathrm{pen}_{l_1}(\mathcal{T}) := \beta \sum_{k=1}^{|\mathcal{T}|} \left\| \bar{y}_{t_{k-1}.t_k} - \bar{y}_{t_k.t_{k+1}} \right\|_1 \tag{32}$$

where $\beta > 0$ is the smoothing parameter, the t_k are the elements of $\mathcal{T}$ and $\bar{y}_{t_{k-1}.t_k}$ is the empirical mean of sub-signal $y_{t_{k-1}.t_k}$.

This relaxation strategy (from l_0 to l_1) is shared with many developments in machine learning, for instance sparse regression, compressive sensing, sparse PCA, dictionary learning [84], where pen_{l_1} is also referred to as the fused lasso penalty. In numerical analysis and image denoising, it is also known as the total variation regularizer [13,18,48]. Thanks to this relaxation, the optimization of the penalized sum of costs (1) in (P2) is transformed into a convex optimization problem, which can be solved efficiently using Lars (for "least absolute shrinkage and selection operator") [18,48]. The resulting complexity is of this order of $\mathcal{O}(T\log T)$ [84,142]. From a theoretical standpoint, under the mean-shift model (piecewise constant signal with Gaussian white noise), the estimated change point fractions are asymptotically consistent [48]. This result is demonstrated for an appropriately converging sequence of values

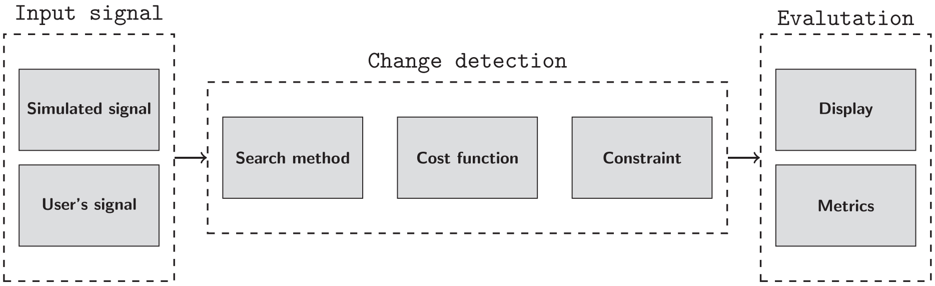

Fig. 12. Schematic view of the `ruptures` package.

of β. This consistency property is obtained even though classical assumptions from the Lasso regression framework (such as the irrepresentable condition) are not satisfied [48]. In the literature, pen_{l_1}, combined with c_{L_2}, is applied on DNA sequences [16,18], speech signals [12] and climatological data [143].

6.3. Complex penalties

Several other penalty functions can be found in the literature. However they are more complex, in the sense that the optimization of the penalized sum of cost is not tractable. In practice, the solution is found by computing the optimal segmentations with K change points, with $K = 1, 2, \ldots, K_{\max}$ for a sufficiently large $K_{\max}$, and returning the one that minimizes the penalized sum of costs. When possible, the penalty can also be approximated by a linear penalty, in which case, `Pelt` can be used. In this section, we describe two examples of complex penalties. Both originate from theoretical considerations, under the univariate mean-shift model, with the cost function c_{L_2}. The first example is the modified BIC criterion (mBIC) [144], which consists in maximizing the asymptotic posterior probability of the data. The resulting penalty function, denoted pen_{mBIC}, depends on the number and repartition of the change point indexes: intuitively, it favours evenly spaced change points.

Penalty 6 (pen_{mBIC})**.** The penalty function pen_{mBIC} is given by

$$\text{pen}_{\text{mBIC}}(\mathcal{T}) := 3|\mathcal{T}| \log T + \sum_{k=0}^{|\mathcal{T}|+1} \log\left(\frac{t_{k+1} - t_k}{T}\right) \tag{33}$$

where the t_k are the elements of $\mathcal{T}$.

In [145], a model selection procedure leads to another complex penalty function, namely pen_{Leb} (where "Leb" is short for "Lebarbier", the author of [145]). Upon using this penalty function, the penalized sum of costs satisfied a so-called oracle inequality, which holds in a non-asymptotic setting, contrary to the other penalties previously described.

Penalty 7 (pen_{Leb})**.** The cost function pen_{Leb} is given by

$$\text{pen}_{\text{Leb}}(\mathcal{T}) := \frac{|\mathcal{T}|+1}{T}\sigma^2\left(a_1 \log \frac{|\mathcal{T}|+1}{T} + a_2\right) \tag{34}$$

where $a_1 > 0$ and $a_2 > 0$ are positive parameters and σ^2 is the noise variance.

7. Current challenges in change point detection

Change point detection is an active research field that has adapted to the interests of the scientific community. In particular, to cope with the complexity and the current challenges of real-world data, recent methods have encompassed techniques linked to machine learning and artificial intelligence. In this section, we review two hot topics which have received a lot of attention in the recent years: the change detection for high-dimensional data and the use of supervised learning for the automatic calibration of the detection methods.

7.1. Change detection for high-dimensional signals

In this review, the number of components d was assumed to be fixed and the number T of samples, growing to infinity. However, in certain applications (for instance copy number data, financial time series, internet traffic monitoring), d is of the same order of magnitude as T, or even far larger. This high-dimensional setting brings about at least two difficulties, first about the consistency of the breakpoint estimates, and second, about the computationally complexity. For mean-shift detection, total variation regression (cost function c_{L_2} and penalty function pen_{l_1}) remains a popular choice because it enjoys a low complexity. In addition, it has been shown to consistently estimate single change points for the "fixed T increasing d" setting [18]. Nevertheless, several authors have observed that for high dimensional signals, certain statistical quantities are no longer so easily estimated: for instance, the inverse of the sample covariance matrix is not necessarily an unbiased estimator of the precision matrix, when d is larger than T (see [146] for example). Since this quantity is often used to detect mean-shifts in i.i.d. Gaussian signals with arbitrary variance/covariance matrix Σ (cost function c_M with $M = \Sigma^{-1}$), more robust replacements to similar conventional statistics were introduced [147,148]. However, those formulations overlook the fact that, in the high-dimensional setting, the changes to detect often occur in only a fraction of the signal components. As argued in [80], this can lead to underestimation of the number of changes. To that end, procedures that take this sparsity into account have been proposed. One possible solution this problem is to search changes in all possible combinations of components, and aggregate the results in a sensible way. In [149], a procedure for single mean-shifts is described: the estimation is shown to be consistently for T and d increasing to infinity, and under certain conditions (in particular, the number of changing components must be sufficiently large) Also, their method has a linear complexity (up to a logarithmic factor), despite the combinatorial optimization problem. In [150], the authors give a general discussion on how to aggregate (dependant) change detection statistics, each based on a subset of an increasing number of components. Another method consists in finding a data-driven projection of the multivariate signal and detect changes on the resulting univariate signal [151]. This is again a combinatorial problem, which can be relaxed to make it convex. The change point estimation is shown to be consistent, under certain conditions. The more general setting of piecewise stationary signals is also tackled for high dimensional time series. A number of associated procedures, which are meant to detect changes in the second order structure of a random process, consist in transforming the signal in a sensible way, and then search mean-shifts with a conventional method [152]. For instance, the transformation can

Table 2
Summary table of literature review.

Publication	Search method	Cost function	Known K		Scalability (wrt T)	Package	Additional information
			Yes	No			
[53,156]	`BinSeg`	c_{L_2}	✓	–	★★★	✓	
[135]	`Opt`	c_{L_2}	–	✓	★☆☆	–	Bayesian information criterion (BIC)
[3]	`Opt`	$c_{\text{i.i.d.}}, c_{L_2}$	–	–	★★★	–	single change point
[66,72]	`Opt`	c_{linear,L_2}	–	–	★★☆	–	single change point
[67]	`Opt`	c_{linear,L_1}	–	–	★★☆	–	single change point
[110]	`Opt`	c_{AR}	✓	–	★☆☆	–	
[69]	`Opt`	c_{AR}	✓	–	★☆☆	–	
[138,157]	`Opt`	c_{L_2}	–	✓	★☆☆	–	model selection
[75]	`Opt`	c_{L_2}	✓	–	★☆☆	–	
[126,133]	`BinSeg`	c_{L_2}	✓	✓	★★★	✓	
[145]	`Opt`	c_{L_2}	–	✓	★☆☆	–	model selection
[10]	`Win`	c_{kernel}	–	✓	★★★	–	dissimilarity measure (one-class SVM), see Remark 4
[97]	`Opt`	c_{kernel}, c_{rbf}	✓	–	★☆☆	–	
[144]	`Pelt`	c_{L_2}	–	✓	★★☆	–	modified BIC
[11]	`Win`	–	✓	✓	★★★	–	dissimilarity measure (Fisher discriminant), see Remark 4
[28,29]	`Win`	c_{rank}	✓	✓	★★★	✓	dissimilarity measure (rank-based), see Remark 4
[77]	`Opt`	c_{L_2}, c_{Σ}	–	–	★★☆	–	single change point
[18]	Fused Lasso	c_{L_2}	–	✓	★★★	–	Tikhonov regularization
[48]	Fused Lasso	c_{L_2}	–	✓	★★★	–	total variation regression (pen_{l_1})
[104]	`Opt`	c_{kernel}, c_{rbf}	✓	✓	★☆☆	–	
[113]	`Pelt`	any $c(\cdot)$	–	✓	★★☆	✓	
[12]	Fused Lasso	c_{AR}	–	✓	★★★	–	Tikhonov regularization
[123]	`Win`	–	–	✓	★★★	–	dissimilarity measure (density ratio), see Remark 4
[139]	`Pelt`	c_{L_2}	–	✓	★★☆	–	supervised method to learn a penalty level (pen_{l_0})
[128]	`BinSeg`	c_{L_2}	✓	✓	★★★	✓	univariate signal
[30]	`Opt`	c_M	✓	–	★☆☆	–	supervised method to learn a suitable metric
[15]	`BinSeg`	$c_{\text{i.i.d.}}$	✓	✓	★★★	✓	exponential distributions family
[93]	`Opt`	c_{rank}	✓	–	★☆☆	✓	
[103]	`Pelt`	c_{kernel}, c_{rbf}	✓	✓	★☆☆	–	
[9]	`Pelt`	any $c(\cdot)$	–	✓	★★☆	–	
[21]	`Pelt`	c_{AR}	✓	✓	★☆☆	✓	

be based on local periodograms [153], or more recently, on PCA and wavelet transforms [154,155].

7.2. Supervised approaches for automatic calibration of detection algorithms

The calibration of detection methods is an important and complex step. As seen in this review, the search method mostly influences the trade-off between computational complexity and segmentation accuracy, and is relatively easy to choose. Conversely, the cost function and the constraint are related to (often) subjective considerations, which are respectively the nature of the changes to detect, and their amplitudes. Several authors (including the authors of these lines) have applied supervised learning procedures to find adapted cost functions and constraints, for a given data set and task. Generally speaking, such procedures can infer complex decision rules only using relevant examples. In the segmentation context, this supposes that experts are able to provide such examples: they consist in signals, manually segmented. The objective is to use those examples to design a change point detection method able to replicate the segmentation strategy of the experts. Two types of annotations (or labels, in the supervised learning terminology) are often considered: full and partial. For a fully annotated signal, the timestamps of all changes are provided by the expert. For a partially annotated signal, the expert only gives an approximate localization of the change points (exact locations are not known). This setting was first studied in [16], then extended in [31,140]. The objective was to find the optimal constraint to detect an unknown number of changes, using a few breakpoint annotations. As for the calibration of the cost function, the works of [30,107] propose a metric learning approach.

8. Summary table

This literature review is summarized in Table 2. When applicable, each publication is associated with a search method (such as `Opt`, `Pelt`, `BinSeg` or `Win`); this is a rough categorization rather than an exact implementation. Note that `Pelt` (introduced in 113) is sometimes associated with publications prior to 2012. It is because some linear penalties [62,144] were introduced long before `Pelt` was, and authors then resorted to quadratic (at best) algorithms. Nowadays, the same results can be obtained faster with `Pelt`. A guide of computational complexity is also provided. Quadratic methods are the slowest and have only one star while linear methods are given three stars. Algorithms for which the number of change points is an explicit input parameter work under the “known K” assumption. Algorithms that can be used even if the number of change points is unknown work under the “unknown K” assumption. (Certain methods can accommodate both situations.)

9. Presentation of the python package

Most of the approaches presented in this article are included in a Python scientific library for multiple change point detection in multivariate signals called `ruptures` [37]. The `ruptures` library is written in pure Python and available on Mac OS X, Linux and Windows platforms. Source code is available from [37] under the BSD license and deployed with a complete documentation that includes installation instructions and explanations with code snippets on advance use.

A schematic view is displayed on Fig. 12. Each block of this diagram is described in the following brief overview of `ruptures`' features.

- **Search methods** Our package includes the main algorithms from the literature, namely dynamic programming, detection with a l_0 constraint, binary segmentation, bottom-up segmentation and window-based segmentation. This choice is the result of a trade-off between exhaustiveness and adaptiveness. Rather than providing as many methods as possible, only algorithms which have been used in several different settings are included. In particular, numerous "mean-shift only" detection procedures were not considered. Implemented algorithms have sensible default parameters that can be changed easily through the functions' interface.

- **Cost functions** Cost functions are related to the type of change to detect. Within `ruptures`, one has access to parametric cost functions that can detect shifts in standard statistical quantities (mean, scale, linear relationship between dimensions, autoregressive coefficients, etc.) and non-parametric cost functions (kernel-based or Mahalanobis-type metric) that can, for instance, detect distribution changes [30,97].
- **Constraints** All methods can be used whether the number of change points is known or not. In particular, `ruptures` implements change point detection under a cost budget and with a linear penalty term [17,113].
- **Evaluation** Evaluation metrics are available to quantitatively compare segmentations, as well as a display module to visually inspect algorithms' performances.
- **Input** Change point detection can be performed on any univariate or multivariate signal that fits into a *Numpy* array. A few standard non-stationary signal generators are included.
- **Consistent interface and modularity** Discrete optimization methods and cost functions are the two main ingredients of change point detection. Practically, each is related to a specific object in the code, making the code highly modular: available optimization methods and cost functions can be connected and composed. An appreciable by-product of this approach is that a new contribution, provided its interface follows a few guidelines, can be integrated seamlessly into `ruptures`.
- **Scalability** Data exploration often requires to run several times the same methods with different sets of parameters. To that end, a cache is implemented to keep intermediate results in memory, so that the computational cost of running the same algorithm several times on the same signal is greatly reduced. We also add the possibility for a user with speed constraints to sub-sample their signals and set a minimum distance between change points.

10. Conclusion

In this article, we have reviewed numerous methods to perform change point detection, organized within a common framework. Precisely, all methods are described as a collection of three elements: a cost function, a search method and a constraint on the number of changes to detect. This approach is intended to facilitate prototyping of change point detection methods: for a given segmentation task, one can pick among the described elements to design an algorithm that fits its use-case. Most detection procedures described above are available within the Python language from the package `ruptures` [37], which is the most comprehensive change point detection library. Its consistent interface and modularity allow painless comparison between methods and easy integration of new contributions. In addition, a thorough documentation is available for novice users. Thanks to the rich Python ecosystem, `ruptures` can be used in coordination with numerous other scientific libraries.

Declaration of Competing Interest

The authors declare no conflicts of interest.

Acknowledgement

This work was supported by a public grant as part of the Investissement d'avenir project, reference ANR-11-LABX-0056-LMH, LabEx LMH.

References

[1] E.S. Page, Continuous inspection schemes, Biometrika 41 (1954) 100–105.
[2] E.S. Page, A test for a change in a parameter occurring at an unknown point, Biometrika 42 (1955) 523–527.
[3] M. Basseville, I. Nikiforov, Detection of abrupt changes: theory and application, 104, Prentice Hall Englewood Cliffs, 1993.
[4] B.E. Brodsky, B.S. Darkhovsky, Nonparametric methods in change point problems, Springer Netherlands, 1993.
[5] M. Csörgö, L. Horváth, Limit theorems in change-point analysis, 1997. Chichester, New York.
[6] J. Chen, A.K. Gupta, Parametric Statistical Change Point Analysis: With Applications to Genetics, Medicine, and Finance, Springer Science & Business Media, 2011.
[7] M. Lavielle, G. Teyssière, Adaptive detection of multiple change-points in asset price volatility, in: Long-Memory in Economics, Springer Verlag, Berlin, Germany, 2007, pp. 129–156.
[8] V. Jandhyala, S. Fotopoulos, I. Macneill, P. Liu, Inference for single and multiple change-points in time series, J. Time Ser. Anal. 34 (4) (2013) 423–446, doi:10.1111/jtsa.12035.
[9] K. Haynes, I.A. Eckley, P. Fearnhead, Computationally efficient changepoint detection for a range of penalties, J. Comput. Graph. Stat. 26 (1) (2017) 134–143.
[10] F. Desobry, M. Davy, C. Doncarli, An online kernel change detection algorithm, IEEE Tran. Signal Process. 53 (8) (2005) 2961–2974.
[11] Z. Harchaoui, F. Vallet, A. Lung-Yut-Fong, O. Cappé, A regularized kernel-based approach to unsupervised audio segmentation, in: Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2009, pp. 1665–1668. Taipei, Taiwan.
[12] D. Angelosante, G.B. Giannakis, Group lassoing change-points piece-constant AR processes, EURASIP J. Adv. Signal Process. 70 (2012).
[13] N. Seichepine, S. Essid, C. Fevotte, O. Cappé, Piecewise constant nonnegative matrix factorization, in: Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2014, pp. 6721–6725. Florence, Italy.
[14] J. Bai, P. Perron, Estimating and testing linear models with multiple structural changes, Econometrica 66 (1) (1998) 47–78.
[15] K. Frick, A. Munk, H. Sieling, Multiscale change point inference, J. R. Stat. Soc. Ser. B 76 (3) (2014) 495–580.
[16] T. Hocking, G. Schleiermacher, I. Janoueix-Lerosey, V. Boeva, J. Cappo, O. Delattre, F. Bach, J.-P. Vert, Learning smoothing models of copy number profiles using breakpoint annotations, BMC Bioinformat. 14 (1) (2013) 164.
[17] R. Maidstone, T. Hocking, G. Rigaill, P. Fearnhead, On optimal multiple changepoint algorithms for large data, Stat. Comput. 27 (2) (2017) 519–533.
[18] J.-P. Vert, K. Bleakley, Fast detection of multiple change-points shared by many signals using group LARS, in: Advances in Neural Information Processing Systems (NIPS), 1, 2010, pp. 2343–2351. Vancouver, Canada.
[19] F. Picard, S. Robin, M. Lavielle, C. Vaisse, J.-J. Daudin, A statistical approach for array CGH data analysis, BMC Bioinformat. 6 (1) (2005) 27.
[20] Y. Guédon, Exploring the latent segmentation space for the assessment of multiple change-point models, Comput. Stat. 28 (6) (2013) 2641–2678.
[21] S. Chakar, É. Lebarbier, C. Levy-Leduc, S. Robin, A robust approach for estimating change-points in the mean of an AR(1) process, Bernouilli Soc. Math. Stat.Probab. 23 (2) (2017) 1408–1447.
[22] L. Oudre, R. Barrois-Müller, T. Moreau, C. Truong, R. Dadashi, T. Grégory, D. Ricard, N. Vayatis, C. De Waele, A. Yelnik, P.-P. Vidal, Détection automatique des pas à partir de capteurs inertiels pour la quantification de la marche en consultation, Neurophysiologie Clinique/Clin. Neurophysiol. 45 (4–5) (2015) 394.
[23] J. Audiffren, R. Barrois-Müller, C. Provost, É. Chiarovano, L. Oudre, T. Moreau, C. Truong, A. Yelnik, N. Vayatis, P.-P. Vidal, C. De Waele, S. Buffat, D. Ricard, Évaluation de l'équilibre et prédiction des risques de chutes en utilisant une wii board balance, Neurophysiologie Clinique/Clin. Neurophysiol. 45 (4–5) (2015) 403.
[24] S. Liu, A. Wright, M. Hauskrecht, Change-point detection method for clinical decision support system rule monitoring, Artif. Intell. Med. 91 (2018) 49–56.
[25] R. Maidstone, Efficient analysis of complex changepoint problems, Lancaster University, 2016 Ph.D. thesis.
[26] J. Verbesselt, R. Hyndman, G. Newnham, D. Culvenor, Detecting trend and seasonal changes in satellite images time series, Remote Sens. Environ. (114) (2010) 106–115.
[27] J. Reeves, J. Chen, X.L. Wang, R. Lund, Q.Q. Lu, A review and comparison of changepoint detection techniques for climate data, J. Appl. Meteorol. Climatol. 46 (6) (2007) 900–915, doi:10.1175/JAM2493.1.

[28] C. Lévy-Leduc, F. Roueff, Detection and localization of change-points in high--dimensional network traffic data, Ann. Appl. Stat. 3 (2) (2009) 637–662.

[29] A. Lung-Yut-Fong, C. Lévy-Leduc, O. Cappé, Distributed detection/localization of change-points in high-dimensional network traffic data, Stat. Comput. 22 (2) (2012) 485–496.

[30] R. Lajugie, F. Bach, S. Arlot, Large-margin metric learning for constrained partitioning problems, in: Proceedings of the 31st International Conference on Machine Learning (ICML), 2014, pp. 297–395. Beijing, China.

[31] T. Hocking, G. Rigaill, G. Bourque, PeakSeg: constrained optimal segmentation and supervised penalty learning for peak detection in count data, in: Proceedings of the International Conference on Machine Learning (ICML), 2015, pp. 324–332. Lille, France.

[32] R. Barrois-Müller, D. Ricard, L. Oudre, L. Tlili, C. Provost, A. Vienne, P.-P. Vidal, S. Buffat, A. Yelnik, 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/Clin. Neurophysiol. 46 (4) (2016) 244.

[33] R. Barrois-Müller, T. Gregory, L. Oudre, T. Moreau, C. Truong, A. Aram Pulini, A. Vienne, C. Labourdette, N. Vayatis, S. Buffat, A. Yelnik, C. de Waele, S. Laporte, P.-P. Vidal, D. Ricard, An automated recording method in clinical consultation to rate the limp in lower limb osteoarthritis, PLoS One 11 (10) (2016) e0164975.

[34] L. Oudre, R. Barrois-Müller, T. Moreau, C. Truong, A. Vienne-Jumeau, D. Ricard, N. Vayatis, P.-P. Vidal, Template-based step detection with inertial measurement units, Sensors 18 (11) (2018).

[35] C. Truong, L. Oudre, N. Vayatis, Segmentation de signaux physiologiques par optimisation globale, in: Proceedings of the Groupe de Recherche et d'Etudes en Traitement du Signal et des Images (GRETSI), 2015. Lyon, France.

[36] R. Barrois-Müller, L. Oudre, T. Moreau, C. Truong, N. Vayatis, S. Buffat, A. Yelnik, C. de Waele, T. Gregory, S. Laporte, P.P. Vidal, D. Ricard, Quantify osteoarthritis gait at the doctor's office: a simple pelvis accelerometer based method independent from footwear and aging, Comput. Method. Biomech. Biomed. Eng. 18 Suppl 1 (2015) 1880–1881.

[37] C. Truong, ruptures: change point detection in python, 2018, [Online]. http://ctruong.perso.math.cnrs.fr/ruptures.

[38] L. Rabiner, A tutorial on hidden Markov models and selected applications in speech recognition, Proc. IEEE 77 (2) (1989) 257–286.

[39] S.I.M. Ko, T.T.L. Chong, P. Ghosh, Dirichlet process hidden markov multiple change-point model, Bayesian Anal. 10 (2) (2015) 275–296.

[40] A.F. Martínez, R.H. Mena, On a nonparametric change point detection model in markovian regimes, Bayesian Anal. 9 (4) (2014) 823–858.

[41] D. Barry, J.A. Hartigan, Product partition models for change point problems, Ann. Stat. 20 (1) (1992) 260–279.

[42] D. Barry, J.A. Hartigan, A bayesian analysis for change point problems, J. Am. Stat. Assoc. 88 (421) (1993) 309–319.

[43] S. Aminikhanghahi, D.J. Cook, A survey of methods for time series change point detection, Know. Inf. Syst. 51 (2) (2017) 339–367.

[44] Y.S. Niu, N. Hao, H. Zhang, Multiple change-point detection: a selective overview, Statistica Sci. 31 (4) (2016) 611–623.

[45] J. Bai, P. Perron, Multiple structural change models: a simulation analysis, J. Appl. Econom. 18 (2003) 1–22.

[46] S. Chakar, É. Lebarbier, C. Levy-Leduc, S. Robin, AR1seg: segmentation of an autoregressive Gaussian process of order 1, 2014. URL https://cran.r-project.org/package=AR1seg.

[47] L. Boysen, A. Kempe, V. Liebscher, A. Munk, O. Wittich, Consistencies and rates of convergence of jump-penalized least squares estimators, Ann. Stat. 37 (1) (2009) 157–183.

[48] Z. Harchaoui, C. Lévy-Leduc, Multiple change-point estimation with a total variation penalty, J. Am. Stat. Assoc. 105 (492) (2010) 1480–1493.

[49] M. Lavielle, Detection of multiples changes in a sequence of dependant variables, Stochast. ProcessesAppl. 83 (1) (1999) 79–102.

[50] F. Pein, H. Sieling, A. Munk, Heterogeneous change point inference, J. R. Stat. Soc. Ser B (Stat. Methodol.) 79 (4) (2017) 1207–1227.

[51] H. Keshavarz, C. Scott, X. Nguyen, Optimal change point detection in gaussian processes, J. Stat. Plann. Inference 193 (2018) 151–178.

[52] M. Lavielle, É. Moulines, Least-squares estimation of an unknown number of shifts in a time series, J. Time Ser. Anal. 21 (1) (2000) 33–59.

[53] A. Sen, M.S. Srivastava, On tests for detecting change in mean, Ann. Stat. 3 (1) (1975) 98–108.

[54] P.R. Krishnaiah, Review about estimation of change points, Handbook Stat. 7 (1988) 375–402.

[55] A. Aue, L. Horvàth, Structural breaks in time series, J. Time Ser. Anal. 34 (2012) 1–16.

[56] P. Fearnhead, Exact and efficient bayesian inference for multiple changepoint problems, Stat. Comput. 16 (2) (2006) 203–213.

[57] T. Górecki, L. Horváth, P. Kokoszka, Change point detection in heteroscedastic time series, Econ. Stat. 7 (2018) 63–88.

[58] Y.-X. Fu, R.N. Curnow, Maximum likelihood estimation of multiple change points, Biometrika 77 (3) (1990) 563–573.

[59] H. He, T.S. Severini, Asymptotic properties of maximum likelihood estimators in models with multiple change points, Bernoulli 16 (3) (2010) 759–779.

[60] H. Chernoff, S. Zacks, Estimating the current mean of a normal distribution which is subjected to changes in time, Ann. Math. Stat. 35 (3) (1964) 999–1018.

[61] G. Lorden, Procedures for reacting to a change in distribution, Ann. Math. Stat. 42 (6) (1971) 1897–1908.

[62] C.L. Mallows, Some comments on Cp, Technometrics 15 (4) (1973) 661–675.

[63] G. Hébrail, B. Hugueney, Y. Lechevallier, F. Rossi, Exploratory analysis of functional data via clustering and optimal segmentation, Neurocomputing 73 (7–9) (2010) 1125–1141.

[64] S. Chib, Estimation and comparison of multiple change-point models, J. Econ. 86 (2) (1998) 221–241.

[65] M. Lavielle, Using penalized contrasts for the change-point problem, Signal Process. 85 (8) (2005) 1501–1510.

[66] J. Bai, Least squares estimation of a shift in linear processes, J. Time Ser. Anal. 15 (5) (1994) 453–472.

[67] J. Bai, Least absolute deviation of a shift, Econom. Theory 11 (3) (1995) 403–436.

[68] J. Bai, Testing for parameter constancy in linear regressions: an empirical distribution function approach, Econometrica 64 (3) (1996) 597–622.

[69] J. Bai, Vector autoregressive models with structural changes in regression coefficients and in variancecovariance matrices, Ann. Econ. Finance 1 (2) (2000) 301–336.

[70] J. Bai, Estimation of a change-point in multiple regression models, Rev. Econ. Stat. 79 (4) (1997) 551–563.

[71] Z. Qu, P. Perron, Estimating and testing structural changes in multivariate regressions, Econometrica 75 (2) (2007) 459–502.

[72] J.J. Bai, P. Perron, Computation and analysis of multiple structural change models, J. Appl. Econom. 18 (1) (2003) 1–22.

[73] J. Bai, Estimation of multiple-regime regressions with least absolutes deviation, J. Stat. Plan. Inference 74 (1998) 103–134.

[74] J. Bai, Likelihood ratio tests for multiple structural changes, J. Econom. 91 (2) (1999) 299–323.

[75] J. Bai, P. Perron, Critical values for multiple structural change tests, Econom. J. 6 (1) (2003) 72–78.

[76] P. Perron, Dealing with structural breaks, Palgrave HandbookEconom. 1 (2) (2006) 278–352.

[77] J. Bai, Common breaks in means and variances for panel data, J. Econom. 157 (2010) 78–92.

[78] J. Bai, R.L. Lumsdaine, J.H. Stock, Testing for and dating common breaks in multivariate time series, Rev. Econ. Stud. 65 (3) (1998) 395–432.

[79] P. Perron, Z. Qu, Estimating restricted structural change models, J. Econom. 134 (2) (2006) 373–399.

[80] C. Han, A. Taamouti, Partial structural break identification, Oxford Bull. Econ. Stat. 79 (2) (2017) 145–164.

[81] B.M. Doyle, J. Faust, Breaks in the variability and comovement of G-7 economic growth, Rev. Econ. Stat. 87 (4) (2005) 721–740.

[82] C.F.H. Nam, J.A.D. Aston, A.M. Johansen, Quantifying the uncertainty in change points, J. Time Ser. Anal. 33 (2012) 807–823.

[83] P.C. Mahalanobis, On the generalised distance in statistics, Proc. Natl. Inst. Sci. India 2 (1) (1936) 49–55.

[84] T. Hastie, R. Tibshirani, J. Friedman, The Elements of Statistical Learning, 2nd, Springer, 2009.

[85] E.P. Xing, M.I. Jordan, S.J. Russell, Distance metric learning, with application to clustering with side-Information, in: Advances in Neural Information Processing Systems (NIPS), 2003, pp. 521–528.

[86] J.V. Davis, B. Kulis, P. Jain, S. Sra, I.S. Dhillon, Information-theoretic metric learning, in: Proceedings of the 24th International Conference on Machine Learning (ICML), 2007, pp. 209–216. Corvalis, Oregon, USA.

[87] J.H.J. Einmahl, I.W. McKeague, Empirical likelihood based hypothesis testing, Bernoulli 9 (2) (2003) 267–290.

[88] C. Zou, G. Yin, F. Long, Z. Wang, Nonparametric maximum likelihood approach to multiple change-point problems, Ann. Stat. 42 (3) (2014) 970–1002.

[89] K. Haynes, P. Fearnhead, I.A. Eckley, A computationally efficient nonparametric approach for changepoint detection, Stat. Comput. 27 (2017) 1293–1305.

[90] J. Zhang, Powerful two-sample tests based on the likelihood ratio, Technometrics 48 (1) (2006) 95–103.

[91] S. Clemencon, M. Depecker, N. Vayatis, AUC optimization and the two-sample problem, in: Advances in Neural Information Processing Systems (NIPS), 2009, pp. 360–368. Vancouver, Canada.

[92] J.H. Friedman, L.C. Rafsky, Multivariate generalizations of Wald-Wolfowitz and Smirnov two-sample tests, The Annals of Statistics 7 (4) (1979) 697–717.

[93] A. Lung-Yut-Fong, C. Lévy-Leduc, O. Cappé, Homogeneity and change-point detection tests for multivariate data using rank statistics, J. de la Société Française de Statistique 156 (4) (2015) 133–162.

[94] F. Wilcoxon, Individual comparisons by ranking methods, Biometr. Bull. 1 (6) (1945) 80–83.

[95] E.L. Lehman, J.P. Romano, Testing Statistical Hypotheses, 101, 3, springer, 2006.

[96] M.G. Kendall, Rank Correlation Methods, Charles Griffin, London, England, 1970.

[97] Z. Harchaoui, O. Cappé, Retrospective multiple change-point estimation with kernels, in: Proceedings of the IEEE/SP Workshop on Statistical Signal Processing, 2007, pp. 768–772. Madison, Wisconsin, USA.

[98] B. Schölkopf, A. Smola, Learning with kernels, MIT Press, Cambridge, USA, 2002.

[99] A. Gretton, K.M. Borgwardt, M.J. Rasch, B. Schölkopf, A. Smola, A kernel two-sample test, J. Mach. Learn. Res. (JMLR) 13 (2012) 723–773.

[100] A. Celisse, G. Marot, M. Pierre-Jean, G. Rigaill, New efficient algorithms for multiple change-point detection with reproducing kernels, Comput. Stat. Data Analysis 128 (2018) 200–220.

[101] B.K. Sriperumbudur, A. Gretton, K. Fukumizu, G. Lanckriet, B. Schölkopf, Injective Hilbert space embeddings of probability measures, in: Proceedings of the 21st Conference on Learning Theory (COLT), 2008, pp. 9–12. Helsinki, Finland

[102] J. Shawe-Taylor, N. Cristianini, Kernel Methods for Pattern Analysis, Cambridge university press, 2004.

[103] D. Garreau, S. Arlot, Consistent change-point detection with kernels, Electron. J. Stat. 12 (2) (2018) 4440–4486.

[104] S. Arlot, A. Celisse, Z. Harchaoui, Kernel change-point detection, arXiv preprint arXiv:1202.3878 (2012) 1–26.

[105] J. Cabrieto, F. Tuerlinckx, P. Kuppens, F.H. Wilhelm, M. Liedlgruber, E. Ceulemans, Capturing correlation changes by applying kernel change point detection on the running correlations, Inf. Sci. 447 (2018) 117–139.

[106] P. Jain, B. Kulis, J.V. Davis, I.S. Dhillon, Metric and kernel learning using a linear transformation, J. Mach. Learn. Res. (JMLR) 13 (2012) 519–547.

[107] C. Truong, L. Oudre, N. Vayatis, Supervised kernel change point detection with partial annotations, in: Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2019, pp. 1–5. Brighton, UK.

[108] S.M. Kay, A.V. Oppenheim, Fundamentals of Statistical Signal Processing, Volume II: Detection Theory, Prentice Hall, 1993.

[109] R. Bellman, On a routing problem, Q. Appl. Math. 16 (1) (1955) 87–90.

[110] M. Lavielle, Optimal segmentation of random processes, IEEE Trans. Signal Process. 46 (5) (1998) 1365–1373.

[111] G. Rigaill, A pruned dynamic programming algorithm to recover the best segmentations with 1 to k_max change-points., J. de la Société Française de Statistique 156 (4) (2015) 180–205.

[112] B. Hugueney, G. Hébrail, Y. Lechevallier, F. Rossi, Simultaneous clustering and segmentation for functional data, in: Proceedings of 16th European Symposium on Artificial Neural Networks (ESANN), 2009, pp. 281–286. Bruges, Belgium.

[113] R. Killick, P. Fearnhead, I. Eckley, Optimal detection of changepoints with a linear computational cost, J. Amer. Stat. Assoc. 107 (500) (2012) 1590–1598.

[114] J. Chen, A.K. Gupta, Testing and locating variance changepoints with application to stock prices, J. Amer. Stat. Assoc. 92 (438) (1997) 739–747.

[115] H. Vullings, M. Verhaegen, H. Verbruggen, ECG segmentation using time-warping, in: Lecture Notes in Computer Science, Springer, 1997, pp. 275–286.

[116] B.E. Brodsky, B.S. Darkhovsky, A.Y. Kaplan, S.L. Shishkin, A nonparametric method for the segmentation of the EEG, Comput. Method. Program. Biomed. 60 (2) (1999) 93–106.

[117] R. Esteller, G. Vachtsevanos, J. Echauz, B. Litt, A comparison of waveform fractal dimension algorithms, IEEE Trans. Circuit. Syst. I: Fundamental Theory and Applications 48 (2) (2001) 177–183.

[118] K. Karagiannaki, A. Panousopoulou, P. Tsakalides, An online feature selection architecture for Human Activity Recognition, in: Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2017, pp. 2522–2526. New Orleans, LA, USA.

[119] S. Adak, Time-dependent spectral analysis of nonstationary time series, J. Am. Stat. Assoc. 93 (444) (1998) 1488–1501.

[120] S.S. Chen, P.S. Gopalakrishnan, Speaker, environment and channel change detection and clustering via the Bayesian information criterion, in: Proceedings of the DARPA Broadcast News Transcription and Understanding Workshop, 1998, p. 8. Landsdowne, VA.

[121] E. Keogh, S. Chu, D. Hart, M. Pazzani, Segmenting time series: a survey and novel approach, Data Min. Time Ser. Databases 57 (1) (2004) 1–22.

[122] Z. Harchaoui, F. Bach, É. Moulines, Kernel change-point analysis, in: Advances in Neural Information Processing Systems (NIPS), 2008, pp. 609–616. Vancouver, Canada.

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

[124] D. Kifer, S. Ben-David, J. Gehrke, Detecting change in data streams, in: Proceedings of the Thirtieth International Conference on Very Large Data Bases (VLDB) - Volume 30, 2004, pp. 180–191. Toronto, Canada.

[125] R. Prescott Adams, D.J.C. MacKay, Bayesian Online Changepoint Detection, Technical Report, 2007.

[126] A.B. Olshen, E.S. Venkatraman, R. Lucito, M. Wigler, Circular binary segmentation for the analysis of array-based DNA copy number data, Biostatistics 5 (4) (2004) 557–572.

[127] J. Bai, Estimating multiple breaks one at a time, Econom. Theory 13 (3) (1997) 315–352.

[128] P. Fryzlewicz, Wild binary segmentation for multiple change-point detection, Ann. Statistics 42 (6) (2014) 2243–2281, doi:10.1214/14-AOS1245.

[129] J. Himberg, K. Korpiaho, H. Mannila, J. Tikanmaki, H.T. Toivonen, Time series segmentation for context recognition in mobile devices, in: Proceedings of the IEEE International Conference on Data Mining (ICDM), 2001, pp. 203–210.

[130] Y.S. Niu, H. Zhang, The screening and ranking algorithm to detect DNA copy number variations, Ann. Appl. Stat. 6 (3) (2012) 1306–1326.

[131] W.R. Lai, M.D. Johnson, R. Kucherlapati, P.J. Park, Comparative analysis of algorithms for identifying amplifications and deletions in array CGh data, Bioinformatics 21 (19) (2005) 3763–3770.

[132] H. Willenbrock, J. Fridlyand, A comparison study: applying segmentation to array CGH data for downstream analyses, Bioinformatics 21 (22) (2005) 4084–4091.

[133] E.S. Venkatraman, A.B. Olshen, A faster circular binary segmentation algorithm for the analysis of array CGH data, Bioinformatics 23 (6) (2007) 657–663.

[134] E. Keogh, S. Chu, D. Hart, M. Pazzani, An online algorithm for segmenting time series, in: Proceedings of the IEEE International Conference on Data Mining (ICDM), 2001, pp. 289–296. San Jose, California, USA.

[135] Y.-C. Yao, Estimating the number of change-points via Schwarz' criterion, Stat. Probab. Lett. 6 (3) (1988) 181–189.

[136] Y.-C. Yao, S.T. Au, Least-squares estimation of a step function, Sankhy 51 (3) (1989) 370–381.

[137] S. Arlot, A. Celisse, A survey of cross-validation procedures for model selection, Stat. Surv. 4 (2010) 40–79.

[138] L. Birgé, P. Massart, Minimal penalties for gaussian model selection, Probability Theory and Related Fields 138 (1) (2007) 33–73.

[139] T. Hocking, G. Rigaill, J.-P. Vert, F. Bach, Learning sparse penalties for change–point detection using max margin interval regression, in: Proceedings of the International Conference on Machine Learning (ICML), 2013, pp. 172–180. Atlanta, USA.

[140] C. Truong, L. Oudre, N. Vayatis, Penalty learning for changepoint detection, in: Proceedings of the European Signal Processing Conference (EUSIPCO), 2017, pp. 1569–1573. Kos, Greece.

[141] G. Schwarz, Estimating the dimension of a model, Ann Stat 6 (2) (1978) 461–464.

[142] R. Tibshirani, Regression shrinkage and selection via the lasso, J R Stat. Soc. Ser B (Methodological) 58 (1) (1996) 267–288.

[143] J.-J. Jeon, J. Hyun Sung, E.-S. Chung, Abrupt change point detection of annual maximum precipitation using fused lasso, J. Hydrol. 538 (2016) 831–841.

[144] N.R. Zhang, D.O. Siegmund, A modified Bayes information criterion with applications to the analysis of comparative genomic hybridization data., Biometrics 63 (1) (2007) 22–32.

[145] É. Lebarbier, E. Lebarbier, Detecting multiple change-points in the mean of gaussian process by model selection, Signal Process. 85 (4) (2005) 717–736, doi:10.1016/j.sigpro.2004.11.012.

[146] J. Bai, S. Shi, Estimating high dimensional covariance matrices and its applications, Ann. Econom. Finance 12 (2) (2011) 199–215.

[147] Z. Bai, H. Saranadasa, Effect of high dimension: by an example of a two sample problem, Stat. Sinica 6 (2) (1996) 311–329.

[148] S.X. Chen, Y.L. Qin, A two-sample test for high-dimensional data with applications to gene-set testing, Ann. Stat. 38 (2) (2010) 808–835.

[149] F. Enikeeva, Z. Harchaoui, High-dimensional change-point detection with sparse alternatives, arXiv preprint arXiv:1312.1900 (2014) 1–33.

[150] M. Jirak, Uniform change point tests in high dimension, Ann. Stat. 43 (6) (2015) 2451–2483.

[151] T. Wang, R.J. Samworth, High dimensional change point estimation via sparse projection, J. R. Stat. Soc. Ser. B (Stat. Methodol.) 80 (1) (2018) 57–83.

[152] M. Jirak, Change-point analysis in increasing dimension, J. Multivar. Anal. 111 (2012) 136–159.

[153] H. Cho, P. Fryzlewicz, Multiple change-point detection for high dimensional time series via Szparsified binary segmentation, J. R. Stat. Soc 77 (2) (2014) 475–507.

[154] M. Barigozzi, H. Cho, P. Fryzlewicz, Simultaneous multiple change-point and factor analysis for high-dimensional time series, J. Econ. 206 (1) (2018) 187–225.

[155] S. Ma, L. Su, Estimation of large dimensional factor models with an unknown number of breaks, J. Econ. 207 (1) (2018) 1–29.

[156] L.Y. Vostrikova, Detecting disorder in multidimensional random processes, Soviet Math. Dokl. 24 (1981) 55–59.

[157] L. Birgé, P. Massart, Gaussian model selection, J. Eur. Math. Soc. 3 (3) (2001) 203–268.