Detecting linear trend changes in data sequences

Hyeyoung Maeng[1] · Piotr Fryzlewicz[2]

Abstract

We propose TrendSegment, a methodology for detecting multiple change-points corresponding to linear trend changes in one dimensional data. A core ingredient of TrendSegment is a new Tail-Greedy Unbalanced Wavelet transform: a conditionally orthonormal, bottom-up transformation of the data through an adaptively constructed unbalanced wavelet basis, which results in a sparse representation of the data. Due to its bottom-up nature, this multiscale decomposition focuses on local features in its early stages and on global features next which enables the detection of both long and short linear trend segments at once. To reduce the computational complexity, the proposed method merges multiple regions in a single pass over the data. We show the consistency of the estimated number and locations of change-points. The practicality of our approach is demonstrated through simulations and two real data examples, involving Iceland temperature data and sea ice extent of the Arctic and the Antarctic. Our methodology is implemented in the R package `trendsegmentR`, available from CRAN.

Keywords Change-point detection · Bottom-up algorithms · Piecewise-linear signal · Wavelets

1 Introduction

Multiple change-point detection is a problem of importance in many applications; recent examples include automatic detection of change-points in cloud data to maintain the performance and availability of an app or a website (James et al. 2016), climate change detection in tropical cyclone records (Robbins et al. 2011), detecting exoplanets from light curve data (Fisch et al. 2018), detecting changes in the DNA copy number (Olshen et al. 2004; Jeng et al. 2012; Bardwell et al. 2017), estimation of stationary

✉ Hyeyoung Maeng
   hyeyoung.maeng@durham.ac.uk

1   Department of Mathematical Sciences, Durham University, Stockton Rd, Durham DH1 3LE, UK

2   Department of Statistics, London School of Economics, Houghton Street, London WC2A 2AE, UK

intervals in potentially cointegrated stock prices (Matteson et al. 2013), estimation of change-points in multi-subject fMRI data (Robinson et al. 2010) and detecting changes in vegetation trends (Jamali et al. 2015).

This paper considers the change-point model

$$X_t = f_t + \varepsilon_t, \quad t = 1, \ldots, T, \tag{1}$$

where f_t is a deterministic and piecewise-linear signal containing N change-points, i.e. time indices at which the slope and/or the intercept in f_t undergoes changes. These changes occur at unknown locations $\eta_1, \eta_2, \ldots, \eta_N$. In this article, we assume that the ε_t's are iid $N(0, \sigma^2)$ and in the supplementary material, we show how our method can be extended to dependent and/or non-Gaussian noise such as ε_t following a stationary Gaussian AR process or t-distribution. The true change-points $\{\eta_i\}_{i=1}^N$ are such that,

$$\begin{aligned}
&f_t = \theta_{\ell,1} + \theta_{\ell,2}\, t \text{ for } t \in [\eta_{\ell-1} + 1, \eta_\ell], \ \ell = 1, \ldots, N + 1 \\
&\text{where } f_{\eta_\ell} + \theta_{\ell,2} \neq f_{\eta_\ell+1} \text{ for } \ell = 1, \ldots, N.
\end{aligned} \tag{2}$$

This definition permits both continuous and discontinuous changes in the linear trend.

Our main interest is in the estimation of N and $\eta_1, \eta_2, \ldots, \eta_N$ under some assumptions that quantify the difficulty of detecting each η_i; therefore, our aim is to segment the data into sections of linearity in f_t. In detail, a change-point located close to its neighbouring ones can only be detected when it has a large enough size of linear trend change, while a change-point capturing a small size of linear trend change requires a longer distance from its adjacent change-points to be detected. Detecting linear trend changes is an important applied problem in a variety of fields, including climate change, as illustrated in Sect. 5.

The change-point detection procedure proposed in this paper is referred to as TrendSegment; it is designed to work well in the presence of either long or short spacings between neighbouring change-points, or a mixture of both. The engine underlying TrendSegment is a new Tail-Greedy Unbalanced Wavelet (TGUW) transform: a conditionally orthonormal, bottom-up transformation for univariate data sequences through an adaptively constructed unbalanced wavelet basis, which results in a sparse representation of the data. In this article, we show that TrendSegment offers good performance in estimating the number and locations of change-points across a wide range of signals containing constant and/or linear segments. TrendSegment is also shown to be statistically consistent and computationally efficient.

In earlier related work regarding linear trend changes, Bai and Perron (1998) consider the estimation of linear models with multiple structural changes by least-squares and present Wald-type tests for the null hypothesis of no change. Kim et al. (2009) and Tibshirani et al. (2014) consider 'trend filtering' with the L_1 penalty and Fearnhead et al. (2019) detect changes in the slope with an L_0 regularisation via a dynamic programming algorithm. Spiriti et al. (2013) study two algorithms for optimising the knot locations in least-squares and penalised splines. Baranowski et al. (2019) propose a multiple change-point detection device termed Narrowest-Over-Threshold (NOT), which focuses on the narrowest segment among those whose contrast exceeds a pre-specified threshold. Anastasiou and Fryzlewicz (2022) propose the Isolate-Detect (ID)

approach which continuously searches expanding data segments for changes. Yu et al. (2022) propose a two-step algorithm for detecting multiple change-points in piecewise polynomials with general degrees.

Keogh et al. (2004) mention that sliding windows, top-down and bottom-up approaches are three principal categories which most time series segmentation algorithms can be grouped into. Keogh et al. (2004) apply those three approaches to the detection of changes in linear trends in 10 different signals and discover that the performance of bottom-up methods is better than that of top-down methods and sliding windows, notably when the underlying signal has jumps, sharp cusps or large fluctuations. Bottom-up procedures have rarely been used in change-point detection. Matteson and James (2014) use an agglomerative algorithm for hierarchical clustering in the context of change-point analysis. Keogh et al. (2004) merge adjacent segments of the data according to a criterion involving the minimum residual sum of squares (RSS) from a linear fit, until the RSS falls under a certain threshold; but the lack of precise recipes for the choice of this threshold parameter causes the performance of this method to be somewhat unstable, as we report in Sect. 4.

As illustrated later in this paper, our TGUW transform, which underlies TrendSegment, is designed to work well in detecting frequent change-points or abrupt local features in which many existing change-point detection methods for the piecewise-linear model fail. The TGUW transform constructs, in a bottom-up way, an adaptive wavelet basis by consecutively merging neighbouring segments of the data starting from the finest level (throughout the paper, we refer to a wavelet basis as adaptive if it is constructed in a data-driven way). This enables it to identify local features at an early stage, before it proceeds to focus on more global features corresponding to longer data segments.

Fryzlewicz (2018) introduces the Tail-Greedy Unbalanced Haar (TGUH) transform, a bottom-up, agglomerative, data-adaptive transformation of univariate sequences that facilitates change-point detection in the piecewise-constant sequence model. The current paper extends this idea to adaptive wavelets other than adaptive Haar, which enables change-point detection in the piecewise-linear model (and, in principle, to higher-order piecewise polynomials, where the details can be found in Section G of the supplementary material). We emphasise that this extension from TGUH to TGUW is both conceptually and technically non-trivial, due to the fact that it is not a priori clear how to construct a suitable wavelet basis in TGUW for wavelets other than adaptive Haar; this is due to the non-uniqueness of the local orthonormal matrix transformation for performing each merge in TGUW, which does not occur in TGUH. We solve this issue by imposing certain guiding principles in the way the merges are performed, which enables detecting not only long trend segments, but also frequent change-points including abrupt local features. The computational cost of TGUW is the same as TGUH. Important properties of the TGUW transform include orthonormality conditional on the merging order, nonlinearity and "tail-greediness", and will be investigated in Sect. 2. The TGUW transform is the first step of the TrendSegment procedure, which involves four steps.

The remainder of the article is organised as follows. Section 2 gives a full description of the TrendSegment procedure and the relevant theoretical results are presented in Sect. 3. The supporting simulation studies are described in Sect. 4 and our methodology

is illustrated in Sect. 5 through climate datasets. The proofs of our main theoretical results are in Appendix 1. The supplementary material includes theoretical results for dependent and/or non-Gaussian noise, extension to piecewise-quadratic signal, details of robust threshold selection and extra simulation and data application results. The TrendSegment procedure is implemented in the R package `trendsegmentR`, available from CRAN.

2 Methodology

2.1 Summary of TrendSegment

The TrendSegment procedure for estimating the number and the locations of change-points includes four steps. We give the broad picture first and outline details in later sections.

1. *TGUW transformation.* Perform the TGUW transform, a bottom-up unbalanced adaptive wavelet transformation of the input data $X_1, \ldots, X_T$, by recursively applying local conditionally orthonormal transformations. This produces a data-adaptive multiscale decomposition of the data with $T - 2$ detail-type coefficients and 2 smooth coefficients. The resulting conditionally orthonormal transform of the data hopes to encode most of the energy of the signal in only a few detail-type coefficients arising at coarse levels (see Fig. 1 for an example output). This representation sparsity justifies thresholding in the next step.
2. *Thresholding.* Set to zero those detail coefficients whose magnitude is smaller than a pre-specified threshold as long as all the non-zero detail coefficients are connected to each other in the tree structure. This step performs "pruning" as a way of deciding the significance of the sparse representation obtained in step 1.
3. *Inverse TGUW transformation.* Obtain an initial estimate of f_t by carrying out the inverse TGUW transformation of the thresholded coefficient tree. The resulting estimator is discontinuous at the estimated change-points. It can be shown to be l_2-consistent, but not yet consistent for N or $\eta_1, \ldots, \eta_N$.
4. *Post-processing.* Post-process the estimate from step 3 by removing some change-points perceived to be spurious, which enables us to achieve estimation consistency for N and $\eta_1, \ldots, \eta_N$.

Figure 2 illustrates the first three steps of the TrendSegment procedure. We devote the following four sections to describing each step above in order.

2.2 TGUW transformation

2.2.1 Key principles of the TGUW transform

In the initial stage, the data are considered smooth coefficients and the TGUW transform iteratively updates the sequence of smooth coefficients by merging the adjacent sections of the data which are the most likely to belong to the same segment. The merging is done by performing an adaptively constructed orthonormal transformation
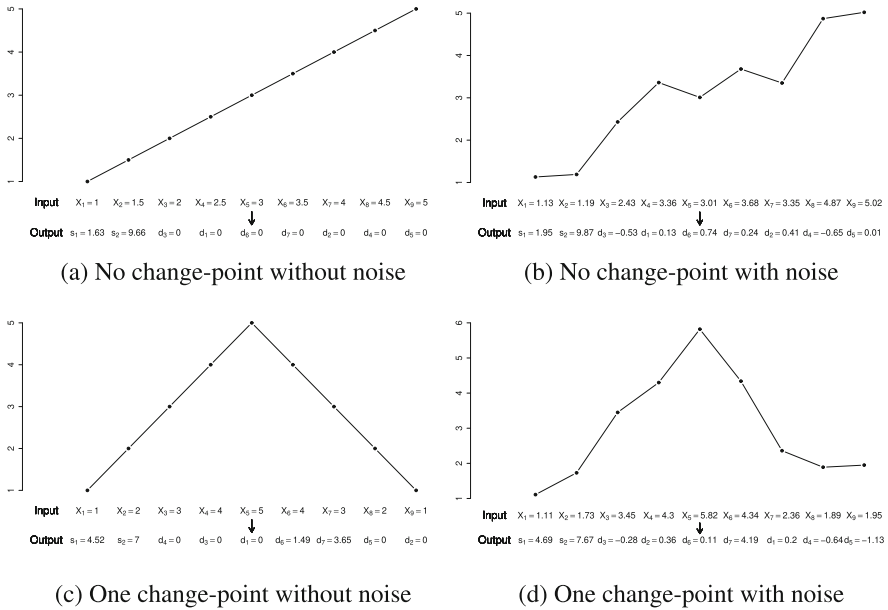
(a) No change-point without noise

(b) No change-point with noise

(c) One change-point without noise

(d) One change-point with noise

Fig. 1 Multiscale decomposition of the data through the TGUW transform when the data has no change-points (**a**, **b**) or one change-point (**c**, **d**). s_1 and s_2 are the smooth coefficients obtained through the TGUW transform and d_k is the detail coefficient obtained in the kth merge. When the data has no noise (**a**, **c**), $d_k = 0$ for all k in **a** while two non-zero coefficients d_6 and d_7 encode the single change in **c**

(a) TGUW transform
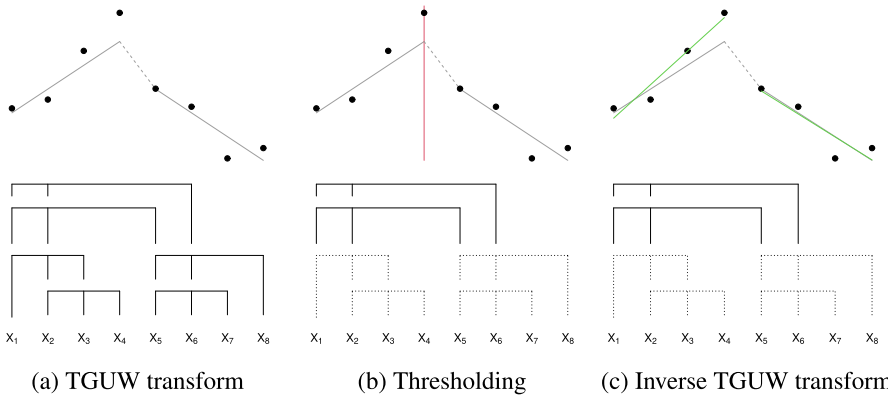
(b) Thresholding

(c) Inverse TGUW transform

Fig. 2 Illustration of the first three steps of the TrendSegment procedure with the observed data X_t (dots), the true signal f_t (grey line) and the tree of mergings; **a** TGUW transform constructs a tree by merging neighbouring segments, **b** In thresholding, surviving coefficients (solid line in the tree) are chosen by a pre-specified threshold, which decides the location of the estimated change point (red), **c** Inverse TGUW transform gives the estimated signal (green) based on the estimated change points obtained in thresholding

Table 1 Notation

X_p	p^{th} element of the observation vector $\mathbf{X} = \{X_1, X_2, \ldots, X_T\}^\top$
$s_{p,p}^0$	p^{th} initial smooth coefficient of the vector $\mathbf{s}^0$ where $\mathbf{X} = \mathbf{s}^0$
$d_{p,q,r}$	detail coefficient obtained from $\{X_p, \ldots, X_r\}$ (merges of Types 1 or 2)
$s_{p,r}^{[1]}, s_{p,r}^{[2]}$	smooth coefficients obtained from $\{X_p, \ldots, X_r\}$, paired under the "two together" rule
$d_{p,q,r}^{[1]}, d_{p,q,r}^{[2]}$	paired detail coefficients obtained by merging two adjacent subintervals, $\{X_p, \ldots, X_q\}$ and $\{X_{q+1}, \ldots, X_r\}$, where $r > q + 2$ and $q > p + 1$ (merge of Type 3)
$\mathbf{s}$	data sequence vector containing the (recursively updated) smooth and detail coefficients from the initial input $\mathbf{s}^0$

See Sect. 2.2.4 for formulae for the terms listed

to the chosen triplet of the smooth coefficients and in doing so, a data-adaptive unbalanced wavelet basis is established. The TGUW transform is completed after $T - 2$ such orthonormal transformations and each merge is performed under the following principles.

1. In each merge, three adjacent smooth coefficients are selected and the orthonormal transformation converts those three values into one detail and two (updated) smooth coefficients. The size of the detail coefficient gives information about the strength of the local linearity and the two updated smooth coefficients are associated with the estimated parameters (intercept and slope) of the local linear regression performed on the raw observations corresponding to the initially chosen three smooth coefficients.
2. *"Two together" rule.* The two smooth coefficients returned by the orthonormal transformation are paired in the sense that both contain information about one local linear regression fit. Thus, we require that any such pair of smooth coefficients cannot be separated when choosing triplets in any subsequent merges. We refer to this recipe as the "two together" rule.
3. To decide which triplet of smooth coefficients should be merged next, we compare the corresponding detail coefficients as their magnitude represents the strength of the corresponding local linear trend; the smaller the (absolute) size of the detail, the smaller the local deviation from linearity. Smooth coefficients corresponding to the smallest detail coefficients have priority in merging.

As merging continues under the "two together" rule, all mergings can be classified into one of three forms:
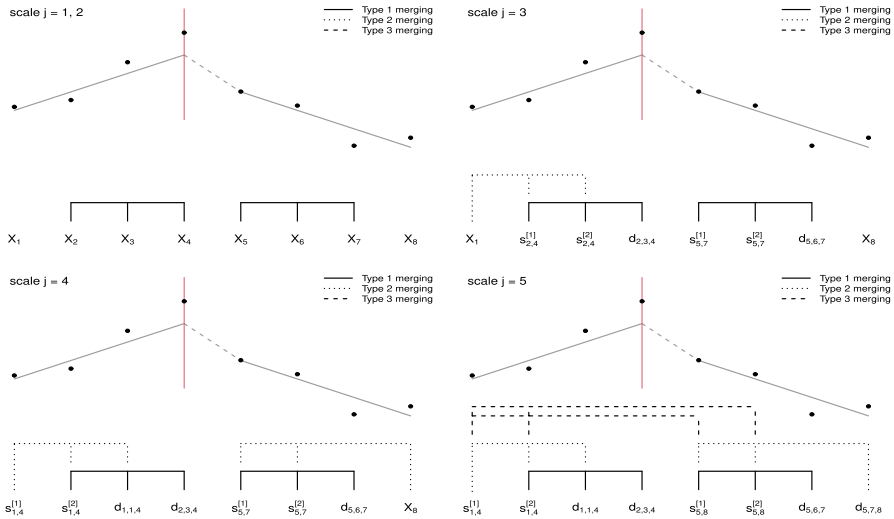
Fig. 3 Construction of tree for the example in Sect. 2.2.2; each diagram shows all merges performed up to the given scale with the data (dot), true signal (grey) and true change point (red)

- Type 1: merging three initial smooth coefficients,
- Type 2: merging one initial and a paired smooth coefficient,
- Type 3: merging two sets of (paired) smooth coefficients,

where Type 3 is composed of two merges of triplets and more details are given in Sect. 2.2.2.

2.2.2 Example

We now provide a simple example of the TGUW transformation; the accompanying illustration is in Fig. 3. The notation for this example and for the general algorithm introduced later is in Table 1. This example shows single merges at each pass through the data when the algorithm runs in a purely greedy way. We will later generalise it to multiple passes through the data, which will speed up computation (this device is referred to as "tail-greediness" as the algorithm merges those triplets corresponding to the lower tail of the distribution of local deviation from linearity in X). We refer to j^{th} pass through the data as scale j. Assume that we have the initial input $s^0 = (X_1, X_2, \ldots, X_8)$, so that the complete TGUW transform consists of 6 merges. We show 6 example merges one by one under the rules introduced in Sect. 2.2.1. This example demonstrates all three possible types of merges.

Scale $j = 1$. From the initial input $s^0 = (X_1, \ldots, X_8)$, we consider 6 triplets $(X_1, X_2, X_3), (X_2, X_3, X_4), (X_3, X_4, X_5), (X_4, X_5, X_6), (X_5, X_6, X_7), (X_6, X_7, X_8)$ and compute the size of the detail for each triplet, where the formula can be found in (7). Suppose that (X_2, X_3, X_4) gives the smallest size of detail, $|d_{2,3,4}|$, then merge (X_2, X_3, X_4) through the orthogonal transformation formulated in (8) and update the data sequence into $s = (X_1, s_{2,4}^{[1]}, s_{2,4}^{[2]}, d_{2,3,4}, X_5, X_6, X_7, X_8)$. We categorise this transformation into Type 1 (merging three initial smooth coefficients).

Scale $j = 2$. From now on, the "two together" rule is applied. Ignoring any detail coefficients in s, the possible triplets for next merging are $(X_1, s_{2,4}^{[1]}, s_{2,4}^{[2]})$, $(s_{2,4}^{[1]}, s_{2,4}^{[2]}, X_5)$, (X_5, X_6, X_7), (X_6, X_7, X_8). We note that $(s_{2,4}^{[2]}, X_5, X_6)$ cannot be considered as a candidate for next merging under the "two together" rule as this triplet contains only one (not both) of the paired smooth coefficients returned by the previous merging. Assume that (X_5, X_6, X_7) gives the smallest size of detail coefficient $|d_{5,6,7}|$ among the four candidates, then we merge them through the orthogonal transformation formulated in (8) and now update the sequence into $s = (X_1, s_{2,4}^{[1]}, s_{2,4}^{[2]}, d_{2,3,4}, s_{5,7}^{[1]}, s_{5,7}^{[2]}, d_{5,6,7}, X_8)$. This transformation is also Type 1.

Scale $j = 3$. We now compare four candidates for merging, $(X_1, s_{2,4}^{[1]}, s_{2,4}^{[2]})$, $(s_{2,4}^{[1]}, s_{2,4}^{[2]}, s_{5,7}^{[1]})$, $(s_{2,4}^{[2]}, s_{5,7}^{[1]}, s_{5,7}^{[2]})$ and $(s_{5,7}^{[1]}, s_{5,7}^{[2]}, X_8)$. The two triplets in middle, $(s_{2,4}^{[1]}, s_{2,4}^{[2]}, s_{5,7}^{[1]})$ and $(s_{2,4}^{[2]}, s_{5,7}^{[1]}, s_{5,7}^{[2]})$, are paired together as they contain two sets of paired smooth coefficients, $(s_{2,4}^{[1]}, s_{2,4}^{[2]})$ and $(s_{5,7}^{[1]}, s_{5,7}^{[2]})$, and if we were to treat these two triplets separately, we would be violating the "two together" rule. The summary detail coefficient for this pair of triplets is obtained as $d_{2,4,7} = \max(|d_{2,4,7}^{[1]}|, |d_{2,4,7}^{[2]}|)$, which is compared with those of the other triplets. Now suppose that $(X_1, s_{2,4}^{[1]}, s_{2,4}^{[2]})$ has the smallest size of detail; we merge this triplet and update the data sequence into $s = (s_{1,4}^{[1]}, s_{1,4}^{[2]}, d_{1,1,4}, d_{2,3,4}, s_{5,7}^{[1]}, s_{5,7}^{[2]}, d_{5,6,7}, X_8)$. This transformation is of Type 2.

Scale $j = 4$. We now have two pairs of paired coefficients: $(s_{1,4}^{[1]}, s_{1,4}^{[2]})$ and $(s_{5,7}^{[1]}, s_{5,7}^{[2]})$. Therefore, with the "two together" rule in mind, the only possible options for merging are: to merge the two pairs into $(s_{1,4}^{[1]}, s_{1,4}^{[2]}, s_{5,7}^{[1]}, s_{5,7}^{[2]})$, or to merge $(s_{5,7}^{[1]}, s_{5,7}^{[2]})$ with X_8. Suppose that the second merging is preferred. Then we perform Type 2 merge and update the data sequence into $s = (s_{1,4}^{[1]}, s_{1,4}^{[2]}, d_{1,1,4}, d_{2,3,4}, s_{5,8}^{[1]}, s_{5,8}^{[2]}, d_{5,6,7}, d_{5,7,8})$.

Scale $j = 5$. The only remaining step is merging $(s_{1,4}^{[1]}, s_{1,4}^{[2]})$ and $(s_{5,8}^{[1]}, s_{5,8}^{[2]})$ into $(s_{1,4}^{[1]}, s_{1,4}^{[2]}, s_{5,8}^{[1]}, s_{5,8}^{[2]})$. This transformation is Type 3 and performed in two stages as follows. In the first stage, we merge $(s_{1,4}^{[1]}, s_{1,4}^{[2]}, s_{5,8}^{[1]})$ and then update the sequence temporarily as $s = (s_{1,8}^{[1']}, s_{1,8}^{[2']}, d_{1,1,4}, d_{2,3,4}, d_{1,4,8}^{[1]}, s_{5,8}^{[2]}, d_{5,6,7}, d_{5,7,8})$. In the second stage, we merge $(s_{1,8}^{[1']}, s_{1,8}^{[2']}, s_{5,8}^{[2]})$, which gives the updated sequence $s = (s_{1,8}^{[1]}, s_{1,8}^{[2]}, d_{1,1,4}, d_{2,3,4}, d_{1,4,8}^{[1]}, d_{1,4,8}^{[2]}, d_{5,6,7}, d_{5,7,8})$. The transformation is now completed with the updated data sequence which contains $T - 2 = 6$ detail and 2 smooth coefficients.

2.2.3 Some important features of TGUW transformation

Before formulating the TGUW transformation in generality, we describe how it achieves sparse representation of the data. Sometimes, we will be referring to a detail coefficient $d_{p,q,r}^{\cdot}$ as $d_{p,q,r}^{(j,k)}$ or $d^{(j,k)}$, where $j = 1, \ldots, J$ is the scale of the transform (i.e. the consecutive pass through the data) at which $d_{p,q,r}^{\cdot}$ was computed, $k = 1, \ldots, K(j)$ is the location index of $d_{p,q,r}^{\cdot}$ within all scale j coefficients, and $d_{p,q,r}^{\cdot}$ is $d_{p,q,r}^{[1]}$ or $d_{p,q,r}^{[2]}$ or $d_{p,q,r}$, depending on the type of merge.

The TGUW transform eventually converts the input data sequence X of length T into the sequence containing 2 smooth and $T-2$ detail coefficients through $T-2$ orthonormal transforms as follows,

$$
\begin{pmatrix}
s_{1,T}^{[1]} \\
s_{1,T}^{[2]} \\
\left(d^{(j,k)} \right)_{j=1,\ldots,J,k=1,\ldots,K(j)}
\end{pmatrix}
=
\begin{pmatrix}
\psi^{(0,1)} \\
\psi^{(0,2)} \\
\left(\psi^{(j,k)} \right)_{j=1,\ldots,J,k=1,\ldots,K(j)}
\end{pmatrix}
\begin{pmatrix}
X_1 \\
X_2 \\
\vdots \\
X_T
\end{pmatrix}
$$

$$
= \Psi_{T \times T}
\begin{pmatrix}
X_1 \\
X_2 \\
\vdots \\
X_T
\end{pmatrix},
\tag{3}
$$

where Ψ is a data-adaptively chosen orthonormal unbalanced wavelet basis for $\mathbb{R}^T$. The detail coefficients $d^{(j,k)}$ can be regarded as scalar products between X and a particular unbalanced wavelet basis $\psi^{(j,k)}$, where the formal representation is given as $\{d^{(j,k)} = \langle X, \psi^{(j,k)} \rangle, _{j=1,\ldots,J,k=1,\ldots,K(j)} \}$ for detail coefficients and $s_{1,T}^{[1]} = \langle X, \psi^{(0,1)} \rangle, s_{1,T}^{[2]} = \langle X, \psi^{(0,2)} \rangle$ for the two smooth coefficients.

The TGUW transform is nonlinear, but it is also conditionally linear and orthonormal given the order in which the merges are performed. The orthonormality of the unbalanced wavelet basis, $\{\psi^{(j,k)}\}$, implies Parseval's identity:

$$
\sum_{t=1}^{T} X_t^2 = \sum_{j=1}^{J} \sum_{k=1}^{K(j)} (d^{(j,k)})^2 + (s_{1,T}^{[1]})^2 + (s_{1,T}^{[2]})^2.
\tag{4}
$$

Furthermore, the filters $(\psi^{(0,1)}, \psi^{(0,2)})$ corresponding to the two smooth coefficients $s_{1,T}^{[1]}$ and $s_{1,T}^{[2]}$ form an orthonormal basis of the subspace $\{(x_1, x_2, \ldots, x_T) \mid x_1 - x_2 = x_2 - x_3 = \cdots = x_{T-1} - x_T\}$ of $\mathbb{R}^T$; see Section E of the supplementary materials for further details. This implies

$$
\sum_{t=1}^{T} X_t^2 - (s_{1,T}^{[1]})^2 - (s_{1,T}^{[2]})^2 = \sum_{t=1}^{T} (X_t - \hat{X}_t)^2
\tag{5}
$$

where $\hat{X} = s_{1,T}^{[1]} \psi^{(0,1)} + s_{1,T}^{[2]} \psi^{(0,2)}$ is the best linear regression fit to X achieved by minimising the sum of squared errors. This, combined with the Parseval's identity above, implies

$$
\sum_{t=1}^{T} (X_t - \hat{X}_t)^2 = \sum_{j=1}^{J} \sum_{k=1}^{K(j)} (d^{(j,k)})^2.
\tag{6}
$$

By construction, the detail coefficients $|d^{(j,k)}|$ obtained in the initial stages of the TGUW transform tend to be small in magnitude. Then the Parseval's identity in (4)

implies that a large portion of $\sum_{t=1}^{T}(X_t - \hat{X}_t)^2$ is explained by only a few large $|d^{(j,k)}|$'s arising in the later stages of the transform; in this sense, the TGUW transform provides sparsity of signal representation.

2.2.4 TGUW transformation: general algorithm

In this section, we formulate in generality the TGUW transformation illustrated in Sect. 2.2.2 by showing how an adaptive orthonormal unbalanced wavelet basis, Ψ in (3), is constructed. One of the important principles is "tail-greediness" (Fryzlewicz 2018) which enables us to reduce the computational complexity by performing multiple merges over non-overlapping regions in a single pass over the data. More specifically, it allows us to perform up to $\max\{2, \lceil \rho \alpha_j \rceil\}$ merges at each scale j, where α_j is the number of smooth coefficients in the data sequence s and $\rho \in (0, 1)$ (the lower bound of 2 is essential to permit a Type 3 transformation, which consists of two merges).

We now describe the TGUW algorithm.

1. At each scale j, find the set of triplets that are candidates for merging under the "two together" rule and compute the corresponding detail coefficients. Regardless of the type of merge, a detail coefficient $d_{p,q,r}^{\cdot}$ is, in general, obtained as

$$d_{p,q,r}^{\cdot} = a s_{p:r}^1 + b s_{p:r}^2 + c s_{p:r}^3, \tag{7}$$

where $p \leq q < r$, $s_{p:r}^k$ is the kth smooth coefficient of the subvector $s_{p:r}$ with a length of $r - p + 1$ and the constants a, b, c are the elements of the detail filter $\boldsymbol{h} = (a, b, c)^{\top}$. We note that (a, b, c) also depends on (p, q, r), but this is not reflected in the notation, for simplicity. The detail filter is a weight vector used in computing the weighted sum of a triplet of smooth coefficients which should satisfy the condition that the detail coefficient is zero if and only if the corresponding raw observations over the merged regions have a perfect linear trend. If $(X_p, \ldots, X_r)$ are the raw observations associated with the triplet of the smooth coefficients $(s_{p:r}^1, s_{p:r}^2, s_{p:r}^3)$ under consideration, then the detail filter $\boldsymbol{h}$ is obtained in such a way as to produce zero detail coefficient only when $(X_p, \ldots, X_r)$ has a perfect linear trend, as the detail coefficient itself represents the extent of non-linearity in the corresponding region of data. This implies that the smaller the size of the detail coefficient, the closer the alignment of the corresponding data section with linearity.

2. Summarise all $d_{p,q,r}^{\cdot}$ constructed in step 1 to a (equal length or shorter) sequence of $d_{p,q,r}$ by finding a summary detail coefficient $d_{p,q,r} = \max(|d_{p,q,r}^{[1]}|, |d_{p,q,r}^{[2]}|)$ for any pair of detail coefficients constructed by Type 3 merges.

3. Sort the size of the summarised detail coefficients $|d_{p,q,r}|$ obtained in step 2 in non-decreasing order.

4. Extract the (non-summarised) detail coefficient(s) $|d_{p,q,r}^{\cdot}|$ corresponding to the smallest (summarised) detail coefficient $|d_{p,q,r}|$ e.g. both $|d_{p,q,r}^{[1]}|$ and $|d_{p,q,r}^{[2]}|$ should be extracted only if $d_{p,q,r} = \max(|d_{p,q,r}^{[1]}|, |d_{p,q,r}^{[2]}|)$. Repeat the extraction until $\max\{2, \lceil \rho \alpha_j \rceil\}$ (or all possible, whichever is the smaller number) detail

coefficients have been obtained, as long as the region of the data corresponding to each detail coefficient extracted does not overlap with the regions corresponding to the detail coefficients already drawn.

5. For each $|d_{p,q,r}|$ extracted in step 4, merge the corresponding smooth coefficients by updating the corresponding triplet in s through the orthonormal transform as follows,

$$
\begin{pmatrix} s_{p,r}^{[1]} \\ s_{p,r}^{[2]} \\ d_{p,q,r} \end{pmatrix} = \begin{pmatrix} \boldsymbol{\ell}_1^\top \\ \boldsymbol{\ell}_2^\top \\ \boldsymbol{h}^\top \end{pmatrix} \begin{pmatrix} s_{p:r}^1 \\ s_{p:r}^2 \\ s_{p:r}^3 \end{pmatrix} = \Lambda \begin{pmatrix} s_{p:r}^1 \\ s_{p:r}^2 \\ s_{p:r}^3 \end{pmatrix}.
\tag{8}
$$

The key step is finding the 3×3 orthonormal matrix, Λ, which is composed of one detail and two low-pass filter vectors in its rows. Firstly the detail filter $\boldsymbol{h}^\top$ is determined to satisfy the condition mentioned in step 1, and then the two low-pass filters $(\boldsymbol{\ell}_1^\top, \boldsymbol{\ell}_2^\top)$ are obtained by satisfying the orthonormality of Λ. There is no uniqueness in the choice of $(\boldsymbol{\ell}_1^\top, \boldsymbol{\ell}_2^\top)$, but this has no effect on the transformation itself. The details of this mechanism can be found in Section E of the supplementary materials.

6. Go to step 1 and repeat at new scale $j = j + 1$ as long as we have at least three smooth coefficients in the updated data sequence s.

More specifically, when Type 1 merge is performed in step 1 (i.e. $\boldsymbol{s}_{p:r}$ in (7) consists of three initial smoothing coefficients, which implies $r = p + 2$), the corresponding detail filter $\boldsymbol{h}$ is obtained as a unit normal vector to the plane $\{(x, y, z)|x - 2y + z = 0\}$, thus the detail coefficient d presents the projection of three initial smoothing coefficients to the unit normal vector. In the same manner, due to the orthonormality of Λ in (8), the two low-pass filters $(\boldsymbol{\ell}_1^\top, \boldsymbol{\ell}_2^\top)$ form an arbitrary orthonormal basis of the plane $\{(x, y, z)|x - 2y + z = 0\}$. In practice, the detail filter $\boldsymbol{h}$ in Step 1 is obtained by updating so-called weight vectors of constancy and linearity in which the initial inputs have a form of $(1, 1, \ldots, 1)^\top$ and $(1, 2, \ldots, T)^\top$, respectively. The details can be found in Section F of the supplementary materials.

We now comment briefly on the computational complexity of the TGUW transform. Assume that α_j smooth coefficients are available in the data sequence s at scale j and we allow the algorithm to merge up to $\lceil \rho \alpha_j \rceil$ many triplets (unless their corresponding data regions overlap) where $\rho \in (0, 1)$ is a constant. This gives us at most $(1 - \rho)^j T$ smooth coefficients remaining in s after j scales. Solving for $(1 - \rho)^j T \leq 2$ gives the largest number of scales J as $\lceil \log(T)/\log((1 - \rho)^{-1}) + \log(2)/\log(1 - \rho) \rceil$, at which point the TGUW transform terminates with two smooth coefficients remaining. Considering that the most expensive step at each scale is sorting which takes $O(T \log(T))$ operations, the computational complexity of the TGUW transformation is $O(T \log^2(T))$.
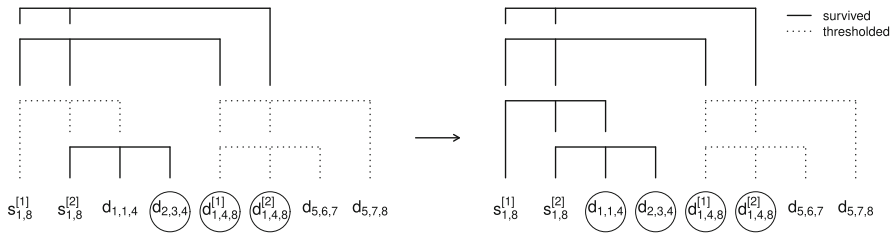
(a) survived detail coefficients before and after applying the "connected" rule
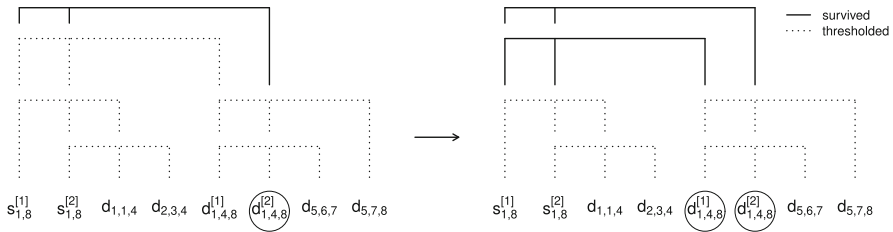
(b) survived detail coefficients before and after applying the "two together" rule

Fig. 4 The tree of mergings in the example of Sect. 2.2.2. Left-hand trees show the examples of tree obtained from initial hard thresholding, the right-hand trees come from processing the respective left-hand ones by applying **a** the "connected" rule and **b** the "two together" rule, respectively, described in Sect. 2.3. The circled detail coefficients are the surviving ones

2.3 Thresholding

Because at each stage, the TGUW transform constructs the smallest possible detail coefficients, but it is at the same time orthonormal and so preserves the l_2 energy of the input data, the variability (= deviation from linearity) of the signal tends to be mainly encoded in only a few detail coefficients computed at the later stages of the transform. The resulting sparsity of representation of the input data in the domain of TGUW coefficients justifies thresholding as a way of deciding the significance of each detail coefficient (which measures the local deviation from linearity).

We propose to threshold the TGUW detail coefficients under two important rules, which should simultaneously be satisfied; we refer to these as the "connected" rule and the "two together" rule. The "two together" rule in thresholding is similar to the one in the TGUW transformation except it targets pairs of detail rather than smooth coefficients, and only applies to pairs of detail coefficients arising from Type 3 merges. Figure 4b shows one such pair in the example of Sect. 2.2.2, $(d_{1,4,8}^{[1]}, d_{1,4,8}^{[2]})$, and the "two together" rule means that both such detail coefficients should be kept if at least one survives the initial thresholding. This is a natural requirement as a pair of Type 3 detail coefficients effectively corresponds to a single merge of two adjacent regions.

The "connected" rule which prunes the branches of the TGUW detail coefficients if and only if the detail coefficient itself and all of its children coefficients fall below a certain threshold in absolute value. This is illustrated in Fig. 4a along with the example

of Sect. 2.2.2; if both $d_{2,3,4}$ and $(d^{[1]}_{1,4,8}, d^{[2]}_{1,4,8})$ were to survive the initial thresholding, the "connected" rule would mean we also had to keep $d_{1,1,4}$, which is the child of $(d^{[1]}_{1,4,8}, d^{[1]}_{1,4,8})$ and the parent of $d_{2,3,4}$ in the TGUW coefficient tree.

Through the thresholding, we wish to estimate the underlying signal f in (1) by estimating $\mu^{(j,k)} = \langle f, \psi^{(j,k)} \rangle$ where $\psi^{(j,k)}$ is an orthonormal unbalanced wavelet basis constructed in the TGUW transform from the data. Throughout the entire thresholding procedure, the "connected" and "two together" rules are applied in this order. We firstly threshold and apply the "connected" rule, which gives us $\hat{\mu}^{(j,k)}_0$, the initial estimator of $\mu^{(j,k)}$, as

$$\hat{\mu}^{(j,k)}_0 = d^{(j,k)}_{p,q,r} \cdot \mathbb{I} \left\{ \exists (j', k') \in \mathcal{C}_{j,k} \quad \left| d^{(j',k')}_{p',q',r'} \right| > \lambda \right\}, \tag{9}$$

where $\mathbb{I}$ is an indicator function and

$$\mathcal{C}_{j,k} = \{ (j', k'), j' = 1, \ldots, j, k' = 1, \ldots, K(j') : d^{(j',k')}_{p',q',r'}$$
$$\text{is such that } [p', r'] \subseteq [p, r] \}. \tag{10}$$

Now the "two together" rule is applied to the initial estimators $\hat{\mu}^{(j,k)}_0$ to obtain the final estimators $\hat{\mu}^{(j,k)}$. We firstly note that two detail coefficients, $d^{(j,k)}_{p,q,r}$ and $d^{(j',k+1)}_{p',q',r'}$ are called "paired" when they are formed by Type 3 mergings and when $(j, p, q, r) = (j', p', q', r')$. The "two together" rule is formulated as below,

$$\hat{\mu}^{(j,k)} = \begin{cases} \hat{\mu}^{(j,k)}_0, & \text{if } d^{(j,k)}_{p,q,r} \text{ is not paired,} \\ \hat{\mu}^{(j,k)}_0, & \text{if } d^{(j,k)}_{p,q,r} \text{ is paired with } d^{(j,k')}_{p,q,r} \text{ and both } \hat{\mu}^{(j,k)}_0 \text{ and } \hat{\mu}^{(j,k')}_0 \text{ are zero or nonzero,} \\ d^{(j,k)}, & \text{if } d^{(j,k)}_{p,q,r} \text{ is paired with } d^{(j,k')}_{p,q,r} \text{ and } \hat{\mu}^{(j,k')}_0 \neq 0 \text{ and } \hat{\mu}^{(j,k)}_0 = 0. \end{cases} \tag{11}$$

It is important to note that the application of the two rules ensures that $\tilde{f}$ is a piecewise-linear function composed of best linear fits (in the least-squares sense) for each interval of linearity. As an aside, we note that the number of survived detail coefficients does not necessarily equal the number of change-points in $\tilde{f}$ as a pair of detail coefficients arising from a Type 3 merge are associated with a single change-point.

2.4 Inverse TGUW transformation

The estimator $\tilde{f}$ of the true signal f in (1) is obtained by inverting (= transposing) the orthonormal transformations in (8) in reverse order to that in which they were originally performed. This inverse TGUW transformation is referred to as TGUW^{-1}, and thus

$$\tilde{f} = \text{TGUW}^{-1} \{ \hat{\mu}^{(j,k)}, j = 1, \ldots, J, k = 1, \ldots, K(j) \| s^{[1]}_{1,T}, s^{[2]}_{1,T} \}, \tag{12}$$

where $\|$ denotes vector concatenation.

2.5 Post processing for consistency of change-point detection

As will be formalised in Theorem 1 of Sect. 3, the piecewise-linear estimator $\tilde{f}$ in (12) possibly overestimates the number of change-points. To remove the spurious estimated change-points and to achieve the consistency of the number and the locations of the estimated change-points, we adopt the post-processing framework of Fryzlewicz (2018). Lin et al. (2017) show that we can usually post-process l_2-consistent estimators in this way as a fast enough l_2 error rate implies that each true change-point has an estimator nearby. The post-processing methodology includes two stages, i) execution of three steps, TGUW transform, thresholding and inverse TGUW transform, again to the estimator $\tilde{f}$ in (12) and ii) examination of regions containing only one estimated change-point to check for its significance.

Stage 1. We transform the estimated function $\tilde{f}$ in (12) with change-points $(\tilde{\eta}_1, \tilde{\eta}_2, \ldots, \tilde{\eta}_{\tilde{N}})$ into a new estimator $\tilde{\tilde{f}}$ with corresponding change-points $(\tilde{\tilde{\eta}}_1, \tilde{\tilde{\eta}}_2, \ldots, \tilde{\tilde{\eta}}_{\tilde{\tilde{N}}})$. Using $\tilde{f}$ in (12) as an input data sequence s, we perform the TGUW transform as presented in Sect. 2.2.4, but in a greedy rather than tail-greedy way such that only one detail coefficient $d^{(j,1)}$ is produced at each scale j, and thus $K(j) = 1$ for all j. We repeat to produce detail coefficients until the first detail coefficient such that $|d^{(j,1)}| > \lambda$ is obtained where λ is the parameter used in the thresholding procedure described in Sect. 2.3. Once the condition, $|d^{(j,1)}| > \lambda$, is satisfied, we stop merging, relabel the surviving change-points as $(\tilde{\tilde{\eta}}_1, \tilde{\tilde{\eta}}_2, \ldots, \tilde{\tilde{\eta}}_{\tilde{\tilde{N}}})$ and construct the new estimator $\tilde{\tilde{f}}$ as

$$\tilde{\tilde{f}}_t = \hat{\theta}_{i,1} + \hat{\theta}_{i,2}\, t \quad \text{for} \quad t \in \left[\tilde{\tilde{\eta}}_{i-1} + 1, \tilde{\tilde{\eta}}_i\right], \quad i = 1, \ldots, \tilde{\tilde{N}}, \tag{13}$$

where $\tilde{\tilde{\eta}}_0 = 0, \tilde{\tilde{\eta}}_{\tilde{\tilde{N}}+1} = T$ and $(\hat{\theta}_{i,1}, \hat{\theta}_{i,2})$ are the OLS intercept and slope coefficients, respectively, for the corresponding pairs $\{(t, X_t), \ t \in [\tilde{\tilde{\eta}}_{i-1} + 1, \tilde{\tilde{\eta}}_i]\}$. The exception is when the region under consideration only contains a single data point X_{t_0}, in which case fitting a linear regression is impossible. We then set $\tilde{\tilde{f}}_{t_0} = X_{t_0}$.

Stage 2. From the estimator $\tilde{\tilde{f}}_t$ in Stage 1, we obtain the final estimator $\hat{f}$ by pruning the change-points $(\tilde{\tilde{\eta}}_1, \tilde{\tilde{\eta}}_2, \ldots, \tilde{\tilde{\eta}}_{\tilde{\tilde{N}}})$ in $\tilde{\tilde{f}}_t$. For each $i = 1, \ldots, \tilde{\tilde{N}}$, compute the corresponding detail coefficient d_{p_i,q_i,r_i} as described in (7), where $p_i = \left\lfloor \frac{\tilde{\tilde{\eta}}_{i-1} + \tilde{\tilde{\eta}}_i}{2} \right\rfloor + 1, q_i = \tilde{\tilde{\eta}}_i$ and $r_i = \left\lceil \frac{\tilde{\tilde{\eta}}_i + \tilde{\tilde{\eta}}_{i+1}}{2} \right\rceil$. Now prune by finding the minimiser $i_0 = \arg\min_i |d_{p_i,q_i,r_i}|$ and removing $\tilde{\tilde{\eta}}_{i_0}$ and setting $\tilde{\tilde{N}} := \tilde{\tilde{N}} - 1$ if $|d_{p_{i_0},q_{i_0},r_{i_0}}| \leq \lambda$ where λ is same as in Sect. 2.3. Then relabel the change-points with the subscripts $i = 1, \ldots, \tilde{\tilde{N}}$ under the convention $\tilde{\tilde{\eta}}_0 = 0, \tilde{\tilde{\eta}}_{\tilde{\tilde{N}}+1} = T$. Repeat the pruning while we can find i_0 which satisfies the condition $|d_{p_{i_0},q_{i_0},r_{i_0}}| < \lambda$. Otherwise, stop, denote by $\hat{N}$ the number of detected change-points and by $\hat{\eta}_i$ – the change-points in increasing order for $i = 0, \ldots, \hat{N} + 1$ where $\hat{\eta}_0 = 0$ and $\hat{\eta}_{\hat{N}+1} = T$. The estimated function $\hat{f}$ is obtained by simple linear

regression for each region determined by the final change-points $\hat{\eta}_1, \ldots, \hat{\eta}_{\hat{N}}$ as in (13), with the exception for the case of single data point as described in Stage 1 above.

Through these two stages of post processing, the estimation of the number and the locations of change-points become consistent, and further details can be found in Sect. 3.

3 Theoretical results

We study the l_2 consistency of $\tilde{f}$ and $\tilde{\tilde{f}}$, and the change-point detection consistency of $\hat{f}$, where the estimators are defined in Sect. 2. The l_2 risk of an estimator $\tilde{f}$ is defined as $\left\| \tilde{f} - f \right\|_T^2 = T^{-1} \sum_{i=1}^{T} (\tilde{f}_i - f_i)^2$, where f is the underlying signal as in (1). We firstly investigate the l_2 behaviour of $\tilde{f}$. The proofs of Theorems 1-3 can be found in Appendix 1.

Theorem 1 X_t *follows model* (1) *with* $\sigma = 1$ *and* $\tilde{f}$ *is the estimator in* (12). *If the threshold* $\lambda = C_1\{2\log(T)\}^{1/2}$ *with a constant* $C_1 \geq \sqrt{3}$, *then we have*

$$\mathbb{P}\left(\left\| \tilde{f} - f \right\|_T^2 \leq C_1^2 \frac{1}{T} \log(T) \left\{ 4 + 8N \lceil \log(T)/\log(1-\rho)^{-1} \rceil \right\} \right) \to 1,$$

(14)

as $T \to \infty$ *and the piecewise-linear estimator* $\tilde{f}$ *contains* $\tilde{N} \leq CN\log(T)$ *change-points where* C *is a constant.*

Thus, $\tilde{f}$ is l_2 consistent under the strong sparsity assumption (i.e. if N is finite) or even under the relaxed condition that N has the order of $\log T$. The crucial mechanism of l_2 consistency is the "tail-greediness" which allows up to $K(j) \geq 1$ smooth coefficients to be removed at each scale j. In other words, consistency is generally unachievable if we proceed in a greedy (as opposed to tail-greedy) way, i.e. if we only merge one triplet at each scale of the TGUW transformation.

We now move onto the estimator $\tilde{\tilde{f}}$ obtained in the first stage of post-processing.

Theorem 2 X_t *follows model* (1) *with* $\sigma = 1$ *and* $\tilde{\tilde{f}}$ *is the estimator in* (13). *Let the threshold* λ *be as in Theorem 1. Then we have* $\left\| \tilde{\tilde{f}} - f \right\|_T^2 = O(NT^{-1}\log^2(T))$ *with probability approaching* 1 *as* $T \to \infty$ *and there exist at most two estimated change-points between each pair of true change-points* (η_i, η_{i+1}) *for* $i = 0, \ldots, N$, *where* $\eta_0 = 0$ *and* $\eta_{N+1} = T$. *Therefore* $\tilde{\tilde{N}} \leq 2(N+1)$.

We see that $\tilde{\tilde{f}}$ is l_2 consistent, but inconsistent for the number of change-points. Now we investigate the final estimators, $\hat{f}$ and $\hat{N}$.

Theorem 3 X_t *follows model* (1) *with* $\sigma = 1$ *and* $(\hat{f}, \hat{N})$ *are the estimators obtained in Sect. 2.5. Let the threshold* λ *be as in Theorem 1 and suppose that the number of true change-points, N, has the order of* $\log T$. *Let* $\Delta_T = \min_{i=1,\ldots,N} \left\{ \left(\underline{f}_T^i \right)^{2/3} \cdot \delta_T^i \right\}$ *where* $\underline{f}_T^i = \min\left(|f_{\eta_i+1} - 2f_{\eta_i} + f_{\eta_i-1}|, |f_{\eta_i+2} - 2f_{\eta_i+1} + f_{\eta_i}| \right)$ *and* $\delta_T^i = \min\left(|\eta_i - \right.$

$\eta_{i-1}|, |\eta_{i+1} - \eta_i|)$. *Assume that* $T^{1/3} R_T^{1/3} = o\left(\Delta_T\right)$ *where* $\left\| \tilde{\tilde{f}} - f \right\|_T^2 = O_p(R_T)$ *is as in Theorem 2. Then we have*

$$\mathbb{P} \left(\hat{N} = N, \quad \max_{i=1,\dots,N} \left\{ |\hat{\eta}_i - \eta_i| \cdot \left(\underline{f}_T^i \right)^{2/3} \right\} \leq C T^{1/3} R_T^{1/3} \right) \rightarrow 1, \qquad (15)$$

as $T \to \infty$ *where C is a constant.*

Our theory indicates that when $\min_i \underline{f}_T^i \sim T^{-1}$, the change-point detection rate of the TrendSegment procedure is $O_p(T^{2/3} \log T)$. If the number of true change-points, N, is finite, then the detection accuracy becomes $O_p(T^{2/3}(\log T)^{2/3})$. Comparing it with the rate of $O_p(T^{2/3}(\log T)^{1/3})$ derived by Baranowski et al. (2019) and Anastasiou and Fryzlewicz (2022) and also with the rate of $O_p(T^{2/3})$ derived by Raimondo (1998), our detection accuracy is different by only a logarithmic factor. In the case in which $\min_i \underline{f}_T^i$ is bounded away from zero, the consistent estimation of the number and locations of change-point is achieved by assuming $T^{1/3} R_T^{1/3} = o(\delta_T)$ where $\delta_T = \min_{i=1,\dots,N+1} |\eta_i - \eta_{i-1}|$ and $R_T = NT^{-1} \log^2(T)$. In addition, when there exists a separate data segment containing only one data point, then the two consecutive change-points, η_{k-1} and η_k, linked via $\eta_{k-1} = \eta_k - 1$ under the definition of a change-point in (2) can be detected exactly at their true locations only if the corresponding $\underline{f}_T^i$s satisfy the condition $\min \left(\underline{f}_T^k, \underline{f}_T^{k-1} \right) \gtrsim \log(T)$.

In the supplementary material, the assumptions of the Gaussianity and the independence on ε_t are relaxed and the corresponding Theorems B.1-B.3 are presented in a setting in which the noise is dependent and/or non-Gaussian.

4 Simulation study

4.1 Parameter choice and setting

4.1.1 Post-processing

In what follows, we disable Stages 1 and 2 of post-processing by default: our empirical experience is that Stage 1 rarely makes a difference in practice but comes with an additional computational cost, and Stage 2 occasionally over-prunes change-point estimates.

4.1.2 Choice of the "tail-greediness" parameter

$\rho \in (0, 1)$ is a constant which controls the greediness level of the TGUW transformation in the sense that it decides how many merges are performed in a single pass over the data. A large ρ can reduce the computational cost but it makes the procedure less adaptive, whereas a small ρ gives the opposite effect. Based on our empirical experience, the best performance is stably achieved in the range $\rho \in (0, 0.05]$ and we use $\rho = 0.04$ as a default in the simulation study and data analyses.

4.1.3 Choice of the minimum segment length

We can give a condition on the minimum segment length of the estimated signal returned by the TrendSegment algorithm. If it is set to 1, two consecutive data-points can be detected as change-points. As theoretically shown in the supplementary material, the minimum length of the estimated segment should have an order of $\log(T)$ to achieve estimation consistency in the case of dependent and/or non-Gaussian errors. To avoid too short segments, and to cover non iid Gaussian noise, we set the minimum segment length to $C \log(T)$ and use the default $C = 0.9$ in the remainder of the paper, otherwise we are not able to detect those short segments in (M6). This constraint can be adjusted by users in the R package `trendsegmentR`.

4.1.4 Continuity at change-points

As described in Sect. 2, the TrendSegment algorithm works by detecting change-points first (in thresholding) and then estimating the best linear fit (in the least-squares sense) for each segment (in the inverse TGUW transform). These procedures normally ensure discontinuity at change-points, however our R package `trendsegmentR` has an option for ensuring continuous change-points by approximating f using the linear spline fit with knots at detected change-points.

4.1.5 Choice of threshold λ

Motivated by Theorem 1, we consider the simplest naïve threshold of the form

$$\lambda^{\text{Naïve}} = C\sigma\sqrt{2 \log T}, \tag{16}$$

where σ can be estimated in different ways depending on the type of noise. Under iid Gaussian noise, we can estimate σ using the Median Absolute Deviation (MAD) estimator (Hampel 1974) defined as $\hat{\sigma} = \text{Median}(|X_1 - 2X_2 + X_3|, \ldots, |X_{T-2} - 2X_{T-1} + X_T|)/(\Phi^{-1}(3/4)\sqrt{6})$ where Φ^{-1} is the quantile function of the Gaussian distribution. We found that under iid Gaussian noise $C = 1.3$ empirically leads to the best performance over a sequence of C, where the details and the relevant results for non-Gaussian and/or dependent errors can be found in Section C of the supplementary material. For completeness, we now present an algorithm for a threshold that works well in all circumstances. When the noise is not generated from iid Gaussian, it is reasonable to assume that the threshold is affected by the serial dependence structure and/or the extent of heavy-tailedness of noise, which motivates us to use threshold of the form:

$$\lambda^{\text{Robust}} = C\mathcal{I}g(\mathcal{K})\sqrt{2 \log T}, \tag{17}$$

where $\mathcal{I}$ is the long-run standard deviation, $\mathcal{K}$ is kurtosis and g is a function. To estimate the unknown parameters in (17), we follow Algorithm 1.

Algorithm 1 Robust threshold selection

INPUT: X, $\lambda^{\text{Naïve}}$, C, $\eta_{\max}$

1. Pre-estimate the fit, $\hat{f}_t$, via the TrendSegment algorithm with $\eta_{\max}$, where $\eta_{\max}$ is a pre-specified maximum number of estimated change-points.
2. Compute the empirical residuals, $\hat{\varepsilon}_t$, from the pre-fit obtained in 1.
3. From $\hat{\varepsilon}_t$, compute the sample kurtosis ($\hat{\mathcal{K}}$) and the long-run standard deviation estimator ($\hat{\mathcal{I}}$) based on AR(1) model. See comments underneath the algorithm for details of this step.
4. Plug in $\hat{\mathcal{I}}$ and $\hat{\mathcal{K}}$ into the threshold formula in (17) with a pre-specified C.
5. To estimate the function g, find a non-parametric regression fit with $X = \hat{\mathcal{K}}$ and $Y = \frac{\lambda^{\text{Naïve}}}{C\hat{\mathcal{I}}\sqrt{2\log T}}$, where $\lambda^{\text{Naïve}}$ is chosen as the best performing threshold by repeating the simulations with a range of threshold constant C over different types of noise.
6. Obtain the threshold in (17) based on the set of estimators, $(\hat{\mathcal{I}}, \hat{\mathcal{K}}, \hat{g})$.

OUTPUT: The robust threshold λ^{Robust}.

We now describe the details of each step in Algorithm 1.

Pre-estimated fit in Step 1. In (17), the heavy-tailedness and dependent structure of the noise are captured by $\mathcal{K}$ and $\mathcal{I}$, respectively. In practice, estimating $\mathcal{I}$ and $\mathcal{K}$ is challenging as the observation includes change-points in its underlying signal. One of the most straightforward way is pre-estimating the fit $\hat{f}_t$ via TrendSegment algorithm with a parameter $\eta_{\max}$, the maximum number estimated change-points. As long as $\eta_{\max}$ is not too large, some extent of overestimation would be acceptable, and we use $\eta_{\max} = \lceil 0.15T \rceil$ as a default in practice, as it empirically led to the best performance and the simulation results do not vary by much over the range $\eta_{\max} \in [\lceil 0.1T \rceil, \lceil 0.2T \rceil]$. The pre-fitting gives us the estimated noise $\hat{\varepsilon}_t = X_t - \hat{f}_t$, from which we can estimate both $\mathcal{I}$ and $\mathcal{K}$.

Pre-specified constant C *in Step 4.* We set $C = 1.3$ as it empirically led to the best performance for iid Gaussian noise with the naive approach in (16). Thus we hope to have both $\hat{\mathcal{I}}$ and $\hat{g}(\hat{\mathcal{K}})$ close to 1 under iid Gaussian noise, but larger than 1 when the noise has serial dependence and/or heavy-tailedness.

$\mathcal{I}$ *and* $\mathcal{K}$ *in Step 4.* $\mathcal{I}$ and $\mathcal{K}$ capture dependency and heavy-tailedness of noise, respectively. First, kurtosis is estimated from the estimated noise as follows:

$$\hat{\mathcal{K}} = \frac{\sum_{t=1}^{T}(\hat{\varepsilon}_t - \bar{\hat{\varepsilon}})^4}{T\hat{s}_{\hat{\varepsilon}}^4}, \tag{18}$$

where $\bar{\hat{\varepsilon}}$ and $\hat{s}_{\hat{\varepsilon}}$ are sample mean and sample standard deviation of $\hat{\varepsilon}$, respectively. For estimating $\mathcal{I}$, we consider the case when Gaussian noise has dependent structure. Then the dependencies increase the marginal variance of CUSUM statistic and one

way of solving this issue is inflating the threshold by the following factor

$$\mathcal{I} = \sqrt{(1 + \phi)/(1 - \phi)}, \tag{19}$$

where ϕ is the true parameter of a AR(1) process (Fearnhead and Fryzlewicz 2022). We can estimate ϕ by fitting AR(1) model to the estimated noise $\hat{\varepsilon}_t = X_t - \hat{f}_t$, and this gives us the estimated long-run standard deviation $\hat{\mathcal{I}}$. Although in theory the inflation factor in (19) is valid only for Gaussian noise, we use the estimator of (19) as an estimated long-run standard deviation even when the noise has both serial dependence and heavy-tailedness, hoping that the heavy-tailedness is captured reasonably well by $\mathcal{K}$.

Kurtosis function g *in Step 5.* We fit a non-parametric regression as described in step 5 of Algorithm 1 over different models and noise scenarios. We found that $g(\hat{\mathcal{K}})$ has no particular functional form in $\hat{\mathcal{K}}$, and is scattered between 0.9 and 1.6 over all noise scenarios and all simulations models considered in the paper. Therefore, the resulting non-parametric fit $\hat{g}(\hat{\mathcal{K}})$ also has a flat shape over a range of $\hat{\mathcal{K}}$, and we use this in finding the robust threshold in practice. This is due to the condition on the minimum segment length described earlier which helps the method to be robust to spikes.

The detailed procedure of estimating g is presented in Section C.2 of the supplementary material. Also, the simulation results using Algorithm 1 for dependent and/or heavy-tailed noise can be found in Tables C.1 - C.10 in Section C.1 of the supplementary material. The proposed robust threshold selection algorithm can also be applied to iid Gaussian noise without any knowledge on type of noise and the corresponding simulation results are given in Sect. 4.3.

We consider iid Gaussian noise and simulate data from model (1) using 8 signals, (M1) wave1, (M2) wave2, (M3) mix1, (M4) mix2, (M5) mix3, (M6) lin.sgmts, (M7) teeth and (M8) lin, shown in Fig. 5. (M1) is continuous at change-points, while (M2) has discontinuities. (M3) contains both constant and linear segments and is continuous at change-points, whereas (M4) is of the similar type but has a mix of continuous and discontinuous change-points. (M5) has three particularly short segments containing 12, 9 and 6 data points, respectively and (M6) has isolated spike-type short segments containing 6 data points each. (M7) is piecewise-constant, and (M8) is a linear signal without change-points. The signals and R code for all simulations can be downloaded from our GitHub repository (Maeng and Fryzlewicz 2021) and the simulation results under dependent or heavy-tailed errors can be found in Section C of the supplementary materials.

4.2 Competing methods and estimators

We perform the TrendSegment procedure based on the parameter choice in Sect. 4.1 and compare the performance with that of the following competitors: Narrowest-Over-Threshold detection [**NOT**, Baranowski et al. (2019)] implemented in the R package not from CRAN, Isolate-Detect [**ID**, Anastasiou and Fryzlewicz (2022)] available in the R package IDetect, trend filtering [**TF**, Kim et al. (2009)] available from https://
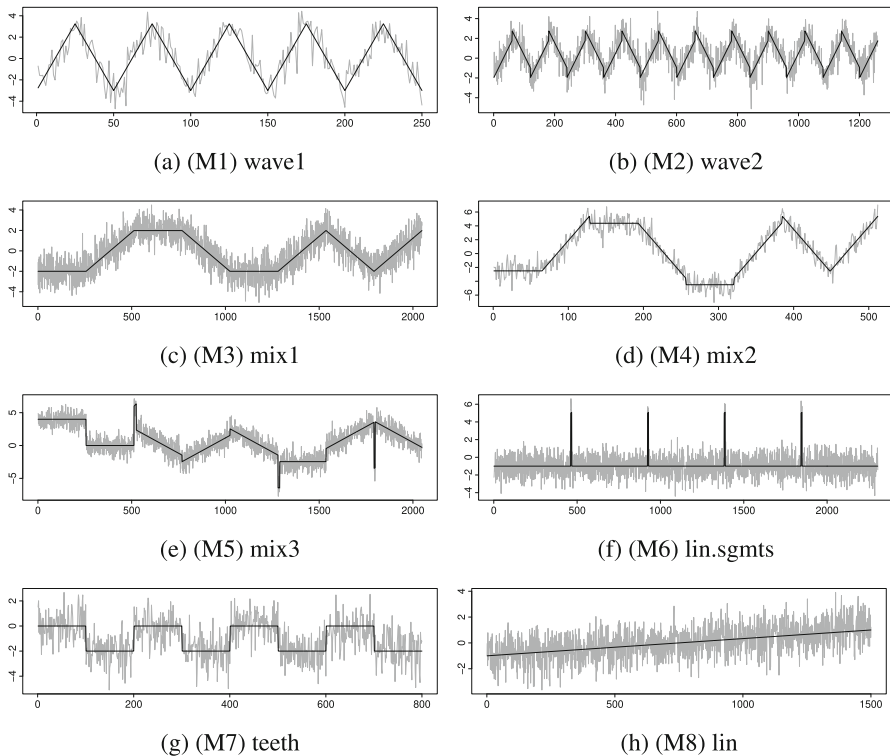
Fig. 5 Examples of data with its underlying signal studied in Sect. 4. **a**–**h** data series X_t (light grey) and true signal f_t (black)

github.com/glmgen/genlasso, Continuous-piecewise-linear Pruned Optimal Partitioning [**CPOP**, Fearnhead et al. (2019)] available from https://www.maths.lancs.ac.uk/~fearnhea/Publications.html and a bottom-up algorithm based on the residual sum of squares (RSS) from a linear fit [**BUP**, Keogh et al. (2004)]. The TrendSegment methodology is implemented in the R package trendsegmentR.

As BUP requires a pre-specified number of change-points (or a well-chosen stopping criterion which can vary depending on the data), we include it in the simulation study (with the stopping criterion optimised for the best performance using the knowledge of the truth) but not in data applications. We do not include the methods of Spiriti et al. (2013) and Bai and Perron (2003) implemented in the R packages freeknotsplines and strucchange as we have found them to be particularly slow. For instance, the minimum segment size in strucchange can be adjusted to be small as long as it is greater than or equal to 3 for detecting linear trend changes. This is suitable for detecting very short segments (e.g in (M6) lin.sgmts), however this setting is accompanied by extremely heavy computation: with this minimum segment size in place, a single signal simulated from (M6) took us over three hours to process on a standard PC.

Out of the competing methods tested, ID, TF and CPOP return continuous change-points, while the estimated signals of Trendsegment and BUP is in principle discontinuous at change-points. For NOT, we use the contrast function for not necessarily continuous piecewise-linear signals. Regarding the tuning parameters for the competing methods, we follow the recommendation of each respective paper or the corresponding R package.

4.3 Results

The summary of the results for all models and methods can be found in Tables 2 and 3. We run 100 simulations and as a measure of accuracy of estimators, we use Monte-Carlo estimates of the Mean Squared Error of the estimated signal defined as MSE=$\mathbb{E}\{(1/T)\sum_{t=1}^{T}(f_t - \hat{f}_t)^2\}$. The empirical distribution of $\hat{N} - N$ is also reported where $\hat{N}$ is the estimated number of change-points and N is the true one. In addition to this, for comparing the accuracy of the locations of the estimated change-points $\hat{\eta}_i$, we show estimates of the scaled Hausdorff distance given by

$$d_H = \frac{1}{T}\mathbb{E}\max\left\{\max_i \min_j |\eta_i - \hat{\eta}_j|, \quad \max_j \min_i |\hat{\eta}_j - \eta_i|\right\} \tag{20}$$

where $i = 0, \ldots, N + 1$ and $j = 0, \ldots, \hat{N} + 1$ with the convention $\eta_0 = \hat{\eta}_0 = 0$, $\eta_{N+1} = \hat{\eta}_{N+1} = T$ and $\hat{\eta}$ and η denote estimated and true locations of the change-points. The smaller the Hausdorff distance, the better the estimation of the change-point locations. For each method, the average computation time in seconds is shown.

We first emphasise that the results with both the naïve and the robust thresholds ($\lambda^{\text{Naïve}}$ in (16) and λ^{Robust} in (17)) are reported for TrendSegment, and the performances are nearly the same except (M7). For simplicity, we call both methods as TrendSegment in the remainder of this section.

The results for (M1) and (M2) are similar. TrendSegment shows comparable performance to NOT, ID and CPOP in terms of the estimation of the number of change-points while it is less attractive in terms of the estimated locations of change-points. TF tends to overestimate the number of change-points throughout all models. When the signal is a mix of constant and linear trends as in (M3) and (M4), TrendSegment, NOT and ID still perform well in terms of the estimation of the number of change-points. CPOP tends to overestimate in (M4) when there exists discontinuity at change-points, however it shows the best performs in terms of localisation (i.e. the smallest mean of Hausdorff distance) as it tends to estimate more than one (and somewhat frequent) change-points at discontinuous change-points. As TrendSegment and NOT deal with the piecewise-linear signals that is not necessarily continuous at change-points, they performs better than others in (M2) and (M4).

We see that TrendSegment has a particular advantage over the other methods especially in (M5) and (M6), when frequent change-points composed of the isolated spike-type short segments of length 6 exist. This is due to the bottom-up nature of TrendSegment which focuses on local features in the early stage of merges and enables TrendSegment to detect those short segments. TrendSegment shows its relative robust-

Table 2 Distribution of $\hat{N} - N$ for models (M1)–(M4) and all methods listed in Sect. 4.1 and 4.2 over 100 simulation runs

Model	Method	$\hat{N} - N$							MSE	$d_H(\times 10^2)$	time
		$\leq$-3	-2	-1	0	1	2	$\geq$3			
(M1)	TS($\lambda^{\text{Naïve}}$)	0	0	2	**98**	0	0	0	0.23	2.96	0.22
	TS(λ^{Robust})	0	0	2	97	1	0	0	0.23	2.97	0.09
	NOT	0	0	0	98	2	0	0	0.19	2.28	0.22
	ID	0	0	0	97	3	0	0	0.14	**1.52**	0.02
	TF	0	0	0	0	0	0	100	0.11	4.50	3.18
	CPOP	0	0	0	97	2	1	0	0.09	**1.09**	0.05
	BUP	100	0	0	0	0	0	0	2.65	10.75	0.35
(M2)	TS($\lambda^{\text{Naïve}}$)	0	0	2	**98**	0	0	0	0.11	1.90	0.50
	TS(λ^{Robust})	0	0	4	96	0	0	0	0.11	1.91	0.24
	NOT	0	0	2	98	0	0	0	0.09	1.56	0.35
	ID	0	0	0	94	6	0	0	0.09	**1.44**	0.23
	TF	0	0	0	0	0	0	100	0.06	2.31	31.34
	CPOP	0	0	0	93	7	0	0	0.06	**1.15**	2.09
	BUP	100	0	0	0	0	0	0	0.75	4.69	2.21
(M3)	TS($\lambda^{\text{Naïve}}$)	0	0	0	99	1	0	0	0.03	3.33	0.61
	TS(λ^{Robust})	0	0	0	**100**	0	0	0	0.03	3.33	0.29
	NOT	0	0	0	**100**	0	0	0	0.02	2.70	0.33
	ID	0	0	0	**100**	0	0	0	0.02	**1.86**	0.02
	TF	0	0	0	0	0	0	100	0.01	5.41	28.89
	CPOP	0	0	0	**100**	0	0	0	0.01	**1.02**	17.38
	BUP	0	0	0	2	22	48	28	0.03	5.46	2.20
(M4)	TS($\lambda^{\text{Naïve}}$)	0	0	0	**100**	0	0	0	0.09	3.24	0.31
	TS(λ^{Robust})	0	0	0	**100**	0	0	0	0.09	3.24	0.09
	NOT	0	0	0	99	1	0	0	0.08	**2.71**	0.23
	ID	0	0	0	97	3	0	0	0.07	**2.04**	0.02
	TF	0	0	0	0	0	0	100	0.05	5.47	8.50
	CPOP	0	0	0	97	3	0	0	0.04	**1.83**	0.39
	BUP	7	64	27	2	0	0	0	0.52	10.66	0.56

Also the average MSE (Mean Squared Error) of the estimated signal $\hat{f}_t$ defined in Sect. 4.3, the average Hausdorff distance d_H given by (20) and the average computational time in seconds using an Intel Core i5 2.9 GHz CPU with 8 GB of RAM, all over 100 simulations. Bold: methods within 10% of the highest empirical frequency of $\hat{N} - N = 0$ or within 10% of the lowest empirical average $d_H(\times 10^2)$. Note that TrendSegment is shortened to TS

Table 3 Distribution of $\hat{N} - N$ for models (M5)-(M8) and all methods listed in Sect. 4.1 and 4.2 over 100 simulation runs

Model	Method	$\hat{N} - N$							MSE	$d_H(\times 10^2)$	time
		$\leq$-3	-2	-1	0	1	2	$\geq$3			
(M5)	TS($\lambda^{\text{Naïve}}$)	0	0	0	**90**	10	0	0	0.03	**1.40**	1.30
	TS(λ^{Robust})	0	0	0	**89**	11	0	0	0.03	**1.41**	0.32
	NOT	0	12	9	75	3	0	1	0.05	**0.73**	0.25
	ID	0	0	0	1	5	25	69	0.29	8.09	0.03
	TF	0	0	0	0	0	0	100	0.14	6.15	28.53
	CPOP	0	0	0	8	27	31	34	0.03	1.42	3.50
	BUP	0	0	0	41	44	13	2	0.10	4.72	2.25
(M6)	TS($\lambda^{\text{Naïve}}$)	0	0	0	**99**	1	0	0	0.01	**0.05**	0.90
	TS(λ^{Robust})	0	3	1	**96**	0	0	0	0.02	0.64	0.34
	NOT	2	13	37	45	2	1	0	0.07	1.74	0.25
	ID	0	0	0	0	0	1	99	0.07	**0.17**	0.04
	TF	0	0	0	0	0	0	100	0.13	9.87	30.72
	CPOP	0	0	0	21	28	40	11	0.03	**0.22**	3.02
	BUP	0	0	0	0	0	0	100	0.12	9.29	2.70
(M7)	TS($\lambda^{\text{Naïve}}$)	0	5	21	40	28	6	0	0.10	7.02	0.31
	TS(λ^{Robust})	1	10	38	31	16	4	0	0.13	8.64	0.13
	NOT	1	1	8	**56**	31	3	0	0.06	**2.62**	0.25
	ID	3	0	16	14	26	13	28	0.32	10.87	0.12
	TF	0	0	0	0	0	0	100	0.10	6.11	23.19
	CPOP	0	0	1	1	3	17	78	0.05	**3.37**	1.19
	BUP	70	25	5	0	0	0	0	0.28	11.89	1.58
(M8)	TS($\lambda^{\text{Naïve}}$)	0	0	0	**100**	0	0	0	0.00	**0.00**	0.43
	TS(λ^{Robust})	0	0	0	**100**	0	0	0	0.00	**0.00**	0.19
	NOT	0	0	0	**100**	0	0	0	0.00	**0.00**	0.17
	ID	0	0	0	**100**	0	0	0	0.00	**0.00**	0.59
	TF	0	0	0	78	5	2	15	0.00	9.08	35.79
	CPOP	0	0	0	**100**	0	0	0	0.00	**0.00**	12.96
	BUP	0	0	0	0	0	0	100	0.01	46.34	2.63

Also the average MSE (Mean Squared Error) of the estimated signal $\hat{f}_t$ defined in Sect. 4.3, the average Hausdorff distance d_H given by (20) and the average computational time in seconds using an Intel Core i5 2.9 GHz CPU with 8 GB of RAM, all over 100 simulations. Bold: methods within 10% of the highest empirical frequency of $\hat{N} - N = 0$ or within 10% of the lowest empirical average $d_H(\times 10^2)$. Note that TrendSegment is shortened to TS

ness in estimating the number and the location of change-points while NOT, ID and CPOP significantly underperform.

For the estimation of the piecewise-constant signal (M7), no methods show good performances and NOT, ID and TrendSegment tend to underestimate the number of change-points while CPOP and TF overestimate. In the case of the no-change-point signal (M8), all methods estimate well except TF and BUP. In summary, TrendSegment is never among the worst methods, is almost always among the best ones, and is particularly attractive for signals containing frequent change-points with short segments. With respect to computation time, NOT and ID are very fast in all cases, TrendSegment is slower than these two but is faster than TF, CPOP and BUP, especially when the length of the time series is larger than 2000.

5 Data applications

5.1 Average January temperatures in Iceland

We analyse a land temperature dataset available from https://www.kaggle.com/berkeleyearth/climate-change-earth-surface-temperature-data, consisting of average temperatures in January recorded in Reykjavik recorded from 1763 to 2013. Figure 6 shows the data; the point corresponding to 1918 appears to be an anomalous point. This is sometimes called point anomaly which can be viewed as a separate data segment containing only one datapoint. Regarding the 1918 observation, Moore and Babij (2017) report that "[t]he winter of 1917/1918 is referred to as the Great Frost Winter in Iceland. It was the coldest winter in the region during the twentieth century. It was remarkable for the presence of sea ice in Reykjavik Harbour as well as for the unusually large number of polar bear sightings in northern Iceland."

Out of the competing methods tested, ID, TF and CPOP are in principle able to classify two consecutive time point as change-points, and therefore they are able to detect separate data segments containing only one data point each. NOT and BUP are not designed to detect two consecutive time point as change-points as their minimum distance between two consecutive change-points is restricted to be at least two. In the TrendSegment algorithm, the minimum segment length can flexibly set by the users as described in Sect. 4. Figures 6a and b show that the change-point estimators depend on the type of threshold we use ($\lambda^{\text{Naïve}}$ or λ^{Robust}) and also vary over conditions on the minimum segment length. Regardless of the minimum segment length, the robust threshold selection tends to detect more change-points than the naïve threshold. When the minimum segment length is set to 1, with both naïve and robust thresholds, TrendSegment commonly identifies change-points in 1917 and 1918, where the temperature in 1918 is fitted as a single point. As shown in Fig. 6d, out of the competing methods, only CPOP detects the temperature in 1918 as an anomalous point. Figures 6b–d show that TrendSegment with λ^{Robust}, NOT and CPOP detect the change of slope in 1974, ID returns an increasing function with no change-points and TF reports 6 points with the most recent one in 1981, but none of them detect the point in 1918 as a separate data segment. When setting the minimum segment length equals to the default
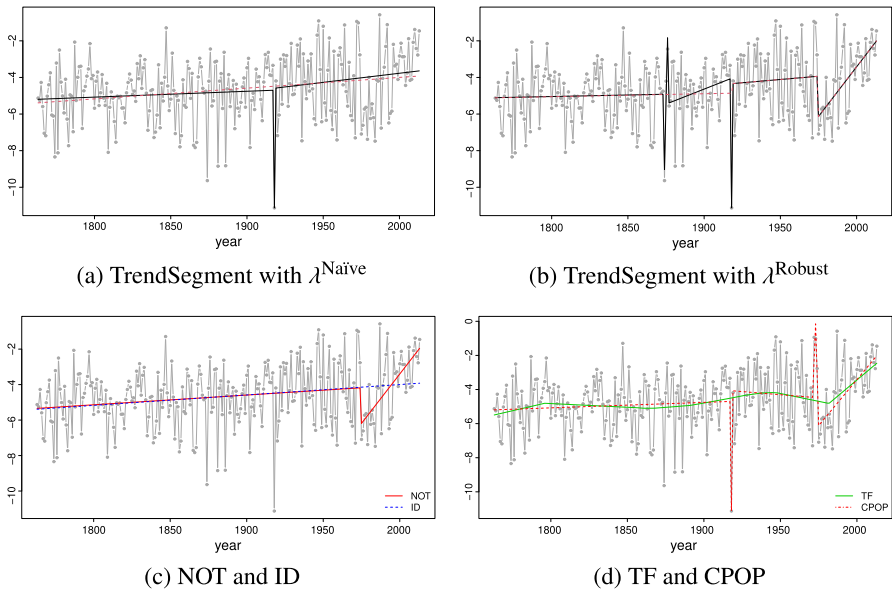
(a) TrendSegment with $\lambda^{\text{Naïve}}$

(b) TrendSegment with λ^{Robust}

(c) NOT and ID

(d) TF and CPOP

Fig. 6 Change-point analysis for January average temperature in Reykjavik from 1763 to 2013 in Sect. 5.1. The data series (grey dots) and estimated signal with change-points returned by **a** TrendSegment using $\lambda^{\text{Naïve}}$ in (16) with minimum segment length equals to 1 (black solid) and equals to $\lfloor 0.9 \log(T) \rfloor$ (red dashed), **b** TrendSegment using λ^{Robust} in (17) with minimum segment length equals to 1 (black solid) and equals to $\lfloor 0.9 \log(T) \rfloor$ (red dashed), **c** NOT and ID, **d** TF and CPOP

($\lfloor 0.9 \log(T) \rfloor$) in TrendSegment with $\lambda^{\text{Naïve}}$ in Fig. 6a, it returns no change-points as ID does. This example illustrates the flexibility of the TrendSegment as it detects not only change-points in linear trend but it can identify a separate data segment at the same time, which the competing methods do not achieve.

5.2 Monthly average sea ice extent of Arctic and Antarctic

We analyse the average sea ice extent of the Arctic and the Antarctic available from https://nsidc.org to estimate the change-points in its trend. As mentioned in Serreze and Meier (2018), sea ice extent is the most common measure for assessing the feature of high-latitude oceans and it is defined as the area covered with an ice concentration of at least 15%. Here we use the average ice extent in February and September as it is known that the Arctic has the maximum ice extent typically in February while the minimum occurs in September and the Antarctic does the opposite.

Serreze and Meier (2018) indicate that the clear decreasing trend of sea ice extent of the Arctic in September is one of the most important indicator of climate change. In contrast to the Arctic, the sea ice extent of the Antarctic has been known to be stable in the sense that it shows a weak increasing trend in the decades preceding 2016 (Comiso et al. 2017; Serreze and Meier 2018). However, Rintoul et al. (2018) warn of a possible collapse of the past stability by citing a significant decline of the sea ice extent in 2016. We now use the most up-to-date records (to 2020) and re-examine the
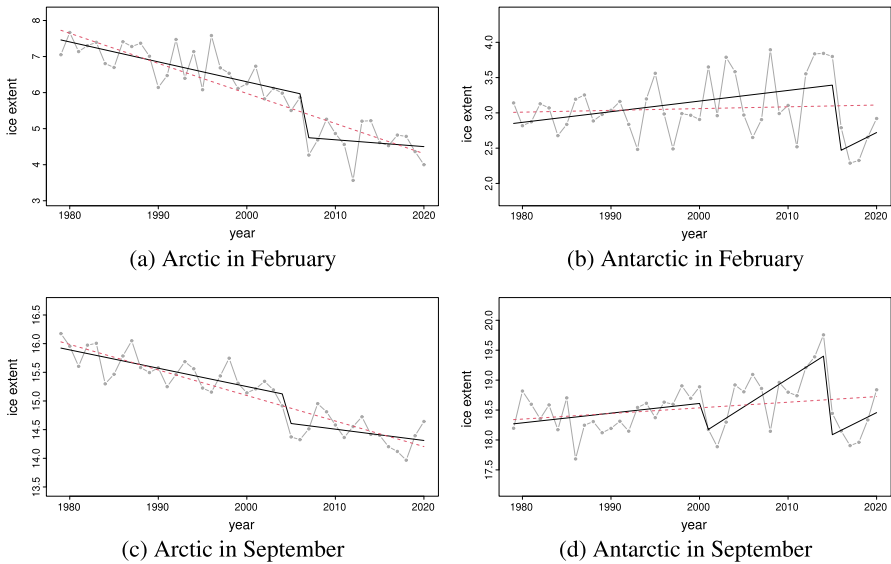
(a) Arctic in February   (b) Antarctic in February

(c) Arctic in September   (d) Antarctic in September

Fig. 7 The TrendSegment estimate of piecewise-linear trend for the monthly average sea ice extent from 1979 to 2020 in Sect. 5.2. **a** the data series (grey dots); the TrendSegment estimate using $\lambda^{\text{Naïve}}$ in (16) (solid black) and TrendSegment estimate using λ^{Robust} (red dashed) for average sea ice extent of the Arctic in February, **b** Antarctic in February, **c** Arctic in September, **d** Antarctic in September

concerns expressed in Rintoul et al. (2018) with the help of our change-point detection methodology.

In this example, the condition on the minimum segment length does not affect the change-point estimation results, thus Fig. 7 shows the results obtained from the default minimum segment length. Also, as shown in Fig. 7, TrendSegment estimate with λ^{Robust} identifies no change-point over all four datasets, thus we focus on giving interpretations for the TrendSegment estimate with $\lambda^{\text{Naïve}}$ in the following.

Figures 7a and c show the well-known decreasing trend of the average sea ice extent in the Arctic both in its winter (February) and summer (September). In Figs. 7a and c, the TrendSegment detects a sudden drop in 2006 and 2004 respectively, which does not clearly differentiate the decreasing speed of ice extent in the Arctic before and after the change-point. As observed in the above-mentioned literature, the sea ice extent of the Antarctic shows a modest increasing trend up until recently (Figs. 7b, d); however, TrendSegment procedure estimates change-point in 2015 which detects a sudden drop during 2015–2017 for the Antarctic summer (February) and estimates two change-points in 2000 and 2014 for the Antarctic winter (September), which is in line with the message of Rintoul et al. (2018). The results of other competing methods can be found in Section D.1 of the supplementary materials.

6 Extension to non-Gaussian and/or dependent noise

Our TrendSegment algorithm can be extended to more realistic settings e.g. when the noise ε_t is possibly dependent and/or non-Gaussian. The extension is performed by slightly altering the estimators $\tilde{f}$, $\tilde{\tilde{f}}$ and $\hat{f}$ and keeping the rate of threshold the same as the one used in Theorems 1-3 (i.e. $\lambda = O((\log T)^{1/2})$) that is established under the iid Gaussian noise. We add an additional step to ensure that only the detail coefficients $d_{p,q,r}^{(j,k)}$ corresponding to a long enough interval $[p, r]$ survive, as this step enables us to apply strong asymptotic normality of $\sum_{t=p}^{r} \varepsilon_t$. Under dependent or non-Gaussian noise, Theorems 1-3 presented in Sect. 3 still hold with a larger rate that is different by only a logarithmic factor, where the corresponding theories and proofs can be found in Section B of the supplementary material.

In Algorithm 1 in Sect. 4.1.5, we propose a robust way of threshold selection that works well in all circumstances including iid Gaussian noise. To demonstrate the robustness of our threshold selection in case the noise has serial dependence and/or heavy-tailedness, additional simulations are performed for five distributions of the noise; (a) $\varepsilon_t \sim$ i.i.d. scaled t_5 distribution with unit-variance, (b) ε_t follows a stationary AR(1) process with $\phi = 0.3$ and Gaussian innovation, (c) the same setting with (b) but with $\phi = 0.6$, (d) ε_t follows a stationary AR(1) process with $\phi = 0.3$ and t_5 innovation and (e) the same setting with (d) but with $\phi = 0.6$, where the results are summarised in Tables C.1-C.10 in Section C.1 of the supplementary material. Lastly, in Section D.2 of the supplementary material, we demonstrate that our TrendSegment algorithm shows a good performance on London air quality data that possibly has some non-negligible autocorrelation.

Appendix A: Technical proofs

The proof of Theorems 1-3 are given below and Lemmas 1 and 2 can be found in Section A of the supplementary materials.

Proof of Theorem 1 Let $\mathcal{S}_j^1$ and $\mathcal{S}_j^0$ as in Lemma 2. From the conditional orthonormality of the unbalanced wavelet transform, on the set A_T defined in Lemma 1, we have

$$\|\tilde{f} - f\|_T^2 = \frac{1}{T} \sum_{j=1}^{J} \sum_{k=1}^{K(j)} \left(d^{(j,k)} \cdot \mathbb{I}\{\exists (j',k') \in \mathcal{C}_{j,k} \ |d^{(j',k')}| > \lambda\} - \mu^{(j,k)} \right)^2 + T^{-1}(s_{1,T}^{[1]}$$

$$- \mu^{(0,1)})^2 + T^{-1}(s_{1,T}^{[2]} - \mu^{(0,2)})^2$$

$$\leq \frac{1}{T} \sum_{j=1}^{J} \left(\sum_{k \in \mathcal{S}_j^0} + \sum_{k \in \mathcal{S}_j^1} \right) \left(d^{(j,k)} \cdot \mathbb{I}\{ \exists (j',k') \in \mathcal{C}_{j,k} \quad |d^{(j',k')}| > \lambda \} \right.$$

$$\left. - \mu^{(j,k)} \right)^2 + 4C_1^2 T^{-1} \log T$$

$$=: I + II + 4C_1^2 T^{-1} \log T, \tag{A1}$$

where $\mu^{(0,1)} = \langle f, \psi^{(0,1)} \rangle$ and $\mu^{(0,2)} = \langle f, \psi^{(0,2)} \rangle$. We note that $\left(s_{1,T}^{[1]} - \mu^{(0,1)} \right)^2 \leq 2C_1^2 \log T$ is simply obtained by combining Lemma 2 and the fact that $s_{1,T}^{[1]} - \mu^{(0,1)} = \langle \boldsymbol{\varepsilon}, \psi^{(0,1)} \rangle$, which can also be applied to obtain $\left(s_{1,T}^{[2]} - \mu^{(0,2)} \right)^2 \leq 2C_1^2 \log T$. By Lemma 2, $\mathbb{I}\{ \exists (j',k') \in \mathcal{C}_{j,k} \quad |d^{(j',k')}| > \lambda \} = 0$ for $k \in \mathcal{S}_j^0$ and also by the fact that $\mu^{(j,k)} = 0$ for $j = 1, \ldots, J, k \in \mathcal{S}_j^0$, we have $I = 0$. For II, we denote $\mathcal{B} = \{ \exists (j',k') \in \mathcal{C}_{j,k} \quad |d^{(j',k')}| > \lambda \}$ and have

$$\begin{aligned} \left(d^{(j,k)} \cdot \mathbb{I}\{\mathcal{B}\} - \mu^{(j,k)} \right)^2 &= \left(d^{(j,k)} \cdot \mathbb{I}\{\mathcal{B}\} - d^{(j,k)} + d^{(j,k)} - \mu^{(j,k)} \right)^2 \\ &\leq (d^{(j,k)})^2 \mathbb{I}(|d^{(j',k')}| \leq \lambda) + 2|d^{(j,k)}| \, \mathbb{I}(|d^{(j',k')}| \leq \lambda) \, |d^{(j,k)} - \mu^{(j,k)}| \\ &\quad + (d^{(j,k)} - \mu^{(j,k)})^2 \\ &\leq \lambda^2 + 2\lambda C_1 \{2 \log T\}^{1/2} + 2C_1^2 \log T. \end{aligned} \tag{A2}$$

Combining with the upper bound of J, $\lceil \log(T)/\log((1-\rho)^{-1}) + \log(2)/\log(1-\rho) \rceil$, and the fact that $|\mathcal{S}_j^1| \leq N$, we have $II \leq 8C_1^2 N T^{-1} \lceil \log(T)/\log((1-\rho)^{-1}) + \log(2)/\log(1-\rho) \rceil \log T$, and therefore $\|\tilde{f} - f\|_T^2 \leq C_1^2 T^{-1} \log(T) \left\{ 4 + 8N \lceil \log(T)/\log((1-\rho)^{-1}) + \log(2)/\log(1-\rho) \rceil \right\}$. As the estimated change-points are obtained through those detail coefficients, thus at each scale, up to N estimated change-points are added. Combining it with the largest scale J whose order is $\log T$, the number of change-points in $\tilde{f}$ returned from the inverse TGUW transformation is up to $CN \log T$ where C is a constant.

Proof of Theorem 2 Let $\tilde{B}$ and $\tilde{\tilde{B}}$ the unbalanced wavelet basis corresponding to $\tilde{f}$ and $\tilde{\tilde{f}}$, respectively. As the change-points in $\tilde{\tilde{f}}$ are a subset of those in $\tilde{f}$, establishing $\tilde{\tilde{f}}$ can be considered as applying the TGUW transform again to $\tilde{f}$ which is just a repetition of procedure done in estimating $\tilde{f}$ in the greediest way. Thus $\tilde{\tilde{B}}$ is classified into two categories, 1) all basis vectors $\psi^{(j,k)} \in \tilde{B}$ such that $\psi^{(j,k)}$ is not associated with the change-points in $\tilde{f}$ and $|\langle X, \psi^{(j,k)} \rangle| = |d^{(j,k)}| < \lambda$ and 2) all vectors $\psi^{(j,1)}$ produced in Stage 1 of post-processing.

We now investigate how many scales are used for this particular transform. First, the detail coefficients $d^{(j,k)}$ corresponding to the basis vectors $\psi^{(j,k)} \in \tilde{B}$ live on no more than $J = O(\log T)$ scales and we have $|\mathcal{S}_j^1| \leq N$ by the argument used in the proof of Theorem 1. In addition, the vectors $\psi^{(j,1)}$ in the second category correspond to different change-points in $\tilde{f}$ and there exist at most $\tilde{N} = O(N \log T)$ change-points in $\tilde{f}$ which we examine one at once (i.e. $|\mathcal{S}_j^1| \leq 1$), thus at most $\tilde{N}$ scales are required

for $d^{(j,1)}$. Combining the results of two categories, the equivalent of quantity II in the proof of Theorem 1 for $\tilde{\tilde{f}}$ is bounded by $II \leq C_3 N T^{-1} \log^2 T$ and this completes the proof of the l_2 result, $\| \tilde{\tilde{f}} - f \|_T^2 = O(N T^{-1} \log^2(T))$ where C_3 is a positive constant large enough.

Finally, we show that there exist at most two change-points in $\tilde{\tilde{f}}$ between true change-points $(\eta_\ell, \eta_{\ell+1})$ for $\ell = 0, \ldots, N$ where $\eta_0 = 0$ and $\eta_{N+1} = T$. Consider the case where three change-point for instance $(\tilde{\tilde{\eta}}_l, \tilde{\tilde{\eta}}_{l+1}, \tilde{\tilde{\eta}}_{l+2})$ lie between a pair of true change-point, $(\eta_\ell, \eta_{\ell+1})$. In this case, by Lemma 2, the maximum magnitude of two detail coefficients computed from the adjacent intervals, $[\tilde{\tilde{\eta}}_l + 1, \tilde{\tilde{\eta}}_{l+1}]$ and $[\tilde{\tilde{\eta}}_{l+1} + 1, \tilde{\tilde{\eta}}_{l+2}]$, is less than λ and $\tilde{\tilde{\eta}}_{l+1}$ would be get removed from the set of estimated change-points. This satisfies $\tilde{\tilde{N}} \leq 2(N+1)$.

Proof of Theorem 3 From the assumptions of Theorem 3, the followings hold.

- Given any $\epsilon > 0$ and $C > 0$, for some T_1 and all $T > T_1$, it holds that $\mathbb{P}\left(\| \tilde{\tilde{f}} - f \|_T^2 > \frac{C^3}{4} R_T \right) \leq \epsilon$ where $\tilde{\tilde{f}}$ is the estimated signal specified in Theorem 2.
- For some T_2, and all $T > T_2$, it holds that $C^{1/3} T^{1/3} R_T^{1/3} (\underline{f}_T^\ell)^{-2/3} < \delta_T^\ell$ for all $\ell = 1, \ldots, N$.

Following the argument used in the proof of Theorem 19 in Lin et al. (2016), we take $T \geq T^*$ where $T^* = \max\{T_1, T_2\}$ and let $r_{\ell,T} = \lfloor C^{1/3} T^{1/3} R_T^{1/3} (\underline{f}_T^\ell)^{-2/3} \rfloor$ for $\ell = 1, \ldots, N$. Suppose that there exist at least one η_ℓ whose closest estimated change-point is not within the distance of $r_{\ell,T}$. Then there are no estimated change-points in $\tilde{\tilde{f}}$ within $r_{\ell,T}$ of η_ℓ which means that $\tilde{\tilde{f}}_j$ displays a linear trend over the entire segment $j \in \{\eta_\ell - r_{\ell,T}, \ldots, \eta_\ell + r_{\ell,T}\}$. Hence

$$\frac{1}{T} \sum_{j=\eta_\ell - r_{\ell,T}}^{\eta_\ell + r_{\ell,T}} (\tilde{\tilde{f}}_j - f_j)^2 \geq \frac{13 r_{\ell,T}^3}{24 T} (\underline{f}_T^\ell)^2 > \frac{C^3}{4} R_T. \tag{A3}$$

The first inequality holds by Lemma 20 of Lin et al. (2016), and the second one holds by the definition of $r_{\ell,T}$. Assuming that at least one η_ℓ does not have an estimated change-point within the distance of $r_{\ell,T}$ implies that the estimation error exceeds $\frac{C^3}{4} R_T$ which is a contradiction as it is an event that we know occurs with probability at most ϵ. Therefore, there must exist at least one estimated change-point within the distance of $r_{\ell,T}$ from each true change point η_ℓ.

Throughout Stage 2 of post-processing, $\tilde{\tilde{\eta}}_{\ell_0}$ is either the closest estimated change-point of any η_ℓ or not. If $\tilde{\tilde{\eta}}_{\ell_0}$ is not the closest estimated change-point to the nearest true change-point on either its left or its right, by the construction of detail coefficients in Stage 2 of post-processing, Lemma 2 guarantees that the corresponding detail coefficient has the magnitude less than λ and $\tilde{\tilde{\eta}}_{\ell_0}$ gets removed. Suppose $\tilde{\tilde{\eta}}_{\ell_0}$ is the closest estimated change-point of a true change-point η_ℓ and it is within the distance of $C T^{1/3} R_T^{1/3} (\underline{f}_T^\ell)^{-2/3}$ from η_ℓ. If the corresponding detail coefficient has the magnitude less than λ and $\tilde{\tilde{\eta}}_{\ell_0}$ is removed, there must exist another $\tilde{\tilde{\eta}}_\ell$ within the

distance of $CT^{1/3}R_T^{1/3}(\underline{f}_T^\ell)^{-2/3}$ from η_ℓ. If there are no such $\tilde{\tilde{\eta}}_\ell$, then by the construction of the detail coefficient, the order of magnitude of $\left|d_{p_{\ell_0},q_{\ell_0},r_{\ell_0}}\right|$ would be such that $\left|d_{p_{\ell_0},q_{\ell_0},r_{\ell_0}}\right| > \lambda$ thus $\tilde{\tilde{\eta}}_{\ell_0}$ would not get removed. Therefore, after Stage 2 of post-processing is finished, each true change-point η_ℓ has its unique estimator within the distance of $CT^{1/3}R_T^{1/3}(\underline{f}_T^\ell)^{-2/3}$.

References

Anastasiou A, Fryzlewicz P (2022) Detecting multiple generalized change-points by isolating single ones. Metrika 85:141–174

Bai J, Perron P (1998) Estimating and testing linear models with multiple structural changes. Econometrica 66:47–78

Bai J, Perron P (2003) Computation and analysis of multiple structural change models. J Appl Economet 18:1–22

Baranowski R, Chen Y, Fryzlewicz P (2019) Narrowest-over-threshold detection of multiple change points and change-point-like features. J R Stat Soc 81:649–672

Bardwell L, Fearnhead P et al (2017) Bayesian detection of abnormal segments in multiple time series. Bayesian Anal 12:193–218

Comiso JC, Gersten RA, Stock LV, Turner J, Perez GJ, Cho K (2017) Positive trend in the Antarctic sea ice cover and associated changes in surface temperature. J Clim 30:2251–2267

Fearnhead P, Fryzlewicz P (2022) Detecting a single change-point. arXiv:2210.07066

Fearnhead P, Maidstone R, Letchford A (2019) Detecting changes in slope with an l 0 penalty. J Comput Graph Stat 28:265–275

Fisch ATM, Eckley IA, Fearnhead P (2018) A linear time method for the detection of point and collective anomalies. arXiv:1806.01947

Fryzlewicz P (2018) Tail-greedy bottom-up data decompositions and fast mulitple change-point detection. Ann Stat 46:3390–3421

Hampel FR (1974) The influence curve and its role in robust estimation. J Am Stat Assoc 69:383–393

Jamali S, Jönsson P, Eklundh L, Ardö J, Seaquist J (2015) Detecting changes in vegetation trends using time series segmentation. Remote Sens Environ 156:182–195

James NA, Kejariwal A, Matteson DS (2016) Leveraging cloud data to mitigate user experience from 'breaking bad'. In: 2016 IEEE International Conference on big data (Big Data), pp. 3499–3508. IEEE

Jeng XJ, Cai TT, Li H (2012) Simultaneous discovery of rare and common segment variants. Biometrika 100:157–172

Keogh E, Chu S, Hart D, Pazzani M (2004) Segmenting time series: a survey and novel approach. In: Data mining in time series databases, pp 1–21. World Scientific

Kim S-J, Koh K, Boyd S, Gorinevsky D (2009) ℓ_1 trend filtering. SIAM Rev 51:339–360

Lin K, Sharpnack J, Rinaldo A, Tibshirani RJ (2016) Approximate recovery in changepoint problems, from ℓ_2 estimation error rates. arXiv:1606.06746

Lin K, Sharpnack JL, Rinaldo A, Tibshirani RJ (2017) A sharp error analysis for the fused lasso, with application to approximate changepoint screening. Adv Neural Inf Process Syst 30:6884–6893

Maeng H, Fryzlewicz P (2021) Detecting linear trend changes and point anomalies in data sequences: simulation code. https://github.com/hmaeng/trendsegment

Matteson DS, James NA (2014) A nonparametric approach for multiple change point analysis of multivariate data. J Am Stat Assoc 109:334–345

Matteson DS, James NA, Nicholson WB, Segalini LC (2013) Locally stationary vector processes and adaptive multivariate modeling. In: 2013 IEEE international conference on acoustics, speech and signal processing (ICASSP), pp 8722–8726. IEEE

Moore G, Babij M (2017) Iceland's great frost winter of 1917/1918 and its representation in reanalyses of the twentieth century. Q J R Meteorol Soc 143:508–520

Olshen AB, Venkatraman E, Lucito R, Wigler M (2004) Circular binary segmentation for the analysis of array-based DNW copy number data. Biostatistics 5:557–572

Raimondo M (1998) Minimax estimation of sharp change points. Ann Stat 26:1379–1397

Rintoul S, Chown S, DeConto R, England M, Fricker H, Masson-Delmotte V, Naish T, Siegert M, Xavier J (2018) Choosing the future of antarctica. Nature 558:233–241

Robbins MW, Lund RB, Gallagher CM, Lu Q (2011) Changepoints in the north Atlantic tropical cyclone record. J Am Stat Assoc 106:89–99

Robinson LF, Wager TD, Lindquist MA (2010) Change point estimation in multi-subject FMRI studies. Neuroimage 49:1581–1592

Serreze MC, Meier WN (2018) The Arctic's sea ice cover: trends, variability, predictability, and comparisons to the Antarctic. Annals of the New York Academy of Sciences, New York

Spiriti S, Eubank R, Smith PW, Young D (2013) Knot selection for least-squares and penalized splines. J Stat Comput Simul 83:1020–1036

Tibshirani RJ et al (2014) Adaptive piecewise polynomial estimation via trend filtering. Ann Stat 42:285–323

Yu Y, Chatterjee S, Xu H (2022) Localising change points in piecewise polynomials of general degrees. Electron J Stat 16:1855–1890