Detecting changes in slope with an L_0 penalty

Paul Fearnhead[1,†], Robert Maidstone[1,2] and Adam Letchford[3]

[1]Department of Mathematics and Statistics, Lancaster University

[2]STOR-i Doctoral Training Centre, Lancaster University

[3]Department of Management Science, Lancaster University

[†]Correspondence: p.fearnhead@lancaster.ac.uk

Abstract

Whilst there are many approaches to detecting changes in mean for a univariate time-series, the problem of detecting multiple changes in slope has comparatively been ignored. Part of the reason for this is that detecting changes in slope is much more challenging: simple binary segmentation procedures do not work for this problem, whilst existing dynamic programming methods that work for the change in mean problem cannot be used for detecting changes in slope. We present a novel dynamic programming approach, CPOP, for finding the "best" continuous piecewise-linear fit to data under a criterion that measures fit to data using the residual sum of squares, but penalises complexity based on an L_0 penalty on changes in slope. We prove that detecting changes in this manner can lead to consistent estimation of the number of changepoints, and show empirically that using an L_0 penalty is more reliable at estimating changepoint locations than using an L_1 penalty. Empirically CPOP has good computational properties, and can analyse a time-series with $10,000$

observations and 100 changes in a few minutes. Our method is used to analyse data on the motion of bacteria, and provides better and more parsimonious fits than two competing approaches.

Keywords: Breakpoints, Functional Pruning, Linear Spline Regression, Narrowest-over-threshold, Trend-filtering

1 Introduction

Changepoint detection and modelling is currently one of the most active research areas in statistics due to its importance across a wide range of applications, including: finance (Fryzlewicz, 2014); bioinformatics (Futschik et al., 2014); environmental science (Killick et al., 2010); target tracking (Nemeth et al., 2014) and fMRI (Aston and Kirch, 2012). It appears to be increasingly important for analysing large scale data streams, as a flexible way of modelling heterogeniety in these streams. This paper focusses on detecting changes in slope: we consider data whose mean varies over time, and we model this mean as a continuous piecewise-linear function of time.

To motivate this work consider the challenge of analysing data of the angular position and velocity of a bacterium, see Figure 1. The movement of the bacterium is driven by the bacterial flagella, a slender thread-like structure that enables it to swim. The movement is circular, and thus the position of the bacterium at any time point can be summarised by its angular position. The data we show comes from Sowa et al. (2005), and consists of a time-series of the amount of rotation that the bacterium has done from its initial position.

The interest in such data is in deriving understanding about the bacterial flagella motor. In particular the angular motion is characterised by stationary periods interspersed by periods of roughly constant angular velocity. The movement tends to be,

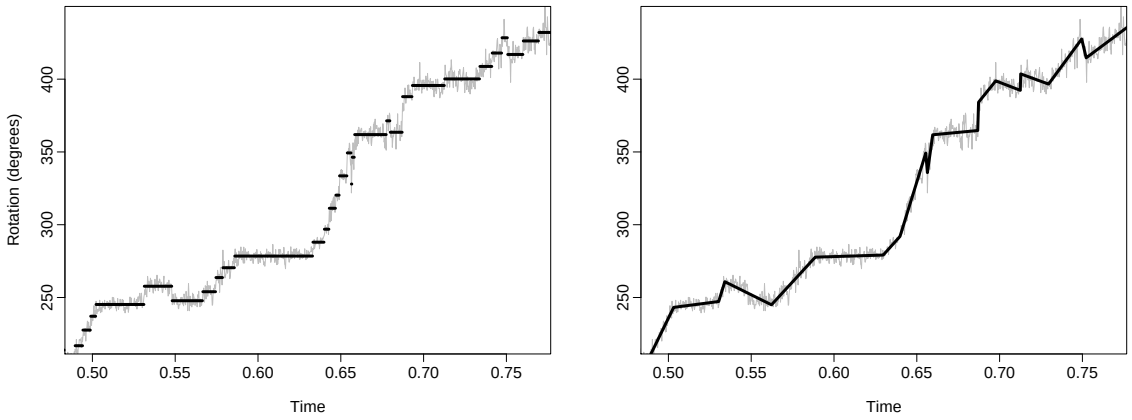

Figure 1: Part of a time-series of angular position of a bacterium (Sowa et al., 2005); best fitting piecewise constant mean (left-hand plot) and continuous piecewise-linear mean (right-hand plot). The former fits data from periods of rotation with a number of short stationary regimes.

though is not exclusively, in one direction.

Sowa et al. (2005) analyse this data using a changepoint model, where the mean is piecewise constant. An example fit from such a model is shown in 1(a). This model is not a natural model given the underlying physics of the application, and this can be seen in how it tries to fit periods of rotation by a number of short stationary regimes. A more natural model is one where we segment the data into periods of constant angular velocity. Such a model is equivalent to fitting a continuous piecewise-linear mean function to the data, with the slope of this function in each segment corresponding to the angular velocity in the segment. Such a fit is shown in 1(b).

Whilst detecting changes in slope seems to be a similar statistical problem to detecting changes in mean, it is fundamentally more challenging. For example, binary segmentation approaches (Scott and Knott, 1974; Fryzlewicz, 2014), which are the most popular generic approach to detecting multiple changepoints, do not work for detecting changes in slope (as shown by Baranowski et al., 2016). Binary segmen-

tation iteratively applies a method for detecting a single changepoint. For change in slope problems one can show that initial estimates of changepoint locations can be midway between actual changepoint locations; binary segmentation is unable to recover from such errors.

A standard approach to detecting changes in mean is to attempt to find the "best" piecewise-constant mean function, where best is defined based on its fit to the data penalised by a measure of complexity of the mean function (Yao, 1988; Lavielle and Moulines, 2000). The most common measure of fit is through the residual sum of squares, and the most natural measure of complexity is the number of changepoints. The latter corresponds to imposing an L_0 penalty on the change in the mean. Dynamic programming can be used to efficiently find the best segmentation of the data under such a criterion for the change in mean problem (Jackson et al., 2005; Killick et al., 2012; Maidstone et al., 2017).

Our statistical approach is to use the same framework to detect changes in slope. We aim to find the best continuous piecewise-linear mean function, where best is defined in terms of the residual sum of squares plus a penalty that depends on the number of changepoints. We present asymptotic results that estimating changepoints in this manner can give consistent estimates of the number of changepoints and can accurately estimate their location.

However using this criteria introduces computational challenges, as standard algorithms cannot be directly applied to minimise our criteria. The reason for this is that the assumption of continuity introduces dependencies in the parameters associated with each segment, and these in turn violate the conditional independence structure that existing dynamic programming algorithms use. Detecting changes in slope under this criterion lies within a class of NP-hard problems (Weinmann and Storath, 2015). It is not clear to us whether our specific problem is NP-hard, but, as far as we are aware, no polynomial-time algorithm has yet been found. Despite this, we present

a dynamic programming algorithm that does find the best segmentation under this criterion, and has practicable computational cost – of the order of minutes when analysing $10,000$ data points with of the order of 100 changepoints.

There has been earlier work on detecting changes in slope using the same or similar statistical criteria. These include Tomé and Miranda (2004) who use an exhaustive search to find the best segmentation – an approach that is only feasible for very small data sets, with perhaps at most 100 to 200 data points. Alternatively, approximate solutions to the true optimal segmentation are found (Horner and Beauchamp, 1996; Goldberg et al., 2014). As we show, our novel dynamic programming approach is guaranteed to find the best segmentation under our criterion, and is still computationally feasible for large data sets.

2 A Penalised Cost Approach to Detecting Changes in Slope

We assume that we have data, $\mathbf{y} = (y_1, \ldots, y_n)$, ordered by time. We will use the notation that, for $t \geq s$, the set of observations from time s to time t is $\mathbf{y}_{s:t} = (y_s, \ldots, y_t)$. If there are m changepoints in the data, this will correspond to the data being split into $m+1$ distinct segments. We let the location of the jth changepoint be τ_j for $j = 1, \ldots, m$, and set $\tau_0 = 0$ and $\tau_{m+1} = n$. The jth segment will consist of data points $y_{\tau_{j-1}+1}, \ldots, y_{\tau_j}$. We let $\boldsymbol{\tau} = (\tau_0, \ldots, \tau_{m+1})$ be the set of ordered changepoints.

We consider the case of fitting a continuous piecewise-linear function to the data. An example of such a fit is given in the right-hand plot of Figure 1. For such a problem, changepoints will correspond to points in time where the slope of the function changes. There are a variety of ways of parameterising the linear function within each segment. Due to the continuity constraint that we wish to enforce it is helpful to parameterise

this linear function by its value at the start and its value at the end of the segment. Our continuity constraint then requires the value for the end of one segment to be equal to the value at the start of the next segment. For the changepoint τ_i we will denote this common value by ϕ_{τ_i}. A continuous piecewise linear function is then defined by the set of changepoints, and these values of the linear function at the changes, ϕ_{τ_i} for $i = 0, \ldots, m+1$. We will simplify notation by letting $\boldsymbol{\phi} = (\phi_{\tau_0}, \ldots, \phi_{\tau_{m+1}})$. In situations where we refer to a subset of this vector we will use the notation $\boldsymbol{\phi}_{j:k} = (\phi_{\tau_j}, \ldots, \phi_{\tau_k})$ for $0 \leq j \leq k \leq m+1$.

Under this parameterisation, we model the data as, for $i = 0, \ldots, m$,

$$Y_t = \phi_{\tau_i} + \tfrac{\phi_{\tau_{i+1}} - \phi_{\tau_i}}{\tau_{i+1} - \tau_i}(t - \tau_i) + Z_t, \quad \text{for } t = \tau_i + 1, \ldots, \tau_{i+1}, \tag{1}$$

where Z_t, for $t = 1, \ldots, n$, are independent, zero-mean, random variables with common variance σ^2.

We infer the set of changepoints with a penalised cost approach, using a squared-error loss function to measure fit to the data. That is, we minimise over m, $\boldsymbol{\tau}$, and $\boldsymbol{\phi}$,

$$\sum_{i=0}^{m} \left[\frac{1}{\sigma^2} \sum_{t=\tau_i+1}^{\tau_{i+1}} \left(y_t - \phi_{\tau_i} - \frac{\phi_{\tau_{i+1}} - \phi_{\tau_i}}{\tau_{i+1} - \tau_i}(t - \tau_i) \right)^2 + h(\tau_{i+1} - \tau_i) \right] + \beta m, \tag{2}$$

for some suitable choice of penalty constant $\beta > 0$ and segment-length penalty function $h(\cdot)$. These penalties are needed to avoid over-fitting of the data. Perhaps the most common choice of penalty is BIC (Schwarz, 1978), where $\beta = 2\log(n)$ and $h(s) = 0$ for all segment lengths s. However, it has been shown that allowing the penalty to depend on segment length can improve the accuracy of penalised cost approaches, and such penalties have been suggested (Zhang and Siegmund, 2007; Davis et al., 2006). The above cost function assumes knowledge of the noise variance, σ^2. In practice this is not known and needs to be estimated, for example using the Median Absolute Deviation estimator (Hampel, 1974); see for example Fryzlewicz (2014).

We can simplify (2) through introducing segment costs. Define the segment cost for fitting the mean of the data $\mathbf{y}_{s+1:t}$ with a linear function that starts at ϕ at time s

and ends at ψ at time t as

$$\mathcal{C}(\mathbf{y}_{s+1:t}, \phi, \psi) = \frac{1}{\sigma^2} \sum_{j=s+1}^{t} \left(y_j - \phi - \frac{\psi - \phi}{t - s}(j - s) \right)^2 .$$

We estimate the number and location of the changepoints, and the underlying continuous piecewise-linear function, through solving the following minimisation problem:

$$\min_{\boldsymbol{\tau}, m, \boldsymbol{\phi}} \left\{ \sum_{i=0}^{m} \left[\mathcal{C}(\mathbf{y}_{\tau_i+1:\tau_{i+1}}, \phi_{\tau_i}, \phi_{\tau_{i+1}}) + h(\tau_{i+1} - \tau_i) \right] + \beta(m+1) \right\}. \tag{3}$$

2.1 Asymptotic Properties of the Penalised Cost Approach

We now consider the asymptotic properties of estimating changepoints by minimising (3). For this we will assume data is generated from the model (1) with $Z_1, Z_2, \ldots,$ being independent and identically distributed Gaussian random variables. Without loss of generality, we will assume their variance is 1.

The properties of our estimates will depend on the choice of both penalties, $h(\cdot)$ and β. To obtain consistency we will need the latter to depend on the number of data points, n, and thus in this section denote its value by β_n. We will further assume that $h(\cdot) = \gamma \log t$ for some constant γ. This covers the common choices of how the penalty depends on segment length (e.g. Zhang and Siegmund, 2007; Davis et al., 2006).

Theorem 2.1 *Fix the true number of changepoints, and denote this as m. For a given n, suppose Y_t is defined by (1) with $Z_1, \ldots, Z_n$ being independent identically distributed standard Gaussian random variables. Let $\delta_n = \min_{i=1,\ldots,m+1}(\tau_i - \tau_{i-1})$ be the minimum segment length, let*

$$\Delta_n^i = \left| \left(\frac{\phi_i - \phi_{i-1}}{\tau_i - \tau_{i-1}} \right) - \left(\frac{\phi_{i+1} - \phi_i}{\tau_{i+1} - \tau_i} \right) \right|,$$

be the change in slope at changepoint i, and let $\Delta_n = \min_i \Delta_n^i$ be the smallest change in slope. Assume that $\delta_n \to \infty$ and $\delta_n^3 \Delta_n^2 / \log n \to \infty$ as $n \to \infty$. Let $\hat{m}_n$ be the

number of changepoints estimated by minimising (3) with $h(t) = \gamma \log t$ *and* β *replaced by* β_n*, and let* $\hat{\tau}_1, \ldots, \hat{\tau}_{\hat{m}_n}$ *be the corresponding estimates of the changepoint locations. There exists constants* C_1, C_2 *such that if* $\beta_n > C_1 \log n$ *and* β_n *is also* $o(\Delta_n^2 \delta_n^3)$ *then as* $n \to \infty$*,*

$$\Pr\left(\hat{m}_n = m, \max_{i=1,\ldots,m} \left\{|\hat{\tau}_i - \tau_i| \left(\Delta_n^i\right)^{2/3}\right\} \leq C_2 (\log n)^{1/3}\right) \to 1. \tag{4}$$

The proof of the theorem is in the Supplementary Material. The result supports the common choice of choosing a penalty, β_n, proportional to $\log n$, but does not specify the constant of proportionality. The argument used in the proof suggests that this constant should increase with the number of true changes – however we believe this is due to the corresponding argument not being tight as it ignores correlation in the fit we will obtain for different, but similar, putative changepoints. The result as stated has strong similarity with those for the Narrowest-over-Threshold procedure of Baranowski et al. (2016) for detecting changes in slope.

The assumption that $Z_{1:n}$ are independent Gaussian random variables is used to bound the tail of the reduction in residual sum of squares that we would obtain by adding a changepoint (see Lemmas B.1 and B.3 in the Supplementary Material). Qualitatively similar tail bounds would be possible with sub-Gaussian noise, or noise with short-range dependence (see Wang and Samworth, 2018, for similar arguments). The impact of such changes would be to change requirements on the constant of proportionality, C_1, of the penalty β_n.

The standard in-fill asymptotic regime, corresponding to sampling data at increasing frequency, would have $\Delta_n = O(1/n)$. In this case the bound on the error of estimates of the changepoint locations is just a logarithmic factor worse than the minimax rate of $n^{2/3}$ (Raimondo, 1998). More generally, the condition that $\delta_n^3 \Delta_n^2 / \log n \to \infty$ means that (4) implies $|\hat{\tau}_i - \tau_i| = o_p(\delta_n)$ for all $i = 1, \ldots, n$: the error in estimating the changepoint locations are asymptotically negligible when compared to the minimum

segment length.

3 Minimising the Penalised Cost

We present a pruned continuous-state dynamic programming approach to calculate the exact solution to (3) efficiently. This approach is much more complicated than other dynamic programming algorithms used in changepoint detection as neighbouring segments share a common parameter: the end-point of the piecewise-linear function for one segment is the start-point for the next segment.

Dynamic programming requires a conditional separability property. We need to be able to choose some information at time s such that, conditional on this information, we can separately minimise the cost related to the data before and after s. For simpler changepoint problems, this information is just the presence of a changepoint at s. For our problem, because neighbouring segments share a parameter, we need to condition on both the location of a changepoint at s and the value of the function at s. Given both these pieces of information we can separately find the best segmentation of the data before s and the best segmentation of the data after s.

3.1 Dynamic Programming Approach

Consider segmenting the data up to time t, $\mathbf{y}_{1:t}$, for $t = 1, \ldots, n$. When segmenting $\mathbf{y}_{1:t}$ with k changepoints, $\tau_1, \ldots, \tau_k$, we use the notation $\tau_0 = 0$ and $\tau_{k+1} = t$. We define the function $f^t(\phi)$ to be the minimum penalised cost for segmenting $\mathbf{y}_{1:t}$ conditional

on the fitted value at time t being ϕ:

$$f^t(\phi) = \min_{\boldsymbol{\tau},k,\boldsymbol{\phi}_{0:k}} \left\{ \sum_{i=0}^{k-1} \left[\mathcal{C}(\mathbf{y}_{\tau_i+1:\tau_{i+1}}, \phi_{\tau_i}, \phi_{\tau_{i+1}}) + h(\tau_{i+1} - \tau_i) \right] + \left[\mathcal{C}(\mathbf{y}_{\tau_k+1:t}, \phi_{\tau_k}, \phi) + h(t - \tau_k) \right] + \beta(k+1) \right\}.$$

Using the initial condition that $f^0(\phi) = 0$, we can construct the following recursion:

$$\begin{aligned} f^t(\phi) &= \min_{\phi',s} \left\{ \min_{\boldsymbol{\tau}_{0:k-1},k,\boldsymbol{\phi}_{0:k-1}} \left\{ \sum_{i=0}^{k-2} \left[\mathcal{C}(\mathbf{y}_{\tau_i+1:\tau_{i+1}}, \phi_{\tau_i}, \phi_{\tau_{i+1}}) + h(\tau_{i+1} - \tau_i) \right] + \right. \right. \\ &\quad \left. \left. + \mathcal{C}(\mathbf{y}_{\tau_{k-1}+1:s}, \phi_{\tau_{k-1}}, \phi') + h(s - \tau_{k-1}) + \beta k \right\} + \mathcal{C}(\mathbf{y}_{s+1:t}, \phi', \phi) + h(t-s) + \beta \right\}, \\ &= \min_{\phi',s} \left\{ f^s(\phi') + \mathcal{C}(y_{s+1:t}, \phi', \phi) + h(t-s) + \beta \right\}. \end{aligned}$$

The idea is that we split the minimisation into first minimising over the time of the most recent changepoint and the fitted value at that changepoint, and then minimising over the earlier changepoints and fitted values. We let s denote the time of the most recent changepoint, and ϕ' the fitted value at s. The inner minimisation is over the number of changepoints, the locations of those changepoints prior to s, and the fitted values at the changepoints prior to s. This inner minimisation gives the minimum penalised cost for segmenting $\mathbf{y}_{1:s}$ conditional on $\phi_s = \phi'$, which is $f^s(\phi')$. The challenge with solving this recursion is that it is in terms of functions of a continuous parameter, ϕ.

To store $f^t(\phi)$ we will write it as the point-wise minimum of a set of cost functions of ϕ, each of which corresponds to a different vector of changepoints, $\boldsymbol{\tau}$. We define each of these functions $f^t_{\boldsymbol{\tau}}(\phi)$ as the minimum cost of segmenting $\mathbf{y}_{1:t}$ with changepoints at $\boldsymbol{\tau} = \tau_1, \ldots, \tau_k$ and fitted value at time t being ϕ:

$$f^t_{\boldsymbol{\tau}}(\phi) = \min_{\boldsymbol{\phi}_{0:k}} \left\{ \sum_{i=0}^{k-1} \left[\mathcal{C}(\mathbf{y}_{\tau_i+1:\tau_{i+1}}, \phi_{\tau_i}, \phi_{\tau_{i+1}}) + h(\tau_{i+1} - \tau_i) \right] + \mathcal{C}(\mathbf{y}_{\tau_k+1:t}, \phi_{\tau_k}, \phi) + h(t - \tau_k) + \beta(k+1) \right\}. \quad (5)$$

Then $f^t(\phi)$ is the point-wise minimum of these functions, $f^t(\phi) = \min_{\boldsymbol{\tau} \in \mathcal{T}_t} f^t_{\boldsymbol{\tau}}(\phi)$, where $\mathcal{T}_t$ is the set of all possible changepoint vectors at time t.

Each of the above functions, $f^t_{\boldsymbol{\tau}}(\phi)$, is a quadratic in ϕ and thus can be represented by a vector of length 3, with the terms in this vector denoting the co-efficients of the quadratic. We can calculate the co-efficients recursively, see Appendix C, and thus can iteratively compute these functions and calculate $f^n(\phi)$.

We calculate the optimal segmentation of $\mathbf{y}_{1:n}$ by minimising $f^n(\phi)$ over ϕ. The value of $\boldsymbol{\tau}$ that achieves the minimum value will be the optimal segmentation. This approach, however, is computationally expensive. To obtain a practicable algorithm we have to use pruning ideas to reduce the number of changepoint vectors, and corresponding functions $f^t_{\boldsymbol{\tau}}(\phi)$, that we need to store. There are two ways in which this can be achieved: functional pruning and inequality based pruning (Rigaill, 2015; Killick et al., 2012; Maidstone et al., 2017). In both cases they are able to remove changepoint vectors whilst still maintaining the guarantee that the resulting algorithm will find the true minimum of the optimisation problem (2).

3.2 Functional Pruning

We can prune candidate changepoint vectors from the minimisation problem if they can be shown to be dominated by other vectors for any given value of ϕ.

Define the set $\overset{*}{\mathcal{T}}_t$ as the set of changepoint vectors that are optimal for some ϕ at time t

$$\overset{*}{\mathcal{T}}_t = \left\{ \boldsymbol{\tau} \in \mathcal{T}_t : f^t(\phi) = f^t_{\boldsymbol{\tau}}(\phi), \text{ for some } \phi \in (-\infty, \infty) \right\}, \tag{6}$$

where $\mathcal{T}_t$ is the set of all possible changepoint vectors at time t. The following theorem shows that if a candidate vector $\boldsymbol{\tau}$ is not in this set at time s then the related candidate vector $(\boldsymbol{\tau}, s)$ is not in the set at time t. Thus at any time s we will need to store only

the functions $f^s_{\boldsymbol{\tau}}(\phi)$ corresponding to segmentations in $\overset{*}{\mathcal{T}}_s$.

Theorem 3.1 *If* $\boldsymbol{\tau} \notin \overset{*}{\mathcal{T}}_s$ *then* $(\boldsymbol{\tau}, s) \notin \overset{*}{\mathcal{T}}_t$ *for all* $t > s$.

Proof: See Appendix D.

The key to an efficient algorithm will be a way of efficiently calculating $\overset{*}{\mathcal{T}}_t$. We can use the above theorem to help us do this. From Theorem 3.1 we can define a set

$$\hat{\mathcal{T}}_t = \left\{ (\boldsymbol{\tau}, s) : s \in \{0, \ldots, t-1\}, \boldsymbol{\tau} \in \overset{*}{\mathcal{T}}_s \right\}, \tag{7}$$

and we will have that $\hat{\mathcal{T}}_t \supseteq \overset{*}{\mathcal{T}}_t$. So assume that we have calculated the sets $\overset{*}{\mathcal{T}}_s$ for $s = 0, \ldots, t-1$. We can calculate $f^t_{\boldsymbol{\tau}}(\phi)$ only for $\boldsymbol{\tau} \in \hat{\mathcal{T}}_t$. When calculating $f^t(\phi)$ we can just minimise over the set of changepoint vectors in $\hat{\mathcal{T}}_t$ rather than the full set. To find $\overset{*}{\mathcal{T}}_t$ we use the fact that ϕ is one-dimensional and perform a line search where we recursively find the quadratic function associated with $\boldsymbol{\tau} \in \hat{\mathcal{T}}_t$ for which $f^t(\phi) = f^t_{\boldsymbol{\tau}}(\phi)$ as we increase ϕ from $-\infty$ to ∞. This method is given in full in Algorithm 2 in the Supplementary Material, and there is a detailed explanation in Appendix E.

3.3 Inequality Based Pruning

A further way pruning can be used to speed up the dynamic programming algorithm is based on the following result.

Theorem 3.2 *Define* $K = 2\beta + h(1) + h(n)$. *If* $h(\cdot)$ *is non-negative, and non-decreasing and if for some* $\boldsymbol{\tau}$,

$$\min_{\phi} f^t_{\boldsymbol{\tau}}(\phi) > \min_{\phi'} \left[f^t(\phi') \right] + K, \tag{8}$$

then at any future time T, $\boldsymbol{\tau}$ *can never be optimal for the data* $\mathbf{y}_{1:T}$.

Proof: See Appendix D.

This result states that for any candidate changepoint vector, if the best cost at time t is worse than the best cost over all changepoint vectors plus K, then the candidate is sub-optimal at all future times as well. Thus we can reduce the size of $\hat{\mathcal{T}}_t$ before the cost functions are updated, discarding candidates from the set if (8) is true. Once discarded, these will remain discarded for all future sets $\hat{\mathcal{T}}_T$ for $T > t$.

Both pruning steps can be used to restrict the set of candidate changepoint vectors that the dynamic program is run over. We call the resulting algorithm CPOP, for Continuous-piecewise-linear Pruned Optimal Partitioning. The pseudocode for the full method with these pruning steps is outlined in Algorithm 1 in the Supplementary Material.

The computational cost of CPOP is studied in detail in Section 4.4.1 of Maidstone (2016). These empirical results suggest the algorithm's computational cost is close to quadratic in n in situations where there is a fixed number of changepoints, and close to linear in n in situations where the number of changepoints increases linearly with n.

4 Statistical Performance of CPOP

We now look empirically at the statistical performance of CPOP, and compare with two other methods for fitting a continuous piecewise-linear mean function to data. All computation was carried out using `R` (R Core Team, 2017). For simplicity we look at minimising our criterion (2) with the BIC penalty, though see Chapter 5 of Maidstone (2016) for results of CPOP when using the modified BIC penalty of Zhang and Siegmund (2007).

The most common, general, approach for detecting changes is to use binary seg-

mentation (Scott and Knott, 1974), but as mentioned in the introduction binary segmentation does not work for this problem: there are examples where even if you observed the underlying mean function without noise, binary segmentation would not correctly identify the changepoints.

To overcome this, Baranowski et al. (2016), present the *narrowest-over-threshold* (NOT) algorithm. The NOT algorithm proceeds by (i) taking a pre-specified number, M, of intervals of data, $\mathbf{y}_{s_i:t_i}$ say; (ii) performing a generalised likelihood ratio test for a change in slope on each $\mathbf{y}_{s_i:t_i}$; (iii) keeping all intervals for which the test-statistic is above some pre-specified threshold; (iv) ordering these intervals, with the shortest interval first and the longest last; (v) running down this list in order, adding changepoints at each of the inferred changepoint locations for an interval providing that interval does not contain any previously inferred changepoints. The idea of the algorithm is that by concentrating on the smallest intervals in (iv), these will be likely to have at most one actual changepoint, and hence the inferred changepoint in step (v) should be close in position to this actual changepoint.

In practice, NOT is run for a continuous range of thresholds in step (iii). This will produce a set of different segmentations of the data. The segmentation that is then chosen is the one that minimises the BIC for a model where the residuals are independent Gaussian with unknown variance σ^2. For a segmentation with m changepoints at locations $\boldsymbol{\tau}$, the BIC corresponds to the minimum, over $\boldsymbol{\phi}$, of

$$n \log \left(\frac{1}{n} \sum_{i=0}^{m} \left[\sum_{t=\tau_i+1}^{\tau_{i+1}} \left(y_t - \frac{\phi_{\tau_{i+1}} - \phi_{\tau_i}}{\tau_{i+1} - \tau_i} (t - \tau_i) \right)^2 \right] \right) + 2m \log n. \tag{9}$$

This is closely related to our criterion (2) with the BIC penalty, except for the assumption of unknown variance, and the fact that this criterion is only minimised over the set of segmentations found by the NOT algorithm. One advantage of this approach is that it avoids the need to have an estimate of σ.

The other approach we compare to is the trend-filtering algorithm (Kim et al., 2009).

Trend-filtering aims to minimise the residual sum of squares of the fitted continuous piecewise-linear mean, but with an L_1 penalty on how the slope changes. One important difference between an L_1 penalty and the L_0 penalty is that the L_1 penalty is the same for multiple consecutive changes in slope of the same sign as it is for one larger change in slope. We believe this means that trend-filtering will tend to over-estimate the number of changepoints.

Trend-filtering requires a choice of penalty, in the same way that we need to choose the penalty β in (2). To mimic the approach of NOT we use a BIC type approach (other approaches to choosing this penalty are considered in Maidstone, 2016, and give qualitatively similar results). This involves running the trend-filtering algorithm for a discrete set of penalty values. For a given penalty value, trend-filtering will output an estimate of the mean at each time point. From this we can infer the changepoint locations as the points where the estimated mean has a change in slope. We evaluate the output from each run of the trend-filtering algorithm using BIC. If the estimated mean is $\hat{\phi}_{1:n}$, and this has m changes in slope, then using the fact that for trend-filtering a segmentation with m changes in slope has an effective degrees of freedom that is $m + 2$ (Tibshirani, 2014), the BIC value is

$$\frac{1}{\sigma^2}\left(\sum_{t=1}^{n}[y_t - \hat{\phi}_t]^2\right) + (m+2)\log(n).$$

Other approaches, including fitting a change in mean to differenced data and ignoring the continuity constraint when detecting changepoints, are considered in Maidstone (2016). However these all perform much worse, across all measures of accuracy, than the three approaches we compare here.

In the comparisons below we implement CPOP for minimising (2) with the BIC penalty. We use the `not` R-package to implement NOT Baranowski et al. (2016), and the code available from `http://stanford.edu/~boyd/l1_tf` to implement trend-filtering. For NOT we set the number of intervals, M in step (i) of the algorithm above, to 10^5. This is larger than recommended in Baranowski et al. (2016), but we

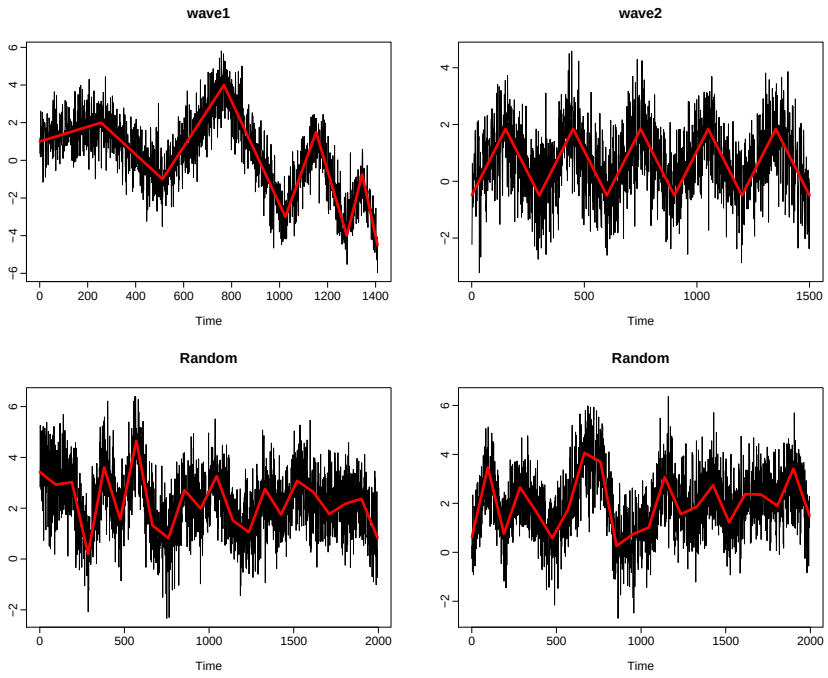

Figure 2: Example data from the three simulation scenarios: `wave1` and `wave2` (top row) have a fixed mean function. For the `Random` scenario (bottom row), the form of the mean is random, and we give two example realisations.

found it gave slightly better results than the default choice of 10^4 intervals. For trend-filtering and CPOP we need an estimate of the variance of the residuals. Within a segment, the variance of the second differences of the data is easily shown to be 6 times the variance of the residuals. Thus we take second differences, and take one-sixth of the median-absolute-deviation estimator of their variance. Of course, being heuristic methods, both NOT and trend-filtering are much faster algorithms than CPOP. Across all the scenarios we considered, trend-filtering and NOT ran in a few seconds, whereas CPOP took between tens of seconds to a few minutes.

The three scenarios that we compared the methods on are shown in Figure 2. The first two of these, `wave1` and `wave2`, are taken from Baranowski et al. (2016). These two scenarios have a fixed mean function. We consider extensions of these two scenarios with higher-frequency observations for `wave1`, where we have twice or four times as

many observations within each segment; and longer time-series for `wave2`, where we have 20 or 40 segments, each of 150 observations, rather than just 10. In the third scenario, which we call `Random`, we simulate the underlying mean for each data set. This setting has segments of equal length, but the value of the mean function at the start/end of each segment is simulated from a Gaussian distribution with variance 4. For this setting we will consider varying both the number of data points and the number of changepoints. In all cases we add independent standard Gaussian noise to the mean.

Following Baranowski et al. (2016), for `wave1` and `wave2` we compare methods using the mean square error (MSE) of the estimates of the mean, and using a scaled Hausdorff distance, d_H, to measure accuracy of the changepoint locations. This distance is defined as

$$d_H = \frac{1}{n_s} \max \left\{ \max_j \min_k |\tau_j - \hat{\tau}_k|, \max_k \min_j |\tau_j - \hat{\tau}_k| \right\},$$

where $\hat{\tau}_k$ are the estimated changepoint locations, τ_j the true changepoint locations, and n_s the length of the largest segment. The idea is that for each true change we find the closest estimated changepoint, and for each estimated changepoint we find the closest true changepoint. We then calculate the distance between each of these pairs of changepoints, and d_H is set to the largest of these distances divided by the length of the longest segment. The smaller d_H the better the estimates of the changespoints, with $d_H = 0$ meaning that all changepoints are detected without error, and no other changepoints are estimated.

First we analyse data from the `wave1` and `wave2` scenarios. We consider different lengths of data with either a fixed number of changepoints (`wave1`) or with the number of changepoints increasing linearly with the number of data points (`wave2`). For both `wave1` and `wave2` there is a substantial change in the slope of the mean at each changepoint. As such, these represent relatively straightforward scenarios for detecting changepoints, and both NOT and CPOP perform well at detecting the

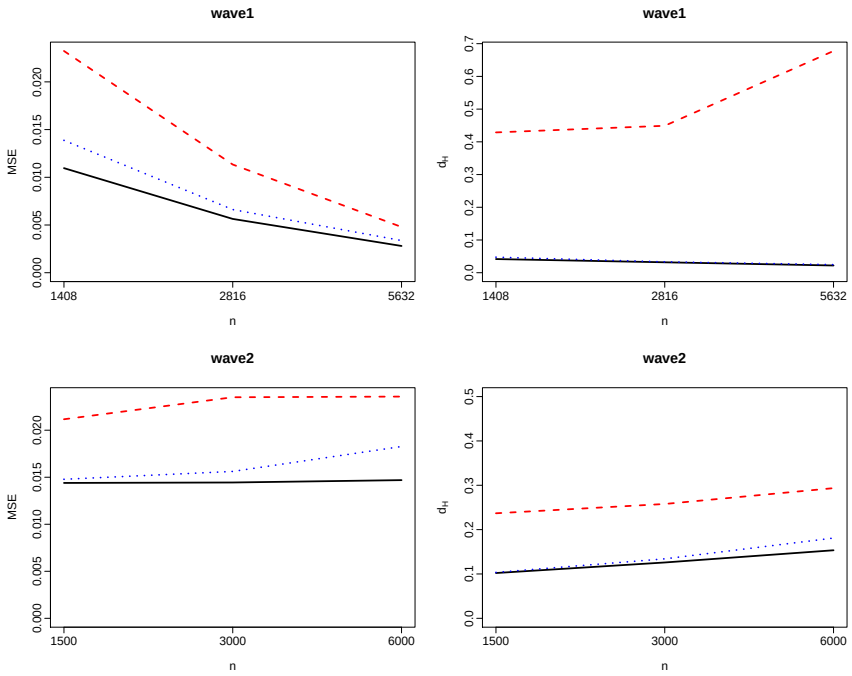

Figure 3: Results for CPOP (black solid line), NOT (blue dotted line) and trend-filtering (red dashed line) for `wave1` (top row) and `wave2` (bottom row). We give results for mean square error of the estimate of the mean (left-hand column) and for the accuracy of the estimates of the changepoint locations, measured via d_H (right-hand column). For `wave1` we had data sets of length $n = 1408$, $n = 2816$ and $n = 5632$,. For `wave2` we had data sets of length $n = 1500$, $n = 3000$ and $n = 6000$. Results are averaged over 100 data sets for each scenario and each value of n.

number of changepoints: NOT correctly identifies the number of changepoints for all 600 simulated data sets, and CPOP correctly identifies the number of changepoints in over 99% of these cases. By comparison trend-filtering substantially over-estimates the number of changepoints in all cases. For `wave1` the average number of changes detected is 16 for $n = 1408$, rising to 29 for $n = 5632$, when the true number of changes is 7. We have similar over-estimation for `wave2`. The reason for this is the use of the L_1 penalty, which is known to lead to algorithms that cannot consistently estimate the number of changepoints for the simpler change in mean setting (Levy-leduc and

Harchaoui, 2008). The L_1 penalty is the same for multiple consecutive changes in slope of the same sign as it is for one large change. As a result trend-filtering tends to introduce multiple changepoints around each actual change.

This over-estimation of the number of changes results in the much larger value of d_H for this method than for NOT and CPOP: see the right-hand plots of Figure 3. Whilst NOT and CPOP perform similarly in terms of accuracy when estimating changepoint location, CPOP is more accurate in terms of estimating the underlying mean: see the MSE results in the left-hand plots of Figure 3. Again both methods perform better than trend-filtering. We believe the reason for this is that trend-filtering shrinks the change in slope towards 0. For signals like `wave1` and `wave2` where all changes in slope are substantial, this causes trend-filtering to under-estimate these changes. This can introduce substantial error at estimating the mean in regions around each changepoint.

We now compare the three methods on the `Random` simulation scenario. We consider data sets of length varying from 1000 to 10000, with either a fixed number of 20 segments or with the segment length fixed to 100. This is a harder scenario, with the change in slope being small in many cases (see the example data sets in the bottom row of Figure 2). As a result there are many changepoints that are hard to detect. In all cases CPOP and NOT underestimate the number of changes, while trend-filtering still over estimates this number. These two different sources of error are masked in the measure d_H, and thus we summarise the accuracy of changepoint detection through true-positive and false-positive proportions. To calculate these we say that an actual change is detected if there is an estimated changepoint within a certain distance of it. The results we show have set this distance to be a fifth of the segment length, though qualitatively similar results are obtained with different choices. We calculate the number of false positives as the number of changepoints detected less the number of true positives. Our results are in terms of the true-positive proportion, which is

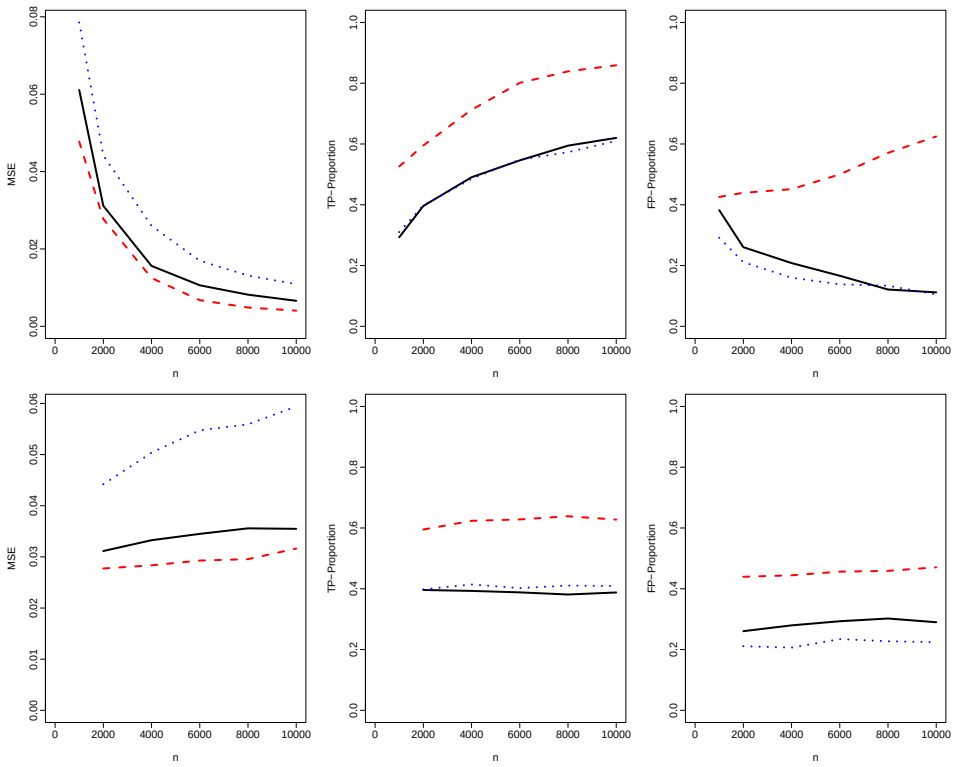

Figure 4: Results for CPOP (black solid line), NOT (blue dotted line) and trend-filtering (red dashed line) for the `Random` scenario with a fixed number of changepoints (top row) and a fixed segment length (bottom row). We give results for mean square error of the estimate of the mean (left-hand column) and for the accuracy of the estimates of the changepoint locations, measured via the proportion of true-positives (middle column) and of false-positives (right-hand column). Results are averaged over 100 data sets for each case and each value of n.

the proportion of actual changepoints detected, and the false-positve proportion, the proportion of detected the changepoints that are false-positive.

Results are shown in Figure 4. These are qualitatively different from the earlier results. For this problem we see that trend-filtering is most accurate in terms of estimating the underlying mean. We believe that trend-filtering is more suited to this scenario as there are a range of values for how much the slope changes at each

changepoint, including many cases where the change is small. Hence the shrinking of the change in slope that trend-filtering induces is actually beneficial. As trend-filtering estimates more changes, it detects a higher proportion of true changepoints, but it has a high false-positive proportion: in all cases over 40% of the changepoints it finds are false-positives. By comparison both NOT and CPOP have lower false positive proportions, and encouragingly, this proportion decreases as the segment length increases (see top right-hand plot in Figure 4). Whilst NOT is marginally better in terms of accuracy of the detected changepoints, CPOP is substantially more accurate in terms of its estimate of the underlying mean.

5 Bacterial Flagella Motor Data

We return to the bacterial flagella motor data we introduced in Section 1 and Figure 1. For more background on these biological systems see Sowa et al. (2005) and Sowa and Berry (2008). Data similar to those we analyse has been collected by Ryu et al. (2000), Chen and Berg (2000) and Sowa et al. (2003) among others. Here we look at how well we can extract the angular motion by fitting change-in-slope models using the CPOP algorithm. The data we analyse comes from Sowa et al. (2005) and is shown in Figure 5. It consists of 11,912 observations.

The aim of our analysis is to fit the underlying angular position. We first compared fitting a continuous piecewise-linear mean to both fitting a piecewise-constant mean and a discontinuous piecewise-linear mean. We fit the latter two by minimising the residual sum of squares plus a penalty times the number of changepoints, using the PELT algorithm (Killick et al., 2012). In all cases we varied the penalty value using the CROPS algorithm (Haynes et al., 2017). Different penalty values lead to optimal segmentations with different numbers of changepoints. For each different segmentation we calculated the actual residual sum of squares of the fit we obtained. A plot

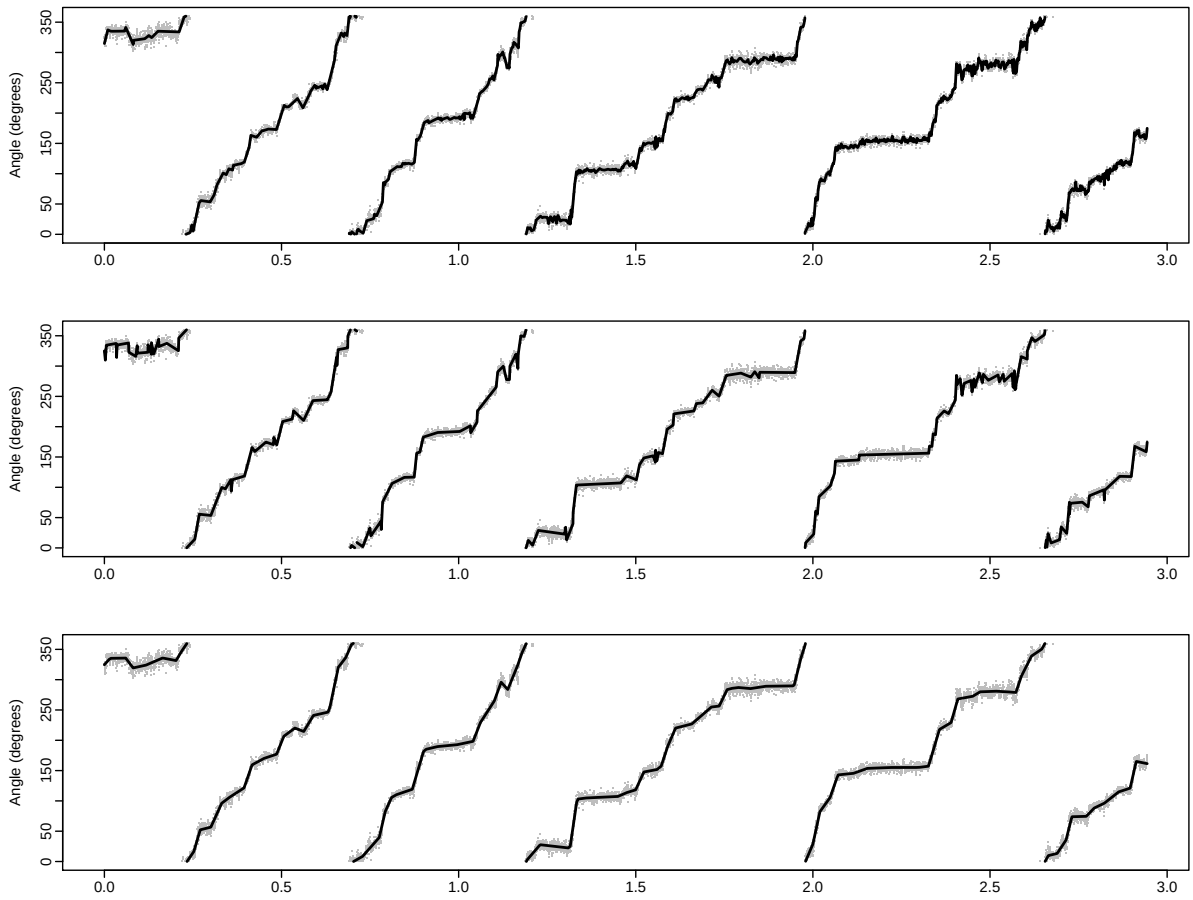

Figure 5: Time-series of angular position (data from Sowa et al., 2005) and example fits obtained by NOT (top); CPOP (middle) and trend-filtering (bottom). The fits by NOT and CPOP are ones which give a similar fit to the data; the NOT fit has 784 changepoints and the fit from CPOP just 182. The fit from trend-filtering has 278 changepoints, though many correspond to very small changes in slope, and a substantially worse fit to the data (see text for more details). For ease of presentation we have plotted the angle of the bacteria, the model we fit assumes continuity of angles of 360 degrees (top of each plot) and 0 degrees (bottom of each plot).

of this against the number of free parameters in the fitted mean is shown in Figure 6. We can see that fitting a continuous piecewise-linear function, which is more natural for this application, leads to a uniformly better fit to the data than the change in

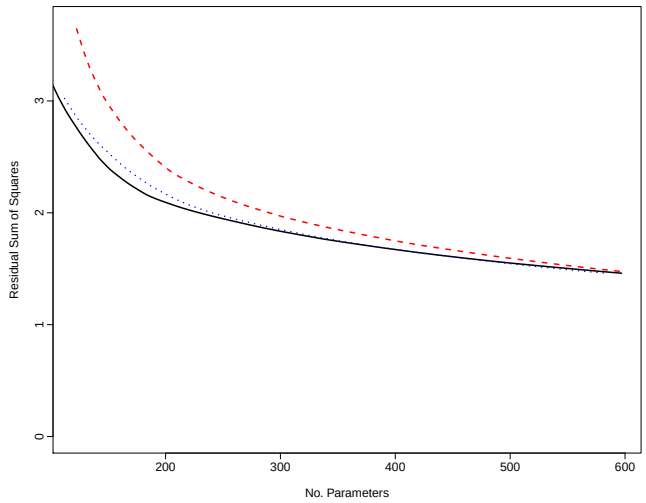

Figure 6: Accuracy of fits of data shown in Figure 5 by a piecewise-constant mean (red dashed line), a continuous piecewise-linear mean (black full line) and a discontinuous piecewise-linear mean (blue dotted line). For each type of line we found the best segmentation, in terms of minimising the residual sum of squares (RSS) of the fit, for a range of the number of changepoints. We plot the RSS against the number of free parameters of the fitted mean function for each case.

mean for any given number of parameters. The assumption of continuity also gives improvements for fitted means with fewer than 400 parameters. While the differences in residual sum of squares looks small, due to the large number of observations, the reduction in log-likelihood, under a model where the residuals are iid Gaussian, is still substantial. For example, for models with fewer than 350 parameters, the best fitting continuous mean has a log-likelihood that is 32.4 units greater than the best fitting discontinuous mean.

We also compared the accuracy of using CPOP to analyse this data to that of using NOT and trend-filtering. A comparison of the fits obtained using NOT, CPOP and trend-filtering are shown in Figure 5. We ran NOT with a total of 10^6 random intervals, and have plotted the segmentation that minimised (9). This segmentation has 794 changepoints, largely because it substantially overfits the latter part of the data. For comparison, an example fit from CPOP is also shown. The segmentation obtained using CPOP has 182 changepoints. Despite fewer changes, it has a smaller residual sum of squares than the segmentation that NOT found: 1.72 as compared to 1.80.

We also ran trend-filtering for a range of penalty values. For all penalty values that gave a reasonable fit to the data, the number of changes in slope was large: with changes at more than half the time-points, but with the majority of changes in slope being small. One example fit is shown in the bottom plot of Figure 6. This has 10,850 changes in slope, though only 278 of these are non-zero if we round the slopes, in degrees, to 3 decimal places. Despite the large number of changepoints, the estimated mean we obtained appears to under-fit the data in a number of places and has a higher residual sum of squares, 2.94, than the fitted mean shown for either CPOP or NOT.

6 Discussion

As with any approach to detecting changepoints, minimising the square error loss of the fit to the data plus an L_0 penalty requires specifying the penalty for adding a changepoint. Whilst using a penalty of $2 \log n$ worked well in simulations, there is currently no theory to support this choice. Furthermore, this choice is only likely to be appropriate for data where the residuals are independent Gaussian with a known variance, or a variance that can accurately be estimated. In practice we would recommend minimising the penalised cost over a range of penalties, using the `CROPS` algorithm Haynes et al. (2017), as we did in Section 5, to investigate the robustness of the segmentations that one obtains by varying the penalty. Furthermore there are approaches to using the output across a range of penalties to help choose an appropriate penalty (Arlot and Massart, 2009). Such an approach would also give robustness to errors in the estimate of the variance of the residuals, as a change in the estimate of the variance is equivalent to keeping the variance fixed and changing the penalty. Alternatively comparing segmentations for different penalty choices on test data, either simulated or real-life, can be used to help make an appropriate choice of penalty (Hocking et al., 2013).

Our dynamic programming approach has the potential to be applied to a much wider range of changepoint problems with dependence across segments. The key requirement is that we can construct a recursion for a set of functions, our $f^t(\phi)$, that are piecewise-quadratic in some univariate parameter ϕ. This requires that we measure fit to the data through the residual sum of squares, that the dependence of the parameters in successive segments is through a univariate quantity ϕ, and that any constraints on parameters in successive segments respect the piecewise-quadratic nature of $f^t(\phi)$. This would cover change in mean or slope under monotonicity constraints (Hocking et al., 2017; Jewell et al., 2018), our change in slope model with an additional L_1 or L_2 penalty on the change in slope, or more general models for the

mean that are piecewise-polynomial and continuous.

The requirement that dependence across segments is through a univariate quantity comes from our functional pruning approach. Such pruning is important for reducing the computational complexity of the algorithm. It is unclear whether functional pruning can be implemented for piecewise-quadratic functions, $f^t(\phi)$, when ϕ is not univariate as the line search approach we take does not generalise beyond the univariate case. Even if not, it may be possible to develop efficient algorithms that implement an approximate version of functional pruning.

Acknowledgements This work was supported by EPSRC grants EP/N031938/1 (StatScale) and EP/H023151/1 (STOR-i). We thank Ashley Nord and Richard Berry for helpful discussions on the analysis of the bacterial flagella motor data; Daniel Grose for help with the CPOP R code; and Rafal Baranowksi, Yining Chen and Piotr Fryzlewicz for advice on using NOT.

7 Supplementary Materials

`CPOP_Supplemantary.pdf` Appendices with proofs and pseudo code for CPOP.

`Rcode.tar.gz` R-code implementing CPOP; R functions used in the simulation study; and data from Section 5.

References

Arlot, S. and Massart, P. (2009). Data-driven calibration of penalties for least-squares regression. *Journal of Machine Learning Research*, 10:245–279.

Aston, J. A. and Kirch, C. (2012). Evaluating stationarity via change-point alter-

natives with applications to fMRI data. *The Annals of Applied Statistics*, pages 1906–1948.

Baranowski, R., Chen, Y., and Fryzlewicz, P. (2016). Narrowest-over-threshold detection of multiple change-points and change-point-like features. *ArXiv:1609.00293*.

Baranowski, R., Chen, Y., and Fryzlewicz, P. (2016). *not: Narrowest-Over-Threshold Change-Point Detection*. R package version 1.0.

Chen, X. and Berg, H. C. (2000). Solvent-isotope and pH effects on flagellar rotation in Escherichia coli. *Biophysical Journal*, 78(5):2280–2284.

Davis, R. A., Lee, T. C. M., and Rodriguez-Yam, G. A. (2006). Structural break estimation for nonstationary time series models. *Journal of the American Statistical Association*, 101:223–239.

Fryzlewicz, P. (2014). Wild binary segmentation for multiple change-point detection. *The Annals of Statistics*, 42(6):2243–2281.

Futschik, A., Hotz, T., Munk, A., and Sieling, H. (2014). Multiscale DNA partitioning: statistical evidence for segments. *Bioinformatics*, 30:2255–2262.

Goldberg, N., Kim, Y., Leyffer, S., and Veselka, T. D. (2014). Adaptively refined dynamic program for linear spline regression. *Computational Optimization and Applications*, 58(3):523–541.

Hampel, F. R. (1974). The influence curve and its role in robust estimation. *Journal of the American Statistical Association*, 69(346):383–393.

Haynes, K., Eckley, I. A., and Fearnhead, P. (2017). Computationally efficient change-point detection for a range of penalties. *Journal of Computational and Graphical Statistics*, 26:134–143.

Hocking, T., Rigaill, G., Vert, J.-P., and Bach, F. (2013). Learning sparse penalties for change-point detection using max margin interval regression. In *International Conference on Machine Learning*, pages 172–180.

Hocking, T. D., Rigaill, G., Fearnhead, P., and Bourque, G. (2017). A log-linear time algorithm for constrained changepoint detection. ArXiv.1703.03352.

Horner, A. and Beauchamp, J. (1996). Piecewise-linear approximation of additive synthesis envelopes : a comparison of various methods. *Computer Music Journal*, 20(2):72–95.

Jackson, B., Scargle, J. D., Barnes, D., Arabhi, S., Alt, A., Gioumousis, P., Gwin, E., Sangtrakulcharoen, P., Tan, L., and Tsai, T. T. (2005). An algorithm for optimal partitioning of data on an interval. *IEEE Signal Processing Letters*, 12:105–108.

Jewell, S., Hocking, T. D., Fearnhead, P., and Witten, D. (2018). Fast Nonconvex Deconvolution of Calcium Imaging Data. ArXiv.1802.07380.

Killick, R., Eckley, I. A., Ewans, K., and Jonathan, P. (2010). Detection of changes in variance of oceanographic time-series using changepoint analysis. *Ocean Engineering*, 37(13):1120–1126.

Killick, R., Fearnhead, P., and Eckley, I. A. (2012). Optimal detection of changepoints with a linear computational cost. *Journal of the American Statistical Association*, 107:1590–1598.

Kim, S. J., Koh, K., Boyd, S., and Gorinevsky, D. (2009). l_1 Trend Filtering. *SIAM Review*, 51(2):339–360.

Lavielle, M. and Moulines, E. (2000). Least-squares estimation of an unknown number of shifts in a time series. *Journal of Time Series Analysis*, 21(1):33–59.

Levy-leduc, C. and Harchaoui, Z. (2008). Catching change-points with lasso. In *Advances in Neural Information Processing Systems*, pages 617–624.

Maidstone, R. (2016). *Efficient analysis of complex changepoint problems*. PhD thesis, Lancaster University. http://eprints.lancs.ac.uk/83055/.

Maidstone, R., Hocking, T., Rigaill, G., and Fearnhead, P. (2017). On optimal multiple changepoint algorithms for large data. *Statistics and Computing*, 27:519–533.

Nemeth, C., Fearnhead, P., and Mihaylova, L. (2014). Sequential Monte Carlo Methods for State and Parameter Estimation in Abruptly Changing Environments. *IEEE Transactions on Signal Processing*, 62(5):1245–1255.

R Core Team (2017). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria.

Raimondo, M. (1998). Minimax estimation of sharp change points. *Annals of Statistics*, pages 1379–1397.

Rigaill, G. (2015). A pruned dynamic programming algorithm to recover the best segmentations with 1 to $K_{\max}$ change-points. *Journal de la Société Française de Statistique*, 156(4):180–205.

Ryu, W. S., Berry, R. M., and Berg, H. C. (2000). Torque-generating units of the flagellar motor of Escherichia coli have a high duty ratio. *Nature*, 403:444–447.

Schwarz, G. (1978). Estimating the dimension of a model. *The Annals of Statistics*, 6(2):461–464.

Scott, A. J. and Knott, M. (1974). A cluster analysis method for grouping means in the analysis of variance. *Biometrics*, 30:507–512.

Sowa, Y. and Berry, R. M. (2008). Bacterial flagellar motor. *Quarterly Reviews of Biophysics*, 41(02):103–132.

Sowa, Y., Hotta, H., Homma, M., and Ishijima, A. (2003). Torque-speed relationship of the Na+-driven flagellar motor of Vibrio alginolyticus. *Journal of Molecular Biology*, 327(5):1043–1051.

Sowa, Y., Rowe, A., Leake, M., Yakushi, T., Homma, M., Ishijima, A., and Berry, R. (2005). Direct observation of steps in rotation of the bacterial flagellar motor. *Nature*, 437(7060):916–9.

Tibshirani, R. J. (2014). Adaptive piecewise polynomial estimation via trend filtering. *The Annals of Statistics*, 42(1):285–323.

Tomé, A. R. and Miranda, P. M. A. (2004). Piecewise linear fitting and trend changing points of climate parameters. *Geophysical Research Letters*, 31(2).

Wang, T. and Samworth, R. J. (2018). High dimensional change point estimation via sparse projection. *Journal of the Royal Statistical Society: Series B*, 80(1):57–83.

Weinmann, A. and Storath, M. (2015). Iterative Potts and Blake–Zisserman minimization for the recovery of functions with discontinuities from indirect measurements. *Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, 471(2176).

Yao, Y.-C. (1988). Estimating the number of change-points via Schwarz' criterion. *Statistics & Probability Letters*, 6(3):181–189.

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

Supplementary Material for Detecting changes in slope with an L_0 penalty

A Proof of Theorem 2.1

Throughout this section m will denote the true number of changepoints. When we consider possible segmentations with a general number of changepoints, we will tend to let d denote the number of changepoints. For data $\mathbf{Y}_{1:n}$ denote the penalised cost of segmenting the data with d changepoints $\hat{\tau}_{1:d}$ by

$$Q(\mathbf{Y}_{1:n}; \hat{\tau}_{1:d}) = \min_{\boldsymbol{\phi}} \left[\sum_{i=0}^{d} \left\{ \mathcal{C}(\mathbf{Y}_{\hat{\tau}_i+1:\hat{\tau}_{i+1}}, \phi_{\hat{\tau}_i}, \phi_{\hat{\tau}_{i+1}}) + h(\hat{\tau}_{i+1} - \hat{\tau}_i) \right\} + \beta_n(d+1) \right]. \quad (10)$$

Further, denote the unpenalised cost by

$$Q_0(\mathbf{Y}_{1:n}; \hat{\tau}_{1:d}) = \min_{\boldsymbol{\phi}} \left\{ \sum_{i=0}^{d} \mathcal{C}(\mathbf{Y}_{\hat{\tau}_i+1:\hat{\tau}_{i+1}}, \phi_{\hat{\tau}_i}, \phi_{\hat{\tau}_{i+1}}) \right\}. \quad (11)$$

We will allow the second argument of both of these functions to be an unordered vector of changepoints, in which case the penalised, or unpenalised, cost is calculated in the obvious way: we remove any duplicate changepoints, order the changepoints and use either (10) or (11) for the ordered changepoints. We also allow the vector of changepoints to include times outside the time-interval for the data – in which case those changepoints are ignored. We write $Q_0(\mathbf{Y}_{1:n})$ for the unpenalised cost if we fit a model with no changepoints.

We base our proof on related proofs for consistency of the number and location of changepoints for change in mean (e.g. Yao, 1988). The extra complication comes from the cost associated with a given segment depending on the location of the other changepoints. To overcome this issue we will use the property of our model that if we add two changepoints at consecutive time-points then the costs associated with segmenting the data before and the data after the pair of changepoints can be

calculated independently of each others. So given a set of d_1 changepoints prior to t, $\hat{\boldsymbol{\tau}}_{1:d_1}$ and a set of d_2 changepoints after $t+1$, $\hat{\boldsymbol{\tau}}_{(d_1+1):(d_1+d_2)}$, then

$$Q_0(\boldsymbol{Y}_{1:n}; \hat{\boldsymbol{\tau}}_{1:(d_1+d_2)}, t, t+1) = Q_0(\boldsymbol{Y}_{1:t}, \hat{\boldsymbol{\tau}}_{1:d_1}) + Q_0(\boldsymbol{Y}_{(t+1):n}, \hat{\boldsymbol{\tau}}_{(d_1+1):(d_1+d_2)}). \tag{12}$$

This can be shown by a simple reparameterisation between the change in slope model fitted for the left-hand side of the equation and the two change in slope models fitted on the right-hand side. As adding changepoints can only lead to a reduction in the unpenalised cost, this gives the following way of bounding the residual sum of squares associated with a given segmentation, which we repeatedly use. For any $s = 1, \ldots, n$,

$$Q_0(\boldsymbol{Y}_{1:n}; \hat{\boldsymbol{\tau}}_{1:d}) - \sum_{t=1}^{n} Z_t^2 \geq \left\{ Q_0(\boldsymbol{Y}_{1:s}; \hat{\boldsymbol{\tau}}_{1:d}) - \sum_{t=1}^{s} Z_t^2 \right\} + \left\{ Q_0(\boldsymbol{Y}_{(s+1):n}; \hat{\boldsymbol{\tau}}_{1:d}) - \sum_{t=s+1}^{n} Z_t^2 \right\}, \tag{13}$$

where, as defined above, we interpret $Q_0(\boldsymbol{Y}_{1:s}, \hat{\boldsymbol{\tau}}_{1:d})$, say, as the unpenalised cost for segmenting $\boldsymbol{Y}_{1:s}$ using just the subset of the changepoints $\hat{\boldsymbol{\tau}}_{1:d}$ that lie between time 1 and time $s-1$.

We define three events which depend on $Y_{1:n}$. The first of these, which we call E_n^1, is the event that, for suitable constants $\alpha > 0$ and $\alpha' > 0$,

$$\max_{i=0,\ldots,m} \left[\max_{d, \hat{\boldsymbol{\tau}}_{1:d}} \left\{ Q_0(\boldsymbol{Y}_{(\tau_i+1):\tau_{i+1}}; \hat{\boldsymbol{\tau}}_{1:d}) - \sum_{t=\tau_i+1}^{\tau_{i+1}} Z_t^2 + d\alpha \log n + \alpha' \sqrt{\log n} \right\} \right] > 0.$$

This event states that if you consider any segment, then the unpenalised cost for fitting just the data in that segment with changepoints $\hat{\boldsymbol{\tau}}_{1:d}$ is less than $d\alpha \log n + \alpha'(\log n)^{1/2}$ lower than the sum of the square of the true residuals for that segment. This holds for all segments and all choices of changepoints.

The second event, E_n^2, is that for $l_n = \lfloor \delta_n/2 \rfloor$

$$\min_{i=1,\ldots,m} \left\{ Q_0(\boldsymbol{Y}_{(\tau_i - l_n + 1):(\tau_i + l_n)}) - Q_0(\boldsymbol{Y}_{(\tau_i - l_n + 1):(\tau_i + l_n)}; \tau_i) \right\} > \frac{1}{50} l_n^3 \Delta_n^2$$

This event states that if you consider the l_n data points either side of any changepoint, then the reduction in the unpenalised cost of fitting a model with the true change,

as compared to fitting a model with no change, to this data is greater than a term proportional to $l_n^3\Delta_n^2$. This holds for all m changepoints.

The final event, E_n^3 is similar to that for E_n^2 but with a different number of data points associated with each changepoint. For $i = 1, \ldots, m$ let $l_n^i = \lfloor C_2(\log n)^{1/3}(\Delta_n^i)^{-2/3} \rfloor$ with C_2 as defined in the statement of Theorem 2.1. The event E_n^3, is that

$$\min_{i=1,\ldots,m} \left[\left\{ Q_0(\boldsymbol{Y}_{(\tau_i - l_n^i+1):(\tau_i+l_n^i)}) - Q_0(\boldsymbol{Y}_{(\tau_i-l_n^i+1):(\tau_i+l_n^i)}; \tau_i) \right\} - \frac{1}{50}(l_n^i)^3(\Delta_n^i)^2 \right] > 0.$$

Lemmas B.1 and B.3, which are stated and proved in Section B, show that each of these three events occurs with probability tending to 1. Thus in the following we will assume they hold, and show that if they do then, for sufficiently large n, the event in the statement of Theorem 2.1 must also hold. We will do this in three stages.

First we show that, for sufficiently large n, $\hat{m}_n \geq m$ if E_n^2 occurs. To do this we consider an arbitrary segmentation of the data $\hat{\boldsymbol{\tau}}_{1:d}$ with $d < m$ changepoints, and show that the penalised cost for this segmentation must be higher than the cost of another segmentation.

For such a segmentation, there must exist at least one true changepoint such that no estimated changepoint lies within half the minimum segment length, $l_n = \lfloor \delta_n/2 \rfloor$, of it. Denoting such a changepoint by τ_i,

$$\begin{aligned} Q_0(\boldsymbol{Y}_{1:n}; \hat{\boldsymbol{\tau}}_{1:d}) &\geq Q_0(\boldsymbol{Y}_{1:(\tau_i-l_n)}; \hat{\boldsymbol{\tau}}_{1:d}) + Q_0(\boldsymbol{Y}_{(\tau_i-l_n+1):(\tau_i+l_n)}) + Q_0(\boldsymbol{Y}_{(\tau_i+l_n+1):n}; \hat{\boldsymbol{\tau}}_{1:d}) \\ &> Q_0(\boldsymbol{Y}_{1:(\tau_i-l_n)}; \hat{\boldsymbol{\tau}}_{1:d}) + Q_0(\boldsymbol{Y}_{(\tau_i-l_n+1):(\tau_i+l_n)}; \tau_i) + Q_0(\boldsymbol{Y}_{(\tau_i+l_n+1):n}; \hat{\boldsymbol{\tau}}_{1:d}) + l_n^3\Delta_n^2/50 \\ &= Q_0(\boldsymbol{Y}_{1:n}; \hat{\boldsymbol{\tau}}_{1:d}, \tau_i - l_n, \tau_i - l_n + 1, \tau_i, \tau_i + l_n, \tau_i + l_n + 1) + l_n^3\Delta_n^2/50 \end{aligned}$$

The first inequality comes from (13). We have then used (12) and the bound on the change of unpenalised cost from adding a true changepoint that comes from event E_n^2. The penalised cost

$$Q(\boldsymbol{Y}_{1:n}; \hat{\boldsymbol{\tau}}_{1:d}) - Q(\boldsymbol{Y}_{1:n}; \hat{\boldsymbol{\tau}}_{1:d}, \tau_i - l_n, \tau_i - l_n + 1, \tau_i, \tau_i + l_n, \tau_i + l_n + 1)$$

is thus bounded below by $\Delta_n^2 l_n^3/50 - 5|\gamma| \log n - 5\beta_n$. By the assumptions on Δ_n and δ_n, $\log n = o(\Delta_n^2 l_n^3)$. If $\beta_n = o(\Delta_n^2 l_n^3)$ this will be positive for sufficiently large n. This argument applies for any segmentation with fewer than m changepoints, and hence for sufficiently large n, if E_n^2 occurs then no segmentation with fewer than m changepoints can minimise the penalised cost.

Next we show that $\hat{m}_n \leq m$ if the event E_n^1 occurs. To do this we consider an arbitrary segmentation of the data $\hat{\boldsymbol{\tau}}_{1:d}$ with $d > m$ changepoints, and show that the penalised cost for this segmentation must be higher than the cost of the true segmentation. First note that

$$\sum_{t=1}^{n} Z_t^2 \geq Q_0(\boldsymbol{Y}_{1:n}; \boldsymbol{\tau}_{1:m}).$$

Hence

$$Q(\boldsymbol{Y}_{1:n}; \hat{\boldsymbol{\tau}}_{1:d}) - Q(\boldsymbol{Y}; \boldsymbol{\tau}_{1:m}) \quad \geq \quad Q_0(\boldsymbol{Y}_{1:n}; \hat{\boldsymbol{\tau}}_{1:d}) - \sum_{t=1}^{n} Z_t^2 - d|\gamma| \log n + (d-m)\beta_n,$$

where we have used a simple bound on the difference in the contribution of the $h(\cdot)$ terms to the two penalised costs. We can bound the first part of the right-hand side by repeated application of (13):

$$\begin{aligned} Q_0(\boldsymbol{Y}_{1:n}; \hat{\boldsymbol{\tau}}_{1:d}) - \sum_{t=1}^{n} Z_t^2 \quad &\geq \quad \sum_{i=0}^{m} \left\{ Q_0(\boldsymbol{Y}_{(\tau_i+1):\tau_{i+1}}; \hat{\boldsymbol{\tau}}_{1:d}) - \sum_{t=\tau_i+1}^{\tau_{i+1}} Z_t^2 \right\} \\ &> \quad -\alpha d \log n - \alpha'(m+1)\sqrt{\log n}. \end{aligned}$$

The last inequality comes from using event E_n^1 to bound the contribution from each term in the sum. If $\beta_n > C_1 \log n$ then

$$Q(\boldsymbol{Y}_{1:n}; \hat{\boldsymbol{\tau}}_{1:d}) - Q(\boldsymbol{Y}; \boldsymbol{\tau}_{1:m}) > \{C_1(d-m) - d|\gamma| - \alpha d\} \log n - \alpha'(m+1)\sqrt{\log n}.$$

For $C_1 > m(|\gamma| + \alpha)$ this is positive for all $d > m$ for sufficiently large n. Hence there exists a constant C_1 such that if $\beta_n > C_1 \log n$ a segmentation with $d > m$ will never minimise the penalised cost.

Taken together, the results shown so far show that $\hat{m}_n = m$ with probability tending to 1. The final part of the proof is to show that there exists a constant, C_2, such that

with probability tending to 1

$$\max_{i=1,\ldots,m} \left\{ |\hat{\tau}_i - \tau_i| \, (\Delta_n^i)^{2/3} \right\} \leq C_2 (\log n)^{1/3}. \tag{14}$$

We show that this is guaranteed, for sufficiently large n, if all events occur. Similar to before, our proof will be to consider an arbitrary segmentation for which (14) does not hold, and show that it cannot minimise the penalised cost. We will consider only n large enough that l_n^i is greater than δ_n for all i. This must be occur for large enough n as l_n increases at rate that is bounded above by a constant times $(\log n/\Delta_n)^{1/3}$, while by the assumptions of the Theorem δ_n increases at a strictly faster rate.

As $\hat{m}_n = m$ with probability tending to 1, we need only consider segmentations with m changes. Let $\hat{\boldsymbol{\tau}}_{1:m}$ be such a segmentation for which (14) does not hold, and let τ_i be a changepoint for which

$$|\hat{\tau}_i - \tau_i| \, (\Delta_n^i)^{2/3} > C_2 (\log n)^{1/3}.$$

Define an event, E_n^4, to be the event that both

$$\max_{\hat{\boldsymbol{\tau}}_{1:d}} \left\{ Q_0(\boldsymbol{Y}_{(\tau_{i-1}+1):(\tau_i - l_n^i)}; \hat{\boldsymbol{\tau}}_{1:d}) - \sum_{t=\tau_{i-1}+1}^{\tau_i - l_n^i} Z_t^2 \right\} + d\alpha \log n + \alpha' \sqrt{\log n} > 0,$$

and

$$\max_{\hat{\boldsymbol{\tau}}_{1:d}} \left\{ Q_0(\boldsymbol{Y}_{(\tau_i + l_n^i + 1):(\tau_{i+1})}; \hat{\boldsymbol{\tau}}_{1:d}) - \sum_{t=\tau_i + l_n^i + 1}^{\tau_{i+1}} Z_t^2 \right\} + d\alpha \log n + \alpha' \sqrt{\log n} > 0,$$

occur for all i. This will occur with probability tending to 1 by Lemma B.1.

We have

$$Q(\boldsymbol{Y}_{1:n}; \hat{\boldsymbol{\tau}}_{1:m}) - Q(\boldsymbol{Y}_{1:n}; \boldsymbol{\tau}_{1:m}) \geq Q_0(\boldsymbol{Y}_{1:n}; \hat{\boldsymbol{\tau}}_{1:m}) - \sum_{t=1}^{n} Z_t^2 - m|\gamma| \log n.$$

Now using (13)

$$Q_0(\boldsymbol{Y}_{1:n};\hat{\boldsymbol{\tau}}_{1:m}) - \sum_{t=1}^{n} Z_t^2 \geq \sum_{j=0}^{i-2}\left\{Q_0(\boldsymbol{Y}_{(\tau_j+1):\tau_{j+1}};\hat{\boldsymbol{\tau}}_{1:m}) - \sum_{t=\tau_j+1}^{\tau_{j+1}} Z_t^2\right\}$$
$$\sum_{j=i+1}^{m}\left\{Q_0(\boldsymbol{Y}_{(\tau_j+1):\tau_{j+1}};\hat{\boldsymbol{\tau}}_{1:m}) - \sum_{t=\tau_j+1}^{\tau_{j+1}} Z_t^2\right\} + \left\{Q_0(\boldsymbol{Y}_{(\tau_{i-1}+1):(\tau_i-l_n^i)};\hat{\boldsymbol{\tau}}_{1:m}) - \sum_{t=\tau_{i-1}+1}^{\tau_i-l_n^i} Z_t^2\right\} +$$
$$\left\{Q_0(\boldsymbol{Y}_{(\tau_i+l_n^i+1):\tau_{i+1}};\hat{\boldsymbol{\tau}}_{1:m}) - \sum_{t=\tau_i+l_n^i+1}^{\tau_{i+1}} Z_t^2\right\} + \left\{Q_0(\boldsymbol{Y}_{(\tau_i+l_n^i+1):(\tau_i+l_n^i)}) - \sum_{t=\tau_i-l_n^i+1}^{\tau_i+l_n^i} Z_t^2\right\}, \quad (15)$$

where we interpret a sum from $j = 0$ to -1, or from $j = m+1$ to m as having the value 0. If E_n^1 and E_n^4 occur then we can lower bound the sum of all terms except the final one by $-m\alpha\log n - (m+1)\alpha'\sqrt{\log n}$

The final term on the right-hand side of (15) can be written as

$$\left\{Q_0(\boldsymbol{Y}_{(\tau_i-l_n^i+1):(\tau_i+l_n^i)}) - Q_0(\boldsymbol{Y}_{(\tau_i-l_n^i+1):(\tau_i+l_n^i)};\tau_i)\right\} + \left\{Q_0(\boldsymbol{Y}_{(\tau_i-l_n^i+1):(\tau_i+l_n^i)};\tau_i) - \sum_{t=\tau_i-l_n^i+1}^{\tau_i+l_n^i} Z_t^2.\right\}$$

Using events E_n^3 and E_n^1, the two bracketed terms on the right-hand side can be bounded below by $\frac{1}{50}(l_n^i)^3(\Delta_n^i)^2$ and $-\alpha\log n - \alpha'\sqrt{\log n}$ respectively.

Thus

$$Q(\boldsymbol{Y}_{1:n};\hat{\boldsymbol{\tau}}_{1:m}) - Q(\boldsymbol{Y}_{1:n};\boldsymbol{\tau}_m) > \frac{1}{50}(l_n^i)^3(\Delta_n^i)^2 - (m+1)\alpha\log n - (m+2)\alpha'\sqrt{\log n} - |\gamma|m\log n. \quad (16)$$

By the definition of l_n^i,

$$(l_n^i)^3(\Delta_n^i)^2 = (C_2)^3\log n + o(\log n),$$

and thus we can choose C_2 such that (16) is positive for large enough n. □

B Lemmas for Proof of Theorem 2.1

Throughout this section $Z_1, Z_2, \ldots$ will denote an infinite set of independent, identically distributed standard Gaussian random variables.

The following lemmas show that each of E_n^1, E_n^2 and E_n^3 occur with probability tending to 1.

Lemma B.1 *Consider data from a segment of length l,*

$$Y_t = \phi_0 + \frac{\phi_1 - \phi_0}{l} t + Z_t, \text{ for } t = 1, \ldots, l.$$

where, without loss of generality, we have assumed this is the first segment. Fix $\epsilon > 0$ and choose any constant $\alpha > 2(1+\epsilon)$. For any set of $d \geq 1$ changepoints $\tau_{1:d}$ with $0 < \tau_1 < \cdots < \tau_d < l$, there exists a constant C independent of l, d and the changepoint locations such that

$$\Pr\left(\sum_{t=1}^{l} Z_t^2 - Q_0(Y_{1:l}; \tau_{1:d}) > d\alpha \log l\right) \leq C l^{-d(1+\epsilon)}; \tag{17}$$

and for any $\alpha' > 0$,

$$\Pr\left(\sum_{t=1}^{l} Z_t^2 - Q_0(Y_{1:l}) > \alpha' \sqrt{\log l}\right) \to 0 \tag{18}$$

as $l \to \infty$.

Furthermore as $l \to \infty$,

$$\Pr\left\{\max_{d, \tau_{1:d}} \left(\sum_{t=1}^{l_n} Z_t^2 - Q_)(Y_{1:l}; \tau_{1:d}) - d\alpha \log l - \alpha' \sqrt{\log l}\right) > 0\right\} \to 0 \tag{19}$$

Proof. For the first set of results $\tau_{1:d}$ is a fixed set of d changepoints. Standard results for the normal linear model give,

$$\sum_{t=1}^{l} Z_t^2 - Q_0(Y_{1:l}; \tau_{1:d}) \sim \chi^2_{d+2},$$

as we are fitting a model with $d+2$ parameters. We can bound the upper tail of this random variable using (see e.g. Lemma 8.1 of Birgé, 2001)

$$\Pr\left\{\chi^2_{d+2} > (d+2) + 2\sqrt{(d+2)x} + 2x\right\} \leq \exp(-x). \tag{20}$$

For any $\alpha > 2(1+\epsilon)$, for large enough l and any integer $d > 0$

$$d\alpha \log l > (d+2) + 2\sqrt{(d+2)d(1+\epsilon)\log l} + 2d(1+\epsilon)\log l,$$

and hence there exists an L_0 such that for $l > L_0$, using (20) with $x = d(1+\epsilon)\log l$,

$$\Pr(\chi^2_{d+2} > d\alpha \log l) \leq \exp\{-d(1+\epsilon)\log l\} = l^{-d(1+\epsilon)}.$$

As we can choose an L_0 independent of d, this is sufficient to prove (17).

To show (18) we use (20) with $d = 0$. For any $\alpha' > 0$

$$\alpha'\sqrt{\log l} > 2 + 2\sqrt{2x} + 2x,$$

where $x = (\alpha'/3)(\log l)^{1/2}$, for large enough l. Hence for large enough l

$$\Pr\left(\sum_{t=1}^{l} Z_t^2 - Q_0(Y_{1:l}) > \alpha'\sqrt{\log l}\right) \leq \exp\left(a\frac{\alpha'}{3}\sqrt{\log l}\right),$$

and the right-hand side tends to 0 as $l \to \infty$.

To show (19) holds it is sufficient to sum the probabilities in (17) over all segmentations of $Y_{1:l_n}$ and show this sum tends to 0. To do this note that we can bound the number of segmentations with d changepoints by l^d. Thus

$$\Pr\left\{\max_{d,\tau_{1:d}}\left(\sum_{t=1}^{l} Z_t^2 - Q_0(Y_1 : l; \tau_{1:d}) - d\alpha\log l - \alpha'\sqrt{\log l}\right) > 0\right\} \leq \sum_{d=1}^{l-1} l^d C l^{-d(1+\epsilon)} < C\sum_{d=1}^{\infty} l^{-d\epsilon}.$$

This is just $Cl^{-\epsilon}/(1-l^{-\epsilon})$ which, as $\epsilon > 0$, tends to 0 as $l \to \infty$ as required. □

Corollary B.2 *Event $E_n^{(1)}$ occurs with probability tending to 1 as $n \to \infty$.*

Proof. This follows immediately from using (19) for each of the $m+1$ segments. □

Lemma B.3 *For a given l and any ϕ_0, ϕ_1 and ϕ_2 with*

$$\Delta = \left|\frac{\phi_1 - \phi_0}{l} - \frac{\phi_2 - \phi_1}{l}\right|$$

let

$$Y_t = \phi_0 + \frac{\phi_1 - \phi_0}{l}t + Z_t, \text{ for } t = 1, \ldots, l, \text{ and}$$
$$Y_t = \phi_1 + \frac{\phi_2 - \phi_1}{l}(t - l) + Z_t, \text{ for } t = l + 1, \ldots, 2l.$$

Then for $l > 2$

$$\Pr\left(Q_0(\boldsymbol{Y}_{1:2l}) - Q_0(\boldsymbol{Y}_{1:2l}; l) < \frac{1}{50}\Delta^2 l^3\right) \leq \exp\left\{-\frac{1}{800}\Delta^2 l^3\right\}.$$

Proof. Standard results for the normal linear model (e.g Theorem 15.8 of Muller and Stewart, 2006) give that, for $l > 2$, $Q_0(\boldsymbol{Y}_{1:2l}) - Q_0(\boldsymbol{Y}_{1:2l}; l)$ has a non-central chi-squared distribution with 1 degree of freedom, and non-centrality parameter

$$\nu = \Delta^2 \frac{l(l+1)(l-1)}{24}\left\{\frac{4l^2+2}{4l^2-1}\right\}.$$

For $l > 2$, $\nu > \Delta^2 l^3/25$. We can bound the lower tail of such a random variable, $\chi^2_1(\nu)$, using (see e.g. Lemma 8.1 of Birgé, 2001)

$$\Pr\left(\chi^2_1(\nu) < 1 + \nu - 2\sqrt{(1+2\nu)x}\right) \leq \exp\{-x\}.$$

Taking $x = (1+2\nu)/64$, and noting that for such an x, $(\nu+1) - 2\sqrt{(1+2\nu)x} > \nu/2$, we get

$$\Pr\left(Q_0(\boldsymbol{Y}_{1:2l}) - Q_0(\boldsymbol{Y}_{1:2l}; l) < \frac{1}{50}\Delta^2 l^3\right) \leq \Pr\left(Q_0(\boldsymbol{Y}_{1:2l}) - Q_0(\boldsymbol{Y}_{1:2l}; l) < \nu/2\right) \leq \exp\{-\nu/32\}.$$

The result follows by noting that $\nu > l^3\Delta^2/25$ for $l > 2$. □

Corollary B.4 *Events* $E_n^{(2)}$ *and* $E_n^{(3)}$ *occur with probability tending to 1 as* $n \to \infty$.

Proof. We can apply Lemma B.3 to each region around a changepoint as $l_n > 2$ for sufficiently large n. For event E_n^2, as $\Delta_n^2 l_n^3 \to \infty$ the probability of

$$Q_0(\boldsymbol{Y}_{\tau_i - l_n + 1:\tau_i + l_n}) - Q_0(\boldsymbol{Y}_{\tau_i - l_n + 1:\tau_i + l_n}; \tau_i) > \frac{1}{50} l_n^3 \Delta_n^2$$

for a given changepoint, τ_i, tends to 1. As there are a fixed number of changepoints, we get that this must hold for all changepoints with probability tending to 1, as required. A similar argument holds for event E_n^3. □

C Updates for Quadratic Functions

In Section 3 (equation 5) we define a function, $f^t_{\boldsymbol{\tau}}(\phi)$, as the minimum cost of segmenting $\mathbf{y}_{1:t}$ with changepoints at $\boldsymbol{\tau} = \tau_1, \ldots, \tau_k$ and fitted value $\phi_t = \phi$ at time t. We then derived a recursion for these functions as follows

$$f^t_{\boldsymbol{\tau}}(\phi) = \min_{\phi'} \left\{ f^{\tau_k}_{\tau_1,\ldots,\tau_{k-1}}(\phi') + \mathcal{C}(y_{\tau_k+1:t}, \phi', \phi) + \beta + h(\tau_{i+1} - \tau_i) \right\}. \tag{21}$$

The functions $f^t_{\boldsymbol{\tau}}(\phi)$ are quadratics in ϕ, and we denote $f^t_{\boldsymbol{\tau}}(\phi)$ as follows

$$f^t_{\boldsymbol{\tau}}(\phi) = a^t_{\boldsymbol{\tau}} + b^t_{\boldsymbol{\tau}}\phi + c^t_{\boldsymbol{\tau}}\phi^2, \tag{22}$$

for some constants $a^t_{\boldsymbol{\tau}}$, $b^t_{\boldsymbol{\tau}}$ and $c^t_{\boldsymbol{\tau}}$. We then wish to calculate these coeffcients by updating the coefficients that make up $f^{\tau_k}_{\tau_1,\ldots,\tau_{k-1}}(\phi')$ using (21). To do this we need to write the cost for the segment from $\tau_k + 1$ to t in quadratic form. Defining the length of the segment as $s = t - \tau_k$ this cost can be written as

$$\begin{aligned}
\mathcal{C}(\mathbf{y}_{\tau_k+1:t}, \phi', \phi) =& \frac{(s+1)(2s+1)}{6s\sigma^2}\phi^2 + \left(\frac{(s+1)}{\sigma^2} - \frac{(s+1)(2s+1)}{3s\sigma^2} \right) \phi'\phi \\
&- \left(\frac{2}{s\sigma^2} \sum y_j(j - \tau_k) \right) \phi + \left(\frac{1}{\sigma^2} \sum y_i^2 \right) \\
&+ 2 \left(\frac{1}{s\sigma^2} \sum y_j(j - \tau_k) - \frac{1}{\sigma^2} \sum y_i \right) \phi' + \frac{(s-1)(2s-1)}{6s\sigma^2}\phi'^2.
\end{aligned} \tag{23}$$

Writing (23) as $A\phi^2 + B\phi'\phi + C\phi + D + E\phi' + F\phi'^2$ for constants A, B, C, D and E, substituting (23) into (21) and minimising out ϕ' we can get the formula for the updating the coefficients of the quadratic $f^t_{\boldsymbol{\tau}}(\phi)$:

$$
\begin{aligned}
a^t_{\boldsymbol{\tau}} &= A - \frac{B^2}{4\left(a^{\tau_k}_{(\tau_1,\ldots,\tau_{k-1})} + F\right)}, \\
b^t_{\boldsymbol{\tau}} &= C - \frac{\left(b^{\tau_k}_{(\tau_1,\ldots,\tau_{k-1})} + E\right)B}{2\left(a^{\tau_k}_{(\tau_1,\ldots,\tau_{k-1})} + F\right)}, \\
c^t_{\boldsymbol{\tau}} &= c^{\tau_k}_{(\tau_1,\ldots,\tau_{k-1})} + D - \frac{\left(b^{\tau_k}_{(\tau_1,\ldots,\tau_{k-1})} + E\right)^2}{4\left(a^{\tau_k}_{(\tau_1,\ldots,\tau_{k-1})} + F\right)} + \beta + h(t-\tau_k).
\end{aligned} \tag{24}
$$

D Proofs from Section 3

D.1 Proof of Theorem 3.1

The proof of Theorem 3.1 works by contrapositive. We show that if $(\boldsymbol{\tau}, s) \in \overset{*}{\mathcal{T}}_t$ then a necessary condition of this is that $\boldsymbol{\tau} \in \overset{*}{\mathcal{T}}_s$, taking the contrapositive of this gives Theorem 3.1.

proof Assume $(\boldsymbol{\tau}, s) \in \overset{*}{\mathcal{T}}_t$, then there exists ϕ such that

$$
f^t(\phi) = f^t_{(\boldsymbol{\tau},s)}(\phi),
$$

Now for any ϕ^*,

$$
\begin{aligned}
f^s(\phi^*) + \mathcal{C}(\mathbf{y}_{s+1:t}, \phi^*, \phi) + \beta &\geq \min_{\phi', r} \left[f^r(\phi') + \mathcal{C}(\mathbf{y}_{r+1:t}, \phi', \phi) + \beta \right], \\
&= f^t(\phi), \\
&= f^t_{(\boldsymbol{\tau},s)}(\phi), \\
&= \min_{\phi''} \left\{ f^s_{\boldsymbol{\tau}}(\phi'') + \mathcal{C}(\mathbf{y}_{s+1:t}, \phi'', \phi) + \beta \right\}, \qquad (25) \\
&= f^s_{\boldsymbol{\tau}}(\phi^A) + \mathcal{C}(\mathbf{y}_{s+1:t}, \phi^A, \phi) + \beta,
\end{aligned}
$$

where ϕ^A is the value of ϕ'' which minimises (25). As ϕ^* can be chosen as any value, we can choose it as ϕ^A. By cancelling terms we get $f^s(\phi^A) \geq f^s_{\boldsymbol{\tau}}(\phi^A)$ and hence

$f^s(\phi^A) = f^s_{\boldsymbol{\tau}}(\phi^A)$ and therefore $\boldsymbol{\tau} \in \overset{*}{\mathcal{T}}_s$. We have shown that if $(\boldsymbol{\tau}, s) \in \overset{*}{\mathcal{T}}_t$ then $\boldsymbol{\tau} \in \overset{*}{\mathcal{T}}_s$, by taking the contrapositive the theorem holds. □

D.2 Proof of Theorem 3.2

The proof for Theorem 3.2 follow a similar argument to the corresponding proof in Killick et al. (2012). However we have to add a segment consisting of the single point y_{t+1} to deal with the dependence between the segments.

Proof Let $\boldsymbol{\tau}^*$ denote the optimal segmentation of $\mathbf{y}_{1:t}$. We will repeatedly use the fact that

$$\mathcal{C}(y_{t+1}, \phi', \phi) = \frac{1}{\sigma^2}(y_{t+1} - \phi)^2,$$

and this does not depend on ϕ'.

First consider $T = t+1$. As adding a changepoint without penalty will always reduce the cost, it is straightforward to show

$$\begin{aligned}
f^T_{\boldsymbol{\tau}}(\phi) &\geq \min_{\phi'} \left[f^t_{\boldsymbol{\tau}}(\phi') + \mathcal{C}(y_{t+1}, \phi', \phi) \right], \\
&= \min_{\phi'}[f^t_{\boldsymbol{\tau}}(\phi')] + \min_{\phi'}[\mathcal{C}(y_{t+1}, \phi', \phi)], \\
&> \min_{\phi'} \left[f^t(\phi') \right] + K + \min_{\phi'}[\mathcal{C}(y_{t+1}, \phi', \phi)], \\
&\geq \min_{\phi'} \left[f^t(\phi') + \mathcal{C}(y_{t+1}, \phi', \phi) + \beta + h(1) \right].
\end{aligned}$$

Thus segmenting $\mathbf{y}_{1:T}$ with changepoints $\boldsymbol{\tau}$ always has a greater cost than segmenting $\mathbf{y}_{1:T}$ with changepoints $(\boldsymbol{\tau}^*, t)$.

Now we consider $T > t+1$. We start by noting that by adding changes, at any point, without the penalty term and minimising over the corresponding ϕ values will also decrease the cost. Therefore

$$f^T_{\boldsymbol{\tau}}(\phi) \geq \min_{\phi', \phi''} \left[f^t_{\boldsymbol{\tau}}(\phi') + \mathcal{C}(y_{t+1}, \phi', \phi'') + \mathcal{C}(\mathbf{y}_{t+2:T}, \phi'', \phi) \right]. \tag{26}$$

So from (26) and using (8),

$$
\begin{aligned}
f_{\boldsymbol{\tau}}^{T}(\phi) &\geq \min_{\phi',\phi''}\left[f_{\boldsymbol{\tau}}^{t}(\phi') + \mathcal{C}(y_{t+1},\phi',\phi'') + \mathcal{C}(\mathbf{y}_{t+2:T},\phi'',\phi)\right], \\
&\geq \min_{\phi'}[f_{\boldsymbol{\tau}}^{t}(\phi')] + \min_{\phi',\phi''}[\mathcal{C}(y_{t+1},\phi',\phi'') + \mathcal{C}(\mathbf{y}_{t+2:T},\phi'',\phi)], \\
&> \min_{\phi'}\left[f^{t}(\phi')\right] + K + \min_{\phi',\phi''}[\mathcal{C}(y_{t+1},\phi',\phi'') + \mathcal{C}(\mathbf{y}_{t+2:T},\phi'',\phi)], \\
&\geq \min_{\phi',\phi''}\left[f^{t}(\phi') + \mathcal{C}(y_{t+1},\phi',\phi'') + \beta + h(1) + \mathcal{C}(\mathbf{y}_{t+2:T},\phi'',\phi) + \beta + h(T-t+1)\right].
\end{aligned}
$$

Therefore the cost of segmenting $\mathbf{y}_{1:T}$ with changepoints $\boldsymbol{\tau}$ is always greater than the cost of segmenting $\mathbf{y}_{1:T}$ with changepoints $(\boldsymbol{\tau}^*, t, t+1)$ (where $\boldsymbol{\tau}^*$ is the optimal segmentation of $\mathbf{y}_{1:t}$) and this holds for all $T > t+1$ and hence $\boldsymbol{\tau}$ can be pruned. □

E Pseudo-Code for CPOP

The CPOP algorithm uses Algorithm 2 to calculate the intervals on which each function is optimal. This then enables the functions that are not optimal for any value of ϕ to be removed. The idea of this algorithm is as follows.

We initialise the algorithm by setting the current parameter value as $\phi_{curr} = -\infty$ and comparing the cost functions in our current set of candidates (which we initialise as $\mathcal{T}_{temp} = \hat{\mathcal{T}}_t$) to get the optimal segmentation for this value, $\boldsymbol{\tau}_{curr}$. This can be optimisation can be done my noting that the quadratic with smallest cost will have the smallest coefficient of the quadratic term. If more than one quadratic has the smallest coefficient, we then choose the quadratic with the largest coefficient of the linear term; and if necessary, then choose the quadratic with the smallest constant term.

For each $\boldsymbol{\tau} \in \mathcal{T}_{curr}$ we calculate where $f_{\boldsymbol{\tau}}^{t}$ next intercepts with $f_{\boldsymbol{\tau}_{curr}}^{t}$ (smallest value of ϕ for which $f_{\boldsymbol{\tau}}^{t}(\phi) = f_{\boldsymbol{\tau}_{curr}}^{t}(\phi)$ and $\phi > \phi_{curr}$) and store this as $x_{\boldsymbol{\tau}}$. If for a $\boldsymbol{\tau} \in \mathcal{T}_{temp}$ we have $x_{\boldsymbol{\tau}} = \emptyset$ (i.e. $f_{\boldsymbol{\tau}}^{t}$ doesn't intercept with $f_{\boldsymbol{\tau}_{curr}}^{t}$ for any $\phi > \phi_{curr}$) then we

Algorithm 1: Algorithm for Continuous Piecewise-linear Optimal Partitioning (CPOP)

Input : Set of data of the form $\mathbf{y}_{1:n} = (y_1, \ldots, y_n)$.

A positive penalty constant, β, and a non-negative, non-decreasing penalty function $h(\cdot)$.

Let $n =$ length of data;

set $\hat{\mathcal{T}}_1 = \{0\}$;

and set $K = 2\beta + h(1) + h(n)$;

for $t = 1, \ldots, n$ **do**

  for $\boldsymbol{\tau} \in \hat{\mathcal{T}}_t$ **do**

    if $\tau = \{0\}$ **then**

      $f^t_{\boldsymbol{\tau}}(\phi) = \min_{\phi'} \mathcal{C}(\mathbf{y}_{1:t}, \phi', \phi) + h(t);$

    else

      $f^t_{\boldsymbol{\tau}}(\phi) = \min_{\phi'} \left\{ f^{\tau_k}_{\tau_1,\ldots,\tau_{k-1}}(\phi') + \mathcal{C}(\mathbf{y}_{\tau_k+1:t}, \phi', \phi) + h(t - \tau_k) + \beta \right\};$

  for $\boldsymbol{\tau} \in \hat{\mathcal{T}}_t$ **do**

    $Int^t_{\boldsymbol{\tau}} = \left\{ \phi : f^t_{\boldsymbol{\tau}}(\phi) = \min_{\boldsymbol{\tau}' \in \hat{\mathcal{T}}_t} f^t_{\boldsymbol{\tau}'}(\phi) \right\};$

    $\overset{*}{\mathcal{T}}_t = \{\boldsymbol{\tau} : Int^t_{\boldsymbol{\tau}} \neq \emptyset\};$

    $\hat{\mathcal{T}}_{t+1} = \hat{\mathcal{T}}_t \cup \left\{ (\boldsymbol{\tau}, t) : \boldsymbol{\tau} \in \overset{*}{\mathcal{T}}_t \right\};$

  $\hat{\mathcal{T}}_{t+1} = \left\{ \boldsymbol{\tau} \in \hat{\mathcal{T}}_{t+1} : \min_{\phi} f^t_{\boldsymbol{\tau}}(\phi) \leq \min_{\phi', \boldsymbol{\tau}'} \left[f^t_{\boldsymbol{\tau}'}(\phi') \right] + K \right\};$

$f_{opt} = \min_{\boldsymbol{\tau}, \phi} f^n_{\boldsymbol{\tau}}(\phi);$

$\boldsymbol{\tau}_{opt} = \arg\min_{\boldsymbol{\tau}} \left[\min_{\phi} f^n_{\boldsymbol{\tau}}(\phi) \right];$

Output: The optimal cost, f_{opt}, and the corresponding changepoint vector, $\boldsymbol{\tau}_{opt}$.

remove $\boldsymbol{\tau}$ from $\mathcal{T}_{temp}$. We take the minimum of $x_{\boldsymbol{\tau}}$ (the first of the intercepts) and set it as our new ϕ_{curr} and the corresponding changepoint vector that produces it as $\boldsymbol{\tau}_{curr}$. We repeat this procedure until the set $\mathcal{T}_{temp}$ consists of only a single value $\boldsymbol{\tau}_{curr}$

which is the optimal segmentation for all future $\phi > \phi_{curr}$.

As written, our algorithm assumes there is a unique quadratic that is optimal for each interval – which we believe will happen with probability 1. If this is not the case, we can interpret the algorithm as choosing one of the optimal quadratics, and outputing an optimal, as opposed to the unique optimal, segmentation. Obviously the algorithm could be re-written to store and output multiple optimal segmentations if they exist.

Algorithm 2: Algorithm for calculation of $Int^t_{\boldsymbol{\tau}}$ at time t

Input : Set of changepoint candidate vectors $\hat{\mathcal{T}}_t$ for current timestep, t,
Optimal segmentation functions $f^t_{\boldsymbol{\tau}}(\phi)$ for current time step t and $\boldsymbol{\tau} \in \hat{\mathcal{T}}_t$.

$\mathcal{T}_{temp} = \hat{\mathcal{T}}_t$;
$Int^t_{\boldsymbol{\tau}} = \emptyset$ for $\boldsymbol{\tau} \in \hat{\mathcal{T}}_t$;
$\phi_{curr} = -\infty$;
$\boldsymbol{\tau}_{curr} = \underset{\boldsymbol{\tau} \in \mathcal{T}_{temp}}{\arg\min} \left[f^t_{\boldsymbol{\tau}}(\phi_{curr}) \right]$;
while $\mathcal{T}_{temp} \backslash \{\boldsymbol{\tau}_{curr}\} \neq \emptyset$ **do**
  for $\boldsymbol{\tau} \in \mathcal{T}_{temp} \backslash \{\boldsymbol{\tau}_{curr}\}$ **do**
    $x_{\boldsymbol{\tau}} = \min\{\phi : f^t_{\boldsymbol{\tau}}(\phi) - f^t_{\boldsymbol{\tau}_{curr}}(\phi) = 0 \ \ \& \ \ \phi > \phi_{curr}\}$;
    if $x_{\boldsymbol{\tau}} = \emptyset$ **then**
      $\mathcal{T}_{temp} = \mathcal{T}_{temp} \backslash \{\boldsymbol{\tau}\}$
  $\boldsymbol{\tau}_{new} = \underset{\boldsymbol{\tau}}{\arg\min}(x_{\boldsymbol{\tau}})$;
  $\phi_{new} = \underset{\boldsymbol{\tau}}{\min}(x_{\boldsymbol{\tau}})$;
  $Int^t_{\boldsymbol{\tau}_{curr}} = [\phi_{curr}, \phi_{new}] \cup Int^t_{\boldsymbol{\tau}_{curr}}$;
  $\boldsymbol{\tau}_{curr} = \boldsymbol{\tau}_{new}$;
  $\phi_{curr} = \phi_{new}$;

Output: The intervals $Int^t_{\boldsymbol{\tau}}$ for $\boldsymbol{\tau} \in \hat{\mathcal{T}}_t$

References

Birgé, L. (2001). An alternative point of view on Lepski's method. *Lecture Notes-Monograph Series*, pages 113–133.

Killick, R., Fearnhead, P., and Eckley, I. A. (2012). Optimal detection of changepoints with a linear computational cost. *Journal of the American Statistical Association*, 107:1590–1598.

Muller, K. E. and Stewart, P. W. (2006). *Linear model theory: univariate, multivariate, and mixed models*. John Wiley & Sons.

Yao, Y.-C. (1988). Estimating the number of change-points via Schwarz' criterion. *Statistics & Probability Letters*, 6(3):181–189.