



Detecting Changes in Slope With an L_0 Penalty

Paul Fearnhead, Robert Maidstone & Adam Letchford

To cite this article: Paul Fearnhead, Robert Maidstone & Adam Letchford (2019) Detecting Changes in Slope With an L_0 Penalty, Journal of Computational and Graphical Statistics, 28:2, 265-275, DOI: [10.1080/10618600.2018.1512868](https://doi.org/10.1080/10618600.2018.1512868)

To link to this article: <https://doi.org/10.1080/10618600.2018.1512868>



© 2018 The Author(s). Published with license by Taylor & Francis.



View supplementary material [↗](#)



Published online: 29 Oct 2018.



Submit your article to this journal [↗](#)



Article views: 6811



View related articles [↗](#)



View Crossmark data [↗](#)



Citing articles: 17 View citing articles [↗](#)

Detecting Changes in Slope With an L_0 Penalty

Paul Fearnhead^a, Robert Maidstone^{a,b}, and Adam Letchford^c

^aDepartment of Mathematics and Statistics, Lancaster University, Lancaster, United Kingdom; ^bSTOR-i Doctoral Training Centre, Lancaster University, Lancaster, United Kingdom; ^cDepartment of Management Science, Lancaster University, Lancaster, United Kingdom

ABSTRACT

While 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, while 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 penalizes 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 analyze a time series with 10,000 observations and 100 changes in a few minutes. Our method is used to analyze data on the motion of bacteria, and provides better and more parsimonious fits than two competing approaches. Supplementary material for this article is available online.

ARTICLE HISTORY

Received November 2017
Revised July 2018

KEYWORDS

Breakpoints; Functional pruning; Linear spline regression; Narrowest-over-threshold; Trend-filtering

1. Introduction

Changepoint detection and modeling 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, Fearnhead, and Mihaylova 2014); and fMRI (Aston and Kirch 2012). It appears to be increasingly important for analyzing large-scale data streams, as a flexible way of modeling heterogeneity in these streams. This article focuses 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 analyzing 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 summarized by its angular position. The data we show come from Sowa et al. (2005), and consist 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 characterized by stationary periods interspersed by periods of

roughly constant angular velocity. The movement tends to be, though is not exclusively, in one direction.

Sowa et al. (2005) analyzed these data using a changepoint model, where the mean is piecewise constant. An example fit from such a model is shown in Figure 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 Figure 1(b).

While 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, Chen, and Fryzlewicz 2016a). Binary segmentation 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.

CONTACT Paul Fearnhead  p.fearnhead@lancaster.ac.uk  Department of Mathematics and Statistics, Lancaster University, Lancaster LA1 4YW, United Kingdom. Color versions of one or more of the figures in the article can be found online at www.tandfonline.com/r/JCGS.

 Supplementary materials for this article are available online. Please go to www.tandfonline.com/r/JCGS.

© 2018 The Author(s). Published with license by Taylor & Francis.

This is an Open Access article distributed under the terms of the Creative Commons Attribution-NonCommercial-NoDerivatives License (<http://creativecommons.org/licenses/by-nc-nd/4.0/>), which permits non-commercial re-use, distribution, and reproduction in any medium, provided the original work is properly cited, and is not altered, transformed, or built upon in any way.

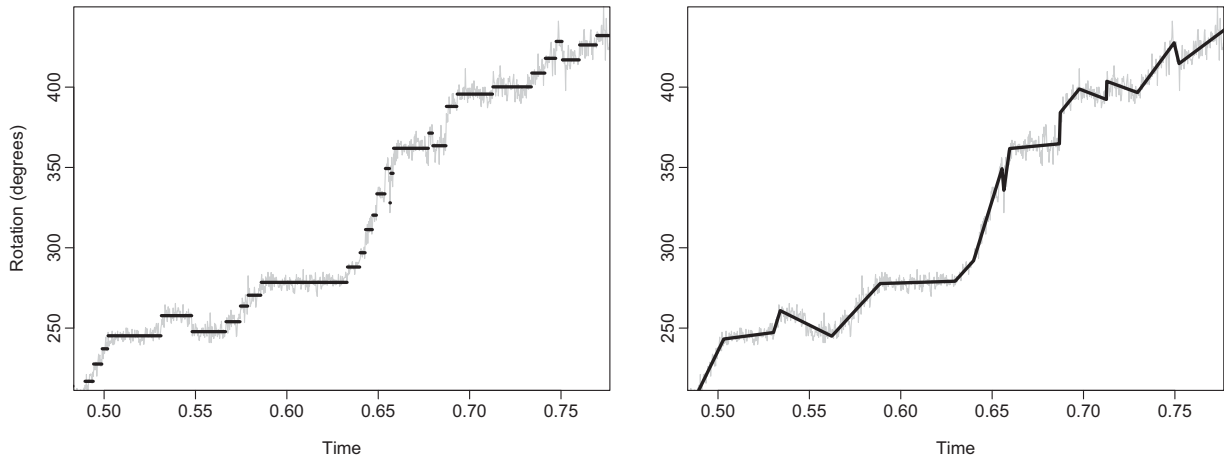


Figure 1. Part of a time series of angular position of a bacterium (Sowa et al. 2005); best-fitting piecewise constant mean (left-hand plot) and continuous piecewise linear mean (right-hand plot). The former fits data from periods of rotation with a number of short stationary regimes.

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 penalized 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, Fearnhead, and Eckley 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 minimize 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 analyzing 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 used an exhaustive search to find the best segmentation—an approach that is only feasible for very small datasets, 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 datasets.

2. A Penalized Cost Approach to Detecting Changes in Slope

We assume that we have data, $\mathbf{y} = (y_1, \dots, y_n)$, ordered by time. We will use the notation that, for $t \geq s$, the set of observations from time s to time t is $\mathbf{y}_{s:t} = (y_s, \dots, y_t)$. 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 j th changepoint be τ_j for $j = 1, \dots, m$, and set $\tau_0 = 0$ and $\tau_{m+1} = n$. The j th segment will consist of data points $y_{\tau_{j-1}+1}, \dots, y_{\tau_j}$. We let $\boldsymbol{\tau} = (\tau_0, \dots, \tau_{m+1})$ 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 parameterizing the linear function within each segment. Due to the continuity constraint that we wish to enforce, it is helpful to parameterize 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 $i = 0, \dots, m + 1$. We will simplify notation by letting $\boldsymbol{\phi} = (\phi_{\tau_0}, \dots, \phi_{\tau_{m+1}})$. In situations where we refer to a subset of this vector, we will use the notation $\boldsymbol{\phi}_{j:k} = (\phi_{\tau_j}, \dots, \phi_{\tau_k})$ for $0 \leq j \leq k \leq m + 1$.

Under this parameterization, we model the data as, for $i = 0, \dots, m$,

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

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

We infer the set of changepoints with a penalized cost approach, using a squared-error loss function to measure fit to the data. That is, we minimize over m , τ , and ϕ ,

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

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 the Bayesian information criterion (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 penalized cost approaches, and such penalties have been suggested (Davis, Lee, and Rodriguez-Yam 2006; Zhang and Siegmund 2007). 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 $y_{s+1:t}$ with a linear function that starts at ϕ at time s and ends at ψ at time t as

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

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

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

2.1. Asymptotic Properties of the Penalized Cost Approach

We now consider the asymptotic properties of estimating changepoints by minimizing (3). For this we will assume data are generated from the model (1) with Z_1, Z_2, \dots , 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 $h(t) = \gamma \log t$ for some constant γ . This covers the common choices of how the penalty depends on segment length (e.g., Davis, Lee, and Rodriguez-Yam 2006; Zhang and Siegmund 2007).

Theorem 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, \dots, Z_n being independent identically distributed standard Gaussian random variables. Let $\delta_n = \min_{i=1, \dots, m+1} (\tau_i - \tau_{i-1})$

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 $\delta_n \rightarrow \infty$ and $\delta_n^3 \Delta_n^2 / \log n \rightarrow \infty$ as $n \rightarrow \infty$. Let \hat{m}_n be the number of changepoints estimated by minimizing (3) with $h(t) = \gamma \log t$ and β replaced by β_n , and let $\hat{\tau}_1, \dots, \hat{\tau}_{\hat{m}_n}$ 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 $n \rightarrow \infty$,

$$\Pr(\hat{m}_n = m, \max_{i=1, \dots, m} \{ |\hat{\tau}_i - \tau_i| (\Delta_n^i)^{2/3} \} \leq C_2 (\log n)^{1/3}) \rightarrow 1. \quad (4)$$

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, Chen, and Fryzlewicz (2016a) 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 $\delta_n^3 \Delta_n^2 / \log n \rightarrow \infty$ means that (4) implies $|\hat{\tau}_i - \tau_i| = o_p(\delta_n)$ for all $i = 1, \dots, n$: the error in estimating the changepoint locations are asymptotically negligible when compared to the minimum segment length.

3. Minimizing the Penalized 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 neighboring segments share a common parameter: the endpoint 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

minimize 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 neighboring 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 $t = 1, \dots, n$. When segmenting $\mathbf{y}_{1:t}$ with k changepoints, τ_1, \dots, τ_k , we use the notation $\tau_0 = 0$ and $\tau_{k+1} = t$. We define the function $f^t(\phi)$ to be the minimum penalized cost for segmenting $\mathbf{y}_{1:t}$ conditional on the fitted value at time t being ϕ :

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

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

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

The idea is that we split the minimization into first minimizing over the time of the most recent changepoint and the fitted value at that changepoint, and then minimizing 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 minimization 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 minimization gives the minimum penalized 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, τ . We define each of these functions $f_\tau^t(\phi)$ as the minimum cost of segmenting $\mathbf{y}_{1:t}$ with changepoints at $\tau = \tau_1, \dots, \tau_k$ and fitted value at time t being ϕ :

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

$$\left. + C(\mathbf{y}_{\tau_k+1:t}, \phi_{\tau_k}, \phi) + h(t - \tau_k) + \beta(k+1) \right\}. \quad (5)$$

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

Each of the above functions, $f_\tau^t(\phi)$, is a quadratic in ϕ and thus can be represented by a vector of length 3, with the terms in this vector denoting the coefficients of the quadratic. We can calculate the coefficients 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 minimizing $f^n(\phi)$ over ϕ . The value of τ 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 $f_\tau^t(\phi)$, that we need to store. There are two ways in which this can be achieved: functional pruning and inequality-based pruning (Killick, Fearnhead, and Eckley 2012; Rigai 2015; Maidstone et al. 2017). In both cases, they are able to remove changepoint vectors while still maintaining the guarantee that the resulting algorithm will find the true minimum of the optimization problem (2).

3.2. Functional Pruning

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

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

$$\mathcal{T}_t^* = \{ \tau \in \mathcal{T}_t : f^t(\phi) = f_\tau^t(\phi), \text{ for some } \phi \in (-\infty, \infty) \}, \quad (6)$$

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

Theorem 2. If $\tau \notin \mathcal{T}_s^*$, then $(\tau, s) \notin \mathcal{T}_t^*$ for all $t > s$.

Proof: See Appendix D.

The key to an efficient algorithm will be a way of efficiently calculating \mathcal{T}_t^* . We can use the above theorem to help us do this. From Theorem 2, we can define a set

$$\hat{\mathcal{T}}_t = \{ (\tau, s) : s \in \{0, \dots, t-1\}, \tau \in \mathcal{T}_s^* \}, \quad (7)$$

and we will have that $\hat{\mathcal{T}}_t \supseteq \mathcal{T}_t^*$. So assume that we have calculated the sets \mathcal{T}_s^* for $s = 0, \dots, t-1$. We can calculate $f_\tau^t(\phi)$ only for $\tau \in \hat{\mathcal{T}}_t$. When calculating $f^t(\phi)$ we can just minimize over the set of changepoint vectors in $\hat{\mathcal{T}}_t$ rather than the full set. To find \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 $\tau \in \hat{\mathcal{T}}_t$ for which $f^t(\phi) = f_\tau^t(\phi)$ 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. \square

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. Define $K = 2\beta + h(1) + h(n)$. If $h(\cdot)$ is nonnegative and nondecreasing and if for some τ ,

$$\min_{\phi} f_{\tau}^t(\phi) > \min_{\phi'} [f^t(\phi')] + K, \quad (8)$$

then at any future time T , τ can never be optimal for the data $y_{1:T}$.

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 sec. 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 . \square

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 minimizing our criterion (2) with the BIC penalty, though see Maidstone (2016, chap. 5) 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 segmentation (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, Chen, and Fryzlewicz (2016a) presented the *narrowest-over-threshold* (NOT) algorithm. The NOT algorithm proceeds by (i) taking a prespecified number, M , of intervals of data, $y_{s_i:t_i}$; say; (ii) performing a generalized likelihood ratio test for a change in slope on each $y_{s_i:t_i}$; (iii) keeping all intervals for which the test statistic is above some prespecified 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 minimizes the BIC for a model where the residuals are independent Gaussian with unknown variance σ^2 . For a segmentation with m changepoints at locations τ , the BIC corresponds to the minimum, over ϕ , of

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

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 minimized 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 minimize 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.

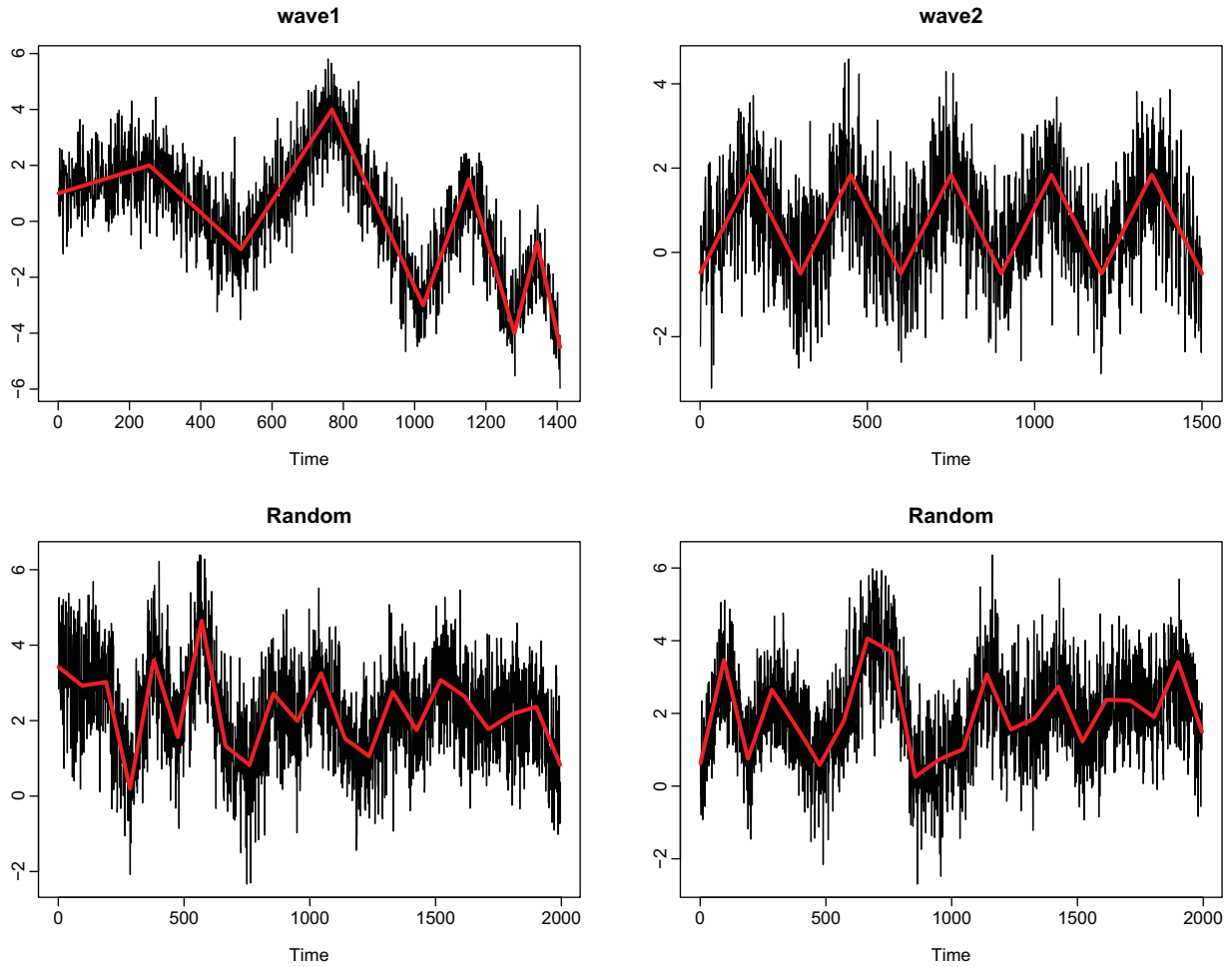


Figure 2. Example data from the three simulation scenarios: *wave1* and *wave2* (top row) have a fixed mean function. For the *Random* scenario (bottom row), the form of the mean is random, and we give two example realizations.

In the comparisons below, we implement CPOP for minimizing (2) with the BIC penalty. We use the `not` R-package to implement NOT (Baranowski, Chen, and Fryzlewicz 2016b), 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, Chen, and Fryzlewicz (2016a), but we 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 six 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, Chen, and Fryzlewicz (2016a). 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 dataset. 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, Chen, and Fryzlewicz (2016a), 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 are the true changepoint locations, and n_s is 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

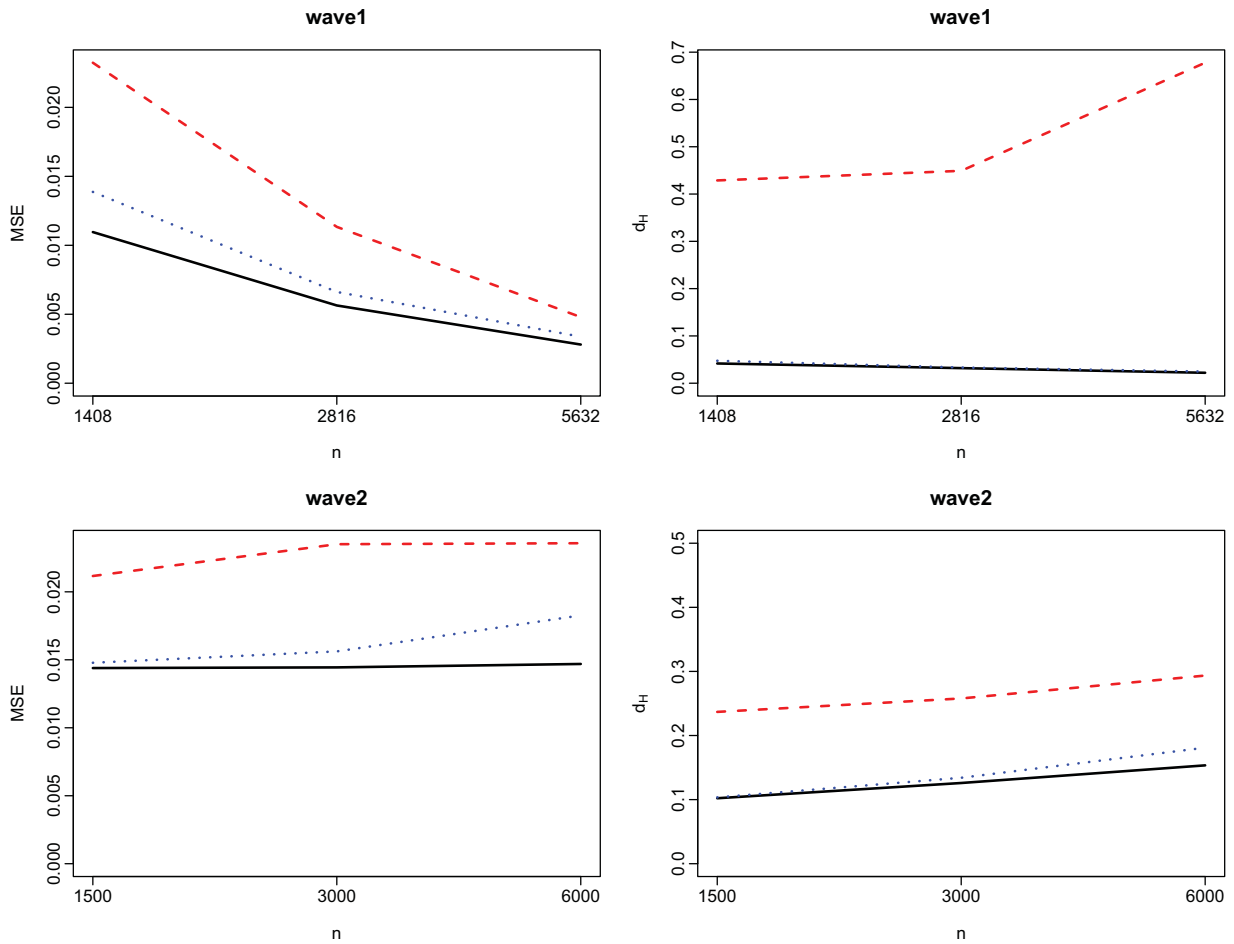


Figure 3. Results for CPOP (black solid line), NOT (blue dotted line), and trend-filtering (red dashed line) for wave1 (top row) and wave2 (bottom row). We give results for mean square error of the estimate of the mean (left-hand column) and for the accuracy of the estimates of the changepoint locations,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

segment. The smaller d_H the better the estimates of the changepoints, with $d_H = 0$ meaning that all changepoints are detected without error, and no other changepoints are estimated.

First, we analyze 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 number of changepoints: NOT correctly identifies the number of changepoints for all 600 simulated datasets, 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. While 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 toward 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 datasets of length varying from 1000 to 10,000, 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 datasets 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

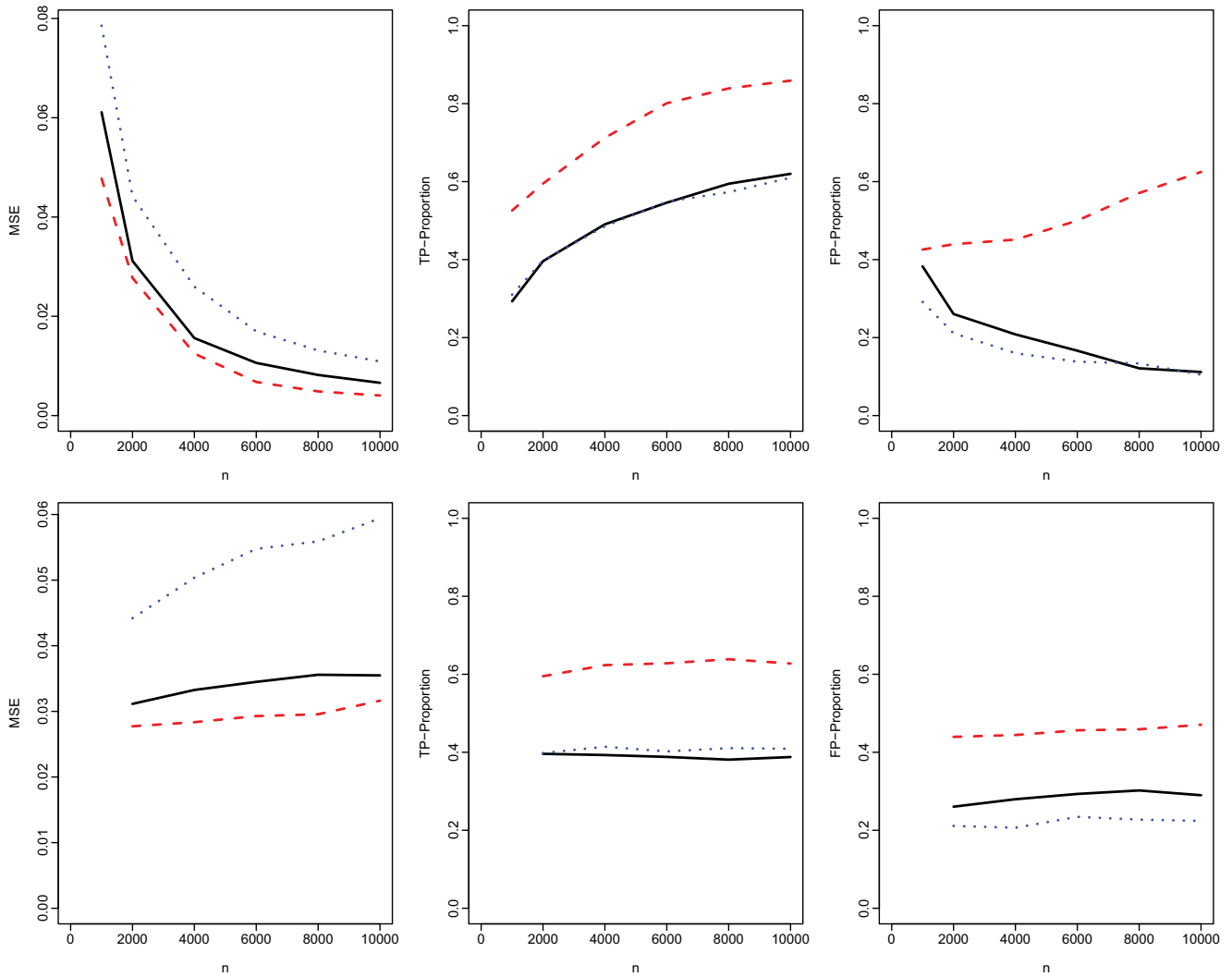


Figure 4. Results for CPOP (black solid line), NOT (blue dotted line), and trend-filtering (red dashed line) for the Random scenario with a fixed number of changepoints (top row) and a fixed segment length (bottom row). We give results for mean square error of the estimate of the mean (left-hand column) 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 datasets for each case and each value of n .

sources of error are masked in the measure d_H , and thus we summarize 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 the proportion of actual changepoints detected, and the false positive 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 is 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). While 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 analyze have been collected by Ryu, Berry, and Berg (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 analyze come

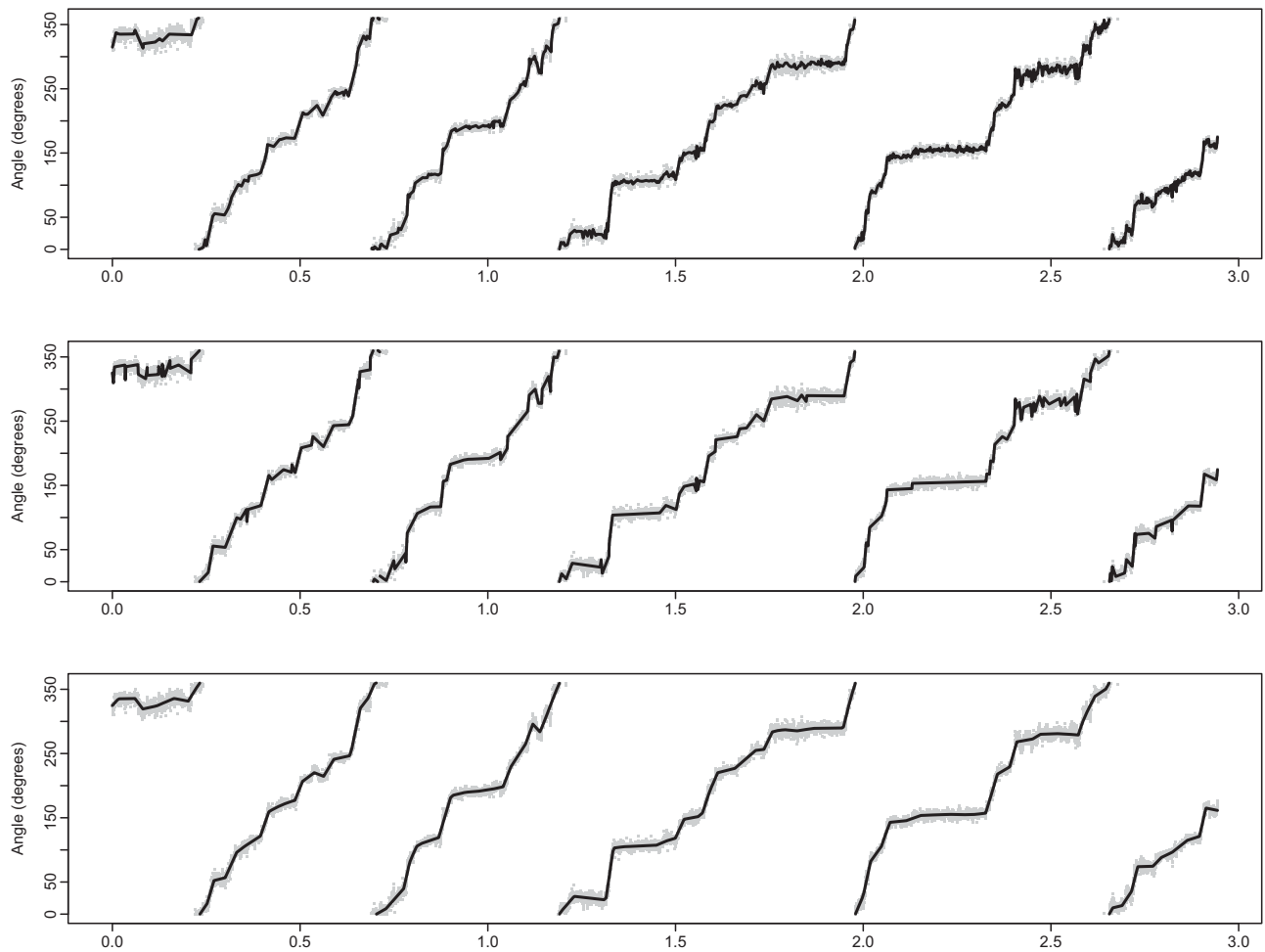


Figure 5. Time series of angular position (data from Sowa et al. 2005) and example fits obtained by NOT (top); CPOP (middle); and trend-filtering (bottom). The fits by NOT and CPOP are ones which give a similar fit to the data; the NOT fit has 784 changepoints and the fit from CPOP just 182. The fit from trend-filtering has 278 changepoints, though many correspond to very small changes in slope, and a substantially worse fit to the data (see the 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).

from Sowa et al. (2005) and are 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 minimizing the residual sum of squares plus a penalty times the number of changepoints, using the PELT algorithm (Killick, Fearnhead, and Eckley 2012). In all cases, we varied the penalty value using the CROPS algorithm (Haynes, Eckley, and Fearnhead 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 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 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.

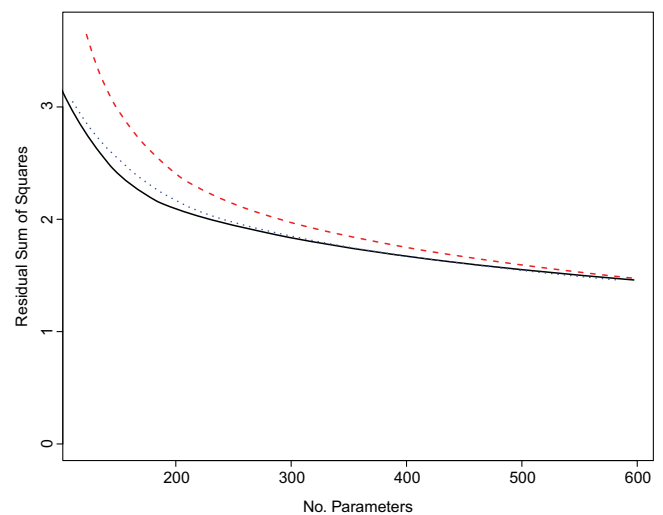


Figure 6. Accuracy of fits of data shown in Figure 5 by a piecewise constant mean (red dashed line), a continuous piecewise linear mean (black full line), and a discontinuous piecewise linear mean (blue dotted line). For each type of line, we found the best segmentation, in terms of minimizing the residual 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.

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 analyze 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 minimized (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 nonzero if we round the slopes, in degrees, to three 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, minimizing the square error loss of the fit to the data plus an L_0 penalty requires specifying the penalty for adding a changepoint. While 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 minimizing the penalized cost over a range of penalties, using the CROPS algorithm (Haynes, Eckley, and Fearnhead 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 generalize beyond the univariate case. Even if not, it may be possible to develop efficient algorithms that implement an approximate version of functional pruning.

Supplementary Materials

CPOP_Supplementary.pdf Appendices with proofs and pseudo code for CPOP.

cpop_0.0.3.tar.gz R package with CPOP code.

Rcode.tar.gz R functions used in the simulation study; and data from Section 5.

Acknowledgments

The authors 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 Baranowski, Yining Chen and Piotr Fryzlewicz for advice on using NOT.

Funding

This work was supported by EPSRC grants EP/N031938/1 (StatScale) and EP/H023151/1 (STOR-i).

References

- Arlot, S., and Massart, P. (2009), “Data-Driven Calibration of Penalties for Least-Squares Regression,” *Journal of Machine Learning Research*, 10, 245–279. [274]
- Aston, J. A., and Kirch, C. (2012), “Evaluating Stationarity via Change-Point Alternatives with Applications to fMRI Data,” *The Annals of Applied Statistics*, 6, 1906–1948. [265]
- Baranowski, R., Chen, Y., and Fryzlewicz, P. (2016a), “Narrowest-Over-Threshold Detection of Multiple Change-points and Change-point-like Features,” ArXiv:1609.00293. [265,267,269,270]
- (2016b), “Not: Narrowest-Over-Threshold Change-Point Detection,” R Package Version 1.0. [270]
- Chen, X., and Berg, H. C. (2000), “Solvent-Isotope and pH Effects on Flagellar Rotation in Escherichia Coli,” *Biophysical Journal*, 78, 2280–2284. [272]
- 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. [267]
- Fryzlewicz, P. (2014), “Wild Binary Segmentation for Multiple Change-Point Detection,” *The Annals of Statistics*, 42, 2243–2281. [265,267]
- Futschik, A., Hotz, T., Munk, A., and Sieling, H. (2014), “Multiscale DNA Partitioning: Statistical Evidence for Segments,” *Bioinformatics*, 30, 2255–2262. [265]
- Goldberg, N., Kim, Y., Leyffer, S., and Veselka, T. D. (2014), “Adaptively Refined Dynamic Program for Linear Spline Regression,” *Computational Optimization and Applications*, 58, 523–541. [266]

- Hampel, F. R. (1974), "The Influence Curve and its Role in Robust Estimation," *Journal of the American Statistical Association*, 69, 383–393. [267]
- 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. [273,274]
- Hocking, T., Rigai, 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*, pp. 172–180. [274]
- Hocking, T. D., Rigai, G., Fearnhead, P., and Bourque, G. (2017), "A Log-Linear Time Algorithm for Constrained Change-point Detection," ArXiv.1703.03352. [274]
- Horner, A., and Beauchamp, J. (1996), "Piecewise-Linear Approximation of Additive Synthesis Envelopes: A Comparison of Various Methods," *Computer Music Journal*, 20, 72–95. [266]
- Jackson, B., Scargle, J. D., Barnes, D., Arabhi, S., Alt, A., Gioumoussis, 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. [266]
- Jewell, S., Hocking, T. D., Fearnhead, P., and Witten, D. (2018), "Fast Non-convex Deconvolution of Calcium Imaging Data," ArXiv.1802.07380. [283]
- Killick, R., Eckley, I. A., Ewans, K., and Jonathan, P. (2010), "Detection of Changes in Variance of Oceanographic Time-Series using Change-point Analysis," *Ocean Engineering*, 37, 1120–1126. [265]
- Killick, R., Fearnhead, P., and Eckley, I. A. (2012), "Optimal Detection of Change-points with a Linear Computational Cost," *Journal of the American Statistical Association*, 107, 1590–1598. [266,268,273]
- Kim, S. J., Koh, K., Boyd, S., and Gorinevsky, D. (2009), " l_1 Trend Filtering," *SIAM Review*, 51, 339–360. [269]
- 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, 33–59. [266]
- Levy-leduc, C., and Harchaoui, Z. (2008), "Catching Change-Points with Lasso," in *Advances in Neural Information Processing Systems*, pp. 617–624. [271]
- Maidstone, R. (2016), "Efficient Analysis of Complex Change-point Problems," Ph.D. dissertation, Lancaster University, United Kingdom, available at <http://eprints.lancs.ac.uk/83055/>. [269]
- Maidstone, R., Hocking, T., Rigai, G., and Fearnhead, P. (2017), "On Optimal Multiple Change-point Algorithms for Large Data," *Statistics and Computing*, 27, 519–533. [266,268]
- 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, 1245–1255. [265]
- Raimondo, M. (1998), "Minimax Estimation of Sharp Change Points," *Annals of Statistics*, 26, 1379–1397. [267]
- R Core Team (2017), *R: A Language and Environment for Statistical Computing*, Vienna, Austria: R Foundation for Statistical Computing. [269]
- Rigai, 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, 180–205. [268]
- 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. [272]
- Schwarz, G. (1978), "Estimating the Dimension of a Model," *The Annals of Statistics*, 6, 461–464. [267]
- Scott, A. J., and Knott, M. (1974), "A Cluster Analysis Method for Grouping Means in the Analysis of Variance," *Biometrics*, 30, 507–512. [265,269]
- Sowa, Y., and Berry, R. M. (2008), "Bacterial Flagellar Motor," *Quarterly Reviews of Biophysics*, 41, 103–132. [272]
- 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, 1043–1051. [272]
- 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, 916–919. [265,266,272,273]
- Tibshirani, R. J. (2014), "Adaptive Piecewise Polynomial Estimation via Trend Filtering," *The Annals of Statistics*, 42, 285–323. [269]
- Tomé, A. R., and Miranda, P. M. A. (2004), "Piecewise Linear Fitting and Trend Changing Points of Climate Parameters," *Geophysical Research Letters*, 31, L02207. [266]
- Wang, T., and Samworth, R. J. (2018), "High Dimensional Change Point Estimation via Sparse Projection," *Journal of the Royal Statistical Society, Series B*, 80, 57–83. [267]
- 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, 20140638. [266]
- Yao, Y.-C. (1988), "Estimating the Number of Change-Points via Schwarz' Criterion," *Statistics & Probability Letters*, 6, 181–189. [266]
- 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. [267,269]