Efficient search methods for high dimensional time-series

Lawrence Bardwell, B.Sc.(Hons.), M.Res

Submitted for the degree of Doctor of Philosophy

at Lancaster University.

September 2017

STOR-i
excellence with impact

Abstract

This thesis looks at developing efficient methodology for analysing high dimensional time-series, with an aim of detecting structural changes in the properties of the time series that may affect only a subset of dimensions.

Firstly, we develop a Bayesian approach to analysing multiple time-series with the aim of detecting abnormal regions. These are regions where the properties of the data change from some normal or baseline behaviour. We allow for the possibility that such changes will only be present in a, potentially small, subset of the time-series. A motivating application for this problem comes from detecting copy number variation (CNVs) in genetics, using data from multiple individuals.

Secondly, we present a novel approach to detect sets of most recent changepoints in panel data which aims to pool information across time-series, so that we preferentially infer a most recent change at the same time point in multiple series.

Lastly, an approach to fit a sequence of piece-wise linear segments to a univariate time series is considered. Two additional constraints on the resulting segmentation are imposed which are practically useful: (i) we require that the segmentation is robust to the presence of outliers; (ii) that there is an enforcement of continuity between the linear segments at the

changepoint locations. These constraints add significantly to the computational complexity of the resulting recursive solution. Several steps are investigated to reduce the computational burden.

Acknowledgements

First and foremost I would like to thank my supervisors, Idris Eckley and Paul Fearnhead. It has been an honour and privilege to have worked with such fine researchers (and people) over the past few years. I am very grateful for all the time and effort they have spent on my project whilst showing unbounded patience and knowledge.

I am very grateful for the financial support provided by the EPSRC and BT. I would like to thank my industrial supervisor Martin Spott for all the helpful conversations, guidance and making my visits to BT very useful and enjoyable.

Thanks also to all the staff and students at the STOR-i Centre for Doctoral Training for providing such an enjoyable working environment of which I am very lucky to have been part of. Jonathan Tawn deserves a special thank you for always going above and beyond for everyone at STOR-i.

Finally, to my family and Kirsti, whose continual support, encouragement and love throughout this process helped me get through it relatively unscathed, words cannot express how grateful I am.

Declaration

I declare that the work in this thesis has been done by myself and has not been submitted elsewhere for the award of any other degree.

Lawrence Bardwell

Chapter 4 has been accepted for publication as Bardwell, L., and Fearnhead, P. (2017). Bayesian Detection of Abnormal Segments in Multiple Time Series. *Bayesian Analysis.*

Chapter 5 has been submitted to *Technometrics* as Bardwell, L., Eckley, I., Fearnhead, P., Smith, S., and Spott, M. (2017). Most recent changepoint detection in Panel data.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The work presented in this thesis considers the detection of changepoints in multivariate time series. There has been a great deal of work in recent years on changepoint detection in univariate time series and the development of many efficient algorithms to solve these problems. However there is much less work on the corresponding problem for multivariate time series due to the increased complexity in modelling such data, together with substantial computational complexity.

The main contribution of this thesis is the development of methodology to detect changepoints in high-dimensional time series where we assume that only a subset of the dimensions are affected by each changepoint. There can be significant benefits in solving problems like this as weaker changes that may not be detectable in the univariate case can now be detected by pooling information across the dimensions of the series.

Our work has been motivated by data sets from Genetics, Finance and the Telecommunications sector. These data sets are explained in their respective chapters, however, the Telecommunications data is considered in two chapters and its structure is somewhat com-

plex so we describe it briefly below.

1.1 Telecommunications event data

This data set contains information about the number of events that occur in a telecommunications network per week. The time series in Figure 1.1.1 shows the number of events that occur over the entire network per week for 175 weeks.

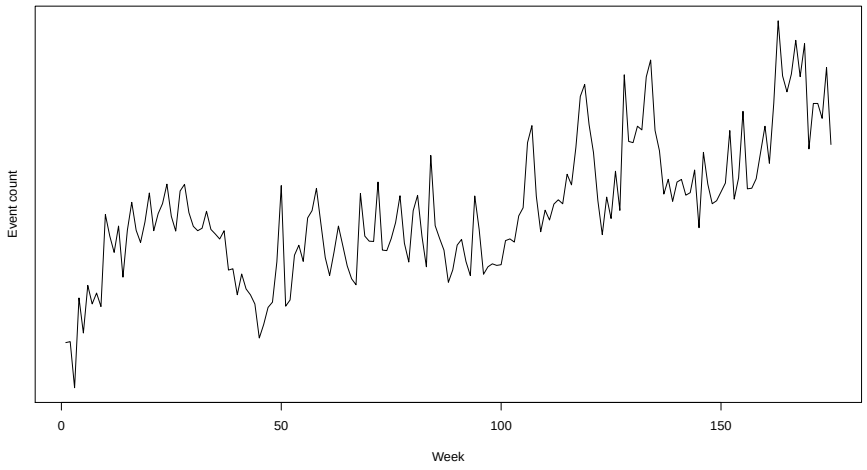


Figure 1.1.1: The number of events that occur per week over the entire telecommunications network measured over a three and a half year time period (175 weeks).

Whenever an event occurs it is automatically logged in a database with the time it occurs. In addition, many other attributes about the event are also recorded including the geographical location of where it occurred on the network and many other attributes known as event types. We can subdivide the number of events per week into different classes. For example the number of events that occur in each of ten Regions of the UK. These regions partition the UK so if we take the pointwise sum over all regions per week we get the total number i.e. the time series in Figure 1.1.1. We can subdivide further and for example look at the number of

events in each Region with a given event type combination. The series are shown in Figure 1.1.2 where there are 80 possible combinations, however, for some combinations no events occur throughout the entire period and these are shown by horizontal lines in the centre of the grid square.

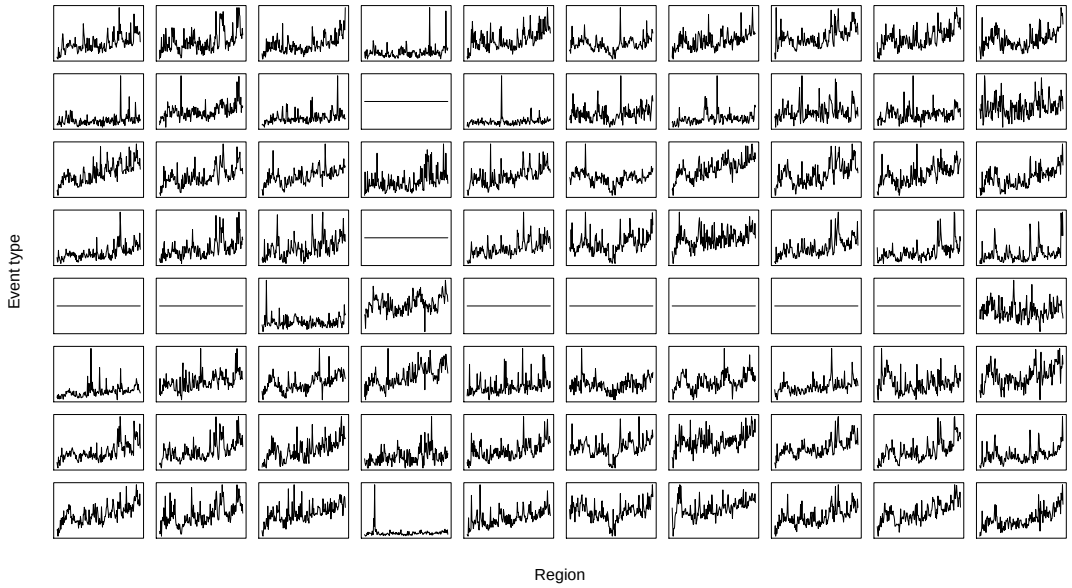


Figure 1.1.2: A grid showing all possible combinations of Regions and Event types and the number of events that occur per week for the combination considered.

The primary goal here is to understand the most recent behaviour of the data by analysing all series in the hierarchical time series and fitting a changepoint model to the data.

1.2 Thesis structure

We begin in Chapters 2 and 3 by reviewing existing literature from both the univariate and multivariate changepoint detection settings respectively. The different problem formulations and detection methods are discussed and special attention is reserved for those state of the art methods which we use as benchmark methods later in the thesis.

In Chapter 4 we present a novel Bayesian approach to analysing multiple time-series with the aim of detecting abnormal regions. These are regions where the properties of the data change from some normal or baseline behaviour. We allow for the possibility that such changes will only be present in a, potentially small, subset of the time-series. We develop a general model for this problem, and show how it is possible to accurately and efficiently perform Bayesian inference, based upon recursions that enable independent sampling from the posterior distribution. A motivating application for this problem comes from detecting copy number variation (CNVs), using data from multiple individuals. Pooling information across individuals can increase the power of detecting CNVs, but often a specific CNV will only be present in a small subset of the individuals. We evaluate the Bayesian method on both simulated and real CNV data, and give evidence that this approach is more accurate than a recently proposed method for analysing such data.

In Chapter 5 we present a novel approach to detect sets of most recent changepoints (MRC) in panel data. A panel is made up of a number of univariate time series and our method is described firstly for the univariate case where finding the most recent changepoint (prior to the end of the data) is straightforward. We then extend this to panel data as there may be a number of MRC's due to different subsets of the series that make up the panel having different behaviours. These MRC's affect disjoint subsets of the series as any one series can only have one MRC. We seek a parsimonious model for the data that gives a reasonable number of MRC's and doesn't over-fit the data. Focusing on the most recent changepoints makes sense for forecasting & understanding recent behaviour of the panel and the specific subsets of series within it. The possibility that such changes will only be present in a, potentially small, subset of the series which many existing methodologies do not account for is a key

consideration in this work. This involves pooling information across individual series of the panel to increase the power of detection. We develop a general model for this problem, and show how it is possible to accurately and efficiently perform inference. We present simulations showing that this approach is more accurate than other proposed method for analysing such data. Two real data sets are considered, regarding the number of events that occur over time in a telecommunications network and a data set from the field of Corporate finance where our method gives insights into the different subsets of series that change.

In Chapter 6 we present a novel approach to detect changepoints and fit a sequence of piecewise linear segments to a univariate time series. Two additional constraints on the resulting segmentation are imposed which are practically useful: (i) we require that the segmentation is robust to the presence of outliers; (ii) that there is an enforcement of continuity between the linear segments at the changepoint locations. These constraints have been considered separately in the context of changepoint detection before but here we develop a set of recursions to segment a series with both criteria. Solving these recursions exactly proves to have a computational cost that is exponential in the length of the data so we apply two pruning techniques, one of which is heuristic that enables us to evaluate the recursions in a reasonable time. We present simulations showing that this approach performs well in practice and describe the improvements over using simpler models such as Ordinary Least Squares.

Chapter 2

Changepoint detection for univariate time series

This chapter focuses on changepoint detection in univariate time series. Many of the techniques used to analyse multivariate time series described later in this thesis are based upon these univariate methods. Thus understanding them allows us to see their limitations and how they can possibly be extended.

Firstly, in Section 2.2, the changepoint problem is posed as an optimisation problem. Several different formulations are discussed as well as different solution methods. One of the main concerns is the computational efficiency of the solution methods. We describe several techniques from the literature which are used to reduce the computational complexity of segmenting a time series by an order of magnitude.

In Section 2.3 we then look at modelling changepoints using the Bayesian paradigm, this allows more quantification of uncertainty about the locations of changes. However, performing inference for time series of moderate length proves challenging and we find that an acceptably

efficient algorithm comes at the cost of performing approximate inference.

Many of the methods mentioned in this Chapter are implemented in the Changepoint `R` package available on CRAN [Killick and Eckley, 2014].

2.1 Notation

Assume we have an ordered sequence of data of length n which we denote $y_{1:n} = (y_1, y_2, \ldots, y_n)$. Firstly for simplicity consider a single changepoint in the data. Then for some time point $\tau \in \{2, \ldots, n-1\}$, we can split the data into two segments, with the data in the first segment being $y_{1:\tau}$, and that in the second $y_{(\tau+1):n}$. A changepoint model assumes a common model for data within the same segment but allows different models for data in different segments. Possibly the simplest example is a change in mean model with Normally distributed residuals with some variance σ^2. Assume the data in the first segment $y_{1:\tau}$ has a common mean of μ_1 and that data in the second segment $y_{(\tau+1):n}$ has a different common mean of μ_2. The distribution for this data is as follows $Y_i \sim \mathcal{N}(\mu_1, \sigma^2)$ for $i \in \{1, \ldots, \tau\}$ and $Y_j \sim \mathcal{N}(\mu_2, \sigma^2)$ for $j \in \{\tau + 1, \ldots, n\}$.

To extend this to the multiple changepoint case, assume there are m changepoints with locations $\boldsymbol{\tau}_{1:m} = (\tau_1, \tau_2, \ldots, \tau_m)$ where $\tau_i < \tau_j$ iff $i < j$. For notational convenience we define $\tau_0 = 0$ and $\tau_{m+1} = n$. These m changepoints segment the data into $m + 1$ segments, with the data in the ith segment being $y_{(\tau_{i-1}+1):\tau_i}$.

We have described a change in mean model, however, many other types of changes can be modelled. In the three time series shown in Figure 2.1.1 we can see a change in mean model, a change in variance model and a change in regression model on the right.

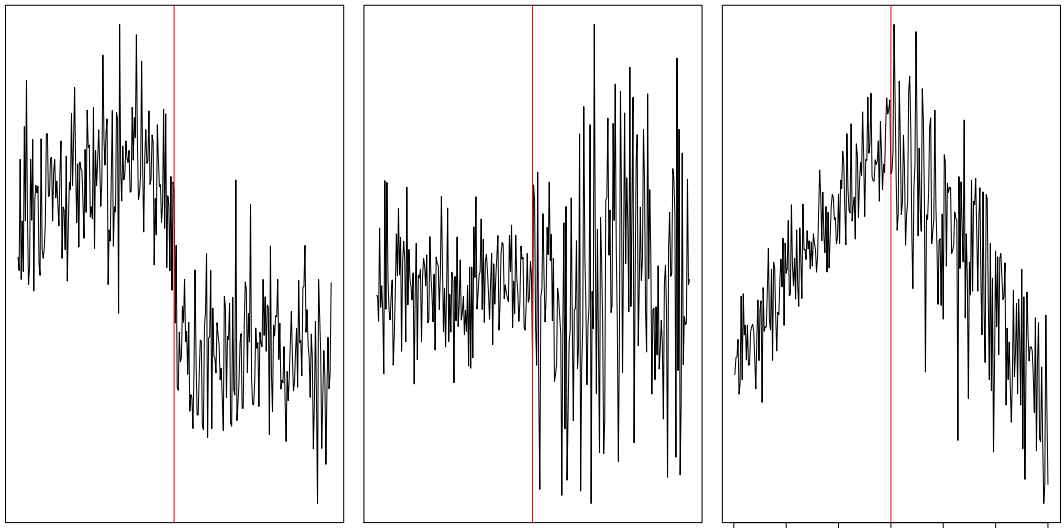


Figure 2.1.1: Three time series having a single change. A mean change on the left, variance change in the middle and change in regression on the right.

2.2 Optimisation methods for changepoint detection

The problem of finding changepoints or, equivalently segmenting a time series into contiguous segments can be formulated as an optimisation problem. This is a popular formulation of the mutiple changepoint detection problem and is the one that the methods in this thesis build on. This formulation gives rise to a relatively simple set of recursions that can be solved exactly to give the location of the changepoints in a time series.

Firstly we discuss two different formulations of the optimisation problem in Sections 2.2.1 and 2.2.2. Efficient methods to perform inference on these two formulations are described in Section 2.2.3.

We only concern ourselves with the case where the data in individual segments are assumed to have some known parametric distribution. However, the formulations we cover can easily be extended to allow non-parametric models for the data within each segment, see for example Haynes et al. [2017b].

A penalised cost approach to detecting changepoints involves introducing a cost associated with each putative segment. This cost is often derived by modelling the data within a segment, and then setting the cost to be proportional to minus the maximum likelihood value for fitting that model to a segment of data. If our model for data in a segment is that they are Independently and identically distributed (IID) with some density $f(y|\theta)$, where θ is a segment-specific parameter, then we can define a cost for a segment $y_{s:t}$ as

$$\mathcal{C}(y_{s:t}) = -2 \max_{\theta} \sum_{i=s}^{t} \log f(y_i|\theta).$$

To make this idea concrete we give an example of a cost function used to model changes in mean. A simple model is that the data in a segment are IID Gaussian with common known variance, σ^2, and segment specific mean, θ. In this case we get

$$\mathcal{C}(y_{s:t}) = -2 \max_{\theta} -\frac{1}{2\sigma^2} \sum_{i=s}^{t} (y_i - \theta)^2 = \frac{1}{\sigma^2} \sum_{i=s}^{t} \left(y_i - \frac{\sum_{j=s}^{t} y_j}{t - s + 1} \right)^2. \tag{2.2.1}$$

Once we have defined a segment cost, we then define a cost for a segmentation as the sum of the segment costs for that segmentation.

To segment the data, and find the changepoints, we then want to minimise this cost over all segmentations. Whenever a new changepoint is added into the model the overall cost of the segmentation decreases. Therefore if we just directly minimise this cost the resulting segmentation would include a changepoint at every time point. We obviously want to avoid such over-fitting which implies we have a trade-off between a reduced cost and a parsimonious model. There are two ways in which we can achieve this trade off, the first is to constrain the

model to some maximum number of changepoints and the second is to introduce a penalty value each time a changepoint is added. This means that the overall 'best' model will provide a good fit using a reasonable amount of changepoints.

The first approach to overcome over-fitting is to consider a constrained optimisation problem where the segmentation is constrained to have a certain number of changepoints (in this case m)

$$Q_m(y_{1:n}) = \min_{\tau_{1:m}} \left\{ \sum_{i=1}^{m+1} \left[\mathcal{C}(y_{(\tau_{i-1}+1):\tau_i}) \right] \right\}. \tag{2.2.2}$$

Usually the number of changes m is unknown so that the number of changes is estimated by minimising the constrained cost plus some penalty which is a function of the number of changes

$$\min_m \{Q_m(y_{1:n}) + \beta f(m)\}. \tag{2.2.3}$$

The general form of the penalty function is $\beta f(m)$. This can be broken down into two parts. Firstly $f(m)$, which is a function of the number of changepoints m. This function is usually chosen to be linear in m. Secondly, β is the term used for model selection (the number of changepoints to be added) which is usually an information theoretic measure. Choices for this term are described in more detail in Section 2.2.4.

Solving the constrained problem is computationally intensive and can be carried out using the Segment Neighbourhood search algorithm [Auger and Lawrence, 1989] which we describe in Section 2.2.1. A more efficient approach which eliminates the need to select an m for the

constrained problem is known as the penalised optimisation problem. This requires us to take $f(m)$ to be linear in m.

So if $f(m) = m$ and for some $\beta > 0$, solving (2.2.3) is equivalent to solving

$$\min_{m,\tau_{1:m}} \left\{ \sum_{i=1}^{m+1} \left[\mathcal{C}(y_{(\tau_{i-1}+1):\tau_i}) \right] + \beta f(m) \right\}, \tag{2.2.4}$$

which in turn can be written as

$$\min_{m,\tau_{1:m}} \sum_{i=1}^{m+1} \left[\mathcal{C}(y_{(\tau_{i-1}+1):\tau_i}) + \beta \right]. \tag{2.2.5}$$

To solve this problem the Optimal Partitioning algorithm described in Section 2.2.2 can be used. Variants of this method are proposed in Section 2.2.3 to decrease the computational cost.

In the following sections we consider the solution of these problems. Both the Segment Neighbourhood and Optimal partitioning methods depend on us being able to break the original problem which is difficult to solve into sub problems which are progressively easier to solve via a relatively simple set of recursions. This approach is known as Dynamic programming [Bellman, 1957].

2.2.1 Segment Neighbourhood

The Segment Neighbourhood search algorithm (SN) [Auger and Lawrence, 1989] was developed to solve the constrained optimisation problem described in (2.2.2). In (2.2.2) we defined $Q_m(y_{1:t})$ as the minimum cost of putting m changepoints into the segment of data $y_{1:t}$. In

the SN recursions we optimise for m changepoints based on the optimal solution for $m-1$ changepoints. We do this by conditioning on the last changepoint being at s where $s < t$. Then we can relate $Q_m(y_{1:t})$ to the segmentation of $y_{1:s}$ with $m-1$ changepoints $Q_{m-1}(y_{1:s})$

$$\begin{aligned} Q_m(y_{1:t}) &= \min_{\tau_{1:m}} \left\{ \sum_{i=1}^{m+1} \left[\mathcal{C}(y_{(\tau_{i-1}+1):\tau_i}) \right] \right\} \\ &= \min_{s \in \{\tau_{m-1}, \ldots, t-1\}} \left[Q_{m-1}(y_{1:s}) + \mathcal{C}(y_{(s+1):t}) \right]. \end{aligned} \tag{2.2.6}$$

Solving this recursion proceeds by going forwards through the data. We need to specify a maximum number of changepoints that we want to consider, say M. We then compute the cost for all possible segmentations, with between 0 and M changepoints.

For each $t \in 1, \ldots, n$ we calculate (2.2.6) for all possible change locations, $s \in m, \ldots, t-1$. For the full n data points this has computational time of $\mathcal{O}(n^2)$. This is repeated M times, therefore SN has an overall computational cost of $O(Mn^2)$. If, as the observed data increases, the number of changepoints increases linearly, then $M = \mathcal{O}(n)$ and the method will have a computational cost that is cubic in n. This is prohibitive if n is large. One advantage to the SN approach is the ability to use an arbitrary penalty of the form, $\beta f(m)$ where $f(m)$ does not have to be linear in m, unlike the Optimal Partitioning method.

2.2.2 Optimal Partitioning

Optimal Partitioning (OP) is a dynamic programming approach to solving (2.2.5) and was first developed in Jackson et al. [2005]. The idea behind OP is to recursively condition on the last changepoint prior to a given time until the end of the data is reached. As an example if the last changepoint prior to time t is at s, then the optimal cost of the segmentation up

to t is the optimal cost up to s plus the cost of adding a segment from $s+1$ to t (with penalty added as well). Of course we do not know the value of s but we can calculate it via minimising over a set of candidate changepoints for each t.

In order to do this we need to be able to calculate segment costs independently of other segments. This implies that there can be no dependence between the parameters in different segments.

More formally let $\mathcal{T}_t = \{\boldsymbol{\tau} : 0 = \tau_0 < \tau_1 < \ldots < \tau_m < \tau_{m+1} = t\}$ be a vector of all possible segmentations with m changepoints and let $F(t)$ denote the minimisation from (2.2.5) for data $y_{1:t}$

$$F(t) = \min_{\boldsymbol{\tau} \in \mathcal{T}_t} \left\{ \sum_{i=1}^{m+1} \left[\mathcal{C}(y_{(\tau_{i-1}+1):\tau_i}) + \beta \right] \right\}.$$

Then we can devise a recursion for $F(t)$ as

$$\begin{aligned} F(t) &= \min_s \left\{ \min_{\boldsymbol{\tau} \in \mathcal{T}_s} \sum_{i=1}^{m} \left[\mathcal{C}(y_{(\tau_{i-1}+1):\tau_i}) + \beta \right] + \mathcal{C}(y_{(s+1):t}) + \beta \right\}, \\ &= \min_s \left\{ F(s) + \mathcal{C}(y_{(s+1):t}) + \beta \right\}. \end{aligned} \tag{2.2.7}$$

Note that the argmin we find in (2.2.7) is the location of the change prior to time t. We therefore record these argmin and to find the full set of changepoints we look back from the end of the data until we encounter an argmin of 0. The pseudocode for this method is shown in Algorithm 1. To initialise the recursion we define $F(0) = -\beta$ so that $F(1) = \mathcal{C}(y_1)$.

Algorithm 1: Optimal Partitioning algorithm.

Input: A data set $y_{1:n} = (y_1, y_2, \ldots, y_n)$.

A cost function $\mathcal{C}(\cdot)$ dependent on the data.

A penalty term β.

Initialize: Let $n =$ the length of the data and set $F(0) = -\beta$, $cp(0) = NULL$.

for $t = 1$ **to** n **do**

1. Calculate $F(t) = \min_{0 \leq s < t} \left[F(s) + \mathcal{C}(y_{(s+1):t}) + \beta \right]$.
2. Let $\tau = \arg\min_{0 \leq s < t} \left[F(s) + \mathcal{C}(y_{(s+1):t}) + \beta \right]$.
3. Set $cp(\tau) = (cp(\tau), \tau)$

end

Output: The changepoints recorded in $cp(n)$.

In Step 1. of Algorithm 1 for each time step t, we must calculate $F(t)$ by minimising over all the integers between 0 and $t-1$. Thus for each t we have to calculate t different expressions and find the minimum. If we have a time series of length n the total number of operations in the full algorithm is of the order of $\sum_{t=1}^{n} t \sim n^2$. The next section explores how we can make OP more efficient by removing some integers from consideration.

2.2.3 Pruning

Pruning is a technique applied in the solution of both the SN and OP methods. It greatly increases the efficiency of both the methods while retaining the exact nature of the solution. There are two types of pruning, inequality pruning which was first developed by Killick et al. [2012] and functional pruning which was introduced by Rigaill [2015]. Both of these pruning techniques can be applied to both the SN and OP methods [Maidstone et al., 2016b]. The

simplest exposition and most efficient algorithm to describe is the OP method.

The intuition behind pruning is quite simple if we consider the full OP method.

In the OP algorithm to find $F(t)$ we condition on the most recent changepoint s, prior to t. This involves searching through all the integers from 0 to $t-1$ in order to find the best location in which to place the changepoint. Denote the set of candidate changepoints that we must consider at time t as $R_t = \{0, 1, \ldots, t-1\}$. Searching through this entire set of candidate changepoints R_t at every time step t can be extremely wasteful. For example if we know that the most recent changepoint prior to t is at s then when looking for the most recent changepoint prior to time $t+1$, we should not have to search through the entire list again.

This is where the concept of pruning comes in, we 'prune' the set R_t removing those candidate changepoints that can never be optimal in the future and are left with a smaller subset of R_t to propagate to the next time step. Intuitively this is clear, however we want the pruned OP method to remain optimal so we must be careful how we prune R_t.

We now consider two pruning methods that can be used to increase the computational efficiency of the OP method whilst ensuring the global minimum of (2.2.5) is still found.

Inequality pruning

The first method of pruning from Killick et al. [2012] is based on an inequality which determines whether a candidate changepoint can ever be optimal in the future. Once we have the inequality it is very easy and efficient to implement this within the OP method. The following theorem proves that for any pruning based on the given inequality we retain the optimal solution from the OP algorithm.

Theorem 2.2.1. *Killick et al. [2012] Assume there exists a constant K such that for all $t < s < T$,*

$$\mathcal{C}(y_{(s+1):t}) + \mathcal{C}(y_{(t+1):T}) + K \leq \mathcal{C}(y_{(s+1):T}). \tag{2.2.8}$$

Then, if

$$F(s) + \mathcal{C}(y_{(s+1):t}) + K \geq F(t) \tag{2.2.9}$$

holds, at a future time $T > t$, then s can never be the optimal last changepoint prior to T and so can be removed from all the candidate changepoint sets in the future with no effect on the exact solution.

Proof. For a proof of this see Section 5 of the supplementary material of Killick et al. [2012]. □

In condition (2.2.8) if we take cost functions that are based on the log likelihood, such as those described in (2.2.1) we can set $K = 0$.

Inequality pruning was combined with the OP algorithm in Killick et al. [2012] and the resulting algorithm is known as Pruned Exact Linear Time (PELT). We give pseudocode in Algorithm 2.

Algorithm 2: The PELT algorithm.

Input: A data set $y_{1:n} = (y_1, y_2, \ldots, y_n)$.

A cost function $\mathcal{C}(\cdot)$ dependent on the data.

A penalty term β.

A constant K that satisfies Equation (2.2.8).

Initialize: Let $n =$ the length of the data and set $F(0) = -\beta$, $cp(0) = NULL$, $R_1 = \{0\}$.

for $t = 1$ **to** n **do**

1. Calculate $F(t) = \min_{s \in R_t} \left[F(s) + \mathcal{C}(y_{(s+1):t}) + \beta\right]$.
2. Let $\tau = \arg\min_{s \in R_t} \left[F(s) + \mathcal{C}(y_{(s+1):t}) + \beta\right]$.
3. Set $cp(\tau) = (cp(\tau), \tau)$.
4. Set $R_{t+1} = \{s \in R_t : F(s) + \mathcal{C}(y_{(s+1):t}) + K < F(t)\} \cup \{t\}$

end

Output: The changepoints recorded in $cp(n)$.

Note how similar this is to the OP algorithm, where the only difference is the addition of the pruning inequality in Step 4. which reduces the size of the candidate changepoint set.

In Figure 2.2.1 we show a time series that undergoes a change in mean process which has been segmented with the OP and PELT methods which give identical locations for the changepoints.

Figure 2.2.2 shows the size of the set of candidate changepoints over time i.e. $|R_t|$ for each time t. For the OP method this line is shown in black and increases linearly with time as $|R_t| = t$. However, for PELT the corresponding line is in red and with the pruning step added to the OP method it increases at a much slower rate. Note also that large reductions in the size of the sets R_t occur at or near changepoints, shown by the vertical lines.

The PELT method is always faster than OP and in certain circumstances when the number of changepoints increases linearly with the length of the data PELT can be shown to be $\mathcal{O}(n)$. It performs well in cases where there are a large number of changepoints in the data as lots of pruning can occur throughout time so that the sets R_t cannot grow to be too large.

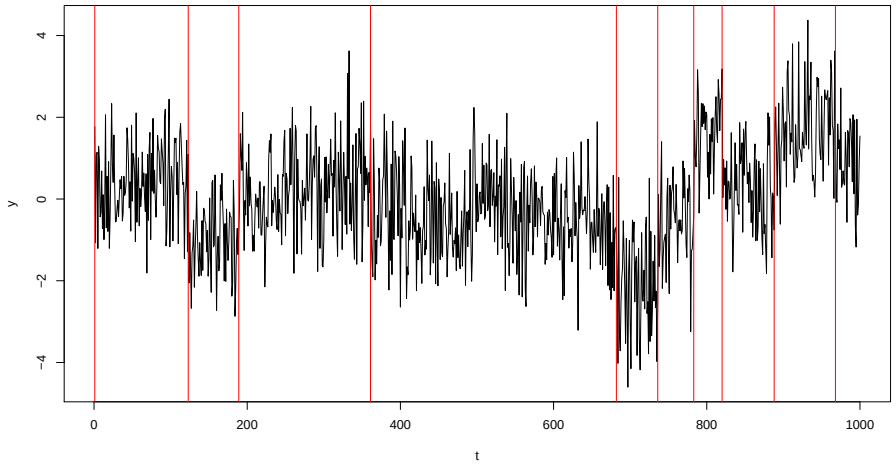


Figure 2.2.1: A time series of length 1000 with the changepoints found using the OP/PELT methods highlighted in red.

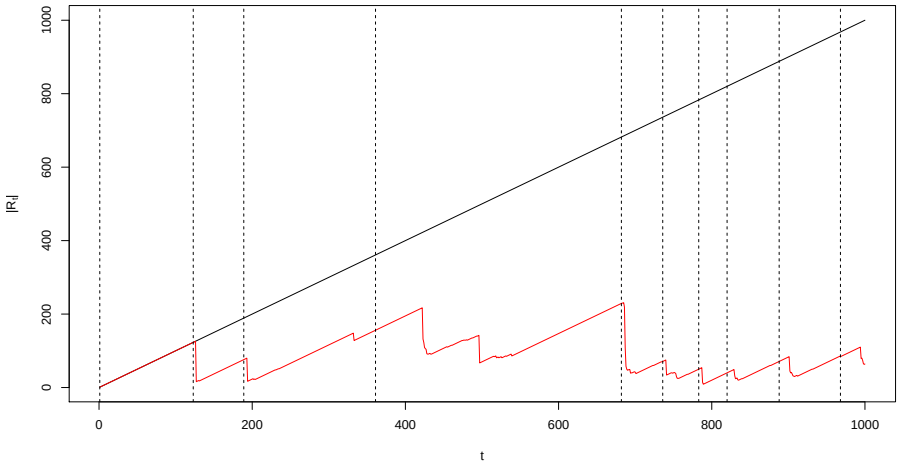


Figure 2.2.2: The number of candidate changepoints at each time point t for OP (in black) and PELT (in red). The vertical dashed lines give the location of the changepoints found (identical for both methods).

Functional pruning

Functional pruning, which was first developed by Rigaill [2015] was originally applied to the SN algorithm.

The basic idea behind functional pruning is that conditional on knowing the parameter for the current segment the best location for the most recent changepoint can be calculated easily. Thus only the most recent changepoints that are optimal for some value(s) of the parameter of the current segment need to be considered and the rest can be pruned.

Firstly we define the segmentation cost as a function of the segment parameter which we denote here as θ. A key assumption that underlies functional pruning is that the segmentation costs defined in (2.2.1) can be split into component parts $\gamma(y_i, \theta)$ that depend on some parameter which we minimise over to obtain the overall cost of a segment

$$\mathcal{C}(y_{s:t}) = \min_{\theta} \sum_{i=s}^{t} \gamma(y_i, \theta). \tag{2.2.10}$$

For the change in mean model for Gaussian data the function $\gamma(\cdot)$ is a quadratic

$$\gamma(y_i, \theta) = \frac{(y_i - \theta)}{\sigma^2}.$$

To develop a set of recursions we follow Maidstone et al. [2016b] and define new cost functions $Cost_t^{\tau}(\theta)$ as the minimal cost of segmenting the data $y_{1:t}$ with the most recent changepoint at τ and the parameter in the final segment being θ. This can be written as the optimal segmentation up to time τ plus the cost of a segment from $\tau + 1$ to t and the addition of the

penalty term for a new segment

$$Cost_t^\tau(\theta) = F(\tau) + \beta + \sum_{i=\tau+1}^{t} \gamma(y_t, \theta). \tag{2.2.11}$$

Given these cost functions $Cost_t^\tau(\theta)$ we can find $F(t)$ by minimising over both τ and θ by interchanging the order of minimisation

$$\begin{aligned}
\min_\tau \min_\theta Cost_t^\tau(\theta) &= \min_\tau \min_\theta \left[F(\tau) + \beta + \sum_{i=\tau+1}^{t} \gamma(y_i, \theta) \right], \\
&= \min_\tau \left[F(\tau) + \beta + \min_\theta \sum_{i=\tau+1}^{t} \gamma(y_i, \theta) \right], \\
&= \min_\tau \left[F(\tau) + \beta + C(y_{(\tau+1):t}) \right], \\
&= F(t).
\end{aligned}$$

This relationship is key as it shows that values of the potential last changepoint, τ, can be pruned whilst allowing for a varying θ.

Define the function $Cost_t^*(\theta)$ as the minimal cost of segmenting data $y_{1:t}$ conditional on the last segment having parameter θ

$$Cost_t^*(\theta) = \min_\tau Cost_t^\tau(\theta).$$

These functions are updated recursively over time, and then we use the relation $F(t) = \min_\theta Cost_t^*(\theta)$ to obtain the solution of the penalised minimisation problem. The recursions

for $Cost_t^*(\theta)$ are obtained by splitting the minimisation over τ into $\tau \leq t-1$ and $\tau = t$

$$\begin{aligned} Cost_t^*(\theta) &= \min\left\{\min_{\tau \leq t-1} Cost_t^\tau(\theta), Cost_t^t(\theta)\right\} \\ &= \min\left\{Cost_{t-1}^*(\theta) + \gamma(y_t, \theta), F(t) + \beta\right\}. \end{aligned}$$

To implement these recursions we need to be able to efficiently store and update $Cost_t^*(\theta)$. This is done by partitioning the space of possible θ values, into sets where each set corresponds to a value τ for which $Cost_t^*(\theta) = Cost_t^\tau(\theta)$. We then need to be able to update these sets, and store $Cost_t^\tau(\theta)$ just for each τ for which the corresponding set is non-empty. This results in us storing piece-wise quadratics for the change in mean model.

Functional pruning was combined with the OP algorithm in Maidstone et al. [2016b] resulting in the Functional Pruning Optimal Partitioning (FPOP) algorithm. It was shown that functional pruning always prunes at least as much as inequality based pruning. The FPOP algorithm is especially effective for long data sets which contain few changes for which the PELT algorithm performs poorly.

The FPOP algorithm has seen several modifications, these include the R(obust)-FPOP method [Fearnhead and Rigaill, 2016] which is an approach to changepoint detection that is robust to the presence of outliers. Also the CPOP algorithm [Maidstone et al., 2017a] was developed to fit a continuous piece-wise linear process to a data set.

Despite the computational advantages of FPOP there is one serious drawback in that when the segment parameter has dimension greater than one it is not practical to perform functional pruning. This is because the pruning of FPOP involves a line search over the values of the parameter of the current segment to find the set of most recent changepoints that are op-

timum for some value of this parameter. Performing this search is easy for a one-dimensional parameter, but computationally intractable for a higher-dimensional parameter. This would occur if we wanted to detect changes in both the mean and variance of a time series at the same time.

2.2.4 Penalty

The penalty function is used to give a parsimonious model that fits the data adequately using a reasonable number of changepoints and so avoids over-fitting.

The general penalty function is $\beta f(m)$, however, in practice the function that penalises the number of changepoints m, $f(m)$ is linear and increasing in m. For the OP and PELT algorithms we need to take $f(m) = m$ so that we can form the recursion in (2.2.7).

The penalty parameter β has received much more attention in the literature as we rely on this parameter for model selection. In the changepoint problem model selection is basically choosing the "optimal" number of changepoints to put in the data.

The Bayesian Information Criterion (BIC) [Schwarz, 1978] and the Akaike Information Criterion (AIC) [Akaike, 1974] are both widely used across statistics for model selection. However, their use in changepoint problems are not theoretically justified, as the likelihood functions involved do not satisfy the required regularity conditions. In Yao [1988] however, weak consistency results were established for estimating the number and position of changepoints, in normally distributed data using the BIC penalty.

If we have a data set of length n and the segment specific parameter that we model to be changing is of dimension p, i.e. for just the mean parameter μ, $p = 1$ or for both the mean

and variance (μ, σ) then $p = 2$. The AIC and BIC are then defined as

$$AIC = 2(p + 1)$$

$$BIC = (p + 1) \log n.$$

A modified version of the BIC was introduced in Zhang and Siegmund [2007b] which has theoretical justification for one specific model, data consisting of independent normally distributed observations with constant variance and piece-wise constant mean. These information criteria have also been developed for different and more complex models. For example Ding et al. [2016] derive a consistent BIC like criterion for a piece-wise auto-regressive model.

Adaptive procedures for penalty selection were first discussed in Lavielle [August 2005], this leads to the fuller treatment given by the CROPS method [Haynes et al., 2017a]. This paper describes an efficient approach to compare segmentations for different choices of the penalty β which takes values on some specified continuous range. This method allows us to evaluate the various segmentations and so to identify a suitable choice for the penalty given the specific data set we observe.

The CROPS method is not dependent on the model we assume for the data and can be used with any changepoint method that minimises the penalised cost. Computationally, it involves running the chosen algorithm for a set of penalty values. The number of different values that need to be taken is at most one more than the difference in the number of changepoints in the optimal segementations for the lowest and highest penalty values.

2.2.5 Binary Segmentation

The methods we have hitherto described are all exact, meaning they find the optimal solution to the optimisation problem in (2.2.4). Also of interest, and widely applied in practice are approximate methods that do not solve (2.2.4) exactly. These methods can generally be applied more widely to different models and can sometimes be much quicker than their exact counterparts.

A very simple and widely used approximate changepoint method is the Binary segmentation method introduced by Scott and Knott [1974] and Sen and Srivastava [1975]. Binary segmentation extends any single changepoint detection method to detect multiple changepoints by repeated application to different subsets of the data. It can be viewed as a greedy heuristic because it makes the locally optimal choice at each stage of the process.

The first step is to apply the chosen single changepoint detection method to the entire data set, if no changepoint is found then we are done. If a changepoint is detected, call this τ, then the data is split into two segments, $y_{1:\tau}$ and $y_{(\tau+1):n}$. We then apply the single changepoint method to the two segments and repeat iteratively. We stop when no more changepoints are detected.

Assume we are at a step of the algorithm where we consider the segment of data $y_{s:t}$, firstly we locate the best location for a single changepoint in this segment of data by minimising

$$\hat{\tau} = \underset{\tau \in \{s+1,\ldots,t-1\}}{\arg\min} \left[\mathcal{C}(y_{s:\tau}) + \mathcal{C}(y_{(\tau+1):t}) \right]. \tag{2.2.12}$$

Then given the location of this candidate changepoint, we test to see whether it improves

the fit of the model to warrant its inclusion by testing whether the following inequality holds

$$\mathcal{C}(y_{s:\hat{\tau}}) + \mathcal{C}(y_{(\hat{\tau}+1):t}) + \beta < \mathcal{C}(y_{s:t}). \tag{2.2.13}$$

If this inequality holds then $\hat{\tau}$ is added to the changepoints found and we then split the data into two at time $\hat{\tau}$ so that we have two resulting segments $y_{s:\hat{\tau}}$ and $y_{(\hat{\tau}+1):t}$. These two new segments are then analysed using the same procedure, splitting them recursively until the inequality in (2.2.13) does not hold.

Computationally, Binary segmentation is very efficient and is of the order of $\mathcal{O}(n \log n)$. The obvious drawback to its use is that it is only approximate in the sense that it does not find the global minimum of (2.2.5), and in certain situations it can break down as estimated changepoint locations are conditional on previously identified changepoints. Practically, however, it often performs very well and the estimated changepoint locations given by this method have been shown to be consistent in a particular sense described in Fryzlewicz [2014b].

There has been some work to develop variants of Binary segmentation in Fryzlewicz [2014b] and Olshen et al. [2004b].

One problem encountered in practice due to the iterative nature of the algorithm, is that Binary segmentation may fail to detect a small segment which lies inside a larger segment. A modified Binary segmentation algorithm, Circular Binary segmentation (CBS) [Olshen et al., 2004b], attempts to address this issue. CBS still iteratively applies a single changepoint detection method. However, at each step it allows for the identification of either a single changepoint or a small segment made up of two changepoints.

2.3 Bayesian inference for changepoint models

The Bayesian paradigm is also widely used in the changepoint literature. Some examples of these are described in Barry and Hartigan [1993], Green [1995], Fearnhead [2006], Benson and Friel [2016] and Fearnhead and Liu [2007].

The Bayesian approach gives us information regarding the uncertainty in the number of changepoints and their locations from the posterior distribution. This is much more informative than the point estimates given by the optimisation methods mentioned above, however, the disadvantage of Bayesian methods is their larger computational cost.

To perform Bayesian inference we need to specify priors on the parameters of interest, namely the number of changepoints, $\pi(m)$, locations and segment parameters of the changes conditional on the number $\pi(\boldsymbol{\theta}^{(m)}|m)$. The parameter vector $\boldsymbol{\theta}^{(m)}$ contains the locations of the m changepoints and the $m+1$ segment specific parameters θ_i, for $i = 1, 2, \ldots, m+1$

$$\boldsymbol{\theta}^{(m)} = (\tau_1, \ldots, \tau_m, \theta_1, \ldots, \theta_{m+1}).$$

For m changepoints the parameter vector $\boldsymbol{\theta}^{(m)}$ has length $2m+1$.

The joint posterior we are interested in can be factorised as

$$p(m, \boldsymbol{\theta}^{(m)}|\mathbf{y}) \propto \pi(m)\pi(\boldsymbol{\theta}^{(m)}|m)p(\mathbf{y}|\boldsymbol{\theta}^{(m)}, m). \tag{2.3.1}$$

Traditional Markov Chain Monte Carlo (MCMC) methods can be used to explore this posterior if m is known or fixed. However as we want to perform a full Bayesian analysis and have specified some prior $\pi(m)$ on m the sampler needs to be able to move between models with

differing numbers of changepoints. This is a problem for traditional MCMC samplers as the parameter vector we are aiming to explore and draw samples from $\boldsymbol{\theta}$ changes in dimension with each iteration.

One way to perform Bayesian inference on this model is to extend the traditional MCMC methodology and follow the well known Reversible jump Markov chain Monte Carlo (RJMCMC) method described in Green [1995]. This method allows us to jump between parameter spaces with a different number of dimensions. The traditional MCMC step of proposing a new value for a parameter and deciding whether to accept or reject the move is maintained in the RJMCMC method. However, two more steps are added: a "birth" step, used for the addition of a changepoint to the model and a "death" step, to remove a changepoint.

As in standard MCMC the mixing of the chain, autocorrelation of the samples and convergence are still an issue here. For general guidelines on designing RJMCMC algorithms see Armstrong et al. [2007]. The problems in diagnosing convergence and the possible substantial errors was highlighted by the analysis of a data set using the RJMCMC method and another approach which we describe below.

An adaptive algorithm was introduced in Benson and Friel [2016] which uses birth and death steps but also learns from the past states of the Markov chain in order to build proposal distributions which can quickly discover where changepoints are likely to be located. It is demonstrated that this algorithm is viable for large datasets and that the MCMC that targets the stationary distribution is ergodic.

For many changepoint models it is possible to simulate independent realisations directly from the posterior distribution. The ideas for direct simulation are based on exact methods for calculating posterior means which originated in Barry and Hartigan [1993] and are

extended in Fearnhead [2006]. The advantages of direct simulation methods over MCMC based methods are twofold. Firstly there is no need to concern ourselves with convergence and potentially running the chain for many iterations to prove convergence. Secondly samples from the posterior are independent so quantifying uncertainty is simple. An example of the first problem regarding convergence of the RJMCMC method is clear if we compare the inferences obtained for a Coal-mining disaster data set analysed using RJMCMC in Green [1995] and direct simulation in Fearnhead [2006]. Using the same model resulted in different segmentations because the chain wasn't run for a sufficient number of iterations.

The disadvantage of direct simulation methods is that they can only be applied to certain changepoint models that satisfy a conditional independence property. In our setting this means that conditional on the changepoint locations parameters of different segments are independent. Many models satisfy this property, for example a change in mean model where the mean parameters of each segment are i.i.d draws from some distribution. However, there are several useful models in practice that do not satisfy this condition. For example if we want to fit piece-wise functions but wish to enforce continuity between functions in neighbouring segments then this introduces dependence between parameters in neighbouring segments. It should be noted that this sort of dependence across segments is a problem for all changepoint detection methods and not only Bayesian methods. Later work in Fearnhead and Liu [2011] showed how the direct simulation method could be adapted to the case where parameters in consecutive segments have a Markov style dependence structure, however their method only calculates an approximation to the posterior.

While the specification of separate priors for the number and position of changepoints in (2.3.1) may seem intuitive. The direct simulation methods of Barry and Hartigan [1993],

Fearnhead [2006] and Fearnhead and Liu [2007] are simplest to describe when both the number and position of changepoints are jointly specified via a single prior distribution on the length of a segment. This single prior for the length of a segment implies that the sequence of changepoints forms a discrete renewal process with inter-arrival times that are identically distributed. The simplest inter-arrival distribution commonly used is a geometric distribution which results in the number of changepoints being Binomially distributed.

In Section 2.3.1 we show how a series of recursions can be developed following Fearnhead and Liu [2007] to calculate the posterior distribution of interest numerically, and then draw samples from it. We then consider how methods from the particle filtering literature can be used to limit the computational cost of these recursions in Section 2.3.2.

2.3.1 Exact online inference

Following the paper of Fearnhead and Liu [2007] we model the data through a hidden state process, $C_{1:n}$. This hidden state process will contain information about where the changepoints of the data are located. Our model is defined through specifying the distribution of the hidden state process, $p(\mathbf{c}_{1:n})$, and then the conditional distribution of the data given the state process, $p(\mathbf{y}_{1:n}|\mathbf{c}_{1:n})$.

Our interest lies in inference about this hidden state process given the observations which involves calculating the posterior distribution for the states

$$p(\mathbf{c}_{1:n}|\mathbf{y}_{1:n}) \propto p(\mathbf{y}_{1:n}|\mathbf{c}_{1:n})p(\mathbf{c}_{1:n}). \tag{2.3.2}$$

We introduce a state at time t, C_t , which is defined to be the time of the most recent change

point prior to time t.

We model C_t as a Markov process, conditional on C_t, either $C_{t+1} = c_t$, which corresponds to no changepoint at time t, or $C_{t+1} = t$, if there is a changepoint at time t. We need that $p(C_{t+1} = t|c_t)$ only depends on c_t. Thus $C_t \in \{0, \ldots, t-1\}$ with $C_t = 0$ meaning that the current segment is the first segment. This Markov process is determined by a set of transition probabilities which depend only on the distance between the current time t and the last changepoint.

Due to this process being Markov we can decompose $p(\mathbf{c}_{1:n})$ into factors

$$p(\mathbf{c}_{1:n}) = p(C_1 = c_1) \prod_{i=1}^{n-1} p(C_{i+1} = c_{i+1}|C_i = c_i). \tag{2.3.3}$$

The decomposition in (2.3.3) gives us two aspects of the process to define, namely the transition probabilities $p(C_{i+1} = c_{i+1}|c_i)$ and the initial distribution, $p(C_1 = c_1)$.

Firstly consider the transition probabilities. Now either $C_{t+1} = C_t$ or $C_{t+1} = t$ depending on whether a new segment starts between time t and $t + 1$. The probability of a new segment starting is just the conditional probability of a segment being of length $t - C_t$ given that is at least $t - C_t$. The probability of a segment continuing is the conditional probability of a segment having a length greater than $t - C_t$ given that is at least of length $t - C_t$.

Let $G(\cdot)$ be the distribution function of the distance between two successive change points then the transition probabilities can be written down for $p(C_{t+1} = j|C_t = i)$ where $i =$

$1, \ldots, t-1$ as

$$p(C_{t+1} = j | C_t = i) = \begin{cases} \frac{1-G(t-i)}{1-G(t-i-1)} & \text{if } j = i \\ \frac{G(t-i)-G(t-i-1)}{1-G(t-i-1)} & \text{if } j = t \\ 0 & \text{otherwise} \end{cases}$$

This hidden process partitions the time interval into contiguous non-overlapping segments. Using this we want to define a likelihood for the observations conditional on this process $p(\mathbf{y}_{1:n}|\mathbf{c}_{1:n})$, in (2.3.2). To make this model tractable, so we can write down a set of recursions, we assume a conditional independence between segments:

$$p(\mathbf{y}_{1:t}|C_t = j) = p(\mathbf{y}_{1:j}|C_t = j)p(\mathbf{y}_{(j+1):t}|C_t = j)$$

We can then define, for all $t < s$,

$$P(t, s) = p(\mathbf{y}_{t:s}|C_s = t - 1). \tag{2.3.4}$$

Conditional on the hidden states $\mathbf{c}_{1:n}$ the likelihood is

$$\begin{aligned} p(\mathbf{y}_{1:n}|\mathbf{c}_{1:n}) &= \prod_{t=1}^{n} p(y_t|\mathbf{c}_{1:n}, \mathbf{y}_{1:t-1}) \\ &= \prod_{t=1}^{n} p(y_t|c_t, \mathbf{y}_{(c_t+1):(t-1)}). \end{aligned} \tag{2.3.5}$$

The terms in (2.3.5) can be written as

$$p(y_t|c_t, \mathbf{y}_{(c_t+1):(t-1)}) = \frac{P(c_t + 1, t)}{P(c_t + 1, t - 1)}. \tag{2.3.6}$$

The main set of recursions is now derived that enable us to calculate the exact posterior numerically. There are two separate cases the first where $j < t$

$$\begin{aligned} p(C_{t+1} = j|\mathbf{y}_{1:(t+1)}) &\propto p(y_{t+1}|\mathbf{y}_{1:t}, C_{t+1} = j)p(C_{t+1} = j|\mathbf{y}_{1:t}) \\ &= \frac{P(j+1, t+1)}{P(j+1, t)} \Pr(C_{t+1} = j|C_t = j)p(C_t = j|\mathbf{y}_{1:t}). \end{aligned}$$

Then the second where $j = t$

$$\begin{aligned} p(C_{t+1} = t|\mathbf{y}_{1:(t+1)}) &\propto p(y_{t+1}|\mathbf{y}_{1:t}, C_{t+1} = j)p(C_{t+1} = j|\mathbf{y}_{1:t}) \\ &= P(t+1, t+1) \sum_{i=0}^{t-1} \Pr(C_{t+1} = t|C_t = i)p(C_t = i|\mathbf{y}_{1:t}). \end{aligned}$$

If we define

$$w_{t+1}^{(j)} = \begin{cases} \frac{P(j+1,t+1)}{P(j+1,t)} & \text{if } j < t \\ P(t+1, t+1) & \text{if } j = t \end{cases}$$

Then we can rewrite the set of recursions above more simply as

$$p(C_{t+1} = j|\mathbf{y}_{1:(t+1)}) \propto \begin{cases} w_{t+1}^{(j)} \frac{1-G(t-i)}{1-G(t-i-1)} p(C_t = j|\mathbf{y}_{1:t}) & \text{if } j < t \\ w_{t+1}^{(t)} \sum_{i=0}^{t-1} \left(\frac{G(t-i)-G(t-i-1)}{1-G(t-i-1)} p(C_t = i|\mathbf{y}_{1:t}) \right) & \text{if } j = t \end{cases} \tag{2.3.7}$$

Rewriting the recursions in the form shown in (2.3.7) enables us to calculate the posterior distribution of C_{t+1} by propagating the posterior for C_t and adding on another support point for a changepoint at time t.

For many simple models such as a change in mean the weights w_{t+1} can be calculated efficiently because each $P(j+1, t)$ depends on a set of summary statistics of the observations

$y_{j+1:t}$. These summaries can often be calculated and stored before we begin calculating the recursions and then updated recursively. Indeed for such models the computational cost of calculating any such w_{t+1} is fixed, and does not increase with $t - j$.

Simulation

Given that we calculate and store the filtering distributions $p(c_t|\mathbf{y}_{1:t})$ for all $t = 1, \ldots, n$, simulating from the full joint posterior is straightforward. This is done backwards in time by first simulating the last changepoint in the data c_n and repeating this until we get to the beginning of the data.

To simulate one realisation from this joint density:

1. Set $t_0 = n$, and $k = 0$.

2. Simulate t_{k+1} from the filtering density $p(C_{t_k}|\mathbf{y}_{1:t_k})$, and set $k = k + 1$.

3. If $t_k > 0$ return to (2); otherwise output the set of simulated changepoints, $t_{k1}, t_{k2}, \ldots, t_1$.

A simple extension of this algorithm allows for efficient simulation of a large sample of realisations of sets of changepoints in a parallel manner. This is described in more detail in Fearnhead [2006].

2.3.2 Approximate filtering

The computational and memory costs of the recursions for exact inference presented in Section 2.3.1 both increase with time. The filtering distribution $p(c_t|\mathbf{y}_{1:t})$ has t support points as $c_t \in \{0, \ldots, t-1\}$. Thus calculating the full set of filtering distributions exactly, grows

quadratically in n. The memory costs of storing all of the filtering densities necessary to simulate from the joint posterior of all changepoints also increases quadratically with n. For large data sets, these computational and memory costs become prohibitive.

A similar problem of increasing computational cost occurs in the analysis of some hidden Markov models, though generally computational cost increases exponentially with time [Chen and Liu, 2000]. Particle filters have been successfully applied to these problems [Fearnhead and Clifford, 2003] by using a resampling step to limit the computational cost at each time step. Similar resampling ideas can be applied to the online inference of changepoint models. We follow the methods described by Fearnhead and Liu [2007].

For our problem the particle filtering and resampling methods described here attempt to approximate the discrete distribution $p(c_t|\mathbf{y}_{1:t})$ that has t support points with a discrete distribution with fewer support points or "particles". This will of course increase the computational efficiency and decrease storage costs. However, due to the nature of these techniques error is introduced in the approximation. The aim is to come up with a method that is a trade off between increased performance while remaining as accurate as possible.

Essentially a threshold value α is chosen that defines the maximum error (as defined by the Kolmogorov Smirnov distance) that is introduced by the resampling procedure at each time step. Then those support points which have a probability less than α are stochastically removed. Note this value α governs the trade-off between a more precise approximation (smaller α) and speed (larger α). The stochastic removal of support points is required so that the process remains unbiased.

This algorithm is known as the Stratified Rejection Control (SRC) algorithm and more details can be found in Fearnhead and Liu [2007]. Pseudo-code for the SRC method is given

in Algorithm 3.

Theoretical guarantees and simulations showing that the SRC algorithm performs well are described in Liu and Chen [1998] and Fearnhead and Liu [2007].

It was found in the analysis of GC content in DNA by Paul Fearnhead [2009] that a value of $\alpha = 10^{-6}$ introduced negligible error, but greatly increased the speed of the overall algorithm. The resulting algorithm approximated the filtering densities, by distributions with an average of around 200 support points whereas the true distributions had an average of 80,000 support points, meaning this led to a 400-fold reduction in CPU and memory costs.

An alternative approach is to specify the maximum number of particles stored at any time. This is most suitable in an online algorithm used to analyse streaming data where the frequency of observations will place an upper bound on the CPU time that can be used to process each observation. This is known as the Stratified Optimal Resampling (SOR) method. More details can be found in Fearnhead and Liu [2007].

Algorithm 3: Stratified rejection control (SRC)

Input: An arbitrary cut-off $0 < \alpha < 1$
An ordered set of M particles $c_t^{(i)}$ with associated weights $w^{(i)}$ that sum to one, where $i = 1, 2, \ldots, M$.

Initialize: Simulate u a single realisation from Unif$(0, \alpha)$
Set $i = 1$

while $i \leq M$ **do**
  if $w^{(i)} \geq \alpha$ **then**
    Keep particle i with weight $w^{(i)}$
  end
  else
    $u \leftarrow u - w^{(i)}$
    if $u \leq 0$ **then**
      Set $w^{(i)} = \alpha$
      $u \leftarrow u + \alpha$
    end
    else
      Set $w^{(i)} = 0$
    end
  end
  $i \leftarrow i + 1$
end

Output: Remove the particles for which $w^{(i)} = 0$ and renormalise the remaining probabilities. These are the set of resampled particles.

Chapter 3

Changepoint detection for Multivariate time series

Historically, research on changepoint detection methods has focused on the univariate setting with some of the methods we have looked at in Chapter 2 being developed for this problem. More recently there has been an increased focus on multivariate changepoint detection due to the increase in multivariate series now being collected. There are many sources of this data such as the large array of sensors that record large streaming data sets. This data comes from many diverse fields such as finance where large numbers of asset prices are considered [Cho and Fryzlewicz, 2015] to bioinformatics and signal processing [Vert and Bleakley, 2010]. If we believe that the variables are related somehow and that changepoints occur at the same time in different series, then it makes sense to analyse them together in order to make the best possible use of the information available.

In this setting there are several subtleties that make this problem quite different to its uni-

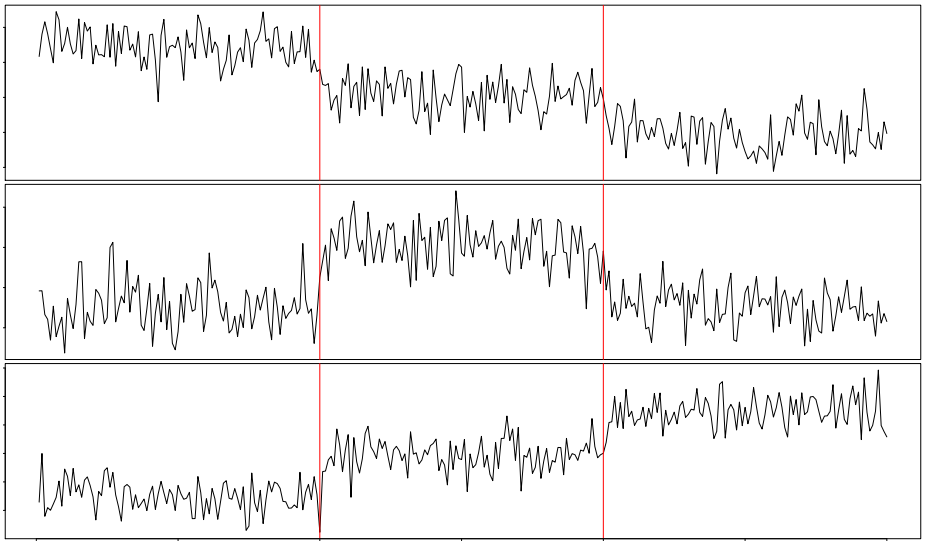


Figure 3.0.1: A multivariate series with three dimensions where any changes that occur affect all three series at the same time. We call this the full change model.

variate analogue. To show this graphically we have three plots in Figures 3.0.1, 3.0.2 and 3.0.3.

In Figure 3.0.1 we show an example of a three dimensional time series in which both of the changepoints affect all of the dimensions in the series. In future we refer to this problem as the full change model. We can compare this situation to that depicted in Figure 3.0.2 where at each changepoint only a subset of the dimensions are affected, we call this the subset change model.

Another possible subtlety in the multivariate setting is that we can allow for the possibility of different dimensions sharing a common change but the precise location in each dimension is perturbed. We call this the lagged change model and show an example of this in Figure 3.0.3. In the changepoint literature to our knowledge there hasn't been any work on the lagged change model.

The focus of the work in this thesis is the subset change model. We briefly review some of the main contributions in the literature to both the full and subset change models separately

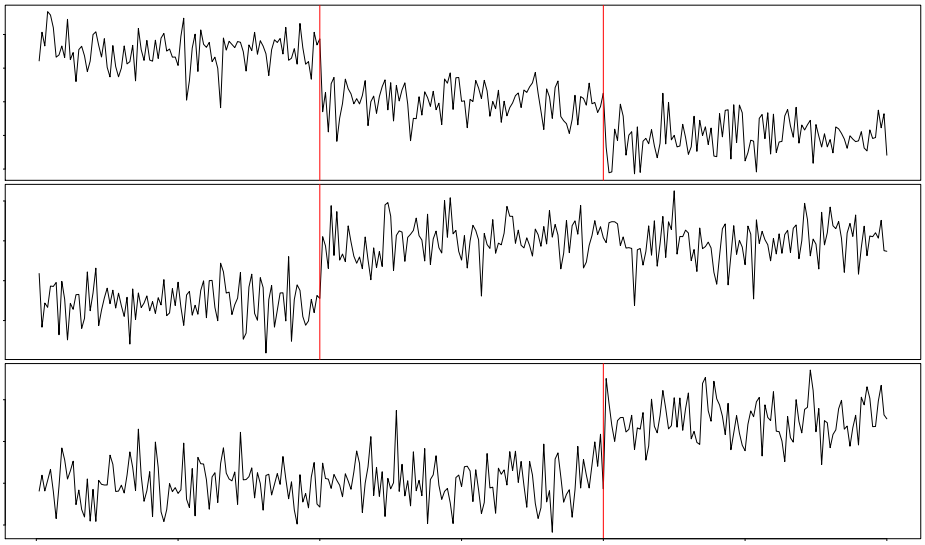


Figure 3.0.2: A multivariate series with three dimensions where the changes that occur only affect a subset of the three series at each changepoint. We call this the subset change model.

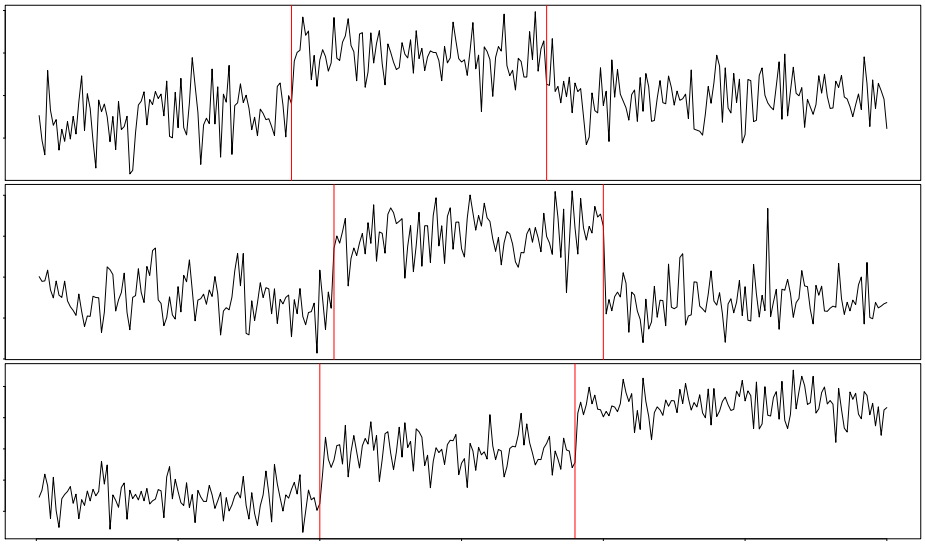


Figure 3.0.3: A multivariate series with three dimensions where any changes that occur affect all three series but at slightly different times. We call this the lagged change model.

so we can understand the different types of approaches that have been successful.

3.1 Full change model

For data that respects the full change model, techniques used for the univariate problem can be used.

A simplistic approach to the problem would thus be to try and apply univariate changepoint methods and analyse each time series separately. This method will lose power when detecting changepoints, as it ignores the information that the different time series have changepoints that occur at the same time.

An alternative approach to analysing data of this form is to treat the data as a single time series with multivariate observations. We then model the multivariate data within a segment, and allow for this model to change, in an appropriate way, between segments. This approach is taken by Lavielle and Teyssière [2006], who model data as multivariate Gaussian but with a mean that can change from segment to segment. Lavielle and Teyssière [2006] proposes dynamic programming recursions similar to those described in Chapter 2.

Matteson and James [2014] present a non-parametric approach which is based on a Euclidean style metric known as an energy statistic which takes the role of a cost function. Estimation of the changepoint locations is based on hierarchical clustering and both divisive and agglomerative algorithms are developed.

3.2 Subset change model

In some applications, when a changepoint occurs only a subset of the dimensions in a given series are affected. Recently there has been lots of work on certain special cases of the subset change model which are of importance in detecting Copy Number Variants (CNV's) in Genetics.

A CNV is a type of structural variation that results in a genome having an abnormal (generally $\neq 2$) number of copies of a segment of DNA, such as a gene. Understanding these is

important as these variants have been shown to account for much of the variability within a population. More details can be found in Jeng et al. [2013] and the references therein.

There are two types of CNV, common CNV which affect all or most of the population and rare CNV which only affect a small subset of the population. Different techniques have been developed to detect the two types of CNV with the methods in Zhang et al. [2010] and Siegmund et al. [2011] designed for the common CNV problem. It was argued in Jeng et al. [2013] that these methods cannot detect rarer CNVs and special methods are required to use the information across the dimensions most efficiently so that it is not drowned out by noise. The Proportion Adaptive Segment Selection (PASS) method developed in Jeng et al. [2013] aims to efficiently detect both common and rare variants without incurring too many false positives.

Such a situation is not uncommon in practice and is not confined to genetics. Consider, for example, the finance setting and a large panel data set consisting of asset returns over time. Here an event may induce a sudden change in the stock prices of companies within one industrial sector but not in those of companies within a different sector. Whereas for the CNV application only a change in mean is considered, for asset returns the second order structure of the data is important. Indeed Mikosch and Stric [2004] note that certain observable features of financial time series, such as long-range dependence of the absolute returns, might be artifacts that are induced by change points in the second order structure.

Both Maboudou-Tchao and Hawkins [2013] and Preuss et al. [2015] present methods which explicitly output both the locations of changepoints and the corresponding sets of affected variables. These methods are quite similar in their approach in that they initially assume the full change model and then perform variable-specific hypothesis tests for each estimated

changepoint to determine the subset of variables each changepoint affects. As estimation is not performed jointly, in that the changepoints and affected variables are not estimated at the same time the methods are approximate. Also when performing the variable specific hypothesis tests problems arise in that certain variables could be falsely claimed to be affected (a Type I error) as well as the converse. For example if many variables were weakly affected by a change but which individually are not significant enough to be flagged by the test (a Type II error). The advantages of both methods, however is that they are relatively computationally efficient and the main idea behind the approaches outlined above is applicable to many different time series models. Indeed Preuss et al. [2015] deviate from the setting of i.i.d. data and aim to detect multiple changes in the autocovariance through the consideration of raw periodograms.

In contrast to the approaches already considered, Pickering [2016] develops a Dynamic programming method for inference in the full subset change model and outputs changepoint locations as well as the variables that are affected. The two methods (for multivariate data) considered in this thesis are Subset Multivariate Optimal Partitioning (SMOP) and Approximate-SMOP (ASMOP).

The general subset change model is formulated in Pickering [2016] using changepoint vectors. For a p-variate series the changepoint vector at time t, $\mathbf{c}_t = (c_t^{(1)}, \ldots, c_t^{(i)}, \ldots, c_t^{(p)})$ has length p. The ith element of the vector $c_t^{(i)}$ is the location of the most recent change in the ith dimension prior to time t. These vectors encapsulate all the information needed to model the full subset change model.

Using these changepoint vectors and a suitably defined cost function a set of Dynamic programming recursions are developed to solve this problem exactly through the optimisation

of a penalised cost function using an exact search. However, this exact search is extremely computationally intensive due to the dependence between segments that exists in the general subset change model. This results in the SMOP method having a computational complexity that is exponential in the length of the data.

An approximate version of this procedure, ASMOP was also presented in this thesis where a substantial reduction in the search space is achieved by identifying "likely" regions in which changepoints are thought to occur and only considering candidates from these areas along with two types of thresholding.

For the univariate and non-parametric case a common and simple approach to detect changepoints is to use the CUSUM test [Page, 1954b]. CUSUM statistics are computed over time and track the cumulative distance from the mean for proposed changepoints. These series of CUSUMs are examined to locate changepoints, often where its maximum in the absolute value is attained. With a binary segmentation (BS) algorithm, the CUSUM statistics can consistently detect multiple change-points in a recursive manner (see e.g. Vostrikova [1981], Venkatraman [1993] and Cho and Fryzlewicz [2012]).

This idea can be extended to multivariate data by combining the CUSUMs of each of the dimensions of the series. Care is needed in choosing the method used to combine these. Standard maximum and average methods for doing so often fail in high dimensions when, changepoints only affect a small subset of the series so that the CUSUM statistics are corrupted by noise.

Cho and Fryzlewicz [2015] and Cho [2016] propose methods that aggregate the cumulative sum statistics by adding only those that pass a certain threshold. This "sparsifying" step reduces the influence of irrelevant noisy contributions, which is particularly beneficial in

high dimensions in order to share information across series. Whenever a thresholding step is introduced in a procedure an additional parameter is usually added. However, the aggregation procedure of Cho [2016] in contrast to Cho and Fryzlewicz [2015], avoids arbitrary choices for the threshold by using a data driven approach which is shown to perform well.

In the next Chapter we describe a Bayesian method that can be used to perform inference for the CNV problem described above.

Chapter 4

Bayesian detection of abnormal segments in multiple time series

4.1 Introduction

In this paper we consider the problem of detecting abnormal (or outlier) segments in multivariate time series. We assume that the series has some normal or baseline behaviour but that in certain intervals or segments of time a subset of the dimensions of the series has some kind of altered or abnormal behaviour. By the term abnormal behaviour we mean some change in distribution of the data away from the baseline distribution. For example, this could include a change in mean, variance or auto-correlation structure. In particular our work is concerned with situations where the size of this subset is only a small proportion of the total number of dimensions. We attempt to do this in a fully Bayesian framework.

This problem is increasingly common across a range of applications where the detection of abnormal segments (sometimes known as recurrent signal segments) is of interest (particu-

larly in high dimensional and/or very noisy data). Some example applications include the analysis of the correlations between sensor data from different vehicles [Spiegel et al., 2011] or for intrusion detection in large interconnected computer networks [Qu et al., 2005]. Another related application involves detecting common and potentially more subtle objects in a number of images, for example Jin [2004] and the references therein look at this in relation to multiple images taken of astronomical bodies.

We will focus in particular on one specific example of this type of problem, namely that of detecting copy number variants (CNV's) in DNA sequences. A CNV is a type of structural variation that results in a genome having an abnormal (generally $\neq 2$) number of copies of a segment of DNA, such as a gene. Understanding these is important as these variants have been shown to account for much of the variability within a population. For a more detailed overview of this topic see Zhang [2010], Jeng et al. [2013] and the references therein.

Data on CNVs for a given cell or individual is often in the form of "log-R ratios" for a range of probes, each associated with different locations along the genome. These are calculated as log base 2 of the ratio of the measured probe intensity to the reference intensity for a given probe. Normal regions of the genome would have log-R ratios with a mean of 0, whereas CNVs would have log-R ratios with a mean that is away from zero.

Figure 4.1.1 gives an example of such data from 6 individuals. We can see that there is substantial noise in the data, and each CNV may cover only a relatively small region of the genome. Both these factors mean that it can be difficult to accurately detect CNVs by analysing data from a single individual or cell. To increase the power to identify CNVs we can pool information by jointly analysing data from multiple individuals. However this is complicated as a CNV may be observed for only a subset of the individuals. For example,

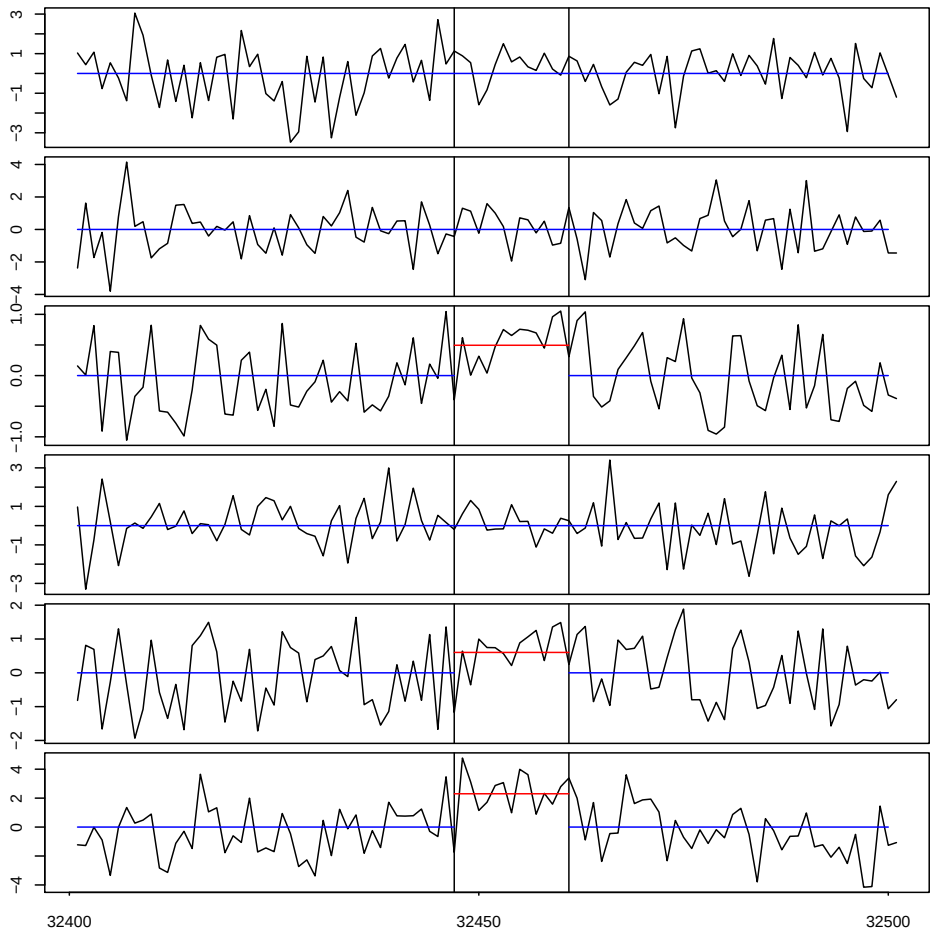


Figure 4.1.1: Log-R ratios from 6 individuals for a small portion of chromosome 16. We indicate the baseline level (mean zero) by a horizontal line in blue and the identified CNV (abnormal region) is highlighted between two vertical black lines with the mean of the affected individuals in red.

for the data in Figure 4.1.1, which shows data from a small portion of chromosome 16, we have identified a single CNV which affects only three individuals. This can seen by the raised means (indicated by the red lines) in these three series for a segment of data. By comparison, the other individuals are unaffected in this segment.

Whilst there has been substantial research into methods for detecting outliers [Tsay et al., 2000, Galeano et al., 2006] or abrupt changes in data [Olshen et al., 2004a, Jandhyala et al., 2013, Wyse et al., 2011, Frick et al., 2014], the problem of identifying outlier regions in just

a subset of dimensions has received less attention. Exceptions include methods described in Zhang et al. [2010] and Siegmund et al. [2011]. However Jeng et al. [2013] argue that these methods are only able to detect common variants, that is abnormal segments for which a large proportion of the dimensions have undergone the change. Jeng et al. [2013] propose a method, the PASS algorithm, which is also able to detect rare variants.

The methods of Siegmund et al. [2011] and Jeng et al. [2013] are based on defining an appropriate test-statistic for whether a region is abnormal for a subset of dimensions, and then recursively using this test-statistic to identify abnormal regions. As such the output of these methods is a list of estimated abnormal regions. Here we introduce a Bayesian approach to detecting abnormal regions. This is able to not only give estimates of the number and location of the abnormal regions, but to also give measures of uncertainty about these. We show how it is possible to efficiently simulate from the posterior distribution of the number and location of abnormal regions, through using recursions similar to those from multiple changepoint detection [Barry and Hartigan, 1992, Fearnhead, 2006, Fearnhead and Vasileiou, 2009]. We call the resulting algorithm, Bayesian Abnormal Region Detector (BARD).

The outline of the paper is as follows. In the next section we introduce our model, both for the general problem of detecting abnormal regions, and also for the specific CNV application. In Section 4.3 we derive the recursions that enable us to draw iid samples from the posterior, as well as a simple approximation to these recursions that results in an algorithm, BARD, that scales linearly with the length of data set. We then present theoretical results that show that BARD can consistently estimate the absence of abnormal segments, and the location of any abnormal segments, and is robust to some mis-specification of the priors. In Section 4.5 we evaluate BARD for the CNV application on both simulated and real data. Our results

suggest that BARD is more accurate than PASS, particularly in terms of having fewer false positives. Furthermore, we see evidence that posterior probabilities are well-calibrated and hence are accurately representing the uncertainty in the inferences. The paper ends with a discussion.

4.2 The Model

We shall now describe the details of our model. Consider a multiple time series of dimension d and length n, $\mathbf{Y}_{1:n} = (\mathbf{Y}_1, \mathbf{Y}_2, \ldots, \mathbf{Y}_n)$ where $\mathbf{Y}_i = (Y_{i,1}, Y_{i,2}, \ldots, Y_{i,d})^T$. We model this data through introducing a hidden state process, $X_{1:n}$. The hidden state process will contain information about where the abnormal segments of the data are. Our model is defined through specifying the distribution of the hidden state process, $p(x_{1:n})$, and the conditional distribution of the data given the state process, $p(\mathbf{y}_{1:n}|x_{1:n})$. These are defined in Sections 4.2.1 and 4.2.2 respectively.

Our interest lies in inference about this hidden state process given the observations. This involves calculating the posterior distribution for the states

$$p(x_{1:n}|\mathbf{y}_{1:n}) \propto p(x_{1:n}, \mathbf{y}_{1:n}) = p(x_{1:n})p(\mathbf{y}_{1:n}|x_{1:n}). \tag{4.2.1}$$

It should be noted that these probabilities will depend on a set of hyper-parameters. These parameters are initially assumed to be known, however we will later discuss performing inference for them.

4.2.1 Hidden State Model

The hidden state process will define the location of the abnormal segments. We will model the location of these segments through a renewal process. The length of a given segment is drawn from some distribution which depends on the segment type, and is independent of all other segment lengths. We assume a normal segment is always followed by an abnormal segment, but allow for either a normal or abnormal segment to follow an abnormal one. The latter is because each abnormal segment may be abnormal in a different way, for example with different subsets of the time-series being affected. This will become clearer when we discuss the likelihood model in Section 4.2.2.

To define such a model we need distributions for the lengths of normal and abnormal segments. We denote the cumulative distribution functions of these lengths by $G_N(t)$ and $G_A(t)$ respectively. We also need to specify the probability that an abnormal segment is followed by either a normal or abnormal segment. We denote these probabilities as π_N and π_A respectively, with $\pi_N = 1 - \pi_A$.

Note that the first segment for the data will have a different distribution to other segments as it may have started at some time prior to when we started collecting data. We can define this distribution in a way that is consistent with our underlying model by assuming the process for the segments is at stationarity and that we start observing it at an arbitrary time. Renewal theory [Cox, 1962] then gives the distribution function for the length of the first segment. If the first segment is normal, then we define its cumulative distribution function as

$$G_{0N}(t) = \sum_{s=1}^{t} \frac{1 - G_N(s)}{E_N},$$

where E_N is the expected length of a normal segment. The cumulative distribution function for the first segment conditional on it being abnormal, $G_{0A}(t)$, is similarly defined.

Formally, we define our hidden state process X_t as $X_t = (C_t, B_t)$ where C_t is the end of the previous segment prior to time t and B_t is the type of the current segment. So $C_t \in \{0, \ldots, t-1\}$ with $C_t = 0$ denoting that the current segment is the first segment. We use the notation that $B_t = N$ if the current segment is normal, and $B_t = A$ if not. This state process is Markov, and thus we can write

$$\begin{aligned} p(x_{1:n}) &= p(c_{1:n}, b_{1:n}) \\ &= \Pr(C_1 = c_1, B_1 = b_1) \prod_{i=1}^{n-1} \Pr(C_{i+1} = c_{i+1}, B_{i+1} = b_{i+1} | C_i = c_i, B_i = b_i). \end{aligned} \tag{4.2.2}$$

The decomposition in (4.2.2) gives us two aspects of the process to define, namely the transition probabilities $\Pr(C_{i+1} = c_{i+1}, B_{i+1} = b_{i+1} | c_i, b_i)$ and the initial distribution, $\Pr(C_1 = c_1, B_1 = b_1)$.

Firstly consider the transition probabilities. Now either $C_{t+1} = C_t$ or $C_{t+1} = t$ depending on whether a new segment starts between time t and $t+1$. The probability of a new segment starting is just the conditional probability of a segment being of length $t - C_t$ given that is at least $t - C_t$. If $C_{t+1} = C_t$, then we must have $B_{t+1} = B_t$, otherwise the distribution of the type of the new segment depends on the type of the previous segment as described above.

Thus for $i = 1, \ldots, t-1$ we have

$$\Pr(C_{t+1} = j, B_{t+1} = k | C_t = i, B_t = N) = \begin{cases} \frac{1-G_N(t-i)}{1-G_N(t-i-1)} & \text{if } j = i \text{ and } k = N, \\ \frac{G_N(t-i)-G_N(t-i-1)}{1-G_N(t-i-1)} & \text{if } j = t \text{ and } k = A, \\ 0 & \text{otherwise,} \end{cases}$$

$$\Pr(C_{t+1} = j, B_{t+1} = k | C_t = i, B_t = A) = \begin{cases} \frac{1-G_A(t-i)}{1-G_A(t-i-1)} & \text{if } j = i \text{ and } k = A, \\ \pi_A \left(\frac{G_A(t-i)-G_A(t-i-1)}{1-G_A(t-i-1)} \right) & \text{if } j = t \text{ and } k = A, \\ \pi_N \left(\frac{G_A(t-i)-G_A(t-i-1)}{1-G_A(t-i-1)} \right) & \text{if } j = t \text{ and } k = N, \\ 0 & \text{otherwise.} \end{cases} \tag{4.2.3}$$

For $i = 0$, that is when $C_t = 0$, we replace $G_N(\cdot)$ and $G_A \cdot$ with $G_{0N}(\cdot)$ and $G_{0A}(\cdot)$ respectively. Finally we need to define the initial distribution for $X_1 = (B_1, C_1)$. Firstly note that $C_1 = 0$ so we need only the distribution of B_1. We define this as the stationary distribution of the B_t process. This is [see for example Theorem 5.6 of Kulkarni, 2012]

$$\Pr(B_1 = N) = \frac{\pi_N E_N}{\pi_N E_N + E_A}, \quad \Pr(B_1 = A) = 1 - \Pr(B_1 = N),$$

where E_N and E_A are the expected lengths of normal and abnormal segments respectively.

4.2.2 Likelihood model

The hidden process $X_{1:n}$ described above partitions the time interval into contiguous non-overlapping segments each of which is either normal, N, or abnormal, A. Now conditional on this process we want to define a likelihood for the observations, $p(\mathbf{y}_{1:n}|x_{1:n})$.

For many applications it is natural to assume a conditional independence property between segments: this means that if we knew the locations of segments and their types then data from different segments are independent. This assumption is key to the algorithms we later introduce to sample from the posterior. Thus when we condition on C_t and B_t the likelihood for the first t observations factorises as follows

$$p(\mathbf{y}_{1:t}|C_t = j, B_t) = p(\mathbf{y}_{1:j}|C_t = j, B_t)p(\mathbf{y}_{j+1:t}|C_t = j, B_t). \tag{4.2.4}$$

The second term in equation (4.2.4) is the marginal likelihood of the data, $\mathbf{Y}_{j+1:t}$, given it comes from a segment that has type B_t. We introduce the following notation for these segment marginal likelihoods, where for $s \geq t$,

$$\begin{aligned} P_N(t, s) &= \Pr(\mathbf{y}_{t:s}|C_s = t - 1, B_s = N), \\ P_A(t, s) &= \Pr(\mathbf{y}_{t:s}|C_s = t - 1, B_s = A), \end{aligned} \tag{4.2.5}$$

and define $P_N(t, s) = 1$ and $P_A(t, s) = 1$ if $s < t$.

Now using the above factorisation we can write down the likelihood conditional on the hidden process. Note that we can condition on X_t rather than the full history $X_{1:n}$ in each of the

factors in (4.2.6) due to the conditional independence assumption on the segments

$$\begin{aligned} p(\mathbf{y}_{1:n}|x_{1:n}) &= \prod_{t=1}^{n} p(\mathbf{y}_t|x_{1:n}, \mathbf{y}_{1:(t-1)}) \\ &= \prod_{t=1}^{n} p(\mathbf{y}_t|C_t, B_t, \mathbf{y}_{(C_t+1):(t-1)}). \end{aligned} \tag{4.2.6}$$

The terms on the right-hand side of equation (4.2.6) can then be written in terms of the segment marginal likelihoods

$$p(\mathbf{y}_t|C_t, B_t, \mathbf{y}_{(C_t+1):(t-1)}) = \frac{P_{B_t}(C_t + 1, t)}{P_{B_t}(C_t + 1, t - 1)}. \tag{4.2.7}$$

Thus our likelihood is specified through defining appropriate forms for the marginal likelihoods for normal and abnormal segments.

Model for data in normal segments

For a normal segment we model that the data for all dimensions of the series are realisations from some known distribution, $\mathcal{D}$, and these realisations are independent over both time and dimension. Denote the density function of the distribution $\mathcal{D}$ as $f_{\mathcal{D}}(\cdot)$. We can write down the segment marginal likelihood as

$$P_N(t, s) = \prod_{k=1}^{d} \prod_{i=t}^{s} f_{\mathcal{D}}(y_{i,k}). \tag{4.2.8}$$

Model for data in abnormal segments

For abnormal segments our model is that data for a subset of the dimensions are drawn from $\mathcal{D}$, with the data for the remaining dimensions being independent realisations from a

different distribution, $\mathcal{P}_\theta$, which depends on a segment specific parameter θ. We denote the density function for this distribution as $f_{\mathcal{P}}(\cdot|\theta)$.

Our model for which dimensions have data drawn from $\mathcal{P}_\theta$ is that this occurs for dimension k with probability p_k, independently of the other dimensions. Thus if we have an abnormal segment with data $\mathcal{Y}_{t:s}$, with segment parameter θ, the likelihood of the data associated with the kth dimension is

$$p_k \prod_{i=t}^{s} f_{\mathcal{P}}(y_{i,k}|\theta) + (1-p_k) \prod_{i=t}^{s} f_{\mathcal{D}}(y_{i,k}).$$

Thus by independence over dimension

$$p(\mathbf{y}_{t:s}|\theta) = \prod_{k=1}^{d} \left(p_k \prod_{i=t}^{s} f_{\mathcal{P}}(y_{i,k}|\theta) + (1-p_k) \prod_{i=t}^{s} f_{\mathcal{D}}(y_{i,k}) \right).$$

Our model is completed by a prior for θ, $\pi(\theta)$. To find the marginal likelihood $P_A(t,s)$ we need to integrate out θ from $p(\mathbf{y}_{t:s}|\theta)$

$$P_A(t,s) = \int p(\mathbf{y}_{t:s}|\theta)\pi(\theta)\,\mathrm{d}\theta. \tag{4.2.9}$$

In practice this integral will need to be calculated numerically, which is feasible if θ is low-dimensional.

CNV example

In Section 4.1 we discussed the copy number variant (CNV) application and showed some real data in Figure 4.1.1. From the framework described above we now need to specify a model for normal and abnormal segments. Following Jeng et al. [2013] we model the data

as being normally distributed with constant variance but differing means either zero or μ depending on whether we are in a normal or abnormal segment. This model also underpins the simulation studies that we present in Section 4.5.

Using the notation from the more general framework discussed above the two distributions for normal and abnormal segments are

$$\mathcal{D} \sim N(0, \sigma^2)$$

$$\mathcal{P}_\mu \sim N(\mu, \sigma^2).$$

We assume that the variance σ^2 is constant and known. In practice we estimate this quantity using the robust median absolute deviation estimator as recommended in Jeng et al. [2013]. Having specified these two distributions we then need to calculate marginal likelihoods for normal and abnormal segments given by equations (4.2.8) and (4.2.9) respectively. Calculating the marginal likelihood for a normal segment is simple because of independence over time and dimension as shown in equation (4.2.8). However calculating $P_A(\cdot, \cdot)$ is more challenging, as there is no conjugacy between $p(\mathbf{y}|\mu)$ and $\pi(\mu)$ so we can only numerically approximate the integral. Calculating the numerical approximation is fast as it is a one-dimensional integral. In the simulation studies and results we take the prior for μ to be uniform on a region that excludes values of μ close to zero. For CNV data such a prior seems reasonable empirically (see Figure 4.5.1c) and also because we expect CNV's to correspond to a change in mean level of at least $\log(3/2)$ and can be both positive or negative.

4.3 Inference

We now consider performing inference for the model described in Section 4.2. Firstly a set of recursions to perform this task exactly are introduced and then an approximation is considered to make this procedure computationally more efficient.

4.3.1 Exact On-line inference

We follow the method of Fearnhead and Vasileiou [2009] in developing a set of recursions for the posterior distribution of the hidden state, the location of the start of the current segment and its type, at time t given that we have observed data upto time t, $p(x_t|\mathbf{y}_{1:t}) = p(c_t, b_t|\mathbf{y}_{1:t})$, for $t \in \{1, 2, \ldots, n\}$. These are known as the filtering distributions. Eventually we will be able to use these to simulate from the full posterior, $p(x_{1:n}|\mathbf{y}_{1:n})$.

To find these filtering distribution we develop a set of recursions that enable us to calculate $p(c_{t+1}, b_{t+1}|\mathbf{y}_{(1:t+1)})$ in terms of $p(c_t, b_t|\mathbf{y}_{1:t})$. These recursions are analogous to the forward-backward equations widely used in analysing Hidden Markov models.

There are two forms of these recursions depending on whether $C_{t+1} = j$ for $j < t$ or $C_{t+1} = t$. We derive the two forms separately. Consider the first case. For $j < t$ and $k \in \{N, A\}$,

$$\begin{aligned} p(C_{t+1} = j, B_{t+1} = k|\mathbf{y}_{1:(t+1)}) &\propto p(\mathbf{y}_{t+1}|\mathbf{y}_{1:t}, C_{t+1} = j, B_{t+1} = k)p(C_{t+1} = j, B_{t+1} = k|\mathbf{y}_{1:t}) \\ &= \left(\frac{P_k(j+1, t+1)}{P_k(j+1, t)}\right) \Pr(C_{t+1} = j, B_{t+1} = k|C_t = j, B_t = k)p(C_t = j, B_t = k|\mathbf{y}_{1:t}), \end{aligned}$$

where the first term in the last expression is the conditional likelihood from equation (4.2.7). The second two terms use the fact that there has not been a new segment and hence $C_{t+1} = C_t$

and $B_{t+1} = B_t$.

Now for the second case, when $C_{t+1} = t$,

$$p(C_{t+1} = t, B_{t+1} = k|\mathbf{y}_{1:t}) = \sum_{i=0}^{t-1} \sum_{l \in \{N,A\}} p(C_t = i, B_t = l|\mathbf{y}_{1:t}) \Pr(C_{t+1} = t, B_{t+1} = k|C_t = i, B_t = l).$$

Thus, as $p(\mathbf{y}_{t+1}|C_{t+1} = t, B_{t+1} = k, \mathbf{y}_{1:t}) = P_k(t+1, t+1)$, the filtering recursion is;

$$p(C_{t+1} = t, B_{t+1} = k|\mathbf{y}_{1:(t+1)}) \propto P_k(t+1, t+1) \sum_{i=0}^{t-1} \sum_{l \in \{N,A\}} p(C_t = i, B_t = l|\mathbf{y}_{1:t}) \Pr(C_{t+1} = t, B_{t+1} = k|C_t = i, B_t = l).$$

These recursions are initialised by $p(C_1 = 0, B_1 = k|\mathbf{y}_1) \propto \Pr(B_1 = k) P_k(1, 1)$ for $k \in \{N, A\}$.

4.3.2 Approximate Inference

The support of the filtering distribution $p(c_t, b_t|\mathbf{y}_{1:t})$ has $2t$ points. Hence, calculating $p(c_t, b_t|\mathbf{y}_{1:t})$ exactly is of order t both in terms of computational and storage costs. The cost of calculating and storing the full set of filtering distributions $t = 1, 2, \ldots, n$ is thus of order n^2. For larger data sets this exact calculation can be prohibitive. A natural way to make this more efficient is to approximate each of the filtering distributions by distributions with a fewer number of support points. In practice such an approximation is feasible as many of the support points of each filtering distribution have negligible probability. If we removed these points then we could greatly increase the speed of our algorithm without sacrificing too much accuracy.

We use the stratified rejection control (SRC) algorithm [Fearnhead and Liu, 2007] to produce an approximation to the filtering distribution with potentially fewer support points at each time-point. This algorithm requires the choice of a threshold, $\alpha \geq 0$. At each iteration the SRC algorithm keeps all support points which have a probability greater than α. For the remaining particles the probability of them being removed is proportional to their associated probability and the resampling is done in a stratified manner. This algorithm has good theoretical properties in terms of the error introduced at each resampling step, measured by the Kolmogorov Smirnov distance, being bounded by α.

4.3.3 Simulation

Having calculated and stored the filtering distributions, either exactly or approximately, simulating from the posterior is straightforward. This is performed by simulating the hidden process backwards in time [Carter and Kohn, 1994]. First we simulate $X_n = (C_n, B_n)$ from the final filtering distribution $p(c_n, b_n|\mathbf{y}_{1:n})$. Assume we simulate $C_n = t$. Then, by definition of the hidden process, we have $C_s = t$ and $B_s = B_n$ for $s = t+1, \ldots, n-1$, as these time-points are all part of the same segment. Thus we next need to simulate C_t, from its

conditional distribution given C_{t+1}, B_{t+1} and $\mathbf{Y}_{1:n}$,

$$
\begin{aligned}
&p(c_t, b_t | C_{t+1} = t, B_{t+1}, \mathbf{y}_{1:n}) \\
&\quad \propto p(c_t, b_t, C_{t+1} = t, B_{t+1}, \mathbf{y}_{1:n}) \\
&\quad = p(c_t, b_t) \Pr(C_{t+1} = t, B_{t+1} | C_t, B_t) p(\mathbf{y}_{1:n} | C_t, B_t, C_{t+1} = t, B_{t+1}) \\
&\quad \propto p(c_t, b_t) \Pr(C_{t+1} = t, B_{t+1} | C_t, B_t) p(\mathbf{y}_{1:t} | C_t, B_t) \\
&\quad \propto p(c_t, b_t | \mathbf{y}_{1:t}) \Pr(C_{t+1} = t, B_{t+1} | C_t, B_t).
\end{aligned}
$$

We then repeat this process, going backwards in time until we simulate $C_t = 0$. From the simulated values we can extract the location and type of each segment.

4.3.4 Hyper-parameters

As mentioned earlier in Section 4.2 the posterior of interest (4.2.1) depends upon a vector of hyper-parameters which we now label as Ψ. In Section 5, Ψ contains the parameters for the LOS distributions for the two differing types of segments which determine the cdf's $G_N(\cdot)$ and $G_A(\cdot)$. However we could extend Ψ to account for the hyperparameters for the prior on μ or, if we did not assume a common and known variance for the data, the variance for each time-series.

We use two approaches to estimating these hyper-parameters. The first is to maximise the marginal-likelihood for the hyper-parameters, which we can do using Monte Carlo EM (MCEM). For general details on MCEM see Levine and Casella [2001]. Although convergence of the hyper-parameters is quite rapid in the examples we look at in Section 4.5, for very large data sets a cruder but faster alternative is to initially segment the data using a different

method to ours and then use information from this segmentation to inform the choice of hyper-parameter values. The alternative method we use is the PASS method of Jeng et al. [2013] and discussed in detail in Section 4.5.

4.3.5 Estimating a Segmentation

We have described how to calculate the posterior density $p(x_{1:n}|\mathbf{y}_{1:n})$ from which we can easily draw a large number of samples. However we often want to report a single estimated "best" segmentation of the data. We can define such a segmentation using Bayesian decision theory [Berger, 1985]. This involves defining a loss function which determines the cost of us making a mistake in our estimate of the true quantity which we then seek to minimise. There are various choices of loss function we could use [see Yau and Holmes, 2010], but we use a loss that is a sum of a loss for estimating whether each location is abnormal or not. If $L(\tilde{b}_t|b_t)$ gives the cost of making the decision that the state at time t is $\tilde{b}_t$ when in fact it is b_t, then:

$$L(\tilde{b}_t|b_t) = \begin{cases} 1 & \text{if } \tilde{b}_t = \text{A and } b_t = \text{N} \\ \gamma & \text{if } \tilde{b}_t = \text{N and } b_t = \text{A} \\ 0 & \text{otherwise} \end{cases} \tag{4.3.1}$$

The inclusion of γ allows us to vary the relative penalty for false positives as compared to false negatives. Under this loss we estimate $\hat{b}_t = N$ if $\pi(b_t = A) < 1/(1+\gamma)$ or $\hat{b}_t = A$ otherwise.

4.4 Asymptotic Consistency

We will now consider the asymptotic properties of the method as d, the number of time-series, increases. Our aim is to study the robustness of inferences to the choice of prior for the abnormal segments, and the estimate of p_d, allowing for abnormal segments that are rare. We will assume that each time-series is of fixed length n. Following Jeng et al. [2013], to consider the influence of rare abnormal segments, we will let the proportion of sequences that are abnormal in an abnormal segment to decrease as d increases.

Our assumptions on how the data is generated is that there are a fixed number and location of abnormal segments. We will assume the model of Section 4.2.2 with, without loss of generality, $\sigma^2 = 1$ (for $\sigma^2 \neq 1$ we can just normalise the data). So if $B_t = N$, then $Y_{i,j} \sim N(0,1)$. If $(t, \ldots, s)$ is an abnormal segment then it has an associated mean, $\mu_0 \neq 0$. For each $j = 1, \ldots, d$, independently with probability α_d, $Y_{i,j} \sim N(\mu, 1)$ for $i = t, \ldots, s$; otherwise $Y_{i,j} \sim N(0,1)$ for $i = t, \ldots, s$.

We fit the model of Section 4.2, assuming the correct likelihood for data in normal and abnormal segments. For each abnormal segment we will have an independent prior for the associated mean, $\pi(\mu)$. Our assumptions on $\pi(\mu)$ is that its support is a subset of $\{[-b, -a], [a, b]\}$ for some $a > 0$ and $b < \infty$, and it places non-zero probability on both positive and negative values of μ. The model we fit will assume a specified probability, p_d, of each sequences being abnormal within each abnormal segment. Note that we do not require $p_d = \alpha_d$, the true probability, but we do allow the choice of this parameter to depend on d.

The Lemmas used in the proof of the following two theorems can be found in the appendices.

Theorem 4.4.1. *Assume the model for the data and the constraints on the prior specified*

above. Let $\mathcal{E}$ be the event that there are no abnormal segments, and $\mathcal{E}^c$ its complement. If there are no abnormal segments and $d \to \infty$, with $1/p_d = O(d^{\frac{1}{2}-\epsilon})$ for some $\epsilon > 0$, then

$$\Pr(\mathcal{E}^c|\mathbf{y}_{1:n}) \to 0,$$

in probability.

Proof. As n is fixed, we have a fixed number of possible segmentations. We will show that the posterior probabilility of each possible segmentation with at least one abnormal segment is $o_p(1)$ as $d \to \infty$.

For time-series k let $P_{N,k}(t,s)$ denote the likelihood of the data $y_{t,k}, \ldots, y_{s,k}$ assuming this is a normal segment; and let $P_{A,k}(t,s;\mu)$ be the marginal likelihood of the same data given that it is drawn from independent Gaussian distributions with mean μ. Then if we have a segmentation with m abnormal segments, with the ith abnormal segment from t_i to s_i, the ratio of the posterior probability of this segmentation to the posterior probability of $\mathcal{E}$ is

$$K \prod_{i=1}^{m} \left(\int \left\{ \prod_{k=1}^{d} \frac{P_{A,k}(t_m, s_m; \mu)}{P_{N,k}(t_m, s_m)} \right\} \pi(\mu) \mathrm{d}\mu \right),$$

where K is the ratio of the prior probabilities of these two segmentations. So it is sufficient to show that for all $t \leq s$,

$$\int \left\{ \prod_{k=1}^{d} \frac{P_{A,k}(t,s;\mu)}{P_{N,k}(t,s)} \right\} \pi(\mu) \mathrm{d}\mu \to 0 \tag{4.4.1}$$

in probability as $d \to \infty$.

Our limit involves treating the data as random. Each term in this product is then random,

and of the form

$$\frac{P_{A,k}(t,s;\mu)}{P_{N,k}(t,s)} = 1 + p_d \left(\exp\left\{ \mu \sum_{u=t}^{s} \left(Y_{k,u} - \frac{\mu}{2} \right) \right\} - 1 \right). \tag{4.4.2}$$

By applying Lemma A.0.4 separately to positive and negative values of μ, we have that this tends to 0 with probability 1 as $d \to \infty$. This is true for all possible segmentations with at least one abnormal segments. As n is fixed, there are a finite number of such segments, so the result follows.

□

Theorem 4.4.2 tells us that the posterior probability of misclassifying a time point as normal when it is abnormal tends to zero as more time-series are observed.

Theorem 4.4.2. *Assume the model for the data and the constraints on the prior specified above. Fix any position t, and consider the limit as $d \to \infty$, with $dp_d^2 \to \infty$ and either*

(i) $p_d = o(\alpha_d)$; *or*

(ii) if μ_0 is the mean associated with the abnormal sequences at position t, then there exists a region A such that the prior probability associated with $\mu \in A$ is non-zero, and for all $\mu \in A$ and for sufficiently large d

$$\alpha_d \left(e^{\mu\mu_0} - 1 \right) - \frac{p_d}{2} \left(e^{\mu^2} - 1 \right) > 0.$$

Then if $B_t = A$

$$\Pr(B_t = N | \mathbf{y}_{1:n}) \to 0.$$

in probability.

Proof. We will show that each segmentation with $B_t = N$ has posterior probability that tends to 0 in probability as $d \to \infty$. For each segmentation with $B_t = N$ we will compare its posterior probability with one which is identical except for the addition of an abnormal segmentant, of length 1, at location t. The ratio of posterior probabilities of these two segmentations will be

$$K\left(\int\left\{\prod_{k=1}^{d}\frac{P_{A,k}(t,t;\mu)}{P_{N,k}(t,t)}\right\}\pi(\mu)\mathrm{d}\mu\right),$$

where K is a constant that depends on the prior for the segmentations. We require that this ratio tends to infinity in probability as $d \to \infty$. Under both conditions (i) and (ii) above this follows immediately from Lemma A.1.2. For case (i) we are using the fact that the prior places positive probability both on μ being positive and negative, and for μ the same sign as μ_0 we have that $e^{\mu\mu_0} > 1$. □

This result shows some robustness of the Bayesian approach to the choice of prior. Consider a prior on the mean for an abnormal segment that has strictly positive density for values in $\{[-b,-a],[a,b]\}$ for $a > 0$. Then for any true mean, μ_0 with $|\mu_0| \geq a$, we will consistently estimate the segment as abnormal provided the assumed or estimated probability of a sequence being abnormal is less than twice the true value. Thus we want to choose a to be the smallest absolute value of the mean of an abnormal segment we expect or wish to detect. The choice of b is less important, in that it does not affect the asymptotic consistency implied by the above theorem.

Furthermore we do not need to specify p_d exactly for consistency – the key is not to overestimate the true proportion of abnormal segments by more than a factor of two. We could

set $p_d = Kd^{-1/2+\epsilon}$ for some constants $K, \epsilon > 0$ and ensure that asymptotically we will consistently estimate the absence of abnormal segments (Theorem 4.4.1) and the location of any abnormal segments (Theorem 4.4.2) the true proportion of abnormal segments decays at a rate that is slower than $d^{-1/2+\epsilon}$.

4.5 Results

We call the method introduced in Sections 4.2 and 4.3 BARD: Bayesian Abnormal Region Detector. We now evaluate BARD on both simulated and real CNV data. Our aim is to both investigate its robustness to different types of model mis-specification, and to compare its performance with a recently proposed method for analysing such CNV data.

The simulation studies we present are based on the concrete example in Section 4.2.2, namely the change in mean model for Normally distributed data. For inference we assume that the LOS distributions, S_N and S_A, to be Negative binomial and the prior probability of a particular dimension k being abnormal p_k as the same for all $k = \{1, 2, \ldots, d\}$. For all the simulation studies we present we used MCEM on a single replicate of the simulated data set to get estimates for the hyper-parameters for the LOS distribution, but fixed p_k. Data for normal segments are IID standard Gaussian, and for abnormal segments data from dimensions that are abnormal are Gaussian with variance 1 but mean μ drawn from some prior $\pi(\mu)$. Below we consider the effect of varying the choice of prior used for simulating the data and that assumed within BARD. In implementing BARD we used the SRC method of resampling described in Section 4.3.2 with a value of $\alpha = 10^{-4}$, we found this value of α gave a good trade off between accuracy and computational cost.

To get an explicit segmentation from BARD we use the asymmetric loss function (4.3.1) with a value of $\gamma = 1/3$.

As a benchmark for comparison we also analyse all data sets using the Proportion Adaptive Segment Selection procedure (PASS) from Jeng et al. [2013]. This was implemented using an R package called `PASS` which we obtained from the authors website. At its most basic level the PASS method involves evaluating a test statistic for different segments of the data. After these evaluations the values of the statistic that exceed a certain pre-specified threshold are said to be significant and the segments that correspond to these values are the identified abnormal segments. This threshold is typically found by simulating data sets with no abnormal segments and then choosing the threshold which gives a desired type 1 error, here we take this error to be 0.05 in the simulation studies. The PASS algorithm considers all segments that are shorter than a pre-defined length. To avoid excessive computational costs this length should be as small as possible, but at least as large as the longest abnormal segment we wish to detect (or believe exists in the data). We ran PASS with this length set to ten-times the largest abnormal segment.

We found that a run of PASS was about twice as fast as one run of BARD. In order to estimate the hyper-parameters using MCEM took between 5 and 20 runs of BARD.

Evaluating a segmentation

To form a comparison between the two methods we must have some way of evaluating the quality of a particular segmentation with respect to the ground truth. We consider the three most important criteria to be the number of true and false positives and the accuracy in detecting the true positives.

We define a segment to be correctly identified or a true positive if it intersects with the

true segment. With this definition in mind then finding the true/false positives is simple. Note that results for false positives are the number of false positive segments per data set. To define the accuracy of an estimated segment compared to the truth it is most intuitive to measure the amount of "overlap" of the segments, this is captured by the dissimilarity measure D_k (4.5.1) defined in Jeng et al. [2013].

Let $\hat{\mathbb{I}}$ be the collection of estimated intervals, the accuracy of estimating the k^{th} true segment I_k is given by D_k

$$D_k = \min_{\hat{I}_j \in \hat{\mathbb{I}}} \left\{ 1 - \frac{|\hat{I}_j \cap I_k|}{\sqrt{|\hat{I}_j||I_k|}} \right\} \tag{4.5.1}$$

$D_k \in [0, 1]$, if $D_k = 0$ then an estimated interval overlaps exactly with segment I_k however if $D_k = 1$ then no estimated intervals overlap with the k^{th} segment, i.e. it hasn't been detected. Smaller values of D indicate a greater overlap.

For all measures we present the average value across the simulated data sets, together with a 95% confidence interval for this average calculated via the bootstrap.

4.5.1 Simulated Data from the Model

Firstly we analysed data simulated from the model assumed by BARD. A soft maximum on the length of the simulated data of $n = 1000$ was imposed and the number of dimensions fixed at $d = 200$. The LOS distributions were

$$S_N \sim \text{NBinom}(10, 0.1) \text{ and } S_A \sim \text{NBinom}(15, 0.3).$$

Two different distributions were used to generate the altered means for the affected dimensions and we also varied π_N (see Table 4.5.1), and for each scenario we implemented the Bayesian method with the correct prior for the abnormal mean, and the correct choice of π_N. The number of affected dimensions for each abnormal segment was fixed at 4% and we fixed p_k to this value. For each scenario we considered we generated 200 data sets.

μ	π_N	Method	Proportion detected	Accuracy	Number of False positives
$U(0.3, 0.7)$	0.5	PASS	0.68 (0.66,0.70)	0.12 (0.11,0.13)	0.80 (0.68,0.93)
		BARD	0.88 (0.87,0.89)	0.077 (0.071,0.084)	0.08 (0.04,0.12)
	0.8	PASS	0.67 (0.65,0.69)	0.13 (0.12,0.14)	1.13 (0.98,1.29)
		BARD	0.78 (0.77,0.80)	0.094 (0.088,0.10)	0.07 (0.04,0.11)
$U(0.5, 0.9)$	0.5	PASS	0.92 (0.91,0.93)	0.074 (0.070,0.078)	1.08 (0.94,1.22)
		BARD	0.98 (0.98,0.99)	0.039 (0.036,0.042)	0.03 (0.01,0.06)
	0.8	PASS	0.94 (0.93,0.95)	0.073 (0.069,0.076)	1.02 (0.88,1.17)
		BARD	0.96 (0.95,0.97)	0.042 (0.040,0.045)	0.02 (0.00,0.04)

Table 4.5.1: Scenarios differed in the prior for μ and the value of π_N used to simulate the data. In BARD these same priors were used for the analysis of the data. The results for each scenario are averages across 200 simulated data sets together with 95% confidence interval in brackets.

Results summarising the accuracy of the segmentations obtained by the two methods are shown in Table 4.5.1. BARD performed substantially better than PASS here especially with regards to the number of false positives each method found, though this is in part because all the modelling assumptions within BARD are correct for these simulated data sets. It is worth noting that both methods do much better when $\mu \sim U(0.5, 0.9)$ due to the stronger signal present.

We next investigated how robust the results were to our choice for p_k. We just consider $\mu \sim U(0.3, 0.7)$ and $\pi_N = 0.8$ and we vary our choice of p_k from 0.5% to 10%. These results are in table 4.5.2. Whilst, as expected, if we take p_k to be the true value for the data we get

p_k	Proportion detected	Accuracy	Number of False positives
$\frac{1}{200}$	0.65 (0.63,0.67)	0.093 (0.086,0.10)	0.03 (0.005,0.06)
$\frac{4}{200}$	0.76 (0.74,0.78)	0.092 (0.086,0.10)	0.09 (0.05,0.12)
$\frac{8}{200}$	0.77 (0.75,0.78)	0.086 (0.081,0.093)	0.06 (0.03,0.09)
$\frac{12}{200}$	0.76 (0.74,0.78)	0.089 (0.083,0.096)	0.06 (0.03,0.095)
$\frac{16}{200}$	0.74 (0.72,0.76)	0.091 (0.084,0.098)	0.06 (0.03,0.09)
$\frac{20}{200}$	0.72 (0.70,0.74)	0.095 (0.088,0.102)	0.05 (0.02,0.08)

Table 4.5.2: The robustness of BARD under a misspecification of p_k taking the prior as $\mu \sim U(0.3, 0.7)$ and $\pi_N = 0.8$ with the true value of p_k being 4%. Values of p_k were varied between 0.5% and 10% and we simulated 200 data sets for each p_k. The results for each scenario are averages across 200 simulated data sets together with 95% confidence interval in brackets.

the best segmentation, the results are clearly robust to mis-specification of p_k. In all cases we still achieve much higher accuracy and fewer false positives than PASS. Apart from the choice $p_k = 1/200$ we also have a higher proportion of correctly detected CNVs than PASS.

We also investigated the robustness to mis-specification of the model for the LOS distribution, and for the distribution of the mean of the abnormal segments. We fixed the position of five abnormal segments at the following time points 200, 300, 500, 600 and 750. Additionally the segments at 200 and 750 were followed by another abnormal segment. Thus we have seven abnormal segments in total. The true LOS distribution for the abnormal segments are in fact Poisson with intensity randomly chosen from the set $\{20, 25, 30, 35, 40\}$. For these abnormal segments the mean value that affected the dimensions was drawn from a Normal distribution with differing means and a fixed variance shown in Table 4.5.3. The number of affected dimensions for each of the abnormal segments was also varied randomly from 3-6% of the total number of dimensions ($d = 200$). For inference, we fixed p_k to 4% for all k and we set the prior for the abnormal mean to be uniform on $(-0.7, -0.3) \cup (0.3, 0.7)$. Our model for the LOS distribution were negative binomials, with MCEM used to estimate the

hyper-parameters of these distributions.

μ	Method	Proportion detected	Accuracy	Number of False positives
$N(0.8, 0.4^2)$	PASS	0.81 (0.78,0.82)	0.065 (0.056,0.068)	1.26 (1.15,1.41)
	BARD	0.85 (0.82,0.86)	0.055 (0.048,0.059)	0.04 (0.02,0.07)
$N(0.7, 0.4^2)$	PASS	0.77 (0.74,0.78)	0.076 (0.069,0.084)	1.11 (1.05,1.33)
	BARD	0.80 (0.78,0.82)	0.066 (0.060,0.073)	0.02 (0.01,0.07)
$N(0.6, 0.4^2)$	PASS	0.69 (0.66,0.71)	0.086 (0.079,0.095)	1.22 (1.08,1.37)
	BARD	0.73 (0.70,0.75)	0.066 (0.061,0.072)	0.06 (0.03,0.09)
$N(0.5, 0.4^2)$	PASS	0.62 (0.60,0.65)	0.10 (0.089,0.11)	1.15 (1.06,1.37)
	BARD	0.65 (0.62,0.68)	0.087 (0.075,0.093)	0.06 (0.02,0.08)
$N(0.4, 0.4^2)$	PASS	0.53 (0.51,0.56)	0.12 (0.10,0.13)	1.07 (0.92,1.22)
	BARD	0.58 (0.55,0.61)	0.093 (0.084,0.10)	0.07 (0.03,0.10)

Table 4.5.3: Results based on 200 simulated data sets as we vary the distribution from which μ was simulated from but keeping the prior $\pi(\mu)$ in BARD uniform. The results for each scenario are averages across 200 simulated data sets together with 95% confidence interval in brackets.

From Table 4.5.3 it can be seen that BARD still outperforms PASS especially in regards to accuracy and the number of false positives. The performance of BARD also shows that it is robust to a misspecification of both the LOS distributions and the distribution from which μ was drawn from as we kept the prior in BARD the same. The performance of both methods was impacted by the decreasing mean of the Normal distributions from which μ was drawn as more of them became close to zero and thus abnormal segments became indistinguishable from normal segments.

4.5.2 Simulated CNV Data

We now make use of the CNV data presented in the Section 4.1, to obtain a more realistic model to simulate data from. We used the PASS method to initially segment one replicate of the data, and then analysed this segmentation to obtain information about the LOS

distributions and the distributions that generate the data in both normal and abnormal segments.

In Figure 4.5.1 we plot some of the empirical data from the segmentation given by PASS. To simulate data sets we either fitted distributions to these quantities or sampled from their empirical distributions. Firstly if we consider the two LOS distributions then for normal segments, see Figure 4.5.1b, we found that a geometric distribution fitted the data well. For the abnormal LOS distribution we took a discrete uniform distribution on $\{1, 2, \ldots, 200\}$. This was partly due to us having specified a maximum abnormal segment length of 200 in the PASS method but is potentially realistic in practice as abnormal segments longer than 200 time points are unlikely to occur. To support this choice we plot the empirical cdf of the ordered data and a straight line which are the quantiles of the uniform distribution we propose. We can see that although the fit is not perfect, this is probably due to the small sample size.

Now consider the distributions that generate the actual observations, we can think of these in two parts, one of them being a distribution for the "noise" in normal segments (Figure 4.5.1d) and then the mean shift parameter for the abnormal segments (Figure 4.5.1c). Up until now we have taken this noise distribution to be standard Normal, however the data suggests that in reality it has heavier tails than the Normal distribution. We found that a t-distribution with 15 degrees of freedom was a better fit to the data so we simulated from this for the noise distribution. For the mean shift parameter μ we took abnormal segments found by the PASS method and looked at the means of each of the dimensions and took the affected dimensions only, this gave the histogram in Figure 4.5.1c. In the study we simulated μ from this empirical distribution.

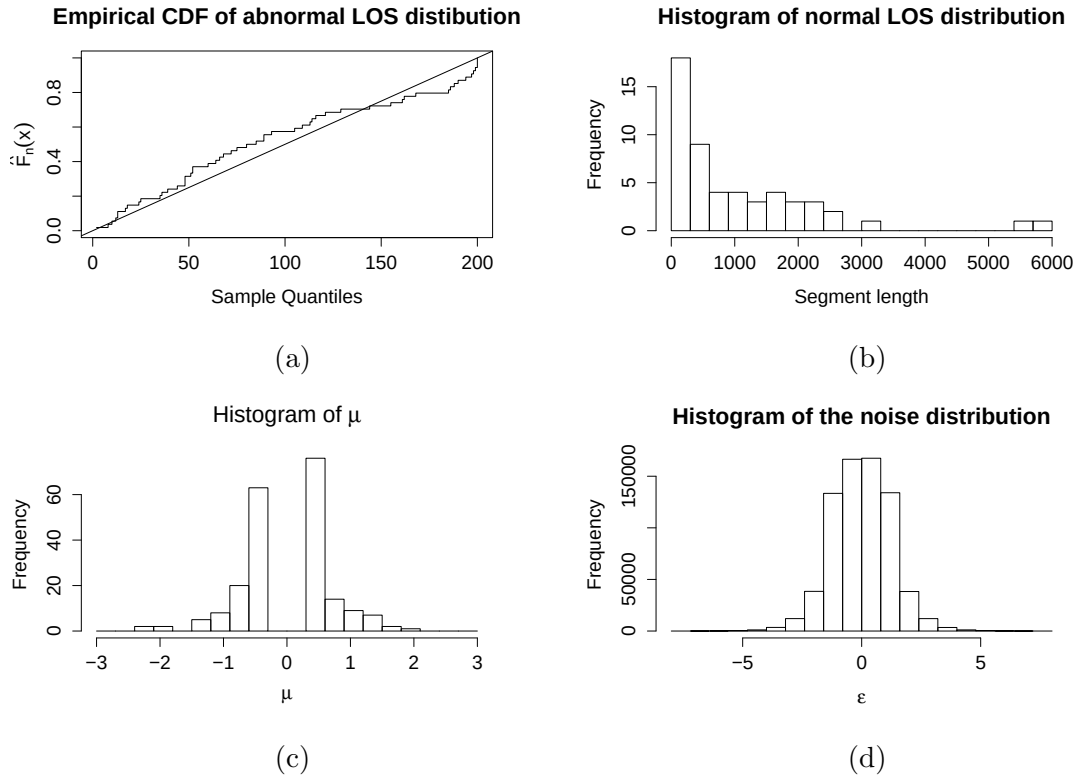


Figure 4.5.1: Empirical distribution of features of the optimal segmentation of CNV data obtained using the PASS method. (a) QQ-plot of length (measured in number of observations) of abnormal segments against a Uniform distribution on $\{1, 2, \ldots, 200\}$; (b) histogram of length (measured in number of observations) of normal segments; (c) histogram of estimated mean for abnormal segments; and (d) histogram of residuals.

Each simulated data set has length of approximately $n = 20,000$ and dimension $d = 50$. We also varied the proportion of affected dimensions between 4% and 6%. The robustness of BARD to the choice of prior for μ introduced in Section 4.4, where $|\mu|$ is uniform on an interval (a, b), was also investigated. We simulated 40 of these data sets for each of the scenarios and used both methods to segment them, results are given in Table 4.5.4.

We can see that the proportion of correct segments identified is decreased in both methods, this is most likely due to the non-Normally distributed noise present. However the two methods report a very different number of false positives. The performance of BARD is encouraging as it gives many fewer false positives than PASS even with heavier tailed ob-

% of dim. affected	Method	Proportion detected	Accuracy	Number of False positives
4%	PASS	0.55 (0.51,0.59)	0.071 (0.061,0.081)	1.23 (0.95,1.53)
	BARD $a = 0$	0.65 (0.62,0.70)	0.064 (0.056,0.074)	0.23 (0.10,0.38)
	BARD $a = 0.15$	0.65 (0.61,0.69)	0.065 (0.056,0.074)	0.23 (0.10,0.38)
	BARD $a = 0.3$	0.65 (0.61,0.69)	0.064 (0.055,0.072)	0.2 (0.08,0.35)
	BARD $a = 0.6$	0.55 (0.50,0.59)	0.070 (0.060,0.081)	0.13 (0.03,0.23)
6%	PASS	0.64 (0.61,0.68)	0.068 (0.062,0.074)	1.38 (0.98,1.80)
	BARD $a = 0$	0.72 (0.69,0.75)	0.058 (0.053,0.064)	0.23 (0.10,0.35)
	BARD $a = 0.15$	0.71 (0.68,0.74)	0.059 (0.053,0.065)	0.2 (0.08,0.35)
	BARD $a = 0.3$	0.70 (0.67,0.73)	0.054 (0.049,0.060)	0.1 (0.00,0.20)
	BARD $a = 0.6$	0.63 (0.59,0.66)	0.071 (0.061,0.081)	0.05 (0.00,0.13)

Table 4.5.4: Results based on 40 simulated data sets for two scenarios where the proportion of dimensions affected for each abnormal segment varied between 4% and 6% (of the total number of dimensions $d = 50$). The prior for $|\mu|$ assumed by BARD is uniform on $(a, 0.7)$ The results for each case are averages across simulated data sets together with 95% confidence interval in brackets.

servations than the standard Gaussian case for all choices of a. The results for BARD are similar for different values of $a \leq 0.3$, but do deteriorate slightly for $a = 0.6$. This is likely to be due to a loss of power in detecting abnormal segments with whose change in mean is less than 0.6.

We also vary the second parameter, b, in the prior for μ. We fix $a = 0.3$ and the number of dimensions as $d = 50$. These figures are reported in Table 4.5.5 and show that our procedure is relatively robust to the choice of b.

We also looked at the effect of varying the dimension d of the data, but keeping the proportion of affected dimensions the same. The results can be seen in Table 4.5.6, these indicate that both the proportion of abnormal segments detected and the accuracy improve as d is increased, due to the extra information with larger d. However the number of false positives gets worse for both methods, as with larger d there is more chance for some dimensions to show evidence for abnormality within normal regions.

b	Method	Proportion detected	Accuracy	Number of False positives
-	PASS	0.55 (0.51,0.59)	0.071 (0.061,0.081)	1.23 (0.95,1.53)
0.5	BARD	0.64 (0.60,0.68)	0.063 (0.055,0.072)	0.13 (0.03,0.23)
1	BARD	0.64 (0.61,0.69)	0.063 (0.055,0.073)	0.3 (0.15,0.48)
2	BARD	0.63 (0.59,0.66)	0.069 (0.059,0.081)	0.2 (0.08,0.35)
4	BARD	0.61 (0.57,0.64)	0.067 (0.057,0.079)	0.13 (0.03,0.25)
10	BARD	0.52 (0.48,0.55)	0.071 (0.060,0.082)	0.10 (0.03,0.20)

Table 4.5.5: Results based on 40 simulated data sets for each scenario where the proportion of dimensions affected for each abnormal segment was fixed at 4% and the number of dimensions $d = 50$. The prior for $|\mu|$ used by BARD was $(0.3, b)$. The results for each case are averages across simulated data sets together with 95% confidence interval in brackets.

d	Method	Proportion detected	Accuracy	Number of False positives
50	PASS	0.55 (0.51,0.59)	0.071 (0.061,0.081)	1.23 (0.95,1.53)
	BARD	0.65 (0.61,0.69)	0.064 (0.055,0.072)	0.2 (0.08,0.35)
100	PASS	0.65 (0.62,0.68)	0.063 (0.056,0.070)	2.1 (1.68,2.58)
	BARD	0.73 (0.70,0.76)	0.051 (0.045,0.059)	0.35 (0.18,0.58)
200	PASS	0.75 (0.72,0.78)	0.055 (0.049,0.062)	3.1 (2.60,3.60)
	BARD	0.85 (0.84,0.87)	0.038 (0.034,0.043)	1.4 (1.05,1.80)

Table 4.5.6: Results based on 40 simulated data sets for each scenario where the proportion of dimensions affected for each abnormal segment was fixed at 4% and the number of dimensions d was varied from 50 to 200. The results for each case are averages across simulated data sets together with 95% confidence interval in brackets.

The final parameter we investigate is γ which is instrumental in getting an explicit segmentation from the BARD method using the loss function in (4.3.1). If γ is small then more evidence is needed for a time point to be classified as abnormal so generally the smaller γ is the smaller the proportion of true positives will be, however the mean number of false positives detected would be large. The reverse is true as γ is increased. We can see from Table 4.5.7 that all values of γ perform similarly.

BARD also allows us to get an estimate of the uncertainty in the position of abnormal segments as from the posterior we can get the probability of each time point belonging to an abnormal segment. If we bin these probabilities into intervals and then find the proportion

γ	Method	Proportion detected	Accuracy	Number of False positives
-	PASS	0.55 (0.51,0.59)	0.071 (0.061,0.081)	1.23 (0.95,1.53)
1/4	BARD	0.64 (0.60,0.68)	0.064 (0.057,0.073)	0.2 (0.05,0.38)
1/3	BARD	0.65 (0.61,0.69)	0.064 (0.055,0.072)	0.2 (0.08,0.35)
1/2	BARD	0.66 (0.62,0.69)	0.063 (0.054,0.072)	0.28 (0.13,0.45)
2/3	BARD	0.67 (0.63,0.71)	0.063 (0.055,0.073)	0.35 (0.18,0.53)
3/4	BARD	0.67 (0.63,0.71)	0.061 (0.053,0.072)	0.4 (0.23,0.60)
1	BARD	0.68 (0.64,0.71)	0.065 (0.056,0.075)	0.48 (0.28,0.68)

Table 4.5.7: Results based on 40 simulated data sets for each scenario where the proportion of dimensions affected for each abnormal segment was fixed at 4%, the number of dimensions $d = 50$ and values of $a = 0.3$ and $b = 0.7$ in the split prior. The parameter γ was varied in the loss function (4.3.1). The results for each case are averages across simulated data sets together with 95% confidence interval in brackets.

of these points that are actually abnormal we can obtain a calibration plot Figure 4.5.2. We can see from this that the model seems to be well calibrated.

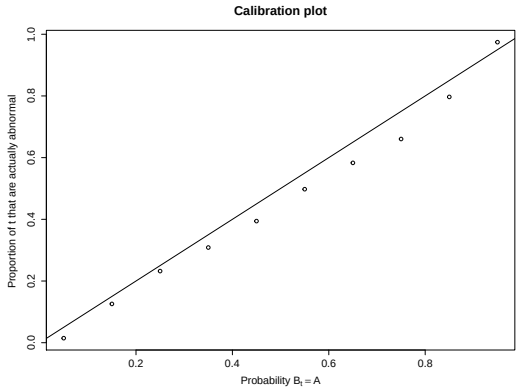


Figure 4.5.2: All the time points t for which the posterior probability lies in a certain interval plotted against the proportion of times t lies in an abnormal segment.

4.5.3 Analysis of CNV Data

We now apply our method to CNV data from Pinto et al. [2011], a subset of which was presented in Section 4.1 and was used to construct a model for the simulated data in Section 4.5.2.

Pinto et al. [2011] undertook a detailed study of the different technologies (platforms) used to obtain the measurements and many of the algorithms currently used to call CNV's. We chose to analyse data from the Nimblegen 2.1M platform and from chromosomes 6 and 16. For both chromosomes we have three replicate data sets, each consisting of measurements from from six genomes. We preprocessed the data to remove experimental artifacts, using the method described in Siegmund et al. [2011], before analysing it. The data from chromosome 16 consisted of 59,590 measurements, and the data from chromosome 6 consisted of 126,695 measurements, for each genome.

Firstly we ran the PASS method on just the first replicate of the data from chromosome 16 and found the most significant segments. Doing this enables us to get an estimate of the parameters for the LOS distributions to use in the Bayesian method without having to do any parameter inference. The maximum length of segment we searched over was 200 (measured in observations not base pairs) as this is greater than the largest CNV we would expect to find. This gave parameters that suggested a geometric distribution for the length of normal segments $S_N \sim \text{Geom}(0.0007)$ and the following Negative Binomial distribution for abnormal segments $S_N \sim \text{NBinom}(2, 0.1)$. We used the same split uniform prior for μ as we did in Section 4.5.2 namely one with equal density on the set $(-0.7, -0.3) \cup (0.3, 0.7)$ and zero elsewhere. We justified the use of this form of prior which excludes values close to zero in Section 4.2.2 and it was shown to perform well on some realistically simulated data in Section 4.5.2.

For both chromosomes we analysed the three replicates separately. Ideally we should infer exactly the same segmentation for each of the replicate data sets. Due to the large amount of noise present in the data this does not happen. However we would expect that a "better"

Truth		PASS			BARD		
Start	Length	Rep 1	Rep 2	Rep 3	Rep 1	Rep 2	Rep 3
2619669	62144	-	-	-	-	✓	✓
21422575	76266	✓	✓	✓	✓	✓	✓
32165010	456897	✓	✓	✓	✓	✓	✓
34328205	286367	✓	✓	✓	✓	✓	✓
54351338	28607	✓	✓	-	✓	✓	✓
70644511	21083	-	✓	✓	-	✓	✓

Table 4.5.8: Known CNV's from HapMap found by either method when analysing different replicates of data from chromosome 16. Ticks indicate whether the particular segment was detected or not.

Truth		PASS			Bayesian		
Start	Length	Rep 1	Rep 2	Rep 3	Rep 1	Rep 2	Rep 3
202353	37484	-	-	-	✓	✓	-
243700	80315	✓	✓	✓	✓	✓	✓
29945167	12079	✓	✓	-	-	-	-
31388080	61239	✓	-	-	✓	-	-
32562253	117686	✓	-	✓	-	-	-
32605094	74845	-	✓	-	✓	✓	✓
32717276	22702	✓	-	-	✓	✓	✓
74648953	9185	✓	✓	-	✓	✓	✓
77073620	10881	-	✓	-	✓	✓	✓
77155307	781	-	-	-	✓	-	-
77496587	12936	-	-	-	✓	✓	✓
78936990	18244	✓	✓	✓	✓	✓	✓
103844669	24085	✓	✓	✓	✓	✓	✓
126225385	3084	✓	✓	-	-	-	✓
139645437	3392	-	-	-	✓	-	-
165647807	4111	-	-	-	✓	-	✓

Table 4.5.9: Known CNV's from HapMap found by either method when analysing different replicates of data from chromosome 6. Ticks indicate whether the particular segment was detected or not.

method would be more consistent across the three replicates, and we use the consistency of the inferred segmentations across the replicates as a measure of accuracy.

We can also use data from the HapMap project to validate some of the CNV's we found to those known experimentally or which have been called by other authors. A list containing

Chromosome	Method	Rep 1 v 2	Rep 1 v 3	Rep 2 v 3
6	PASS	0.474	0.709	0.522
	BARD	0.495	0.457	0.416
16	PASS	0.478	0.507	0.388
	BARD	0.426	0.467	0.682

Table 4.5.10: The average consistency measured using the dissimilarity measure for found CNV's between replicates and methods. A lower value indicates the inferred segmentations for the two replicates were more similar.

these known CNV's by chromosome and sample can be found at `http://hapmap.ncbi.nlm.nih.gov/`.

These validated segments suggest that about 1% of chromosome 16 is abnormal.

To make comparisons between BARD and PASS fair we implemented both of these methods so that they identified the same proportion, 4%, of the chromosome as being abnormal. For BARD this involved choosing γ in the loss function (4.3.1) appropriately and for PASS selecting the most significant segments that give us a total of 4% abnormal time points. We then tested these against the validated CNV's.

The results for chromosome 16 are contained in Tables 4.5.8 and 4.5.10; and those for chromosome 6 in Tables 4.5.9 and 4.5.10. Tables 4.5.8 and 4.5.9 list the known CNV regions that were detected by one or both methods for at least one replicate, whilst Table 4.5.10 gives summaries of the consistency of the inferred segmentations across replicates.

The results show that BARD is more successful at detecting known CNV regions than PASS. In total BARD found 6 CNV regions on chromosome 16 for at least one replicate, and 14 for chromosome 6, while PASS managed 5 and 11 respectively. For the measures of consistency across the different replicates, shown in Table 4.5.10, BARD performed better for 4 of the 6 pairs.

4.6 Discussion

In this paper we have developed novel methodology to detect abnormal regions in multiple time series. Firstly we developed a general model for this type of problem including length of stay distributions and marginal likelihoods for normal and abnormal segments. We then derived recursions that could be used to calculate the posterior of interest and showed how to obtain iid samples from an accurate approximation to this posterior in a way that scales linearly with the length of series.

The resulting algorithm, BARD, was then compared in several simulation studies and some real data to another competing method PASS. These results showed that BARD was consistently more accurate than the PASS benchmark on several important criteria for all of the data sets we considered. Furthermore, being able to accurately and efficiently perform Bayesian inference for large and high dimensional data sets of this type allows us to quantify uncertainty in the location of abnormal segments. Before this with other methods such as PASS this quantification of uncertainty has not been possible.

Whilst we have focused on a specific model of changes in mean from some baseline level, our method could easily be adapted to any model which specifies some normal behaviour and abnormal behaviour. The only restrictions we place on this is the ability to calculate marginal likelihoods for both types of segment. The main computational bottleneck would be in the calculation of the abnormal marginal likelihoods as this involves integration over a prior for the parameter(s) which cannot be done analytically, and for higher dimensional parameters would be computationally intensive. For example, our approach can trivially be extended to allow different but known variances for each time-series. To allow each abnormal segment to

have its own variance as well as mean is possible, but would involve extra computation, as a 2-dimensional integral would be needed to calculate the marginal likelihoods for abnormal segments.

R code to run the BARD method is available at the first authors website. `http://www.lancaster.ac.uk/pg/bardwell/Work.html`. The real CNV data we analysed in Section 6.5 is available publicly and can be downloaded from the GEO accession website `http://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE25893`.

Chapter 5

Most recent changepoint detection in Panel data

5.1 Introduction

There are many modern applications where high-dimensional observations are collected and stored over time. This type of data can be viewed as a (potentially large) collection of time series and in the literature is often known as panel data. For an overview of this area see Wooldridge [2010].

We are interested in structural changes, also known as changepoint detection. For an overview of some of the methods used on univariate time series see Jandhyala et al. [2013]. In this work, however, we will look at structural changes in panel data. Some recent work in this area includes Kirch et al. [2015], Ma and Yau [2016] and Preuss et al. [2015]. Applications of these methods to detect changes occur in many areas such as finance, bioinformatics and signal processing [Cho and Fryzlewicz, 2015, Vert and Bleakley, 2010, Cao and Wu, 2015].

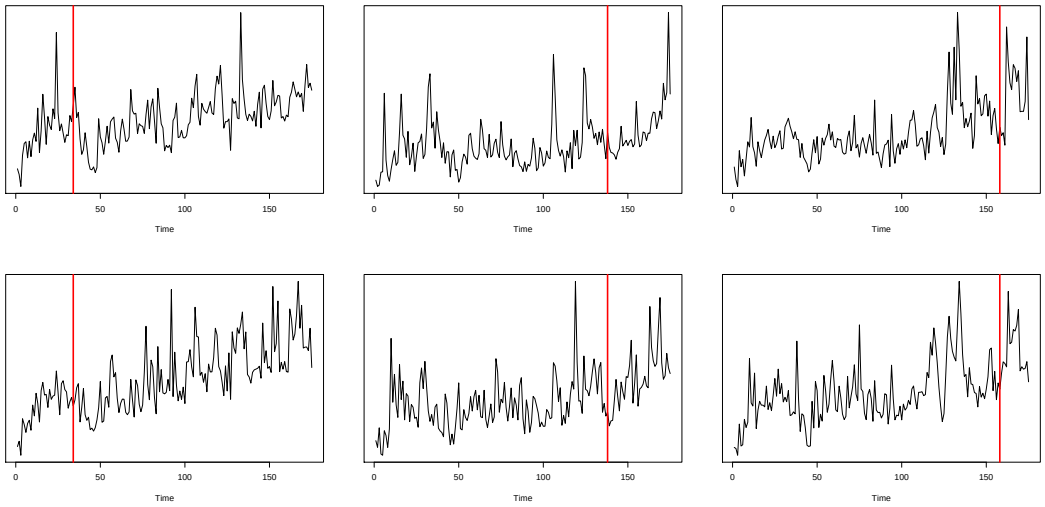


Figure 5.1.1: An example of six of the event count time-series. These show different patterns. The left-hand column has two series consistent with a constant positive trend since around week 40. The middle column show series with evidence for a recent increase in trend around week 140. The right-hand column shows series with evidence for a decrease in the rate of events from around week 160. In each case we show our estimate of the most recent changepoint – see Section 5.5.1 for more detail.

Our work is motivated by a real-life problem of predicting the number of events that occur across a telecommunications network. We have weekly data on the number of events in the network, with this number recorded for each of a set of line types and for each of a set of geographical regions. Being able to make short-term predictions of future event counts is important for planning. These event counts are observed to change over time, often abruptly, and it is natural to model the time-series data using a changepoint model.

The challenge with analysing the data is dealing with the large number of separate time-series, one for each product and region pair. In total there are 160 time-series. Six example time-series are shown in Figure 5.1.1. It is natural to assume that some reasons, such as large external factors, that affect the event count for one time-series may also affect the event counts for other time-series. However, not all time-series may see a changepoint at exactly the same time. We would like a changepoint method that has the flexibility to encapsulate,

but does not force, time-series to share common changepoints. As our primary interest is in short-term prediction, we particularly want a method that is accurate in estimating the location of the most recent change-point for each time-series, so that we can use the data since that change-point to predict the likely number of events in the future.

Detecting changepoints in multiple time-series introduces computational challenges that are not present when analysing a single time-series. A simplistic approach to the problem would thus be to try and apply univariate changepoint methods [Jandhyala et al., 2013]. There are two ways of doing this. One is to analyse each time-series separately. The other is to aggregate the time-series, and analyse the resulting univariate series. Each method has its drawbacks. The former will lose power when detecting changepoints, as it ignores the information that different time-series are likely to have changepoints at similar times. The latter approach can perform poorly if the signal from changepoints that affect a small number of series is swamped by the noise in the remaining series when they are aggregated.

An alternative approach to analysing data of this form is to treat the data as a single time-series with multivariate observations. We then model the multivariate data within a segment, and allow for this model to change, in an appropriate way, between segments. This approach is taken by Lavielle and Teyssière [2006], who model data as multivariate Gaussian but with a mean that can change from segment to segment. Similarly, Matteson and James [2014] present a non-parametric approach to detecting multiple changes in multivariate data. However, like aggregating the data, these methods may lack power if a change only affects a small proportion of the time-series. [Though see Wang and Samworth, 2016, for ideas that try to overcome this problem].

Recently there have been methods specifically designed for detecting changes that affect

only a subset of series. Cho and Fryzlewicz [2015] and Cho [2016] propose a way to detect a single, potentially common, changepoint in such data. They consider a novel, non-linear, way of combining summaries of individual time-series, so-called CUSUM statistics, that contain information about the presence and location of a changepoint. The intuition is to retain CUSUM values from all series that show strong evidence for a change at a given time-point, but down-weight the values from other time-series. Thus they are able to share information across time-series without any signal being swamped by noise from series which do not share the common changepoint.

Similarly, Xie and Siegmund [2013] introduce a generalised likelihood ratio test for detecting a single common changepoint that affects only a subset of series. This test needs an estimate of the proportion of series affected by the change, and this estimate then affects the weight given to evidence for a change from each series. Again the intution of the approach is to give large weight to series that show strong evidence for a change, but lower weight to those with little evidence. These approaches can be used within a binary segmentation procedure to find multiple changes. Empirical results in Cho and Fryzlewicz [2015] and Cho [2016] show this type of approach can be more powerful than either analysing series individually or aggregating them.

Because we are primarily interested in estimating the most recent changepoint for each time-series, we take a different approach. Our approach is focussed primarily on detecting the most recent changepoint in each time-series. It does this by partioning the panel of time-series into groups each of which share the same most recent changepoint, with, potentially, a group corresponding to time-series with no change. This is achieved by analysing each time-series independently, using a penalised cost, or penalised likelihood, approach to detecting changes

[Lavielle, 2005, Killick et al., 2012, Maidstone et al., 2017b]. From each analysis we output a measure of evidence for the most recent changepoint being at each possible time-point, or that the series has no change. We then post-process the output from these analyses in a way that encourages time-series to share a common most recent change. This post-processing step involves trying to partition the time-series into a small number, K, of groups that share the same value for their most recent changepoint. We show that this post-processing step can be formulated in terms of solving a combinatorial optimisation problem, known as the K-median problem. Whilst this problem is NP-hard, we use a heuristic solver that is computationally inexpensive, and, empirically, works well in terms of the estimated most recent changepoints.

The outline of the paper is as follows. Firstly we define the problem of finding the most recent changepoint in a univariate time series using a penalised cost approach, and show how this can be extended to panel data. To infer the most recent changepoints requires solving a combinatorial optimisation problem. We discuss how to solve this in Section 5.3. In Section 6.4 we evaluate our method, and compare to a number of alternatives on simulated data. We then apply our method to two real data applications. The first data set represents a telecommunications event time-series, shown in Figure 5.1.1, where the aim is for improved prediction. Secondly, we analyse financial data from a large number of firms. In this application we are more concerned about detecting the locations of most recent changepoints and the sets of firms that change. The aim of this is to understand the causes of these changes, for example whether they be legal changes that affect specific sectors, or wider economic changes. Finally we end with a discussion on the advantages and limitations of our method.

5.2 A Penalised Cost Approach to Most Recent Changepoint Detection

We begin by assuming we have panel data consisting of N time series of length n. Denote the ith time series by $y_{1,i}, \ldots, y_{n,i}$. Throughout we will use the notation $y_{s:t,i}$ to denote the subset of observation from time s to time t inclusive.

Our approach to detecting the common most recent changepoints is based on a penalised cost approach. We will first describe how this approach can be used to analyse individual time-series, before then explaining how the output from these individual analyses can be combined to estimate a set of common most recent changepoints for our N series.

5.2.1 Analysing a Univariate Time Series

First consider analysing data from one of the N time series in our panel data. To simplify notation we will drop the subscript that denotes which time-series, and instead denote the data by $y_{1:n}$. We we will denote the number and position of changes by m and $\boldsymbol{\tau} = (\tau_1, \ldots, \tau_m)$ respectively. We will assume the changepoints are ordered, and define $\tau_0 = 0$ and $\tau_{m+1} = n$.

A penalised cost approach to detecting changepoints in this time series involves introducing a cost associated with each putative segment. This cost is often derived by modelling the data within a segment, and defining the cost to be proportional to minus the maximum likelihood value for fitting that model to a segment of data. If our model for data in a segment is that they are IID with some density $f(y|\theta)$, where θ is a segment-specific parameter, then we can

define a cost for a segment $y_{s:t}$ as

$$\mathcal{C}(y_{s:t}) = -2 \max_{\theta} \sum_{u=s}^{t} \log f(y_u|\theta).$$

The segment cost function can include a component that depends on the length of segment as is used in some penalised cost approaches [Davis et al., 2006, Zhang and Siegmund, 2007a]. To make this idea concrete we will give two examples of cost functions that we will use later. The first is for detecting a change in mean. A simple model is that the data in a segment is IID Gaussian with common known variance, σ^2, and segment specific mean, θ. In this case we get

$$\mathcal{C}(y_{s:t}) = -2 \max_{\theta} \frac{-1}{2\sigma^2} \sum_{u=s}^{t} (y_u - \theta)^2 = \frac{1}{\sigma^2} \sum_{u=s}^{t} \left(y_u - \frac{\sum_{v=s}^{t} y_v}{t-s+1} \right)^2 .$$

The second is where we model the mean of the data within a segment as a linear function of time, but allow this linear model to vary between segments. Denote $\theta = (\theta_1, \theta_2)$ to be the segment intercept and slope. If the noise for this model is IID Gaussian we then get a segment cost

$$\mathcal{C}(y_{s:t}) = \frac{1}{\sigma^2} \max_{\theta} \sum_{u=s}^{t} (y_u - \theta_1 - u\theta_2)^2 .$$

We use this model for analysing the data presented in the introduction, however in that application some time-series have clear outliers. To make our inferences robust to these outliers we follow Fearnhead and Rigaill [2016] and instead use a segment cost

$$\mathcal{C}(y_{s:t}) = \frac{1}{\sigma^2} \max_{\theta} \sum_{i=s}^{t} \min \left\{ (y_i - \theta_1 - i\theta_2)^2 , 4\sigma^2 \right\} . \tag{5.2.1}$$

This cost limits the impact of outliers if their residuals are greater than 2 standard deviations

away from the segment mean.

For all these costs we require knowledge of σ^2, the residual variance (or in the latter example, the variance of the non-outlier residuals). In practice we use a simple and robust estimator of σ, based on the median absolute deviation of the differenced time-series [Fryzlewicz, 2014a]. Once we have defined a segment cost, we then define a cost for a segmentation as the sum of the segment costs for that segmentation. To segment the data, and find the changepoints, we then want to minimise this cost over all segmentations. However to avoid over-fitting we add a penalty, $\beta > 0$, for each segment. Thus to segment the data we solve the following optimisation problem

$$\min_{m,\boldsymbol{\tau}} \sum_{j=1}^{m+1} \left[\mathcal{C}(y_{(\tau_{j-1}+1):\tau_j}) + \beta\right]. \tag{5.2.2}$$

The choice of β in this approach is important. Higher values for β will mean fewer changepoints detected. There are various suggestions for how to choose β, and the most common for detecting changes in a single time-series is the BIC criteria. If our segment specific parameter is of dimension p, then this corresponds to $\beta = (p+1)\log n$. This has good theoretical properties, if our modelling assumptions are correct [e.g. Yao, 1987]. However care is needed in practice where this is not the case, see Haynes et al. [2017a] for guidance in selecting an optimal value for β for a given a time-series.

Solving (5.2.2) is possible using dynamic programming. This requires the solution of a set of intermediate problems. Define $F(t)$ for $t = 1, 2, \ldots, n$ as

$$F(t) = \min_{\boldsymbol{\tau}} \left\{ \sum_{j=1}^{m+1} \left[\mathcal{C}(y_{(\tau_{j-1}+1):\tau_j}) + \beta\right] \right\}, \tag{5.2.3}$$

where the minimisation is over m and $0 = \tau_0 < \tau_1 < \cdots < \tau_m < \tau_{m+1} = t$. Thus $F(t)$ is the minimum cost for segmenting data $y_{1:t}$. The functions $F(\cdot)$ can be efficiently calculated, for example using the PELT [Killick et al., 2012] or FPOP [Maidstone et al., 2017b] algorithms, as

$$F(t) = \min_{s<t} \{F(s) + \mathcal{C}(y_{s+1:t}) + \beta\}.$$

Recalling that our interest is in detecting the most recent changepoint, let us consider $G(r)$, which we define to be the minimum cost of the data conditional on the most recent changepoint prior to n being at time r. This is related to $F(r)$ as it is just the minimum cost of segmenting $y_{1:r}$ plus the cost of adding a changepoint and the cost for segment $y_{(r+1):n}$,

$$G(r) = F(r) + \mathcal{C}(y_{(r+1):n}) + \beta, \text{ for } r = 1, \ldots, n-1, \tag{5.2.4}$$

with $G(0) = \mathcal{C}(y_{1:n})$, This quantity can be viewed as related to the idea of a profile likelihood, as we have optimised over all nuisance parameters (the number and locations of the changepoints prior to the most recent changepoint). It is trivial to see that our estimate for the most recent changepoint is given by $\arg\min_{r \in \{0,\ldots,n-1\}} G(r)$. If the most recent changepoint is at $r = 0$, then this corresponds to no change within the time-series.

5.2.2 Extension to panel data

We now return to the problem of finding a set of common most recent changepoints in our panel data. Let $G_i(r)$ denote the minimum cost for segmenting series i with a most-recent changepoint at r, defined in (5.2.4). Our idea is to search for a set of K locations for the common most recent changepoints for our N series.

Firstly assume that an appropriate value for K is known. Denote a set of common most recent changepoints as $\mathbf{r}_{1:K} = (r_1, \ldots, r_K)$. For the kth most recent changepoint, located at r_k, then there will exist a set, $I_k \subset \{1, 2, \ldots, N\}$, such that all series $i \in I_k$ the most recent changepoint is located at r_k. The sets $I_1, I_2 \ldots, I_K$ will be disjoint sets that partition the full set of series $\{1, 2, \ldots, N\}$.

It is natural to estimate the $\mathbf{r}_{1:K}$, and the associated sets, by the values that minimise the sum of costs for each series

$$C_K = \min_{I_1,\ldots,I_K \; r_1,\ldots,r_K} \sum_{k=1}^{K} \sum_{i \in I_k} G_i(r_k). \tag{5.2.5}$$

The minimisation of (5.2.5) is challenging, however we will describe a method adopted from the field of combinatorial optimisation to solve it for a given value of K in Section 5.3.

In practice we do not know what value of K to choose. Thus to choose K we resort to minimising a penalised version of (5.2.5). We first solve the optimisation problem in (5.2.5) for a range of K, and then choose the value of K that minimises

$$C_K + N \log_2 K + K \log_2 n,$$

where $\log_2$ is log base two. This uses a minimum description length criteria [Grünwald, 2007], and the penalty can be viewed as the log, in base two, of the model complexity for allowing K most recent changepoints: the number of choices of the K changepoints is approximately n^K and then each of the N time-series can choose which of the K most recent changepoints to have, which gives K^N possible choices.

This approach penalises adding most recent changepoints. Thus when we implement our method we use a value of β, the penalty for adding a change used in calculating $G_i(r)$, which is slightly lower than the BIC choice. Specifically, we suggest using $\beta = (p + 1/2) \log n$, as on simulated data with no change, values of β lower than this produce $G(r)$ functions that on average get smaller as r increases for $r \geq 1$ – which suggests smaller choices of β would be biased towards adding erroneous very recent changepoints. By comparison our choice of β produced $G(r)$ functions whose average value appeared constant for $r \geq 1$.

5.3 Optimal set of most recent changepoints

We now turn to solving the optimisation problem in (5.2.5) for a fixed value of K. Solving this is computationally challenging if a brute force method is applied, due to the exponentially large number of ways of choosing either $\mathbf{r}_{1:K}$ or the sets $I_{1:K}$. However it can be reduced to a well studied problem in the field of combinatorial optimisation.

To formulate this problem we proceed as follows. Let $\mathbf{G}$ be a matrix of the conditional costs that we defined in (5.2.4), so that for $i = 1, 2, \ldots, N$ and $r = 0, 1, \ldots, n-1$, $\mathbf{G}_{ir} = G_i(r)$ the optimal cost of the most recent changepoint of the ith series being at time r. We want to find the K columns of $\mathbf{G}$ such that if, for each row, we take the minimum of elements in these columns, and then sum these across all N rows, the total is minimised. This allocates each of the N series into K disjoint classes according to which series are affected by a specific most recent changepoint. The specific optimisation problem is

$$\min_{S} \sum_{i=1}^{N} \min_{r \in S} \mathbf{G}_{ir}, \text{ where } S \subset \{0, 1, \ldots, n-1\} \text{ and } |S| = K.$$

It turns out that this optimisation problem is mathematically equivalent to the so-called K-median problem [Reese, 2006]. This problem can be formulated, and solved, as an integer program with binary variables x_{ir} and z_r where

$$x_{ir} = \begin{cases} 1 & \text{if series } i \text{ has most recent changepoint at time } r \\ 0 & \text{otherwise,} \end{cases}$$

and

$$z_r = \begin{cases} 1 & \text{if there is a most recent changepoint in any series at time } r \\ 0 & \text{otherwise.} \end{cases}$$

The objective is then simply to solve the following problem:

$$\min \sum_{i=1}^{N} \sum_{r=0}^{n-1} \mathbf{G}_{ir} x_{ir} \tag{5.3.1}$$

$$\text{subject to} \quad \sum_{r=0}^{n-1} x_{ir} = 1, \forall i, \tag{5.3.2}$$

$$x_{ir} \leq z_r, \forall i, r, \tag{5.3.3}$$

$$\sum_{r=0}^{n-1} z_r = K. \tag{5.3.4}$$

Here constraint (5.3.2) ensures each series has only one most recent changepoint, whilst the two remaining constraings, (5.3.3) and (5.3.4), ensure that K different most recent changepoints are selected.

Approaches for solving the K-median problem are discussed in Reese [2006] and references therein. We use the method of Teitz and Bart [1968], available within the `R` package `tbart`.

This is a simple algorithm that tries to improve on a current solution by replacing one of the K values for a most recent changepoint with a value that is not currently in the set of most recent changepoints. It loops over all such pairs, and makes the replacement if it will reduce the objective function (5.3.1). This is repeated until there is no replacement that will improve the objective any further. This method is heuristic, in that it is not guaranteed to find the global optimum to the optimisation problem. However we found that it is computationally efficient and empirically leads to good estimates of the most recent changepoints.

Pseudo-code for full MRC method is shown below.

Algorithm 4: Pseudo-code for the MRC algorithm.

Input: A data set containing N series made up of n observation, $\mathbf{y} \in \mathbb{R}^{N \times n}$.

A cost function $\mathcal{C}(\cdot)$ used as a measure of fit, usually the negative log-likelihood for the data in a segment of the assumed model.

A penalty term β.

A constant K_{max}, the maximum number of most recent changes to find.

Pre-process: Run the PELT algorithm on each series in the panel and calculate the matrix containing the costs for a MRC in each series at a given time point.

for $i = 1$ **to** N **do**

1. Calculate $F_i(t)$ using PELT for all time points $t = 1, 2, \ldots, n$ in series i.

2. Calculate the conditional costs for a MRC in series i at time t

$$\mathbf{G}_{i,t} = F_i(t) + \mathcal{C}(\mathbf{y}_{i,(t+1):n}) + \beta$$

and store in a matrix (use corrected value for β as described in 5.2.2).

end

Optimisation: Apply the K-median problem to the $\mathbf{G}$ matrix.

3. Calculate C_K for all $K = 1, 2, \ldots, K_{max}$.

5. Find the number of most recent changepoints by minimising the MDL,

$$\hat{K} = \underset{K \in \{1,\ldots,K_{max}\}}{\arg\min} \; C_K + N \log_2 K + K \log_2 n.$$

Output : The $\hat{K}$ most recent changepoints along with a matrix ($\mathbf{x}$) from the IP formulation that gives the MRC for each series.

5.4 Simulation study

As described in the introduction, there are a number of methods in the literature that allow us to detect multiple changes in panel data. We compare our method, which we call MRC, to several of these to see empirically how they compare. None of these alternative methods were specifically designed to just estimate the most recent changepoints, and we are unaware of any other methods that focus solely on this. Furthermore some of these methods are able to infer quantities, such as earlier common changepoints, that MRC cannot.

The alternative methods can be split into two groups. The first set of methods estimate common change points for each series. We compare with three such approaches. These are analysing the aggregated data (AGG) and two approaches for detecting common changepoints in multivariate data. The latter two methods are the approach of Lavielle and Teyssière [2006] which models data within a segment as multivariate Gaussian with known covariance (MV); and the ECP method [Matteson and James, 2014], which is a non-parametric change point detection procedure (ECP).

Both the AGG and MV methods require a choice of penalty and we use the BIC penalty. However, for the ECP method every proposed changepoint is tested for statistical significance using a permutation test and a threshold obtained via a bootstrap which is described in Matteson and James [2014].

The second group of alternative methods includes two methods that can estimate common changepoints that affect only a subset of the time-series. The simplest method we consider (IND) involves analysing each series in the panel independently and finding the most recent changepoint in each series. The second method in this group is Double CUSUM Binary

Segmentation (DCBS) [Cho, 2016] which also identifies the subsets affected by each change. We again use the BIC penalty when segmenting each series as part of the IND method. The DCBS method has two parameters that need to be chosen. The first parameter ψ, is related to the expected degree of sparsity or the number of series affected by a change compared to the total number of series. Guidance is available on how to choose this parameter in Cho [2016]. The second parameter π^{ψ} is the threshold for testing whether or not a change is significant as is done in the ECP method mentioned above. This threshold is chosen using a bootstrap style procedure where the null hypothesis of no changepoint is assumed and some empirical quantile of this distribution is taken. We chose this parameter by simulating 100 replications from the null hypothesis, i.e. no changepoints at all, and measured the proportion of false positives for a number of different values for π^{ψ}. In practice, we found that a value of $\pi^{\psi} = 10$ worked well.

Each panel data set we simulate consists of 100 series all having length 500. For a given value of K most recent changepoint we first simulate K distinct values for the most recent changepoints and these are sampled at random from the set $\{300, 320, \ldots, 480\}$. This ensures each most recent changepoint position is at least 20 time-points away from all other positions, which helps interpretation when we measure the accuracy of methods in detecting the location of the changes. We partition our 100 time-series evenly across the K most recent changepoint locations. We then simulate earlier common changepoints by first simulating potential changepoints independently with probability 0.02 at each time-point prior to the earliest most recent common changepoint. For each of these we simulate a probability from a uniform distribution, and then simulate that a changepoint appears in each time-series independently with this probability. The observations in each of the segments are IID and

Normally distributed with mean μ drawn from its prior distribution $\mathcal{N}(0, 2^2)$. For simplicity we keep a fixed variance $\sigma^2 = 1$ for all the observations. In this study the parameter of the last segment differs by ϵ from the mean in the penultimate segment, with the sign of the change being chosen uniformly at random for each time-series. We use $\epsilon = 1$ for the studies in Cases 1, 3 and 4 below, whereas for Case 2 we look at the effect of varying ϵ.

In the first three studies we consider the accuracy of estimates of the most recent changepoints and which series are affected. We only compare IND, DCBS from the second group of methods and our method MRC. We exclude those methods in the first group because they only detect common changes in the panel and are unable to identify which series are affected by each of the different most recent changepoints. We evaluate these three methods on a number of different criteria: the proportion of the true changepoints we detect (PD), the changepoint accuracy (CA) and the location accuracy (LA). A changepoint is defined as being detected if it is within 5 time points of the truth. To define the location accuracy we take only those changepoints that are detected then take the average of their absolute deviations from the true most recent changepoint in each series. For every series in each panel that we simulate we calculate whether or not the most recent change in that series has been detected and the accuracy of the location estimated compared to the truth. We then take the average over the 100 series in that panel and do this for every panel we simulate.

Two of the methods we consider, namely MRC and DCBS, return more information than IND, including the estimated number of most recent changepoints $\hat{K}$ and the subset of series that are affected by each most recent changepoint.

We measure the accuracy of the estimate of the number of most recent changepoints using the absolute error, $|\hat{K} - K|$, and call this the changepoint accuracy (CA).

We then measure the accuracy of the estimates of the subsets of series affected by each of the changepoints using the set coverage

$$D_j = 1 - \frac{|\hat{I}_j \cap I_j|}{\sqrt{|\hat{I}_j||I_j|}},$$

where I_j is the true subset of series affected by the j most recent changepoint and $\hat{I}_j$ is the estimated subset. This measure satisfies $D_j \in [0, 1]$, with $D_j = 0$ indicating that the estimated subset overlaps exactly with the true subset, and $D_j = 1$ if the two subsets are disjoint. More generally, smaller values of D_j indicate a greater overlap. In the simulations presented for each panel we calculate the mean of $D_1, D_2, \ldots, D_{\hat{K}}$.

Case 1. Effect of K.

For the first study we simulated data as described above for a range of values for K from $K = 1$ to $K = 10$. Results are shown in Table 5.4.1.

	IND		DCBS				MRC			
k	PD	LA	PD	CA	LA	D	PD	CA	LA	D
1	0.73 (0.11)	1.46 (0.25)	0.86 (0.23)	2.98 (2.46)	0.05 (0.21)	0.09 (0.15)	0.98 (0.07)	0.10 (0.44)	0.06 (0.31)	0.01 (0.04)
2	0.76 (0.07)	1.43 (0.26)	0.82 (0.19)	2.55 (1.86)	0.04 (0.22)	0.14 (0.16)	0.97 (0.03)	0.04 (0.20)	0.04 (0.19)	0.03 (0.03)
3	0.77 (0.05)	1.39 (0.22)	0.70 (0.19)	2.64 (2.01)	0.13 (0.34)	0.24 (0.15)	0.95 (0.03)	0.05 (0.22)	0.03 (0.15)	0.05 (0.03)
4	0.77 (0.05)	1.37 (0.25)	0.67 (0.13)	2.47 (2.23)	0.18 (0.36)	0.28 (0.11)	0.94 (0.04)	0.03 (0.17)	0.05 (0.15)	0.06 (0.03)
5	0.78 (0.05)	1.38 (0.22)	0.58 (0.12)	2.09 (1.72)	0.24 (0.39)	0.35 (0.12)	0.93 (0.03)	0.03 (0.17)	0.04 (0.11)	0.07 (0.03)
10	0.78 (0.04)	1.41 (0.20)	0.29 (0.07)	1.77 (1.37)	0.76 (0.68)	0.62 (0.10)	0.89 (0.04)	0.10 (0.30)	0.19 (0.17)	0.10 (0.04)

Table 5.4.1: For all of the methods and differing values of K we repeated each experiment 100 times and recorded the proportion of true changes we detected (PD), the accuracy in detecting the number of distinct most recent changes (CA), the accuracy of the estimated location of these changes (LA) and the set coverage (D). These values are averaged over the 100 replications alongside their standard deviation, shown in brackets.

It is clear from Table 5.4.1 that our MRC method outperforms both IND and DCBS across the criteria we consider. The ability to synthesise information across time-series means that MRC is able to more accurately detect changes and locate where they occur than analysing

each time-series independently. Not surprisingly we see that the advantage of using MRC over IND decreases as K increases. We also see that DCBS is more accurate than IND for small values of K, and is consistently more accurate in estimating the position of detected changepoints, but appears less powerful at detecting the most recent changes as K increases.

Case 2. Effect of size of change at final changepoint.

Next we look at how each method is effected by the size of the mean change at the most recent changepoint, ϵ. We fix the number of most recent changepoints as $K = 5$, meaning that there are 20 series affected by each different changepoint. We vary the value of ϵ from $\epsilon = 0.2$ to $\epsilon = 1.6$. Results are shown in Table 5.4.2.

We again see MRC giving consistently stronger performance for all values of ϵ. The advantage of MRC over IND is largest for moderate values of ϵ. For small values of ϵ the information about changes in each time-series is small, and thus the benefit of merging information across time-series is limited. For larger values of ϵ it is relatively easy to detect changes from an individual time-series, and hence the benefit of using MRC over IND is mainly seen in its ability to more accurately locate the position. Surprisingly DCBS does not improve as much as the other methods as we increase ϵ. The DCBS method was not specifically designed to detect most recent changes, and it appears not to be as accurate at identifying which time-series change at each changepoint, which then impacts its accuracy at detecting which changes are most recent for a given time-series.

Case 3. Dependent observations.

One of the key assumptions we made when modelling the most recent change process was the independence of observations, both within and between segments. This greatly simplifies the modelling and especially the inference procedure, however, in many real time series

	IND		DCBS				MRC			
ϵ	PD	LA	PD	CA	LA	D	PD	CA	LA	D
0.2	0.11 (0.12)	1.49 (1.13)	0.09 (0.07)	3.49 (1.61)	0.75 (1.26)	0.66 (0.10)	0.11 (0.15)	2.06 (1.12)	0.47 (0.88)	0.64 (0.16)
0.4	0.22 (0.11)	1.75 (0.53)	0.22 (0.10)	3.43 (1.66)	1.01 (1.12)	0.55 (0.11)	0.36 (0.19)	1.27 (1.07)	0.88 (1.05)	0.42 (0.11)
0.6	0.46 (0.09)	1.76 (0.34)	0.37 (0.10)	3.00 (1.92)	0.64 (0.69)	0.51 (0.09)	0.76 (0.12)	0.30 (0.50)	0.29 (0.33)	0.20 (0.06)
0.8	0.65 (0.07)	1.57 (0.25)	0.48 (0.13)	2.51 (1.96)	0.35 (0.50)	0.45 (0.11)	0.89 (0.06)	0.09 (0.29)	0.12 (0.21)	0.10 (0.04)
1	0.78 (0.05)	1.38 (0.22)	0.58 (0.12)	2.09 (1.72)	0.24 (0.39)	0.35 (0.12)	0.93 (0.03)	0.03 (0.17)	0.04 (0.11)	0.07 (0.03)
1.2	0.86 (0.04)	1.19 (0.17)	0.62 (0.13)	1.63 (1.47)	0.16 (0.25)	0.31 (0.13)	0.95 (0.03)	0.04 (0.20)	0.02 (0.06)	0.05 (0.03)
1.4	0.91 (0.04)	1.01 (0.14)	0.65 (0.12)	1.44 (1.25)	0.13 (0.26)	0.26 (0.13)	0.95 (0.03)	0.04 (0.20)	0.00 (0.03)	0.05 (0.03)
1.6	0.93 (0.03)	0.85 (0.13)	0.66 (0.12)	1.47 (1.25)	0.10 (0.23)	0.25 (0.13)	0.96 (0.03)	0.04 (0.20)	0.00 (0.02)	0.04 (0.02)

Table 5.4.2: For all of the methods with a fixed value of $K = 5$ and differing values of ϵ we repeated each experiment 100 times and recorded the proportion of true changes we detected (PD), the accuracy in detecting the number of distinct most recent changes (CA), the accuracy of the estimated location of these changes (LA) and the set coverage (D). These values are averaged over the 100 replications alongside their standard deviation, shown in brackets.

applications observations are not independent and display serial autocorrelation.

To assess the robustness of the MRC procedure we simulated an MRC process with a piecewise constant mean function as before but instead of adding IID normally distributed 'noise' we simulated an $AR(1)$ noise process, Z_t, with standard normal errors e_t

$$Z_t = \phi Z_{t-1} + e_t.$$

This process was simulated for a range of values of ϕ which represented mild to moderate autocorrelation. The number of most recent changepoints was fixed at $K = 5$ and we set $\epsilon = 1$. Results are shown in Table 5.4.3.

As ϕ increases the dependence between observations increases and the measures for all methods we consider decrease. The impact on both MRC and DCBS is larger than the impact on IND, with IND correctly detecting more recent changepoints for $\phi = 0.4$. Both MRC and DCBS still give more accurate estimates of the position of the changes that they do detect, and MRC is again more accurate than DCBS for all cases we consider. It may be possible to

	IND		DCBS				MRC			
ϕ	PD	LA	PD	CA	LA	D	PD	CA	LA	D
0	0.78 (0.05)	1.38 (0.22)	0.58 (0.12)	2.09 (1.72)	0.24 (0.39)	0.35 (0.12)	0.93 (0.03)	0.03 (0.17)	0.04 (0.11)	0.07 (0.03)
0.1	0.73 (0.04)	1.47 (0.23)	0.55 (0.12)	2.17 (1.95)	0.29 (0.37)	0.40 (0.12)	0.89 (0.04)	0.03 (0.22)	0.07 (0.15)	0.11 (0.03)
0.2	0.64 (0.05)	1.56 (0.26)	0.43 (0.13)	2.66 (1.90)	0.31 (0.48)	0.49 (0.11)	0.75 (0.08)	0.74 (0.81)	0.18 (0.28)	0.21 (0.05)
0.3	0.51 (0.05)	1.59 (0.24)	0.12 (0.09)	3.05 (1.20)	0.66 (1.01)	0.71 (0.12)	0.47 (0.09)	1.71 (1.38)	0.35 (0.38)	0.38 (0.05)
0.4	0.36 (0.05)	1.72 (0.29)	0.04 (0.04)	2.70 (1.06)	1.59 (1.49)	0.80 (0.13)	0.21 (0.09)	1.79 (1.24)	0.82 (0.90)	0.55 (0.07)

Table 5.4.3: For all of the methods and differing values of ϕ we repeated each experiment 100 times and recorded the proportion of true changes we detected (PD), the number of false positives (FP), the accuracy of estimated location of these changes (LA) and the set coverage (D). These values are averaged over the 100 replications alongside their standard deviation, shown in brackets. Fixed values for $K = 5$ and $\epsilon = 1.0$ were used.

improve the performance of all methods by increasing the penalty or threshold the defines when we add a change [see Lavielle and Moulines, 2000, for theoreticaly justification of this].

Case 4. Accuracy of prediction.

Finally, we consider at how each method performs if the aim is to predict $Y_{i,n+1}$ for each time-series. Each method gives an estimate for the most recent changepoint for each time-series. Conditional on this estimate we can estimate the mean in the final segment. This estimated mean is our prediction for the next value(s). We use the same data as in Case 1 but leave out 5 time points at the end of the data. We then predict the last 5 points using the most recent changepoints found by each method and measure the Mean Squared Error (MSE) between the truth and our predictions. Results are shown in Table 5.4.4.

k	IND	AGG	MV	ECP	DCBS	MRC
1	1.04 (0.10)	1.29 (0.84)	1.01 (0.07)	1.09 (0.28)	1.08 (0.19)	1.01 (0.07)
2	1.06 (0.09)	1.27 (0.31)	1.05 (0.08)	1.09 (0.15)	1.08 (0.12)	1.03 (0.07)
3	1.04 (0.07)	1.25 (0.28)	1.06 (0.11)	1.08 (0.13)	1.07 (0.11)	1.02 (0.06)
4	1.04 (0.08)	1.34 (0.37)	1.08 (0.11)	1.09 (0.12)	1.08 (0.12)	1.02 (0.07)
5	1.04 (0.07)	1.23 (0.24)	1.08 (0.10)	1.09 (0.12)	1.07 (0.11)	1.02 (0.06)
10	1.04 (0.06)	1.29 (0.28)	1.12 (0.08)	1.09 (0.09)	1.09 (0.09)	1.02 (0.06)

Table 5.4.4: The average Mean Squared Error (MSE) for predictions of each method. The MSE was calculated for the difference between the truth and predicted values and averaged over 100 replications.

MRC gives the most accurate predictions for all values of K, and is the only method to consistently be more accurate than analysing each time-series individually. The method which treats the N time-series as a multivariate time-series where the mean changes in all components at a change (MV) does well for $K = 1$ and $K = 2$, but loses accuracy for larger K. The method that aggregates the time-series, and then detects changes in the resulting uni-variate time series does particularly poorly. This is because the aggregation step reduces the signal for a change, even when all changes are in the same location, as the sign of the change in mean differs across time-series.

Computational cost and scaling

The computational cost of the different methods is compared in Table 5.4.5 for a typical data set used in the simulation study above. The data we used contained $N = 100$ series and had a length of 500 time points, $n = 500$.

	Methods				
	AGG	ECP	IND	DCBS	MRC
Average run time (seconds)	0.01	9.94	0.96	1.84	5.14

Table 5.4.5: Average run time calculated on 10 replications of the same data set which was simulated with fixed values for $K = 5$ and $\epsilon = 1.0$.

The `ecp` package and function was used with its default values, this method makes no distributional assumptions on the data and has quite a complex procedure for changepoint detection. Our MRC method is reasonably quick on a moderately sized panel data set and is around two and a half times slower than the leading competitor, the DCBS method.

Our MRC method "contains" the IND method in its pre-processing step (see Algorithm 4) as the univariate changepoint method PELT is applied to each series individually. Thus, this pre-processing step will scale approximately linearly in both n and N i.e. $\mathcal{O}(nN)$. This

is because PELT scales linearly in the length of the data n subject to some conditions. Calculating the conditional costs is relatively straightforward as the costs $\mathcal{C}(y_{(t+1):n})$ are derived from summary statistics that are stored.

In Section 5.5.2 we apply the MRC method to data with $N = 7039$ firms over an observation period of $n = 53$ years. Even with this large number of firms the time needed to solve the K-median problem only increased to around 30 seconds. This scaling suggests that the number of dimensions is not particularly important in terms of the time needed to solve the second step of our method. As the length of the data (n) increases however, the computational cost should increase linearly. This is because for each new time point there are another K alternatives used to compute the objective function. As each of the current K MRC's can be swapped with the new time point and the objective evaluated at each of them to see if a swap should be performed.

So far we have fixed the maximum number of most recent changepoints we wanted to search for to $K_{max} = 10$. With this choice the heuristic in the `tbart` package gave fast and efficient results, however we may want to increase this. Figure 5.4.1 shows how the computational cost changes as K_{max} increases. For problems with many most recent changes it may be sensible to get an estimate for the earliest change of interest and analyse only the data after that so that the number of true MRC's is limited to only those of interest.

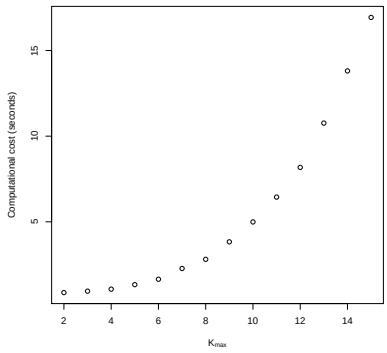


Figure 5.4.1: The computational cost when we changed the maximum number of most recent changepoints to search for.

5.5 Applications

We look at two different applications of our method using real data. These applications differ in their focus and the aim of the analysis. The first is that of the telecommunications event count data introduced in Section 5.1. For this example we segment series assuming a piecewise linear regression model which is robust to the presence of outliers.

Our second application concerns the balance sheets of a large number of firms. In this case we look for changes in a parameter that measures the ratio between the cash holdings of a company and the net assets held on its balance sheet. The goal of this analysis is to explore why the cash holdings of many large firms have increased over time, and if there are any specific events which have caused this. By using our method we can identify in which years a change occurs and for each of these years which firms change. This information helps us to tie in specific legal or economic changes to the years in which they happened and the types of industries that are affected.

5.5.1 Telecommunications event data

Our panel data consists of the number of events that occur each week over a 175 week period. Events are recorded for each of 10 geographical regions, 8 different line types, and 2 different age classes (binary variable for age that exceeds a certain threshold). Thus there are 160 possible series, of which 18 of these show no weekly events over the 175 weeks measured. So we are left with 142 series to analyse.

We can get an overall time series for the number of events per week across the entire network if we aggregate all of these series together for the 175 weekly observations over all line attributes. This fully aggregated series is shown in Figure 5.5.1. We can see that there are distinct changes in the slope of this series and it is segmented into piecewise linear regressions. The exact cost function we use to model the data within a segment is shown in (5.5.1). This is a piecewise linear regression model which uses a bi-weight loss function to limit the impact of outliers on the inference for the slope and intercept parameters

$$\mathcal{C}(y_{s:t}) = \frac{1}{\sigma^2} \max_{\theta} \sum_{i=s}^{t} \min \left\{ (y_i - \theta_1 - i\theta_2)^2 , 4\sigma^2 \right\} . \tag{5.5.1}$$

In the cost function above we have defined a data point to be an outlier if its residual is greater than two standard deviations away from the segment mean. We also require knowledge of σ^2, the variance of the non-outlier residuals). In practice we use a simple and robust estimator of σ, the median absolute deviation of the differenced time-series.

As mentioned in the introduction, the main interest with this data is in making short-term predictions. To do this we use the method described in Section 5.3 to find the number of most recent changes. We estimate that there are five different most recent changepoints.

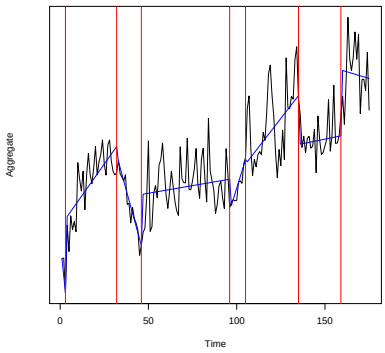


Figure 5.5.1: The aggregate series segmented into piece wise linear regressions.

This means that all of the 142 series can be separated into five groups depending upon which of the five most recent changepoint affects each series.

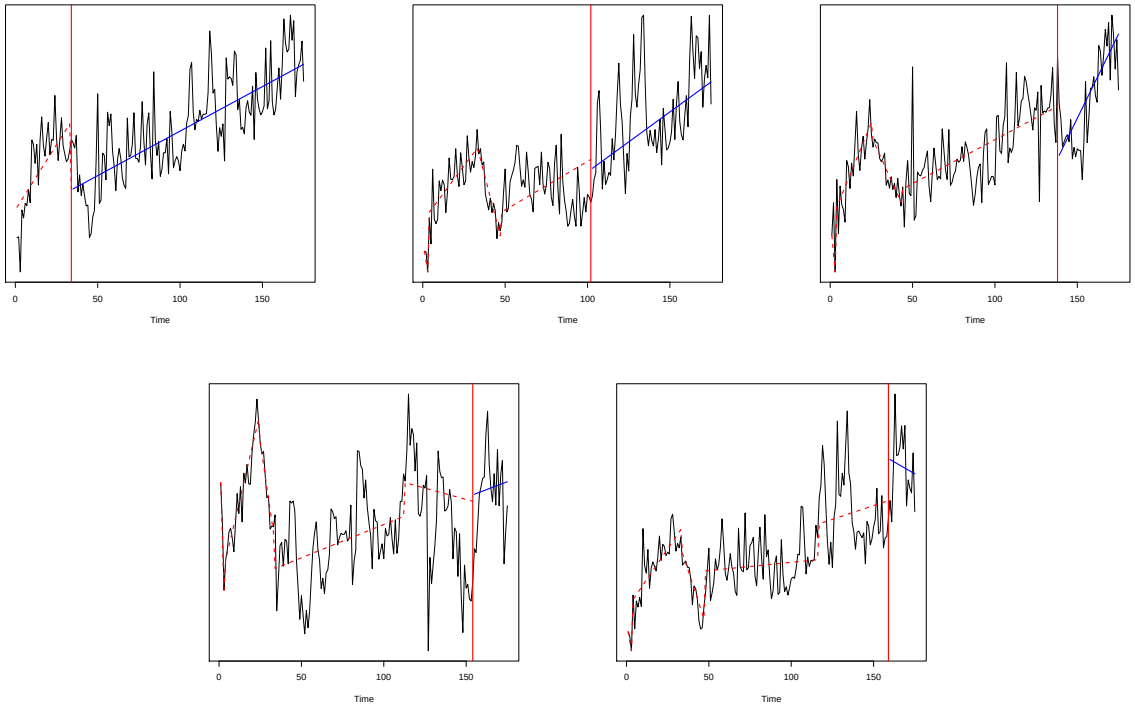


Figure 5.5.2: The aggregate series for each of the five groups of series. Their respective most recent changepoints are added, with the final segment shown in blue. The previous segmentation prior to the most recent change is shown as a red dashed line.

Figure 5.5.2 show the aggregate series for each of the five groups. The groups contain 26, 27, 28, 28 and 33 series from left to right respectively.

All of the aggregated series show an increased trend initially until around the 35^{th} week. This can be seen most prominently in the first series on the left, with a lower consistent gradient after this change. The second series shows that at around the 100^{th} week the gradient of the trend increases slightly. In the third series at around the 140^{th} week the gradient of the final segment increases markedly. The fourth and fifth series both show a most recent change which is close to the end of the series, at around the 160^{th} week, with a marked decrease in trend for the fifth series.

We can see several characteristics of the fully aggregated series in Figure 5.5.1 "stripped" almost into their component parts. The fourth series is somewhat of an anomaly as it is highly variable, upon further inspection this set of series was made up of individual series which all contained a small number of events per week and were quite variable.

When we have found the most recent changepoints, the parameters of the resulting regression line in the last segment can be estimated. These estimates can then be used to predict succeeding time points. We analyse the data up to four data points (weeks) from the end of the data and then use the predictions obtained from the regression model to evaluate the Mean Square Error (MSE) of the prediction for the last four weeks.

We compare predictions using the estimated most recent changepoints from MRC with predictions where we segment each time-series separately. The MSE for the predictions in the latter case is 43442 while for our algorithm (with $K = 5$) it is 41779. This is an improvement of 3.8% in the MSE of the prediction compared to analysing each time-series individually.

5.5.2 Corporate finance data

We now apply our method to a panel data set from the field of Corporate finance. This data set comprises the annual value of a range of different financial indicators for a number of firms. These include, for example, the value of a firm's assets or whether the firm pays dividends or not. This particular data set is known as an unbalanced panel as the observations for each firm do not all begin or end in the same year. We can view this as a longitudinal data problem where the cohort are the firms that are tracked over time. As is common in these problems there is a large (cohort) number of firms, 7039 in this example, but these are observed over a much smaller time frame. In this case there are a maximum 53 observations per firm (annually from 1962 - 2015).

An intriguing phenomenon in corporate finance is the fact that U.S. firms hold considerably more cash nowadays compared with a few decades previously. Specifically, cash as a proportion of total assets held by U.S. firms has more than doubled in the past three decades. The evolution of corporate cash holdings has received a lot of attention from academic researchers, policy makers, and practitioners. Numerous explanations for this have been offered in the literature, inclduing increased cash flow volatility [Bates et al., 2009, 2017], competition [Brown and Petersen, 2011], changes in production technology [Gao, 2017], changes in the cost of carry [Azar et al., 2015].

Azar et al. [2015] argue that changes over time in the cost of carry, that is the net cost of financing one dollar of liquid assets, explains the evolution of corporate cash holdings [see also Graham and Leary, 2016]. They measure the cost of carry as the spread between the risk-free Treasury-bill rate and the return on the portfolio of liquid assets for the corporate sector. However a limitation of existing studies is that they split their data along the time

domain into distinct 'regimes' by eyeballing the data. Such an approach is highly subjective and increases the opportunity for data snooping. It would be preferable to introduce a formal procedure for detecting any distinct regimes.

We therefore re-examine the ability of the cost of carry to capture variation in corporate cash by formally modelling the breakpoint process using our changepoint methodology. Our analysis follows Azar et al. [2015] and therefore uses the same dataset [see Azar et al., 2015, for a detailed description of the dataset]. We control for a number of variables that may affect cash holdings of a firm, such as capital expenditure, spending on R&D and the amount of leverage it has amongst others. Specifically we consider a fixed effects linear model where the response variable, y_{it}, represents the cash to net asset ratio of firm i in year t is regressed against 12 covariates,

$$y_{it} = \alpha_i + \beta_1 X_{1it} + \beta_2 X_{2it} + \ldots + \beta_{12} X_{12it} + \epsilon_{it}. \tag{5.5.2}$$

These covariates are described in Table 5.5.1. The β_js are pooled estimates of the effect of the covariates measured over all 7039 firms and the years in which they are observed. Each fixed effect term, α_i, captures a firm-specific characteristic in terms of a firm specific intercept. These fixed effects can be interpreted as the difference between the predicted cash to net assets ratio and the true value observed. As such, the fixed effects are able to capture differences caused by external changes which can not be explained by the covariates in the model.

For a specific firm the fixed effects term may change due to a number of factors such as a CEO change, a merger or takeover by another firm or some scandal such as a product recall

which requires large amounts of cash to be spent. However, we are more interested in the times at which the fixed effect parameter changes in a significant number of firms at the same time. The causes of these changes would be due to wider economic events such as changes in policy, technological innovation, or regulatory changes such as the Sarbanes-Oxley Act of 2002.

Having estimated the β_js via maximum likelihood estimate, we can rewrite (5.5.2) as a change in mean model

$$y_{it} - \left(\hat{\beta}_1 X_{1it} + \hat{\beta}_2 X_{2it} + \ldots + \hat{\beta}_{12} X_{12it}\right) = \alpha_{it} + \epsilon_{it}, \tag{5.5.3}$$

where the $\hat{\beta}_i$s are the parameter estimates.

Our MRC method can be applied to this problem and aims to find the year(s) in which the most recent changepoint(s) occur and the subsets of firms that are affected. We now follow the method of Section 5.3 to find the optimal number of most-recent changepoints and the sets of firms that are affected by them.

The Estimated Changepoints

We find three most-recent changepoints. These are located in years 1979, 1996 and 2007. The largest subset of firms have their most recent change at 1979. Approximately 70% of the 7039 firms we consider have some evidence for a change in their fixed effect parameter in 1979. This date corresponds to a change in the Federal Reserve's operating procedures, specifically it marks the beginning of the 'monetarist policy experiment', and is identified as a breakpoint in Pettenuzzo and Timmermann [2011] who use a historical time series of

excess returns that are subject to breaks to forecast the equity premium out-of-sample.

Another benefit of our methodology is the ability to observe which firms are undergoing a change and which are not. This is of real interest in economic and finance applications because it may be able to provide information to help identify the underlying cause of the structural break. For instance if the change is experienced predominantly by firms in one industry it could be indicative of an industry-specific shock or regulation change. Conversely, if the change occurs across all firms this might suggest an economy-wide change in policy. In this application the affected firms are roughly equally distributed across each of the broader industry classes strengthening the case for the cause being the change in the Federal Reserve's macroeconomic policy. This policy change led to a decrease in the fixed effects part of the model in the majority of the firms that were affected by the change and thus a decrease in their cash holdings.

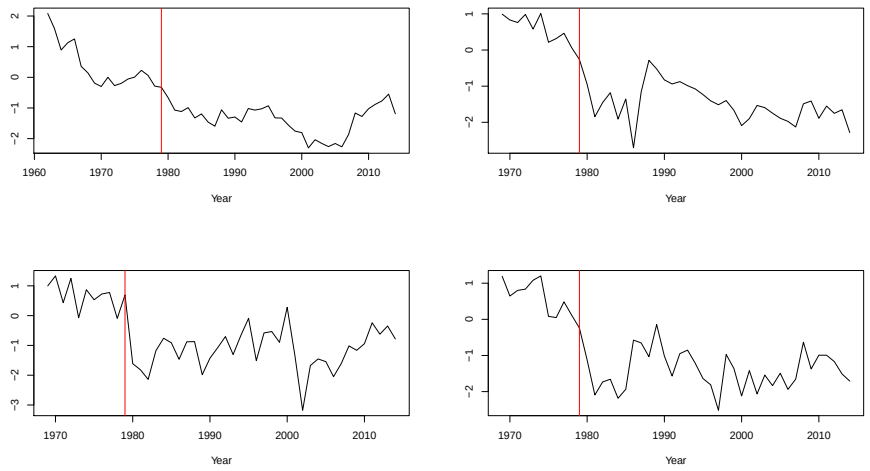


Figure 5.5.3: Some of the affected firms plots of their fixed effects showing a change in 1979.

The changes at 1996 and 2007 each affect around 15% of the firms. The change in 1996 affected mostly Utilities firms and the one in 2007 affected both the Trade and Services sec-

tors. The 1996 changepoint is also found in related work on structural breaks [Pettenuzzo and Timmermann, 2011] and can be attributed to the late 1990's retail bull market in which net assets markedly increased in value. The Telecommunications Act of 1996 which deregulated the U.S. broadcasting and telecommunications markets could explain why the Utilities sector experienced a large shock. The deregulation paved the way for many utilities companies to enter the broadcasting and telecommunications market. The change in 2007 corresponds to the recent financial crisis and the large fluctuations in the value of assets held by many firms in that period.

Covariate	Description
X_{1it}	T-Bill (the rate of return on a 90 day treasury bill)
X_{2it}	Cost of carry
X_{3it}	Log of real assets
X_{4it}	Industry sigma (a measure of the volatility in each sector)
X_{5it}	Cash flow to assets ratio
X_{6it}	Net working capital to assets ratio
X_{7it}	R&D/Sales
X_{8it}	Dividend dummy
X_{9it}	Market to book ratio
X_{10it}	Capital expenditure
X_{11it}	Leverage
X_{12it}	Acquistion activity

Table 5.5.1: A description of the 12 covariates in the model.

5.6 Discussion

In this paper we have developed novel methodology to detect changepoints in panel data. The specific changepoints we aim to detect are the most recent changes that affect different and disjoint subsets of the series that make up the panel. We focus on detecting the most recent changes as this can be useful in forecasting as shown in Section 5.5.1. We are also able to identify which series are affected by different changes which leads to a greater understanding of why and how the changes have occurred.

In our analysis of the two real data sets we used cost functions for segmenting each individual time-series that are based on assuming no temporal dependence in the residuals. This can be justified theoretically by results that show, for example, that detecting changes in mean using a least squares criteria is robust to the presence of temporal dependence in the residuals [Lavielle and Moulines, 2000]. We showed empirically that our method can still detect the most recent changes even in the presence of AR(1) structure. Furthermore, our general approach can easily be extended to allow for modelling of the error structure of the residuals, by using cost functions for the data within each segment that are based on models which allow for autocorrelation.

Our method also ignores any dependence across time-series, either in the form of cross-correlation in the residuals or of similar changes at common changepoints. Whilst the former is an active area of research within the non-stationary time series community [see for example Ombao et al., 2005, Park et al., 2014] this is an open, and intriguing area of future research for the changepoint community. The consquence of ignoring such (time-varying) structure might be that we infer some spurious changes to fit unusual patterns in the residuals that

are seen in multiple time-series. It is not clear how to develop a method that accounts for the latter, but such a method could have greater power at detecting changes than our MRC procedure.

Chapter 6

Changepoint detection for piece-wise linear models in the presence of outliers

6.1 Introduction

A problem often found when performing changepoint detection on real data is being able to distinguish between actual changes that occur in the time series and outliers that exist in the data due to measurement or experimental error.

Currently most changepoint detection methods do not give robust segmentations of data when outliers are present. Those methods that use a penalised likelihood approach [Killick et al., 2012] assume simple parametric models such as Gaussian noise and thus over estimate the number of changes when outliers are present (i.e. the true noise distribution has heavier tails).

A widely used family of changepoint detection methods are based on the cumulative sums of squares test (CUSUM) [Page, 1954a]. These are non-parametric tests and can be applied to many different models, however, they implicitly assume Gaussian errors as well and so are not robust. Alternative non-parametric methods exist which would be expected to have some robustness such as the methods described in Haynes et al. [2017b] and Zou et al. [2014].

Thus, in reality it is often necessary to do some pre-processing to "clean" the data and remove outliers manually before performing changepoint detection. For data collected at a high-frequency, this is obviously a time consuming task and can be prone to error.

Fearnhead and Rigaill [2016], give a concrete example of the effects outliers can cause when attempting to detect changes in mean when outliers are present by considering the well-log data set. This data set, originally presented in O Ruanaidh [1996], has been analysed by a number of authors using different techniques. Whilst many methods perform well when analysing the cleaned dataset, they performed much worse and in many cases were unable to distinguish between changes and outliers when analysing the real data.

In recent years several automatic methods that detect a change in mean which are robust to outliers have been proposed.

One natural approach is to adapt ideas from robust statistics, namely replacing the squared error loss for a change in mean test statistic (assuming Gaussian noise) with an alternative loss function, (i.e. the Huber loss) that is less sensitive to outliers. Hušková [2013] derives CUSUM-like tests using these alternative loss functions to detect a single changepoint. This can be easily extended to find multiple changes using binary segmentation. Fearnhead and Rigaill [2016] incorporate a loss function within a penalised cost approach to estimating multiple changepoints which overcomes the approximate nature of binary segmentation.

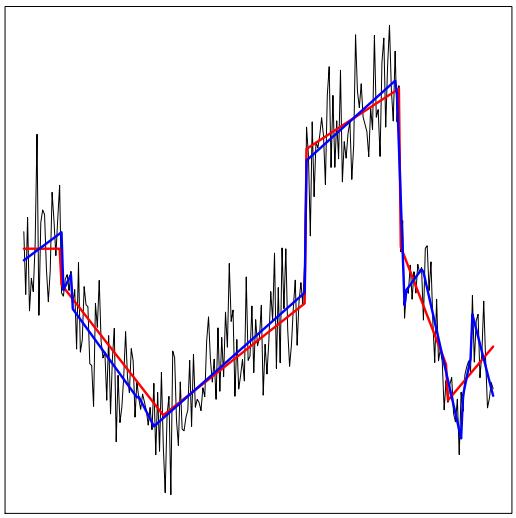
An alternative approach, devised by James et al. [2016], segments time series using a test statistic based on the median instead of the mean. This has the advantage of being robust by design, fast to compute, but approximate if a segmentation of a time series into changing means is required. This method has been employed in a number of areas including Finance, Medicine, and Signal processing as well as being used on a regular basis at Twitter.

What has received less attention is the problem of distinguishing between changepoints and outliers in more complex models, such as the change in regression setting. The differing models we look at in this paper can be summarised graphically in Figure 6.1.1. These are changes in regression, or slope, in which the latter has a constraint to enforce continuity on the underlying linear process. The other feature we consider apart from continuity are outliers. We can see the difference in Figures 6.1.1b and 6.1.1d where the residuals are drawn from a mixture distribution. This is a mixture between a $t-$distribution and a standard Gaussian random variable.

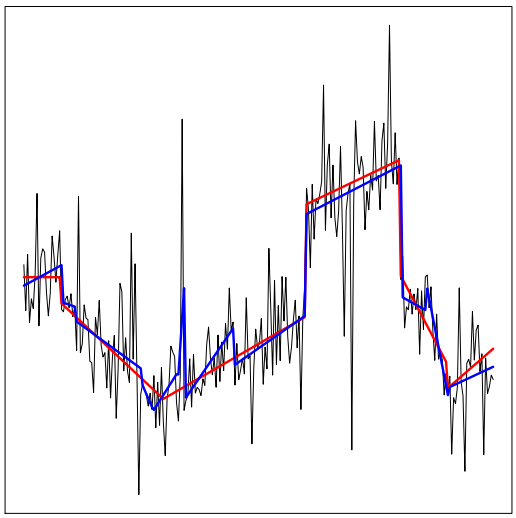
In this paper we aim to develop a changepoint detection technique for data, where the model we assume is made up of continuous piece-wise linear segments and in which outliers are present. To do this we aim to combine two methods from the changepoint literature which we describe below.

The first method, described in Fearnhead and Rigaill [2016], considers the problem of detecting changes in the mean in the presence of outliers. The aim of this work is to provide segmentations that are robust to outliers. There are several applications of this work in diverse fields such as bio-statistics, in estimating Copy Number Variation and of detecting tampering in Wifi transmitters and receivers [Bagci et al., 2015].

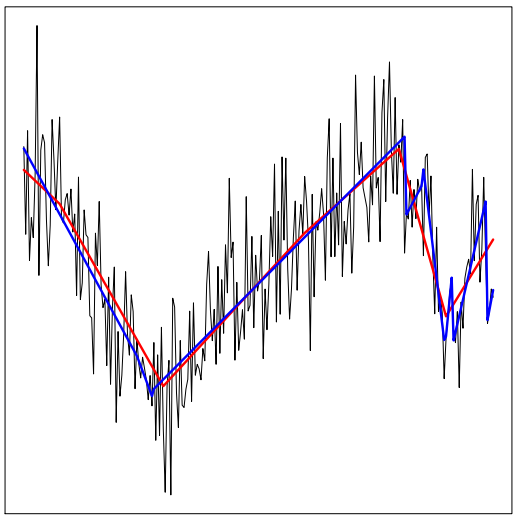
The method, described in Fearnhead and Rigaill [2016], is called R(obust)-FPOP and is



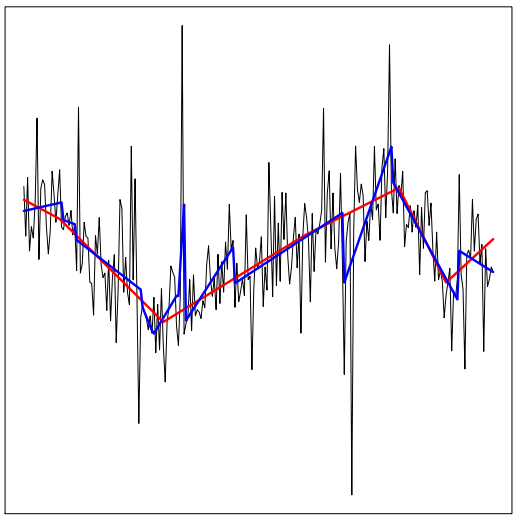
(a) Regression change - OLS



(b) Regression change - Robust



(c) Slope change - OLS



(d) Slope change - Robust

Figure 6.1.1: Four plots showing the different data models we consider in this paper. These include data with or without outliers and an underlying process which is continuous or non-continuous at the changepoints. In each figure we show the true segmentation of the data in red and the standard estimated segmentation under the assumption of Normally distributed residuals (the OLS segmentation) in blue. The standard OLS method works reasonably well for situations with no outliers (Figures 6.1.1a and 6.1.1c). However, for data with outliers (Figures 6.1.1b and 6.1.1d) the standard estimation method is heavily affected. This can be seen from the large deviations from the red and blue lines and the spikes in the blue line at outlier locations. The two figures in the top row show examples of the piece-wise change in regression model whereas the bottom row shows the change in slope model (Figures 6.1.1c and 6.1.1d). The underlying data generating process for the change in slope model is continuous at the changepoints.

based on the FPOP method first introduced in Maidstone et al. [2017b].

The second method we consider [Maidstone et al., 2016a], describes the problem of segmenting time series which are made up of continuous piece-wise linear segments. This continuity constraint adds a great deal of computational complexity to the problem as it means that there is dependence across segments. This increase in computational time is based on the fact that the usual dynamic programming recursions and pruning techniques developed in Killick et al. [2012] and Maidstone et al. [2017b] are predicated on an assumption of conditional independence between segments.

The outline of the chapter is as follows. Firstly, in Section 6.2, we present a penalised cost approach that uses existing methods to detect piece-wise changes in regression using both OLS and robust methods for independent segments. We then proceed to develop a set of recursions for the dependent segment case in Section 6.3.2 with robustness to outliers that update and propagate sets of quadratic equations in two variables. If we attempted to solve this set of recursions the computational complexity would be exponential in the length of the data. To deal with this, we consider two pruning algorithms that attempt to reduce the number of quadratics considered at each time step. One of these pruning methods overestimates the size of the set but is guaranteed to be optimal whereas the other has no such guarantee but is much more efficient in reducing the size of the set.

In Section 6.4 we evaluate our method, and compare it to a number of alternatives on simulated data. We then apply our method on a telecommunications event time series that contains a number of outliers in Section 6.5.

Finally we end with a discussion on the advantages and limitations of our method.

6.2 Problem set-up

Assume we have time-series data, $y_{1:n}$, though the ideas we present apply equally to other univariate data that is ordered, for example with genetic data which is ordered by position along a chromosome. We wish to fit a piecewise-linear mean to this data. We can define such a piecewise-linear mean through a set of changepoints, which will split the data into contiguous segments, and the linear form of the mean on each of the segments. Denote the number and position of changepoints by m and $\boldsymbol{\tau}_{1:m} = (\tau_1, \ldots, \tau_m)$ respectively. We will assume the changepoints are ordered, and define $\tau_0 = 0$ and $\tau_{m+1} = n$. We model the mean of the data within a segment as a linear function of time, but allow this linear model to vary between segments. We denote (ϕ_j, δ_j) to be, respectively, the intercept and slope of the mean for the jth segment.

A standard approach to fitting such a piecewise-linear mean to the data, and hence to estimating the changepoints where the form of the mean changes, is to minimise some cost over all possible piecewise-linear means. This cost is usually defined in terms of some measure of fit to the data plus some penalty for the complexity of the piecewise-linear mean, with the measure of fit being a sum of a loss-function of the residuals and the penalty on complexity being linear in the number of changepoints. So, for some appropriately chosen loss function $\gamma(\cdot)$ and positive constant β we wish to find the piecewise-linear mean that minimises a cost of the form

$$\sum_{j=1}^{m+1} \left[\sum_{t=\tau_j+1}^{\tau_{j+1}} \gamma\left(y_t - \phi_j - \delta_j t\right) + \beta \right]. \tag{6.2.1}$$

We consider two types of changepoint model, corresponding to two different classes of piecewise-linear means that we minimise over. The first, which we call *change in regression*, allows for

any linear mean within each segment, and thus imposes no constraints on the parameters m, $\boldsymbol{\tau}$, $\boldsymbol{\phi}_{1:m_1}$ and $\boldsymbol{\delta}_{1:m+1}$. The second, which we call *change in slope*, requires the piecewise linear mean to be continuous. To enforce continuity at the changepoint locations we must impose a series of constraints linking parameters in adjacent segments at each changepoint

$$\phi_j + \delta_j \tau_j = \phi_{j+1} + \delta_{j+1} \tau_j \qquad \forall j = 1, \ldots, m. \tag{6.2.2}$$

This additional constraint makes solving the optimisation problem in (6.2.1) much more difficult, as it creates dependency across segments.

There is substantial literature on how to choose the penalty constant β as it has an important impact on the accuracy of the estimated segmentation that we obtain. This parameter is involved in model selection and helps to avoid overfitting by penalising the addition of changepoints. Many authors have looked at different choices of penalties. If we let p denote the number of additional parameters introduced by adding a changepoint, then two popular examples used frequently in the literature include $\beta = 2(p+1)$ (Akaike's Information Criterion (AIC); [Akaike, 1974] and $\beta = (p+1)\log n$ (Bayesian Information Criterion (BIC); [Schwarz, 1978]. In both cases, some assumptions are made about the data for these penalty values to work well while an incorrect specification risks over/under-fitting the data. Some theoretical results have been established showing weak consistency for the BIC penalty in a number of models by Yao [1988] amongst others. There have also been several modifications to the BIC with their own advantages and unique features. One of the better known examples is the modified BIC (mBIC) which was introduced in Zhang and Siegmund [2007b] with a strong theoretical justification. We chose the BIC penalty in this work as it is by far the

most commonly used and is very simple to implement. In Section 6.3.1 we comment on alternatives.

The standard choice of loss-function is the square-error loss

$$\gamma(x) = \frac{x^2}{\sigma^2},$$

where σ^2 is the variance of the residuals, or an estimate of this variance. In this case the measure of fit to the data is just the standard residual sum of squares used for Ordinary Least Squares regression (OLS). However using such a loss-function leads to methods that are not robust to outliers. This can be seen in Figure 6.1.1, where all four time series are segmented using the square-error loss with these estimated segmentations shown in blue while the true segmentations are shown in red. In Figures 6.1.1b and 6.1.1d in which outliers are present in the data, it can be seen that the square-error loss performs poorly due to spikes in the estimated segmentation at the outlier locations.

In order to obtain a changepoint method that is robust to the presence of outliers, we will consider a different form for this loss function. In particular we follow Fearnhead and Rigaill [2016] and take $\gamma(\cdot)$ to be

$$\gamma(x) = \min\left\{\frac{x^2}{\sigma^2}, K^2\right\}. \tag{6.2.3}$$

For a suitably chosen value of K this cost limits the impact of outliers on estimating segment specific parameters. In the examples considered in this work we set $K = 2$. Our rationale for doing this is the well known fact that approximately 95% of Normally distributed observations lie within two standard deviations of the mean. This means that we attain robustness to outliers whilst still using the information from the vast majority of the observations.

We now consider how to minimise the cost function (6.2.1) when we use the loss-function (6.2.3) for both the change in regression and the change in slope problems.

6.3 Algorithms

We require two different methods to solve the problems discussed in Section 6.2. Firstly we recall the methods that can be used to infer changepoints for the change in regression model outlined in Section 6.3.1. We then consider the change in slope problem and give two different pruning methods. One of these pruning methods is a heuristic and so will not solve the problem exactly, however, without some form of approximation the computational complexity increases exponentially in the length of the data.

The vast difference in computational complexity between the two problems is caused by the addition of the continuity constraint and the dependence between parameters in different segments this introduces.

6.3.1 Change in regression

For the change in regression case existing penalised likelihood methods [Killick et al., 2012] can be applied. This is because we can write down the likelihood for the data within a putative segment and analytically (or numerically) maximise the likelihood function, enabling us to define a cost function to be used within methods such as the PELT algorithm. To ensure robustness to outliers we can use alternative loss functions discussed in Section 6.2.

We model the mean of the data within a segment as a linear function of time, but allow this linear model to vary between segments. We denote θ_j and δ_j to be the intercept and slope

respectively of the regression line in the jth segment.

The log likelihood for the jth segment of data $y_{(\tau_{j-1}+1):\tau_j}$ assuming a linear regression model with an intercept ϕ_j and slope δ_j can be written as

$$\ell(y_{(\tau_{j-1}+1):\tau_j}; \phi_j, \delta_j, \sigma) = -(\tau_j - \tau_{j-1}) \log \sigma - \sum_{i=\tau_{j-1}+1}^{\tau_j} \gamma(y_i - \phi_j - \delta_j i). \tag{6.3.1}$$

To simplify the notation we will now use $r_i = y_i - \phi_j - \delta_j i$ to denote the residual associated with the ith data point. The residual is a function of the parameters associated with the segment that y_i belongs to, but we suppress this dependence in the notation.

The cost of this segment can then be produced by minimising the negative log likelihood in equation (6.3.1) with respect to the two segment parameters ϕ_j and δ_j,

$$\mathcal{C}(y_{(\tau_{j-1}+1):\tau_j}) = \min_{\phi_j, \delta_j} \left[(\tau_j - \tau_{j-1}) \log(\sigma^2) + \sum_{i=\tau_{j-1}+1}^{\tau_j} \gamma\left(r_i\right) \right]. \tag{6.3.2}$$

If we were to take $\gamma(\cdot)$ to be the square error loss, then we could perform the minimisation required in (6.3.2) analytically, however we would be implicitly assuming Gaussian noise and hence would not be robust to outliers. Using a more robust loss function such as that described in (6.2.3) results in a more complex minimisation problem (6.3.3) that requires the use of numerical methods.

$$\min_{\phi_j, \delta_j} \sum_i \gamma(r_i). \tag{6.3.3}$$

Of these methods, the Iteratively ReWeighted Least Squares (IRWLS) method, is by far the most commonly used. This requires the definition of a weight function which is related to

the first derivative of the loss function $\gamma(\cdot)$. The minimum is found by iteratively computing

$$\min_{\phi_j,\delta_j} \sum_i w(r_i^{(k)})(y_i - \phi_j - \delta_j i)^2. \tag{6.3.4}$$

Where the weight function $w(x) = \frac{\gamma'(x)}{x}$ and $r_i^{(k)}$ is the kth iteration formed by finding the residuals using the kth estimates $\phi_j^{(k)}$ and $\delta_j^{(k)}$ of ϕ_j and δ_j respectively.

Here we are considering a fixed segment, the jth consisting of the data $y_{(\tau_{j-1}+1):\tau_j}$. Initial estimates for ϕ_j, δ_j are taken by fitting an OLS model. Then, for each observation in the jth segment $r_i^{(0)}$ is the residual from fitting this OLS model. The estimates of the segment parameters are then iteratively improved by solving a weighted least squares minimisation problem. Given an estimate of the residuals at step k, $r_i^{(k)}$, we find the values of ϕ_j and δ_j that minimise (6.3.4). Using this method convergence is usually quite rapid. When the minimum has been found this can be substituted into the segment cost (6.3.2).

Having defined a segment cost, we can then use a standard Dynamic Programming algorithm such as PELT [Killick et al., 2012] to find the optimal segmentation with respect to this cost.

6.3.2 Change in slope

In contrast to the change in regression problem the change in slope problem is much more difficult as we cannot form a segment cost in the same way. This is due to the continuity constraint between segments introducing dependence in the parameters in contiguous segments.

We develop an alternative set of Dynamic programming recursions in two variables in order to be able to model and solve this problem. At each time step, t, if we condition on the

current value of the process (mean) and the current slope then we do not need to consider past segmentations prior to time t in order to segment data in the future. This idea is key in developing the recursions in Section 6.3.2.

Recursions

Define $F_t(\theta, \delta)$ as the optimal segmentation of the sequence of data $y_{1:t}$ with the position of the underlying estimated linear process being θ at time t and the gradient of the segment that includes y_t being δ.

We can find $F_t(\theta, \delta)$ by solving the minimisation problem in (6.3.5) for the first t time points where $0 < \tau_1 \ldots < \tau_{p-1} < \tau_p = t$ with all the continuity constraints for the segment parameters added as well

$$\begin{aligned} F_t(\theta, \delta) &= \min_{p, \tau_{1:p}} \sum_{j=1}^{p-1} \left[\sum_{i-\tau_j+1}^{\tau_{j+1}} \gamma(y_i - \phi_j - \delta_j i) + \beta \right] \\ \text{st } \phi_j + \delta_j \tau_j &= \phi_{j+1} + \delta_{j+1} \tau_j \ \forall j = 1, 2, \ldots, p-1 \\ \text{and } \theta &= \phi_p + \delta_p t \\ \delta &= \delta_p. \end{aligned} \tag{6.3.5}$$

We can form a set of recursions to calculate $F_{t+1}(\theta, \delta)$ based on its previous value $F_t(\Theta, \Delta)$. The form of these recursions are simple if we condition on whether or not there is a change-point at time $t-1$

$$F_{t+1}(\theta, \delta) = \min \left\{ F_t(\theta - \delta, \delta) + \gamma(y_{t+1} - \theta), \min_{\delta'} \left[F_t(\theta - \delta, \delta') + \gamma(y_{t+1} - \theta) + \beta \right] \right\}. \tag{6.3.6}$$

To describe the intuition behind (6.3.6), we look at each term in the minimisation on the right separately.

Firstly if we assume no change occurs at time t, then the gradient of the line at time t and $t+1$ are equal. For the linear process to take the value θ at time $t+1$ the position at time t must be $\theta - \delta$ because the gradient at t is δ and we "step back" one time step. As we are considering the observation at time $t+1$, y_{t+1}, then we add on the contribution given by the loss function evaluated at $y_{t+1} - \theta$.

If a change does occur at time t the gradient of the line segment that ended at time t is not the same as the gradient of the new line segment at time $t+1$, i.e. $\delta' \neq \delta$. This gradient is found by minimising the resulting expression with respect to δ'. The values for the position can be argued in the same way as above. The penalty β must also be added for including a new changepoint.

For the recursion in equation (6.3.6) to be useful in computing F_t, it is instructive to see that F_t can be written as the minimum over a set of quadratics S_t. I.e.

$$F_t(\theta, \delta) = \min_{i \in S_t} \left\{ q_t^{(i)}(\theta, \delta) \right\}. \tag{6.3.7}$$

Substituting (6.3.7) into (6.3.6) and switching the order of minimisation gives

$$F_{t+1}(\theta, \delta) = \min_{i \in S_t} \left\{ \min\{ q_t^{(i)}(\theta - \delta, \delta) + \gamma(y_{t+1} - \theta), \min_{\delta'} q_t^{(i)}(\theta - \delta, \delta') + \gamma(y_{t+1} - \theta) + \beta \} \right\}.$$

If we consider the terms in the inner minimisation that depend upon $q_t^{(i)}$ and substitute in

the definition of $\gamma(\cdot)$, these become

$$\min \begin{cases} q_t^{(i)}(\theta-\delta,\delta) + \frac{(y_{t+1}-\theta)^2}{\sigma^2} \\ q_t^{(i)}(\theta-\delta,\delta) + K^2 \\ \min_{\delta'} q_t^{(i)}(\theta-\delta,\delta') + \frac{(y_{t+1}-\theta)^2}{\sigma^2} + \beta \\ \min_{\delta'} q_t^{(i)}(\theta-\delta,\delta') + K^2 + \beta \end{cases} \tag{6.3.8}$$

The first two quadratics in (6.3.8) are produced if we assume no change at t and thus y_{t+1} can either be an ordinary point or an outlier. The third and fourth quadratics are produced when a change occurs at time t and y_{t+1} is an ordinary or outlier point respectively.

To initialise these recursions we set $q_1^1(\theta,\delta) = (y_1-\theta)^2 = \theta^2 - 2\theta y_1 + y_1^2$. The different updates for the coefficients of the quadratics are given in Appendix B.

For each quadratic q_t at time t the recursion in (6.3.6) and quadratic updates given in (6.3.8) imply that there will be four new quadratics propagated to time $t+1$. Thus the number of quadratics considered to calculate $F_{t+1}(\theta,\delta)$ is $4 \times |S_t|$. This is because for every $i \in S_t$ four new quadratics are propagated from each quadratic $q_t^i(\theta,\delta)$ at time t.

Computational cost

It is simple to calculate the number of quadratics stored at each time ($|S_t|$) as $|S_t| = 4|S_{t-1}|$ by the argument above. Initially at time $t = 1$ we only consider a single quadratic so that $|S_1| = 1$. Thus we have that $|S_t| = 4^{t-1}$.

Using this method the number of quadratics we have to store and consider grows exponentially in the length of the data, thus for problems of any interesting size this procedure is infeasible. We consider two ways in which the number of quadratics that are considered at each time

can be pruned so that the total number propagated to the next time step is reduced.

The first method discussed is called inequality pruning, this reduces the set of quadratics at every time step t using an easily computed inequality. This type of pruning is also guaranteed to retain the optimal solution. However, the number of quadratics still grows rapidly over time when using this pruning method albeit at a slower rate than without any pruning, making it infeasible for analysing time series of moderate length.

The second approach discussed is called heuristic pruning as in contrast to inequality pruning the method is not guaranteed to retain the optimal solution. The pruning technique used by this method is more computationally intensive than inequality pruning at each time step, however it prunes much more than inequality pruning and far fewer quadratics are retained at each time step. Tuning parameters are used to balance the approximation to the optimal solution and the efficiency of the pruning method.

Inequality pruning

The first pruning method we present is akin to the PELT pruning method as described in Killick et al. [2012] because it involves some pruning based on an inequality.

Define the global minimum of all the quadratics at time t, as

$$q_t^* = \min_{i \in S_t} \left[\min_{\theta,\delta} q_t^i(\theta, \delta) \right].$$

Then we can show that the following PELT style inequality to remove quadratics that can never be optimal in the future is true.

Theorem 6.3.1. *If for some i,*

$$\min_{\theta,\delta}\left[q_t^i(\theta,\delta)\right] > q_t^* + 2\beta, \tag{6.3.9}$$

holds then at any future time $T > t$, the ith quadratic $q_t^i(\theta,\delta)$ and all of its offspring can never in the future be optimal. Therefore whenever this inequality is satisfied the quadratic can be pruned from the set of quadratics considered.

The advantage of pruning using this inequality is that it can be done very efficiently. The natural logarithm of the number of quadratics considered at each iteration is shown on the right hand side plot in Figure 6.3.1 for the methods with and without inequality pruning. We can see that inequality pruning reduces the number of quadratics that are considered at each iteration, however, the number of quadratics still grows quickly and analysing larger data sets would be infeasible.

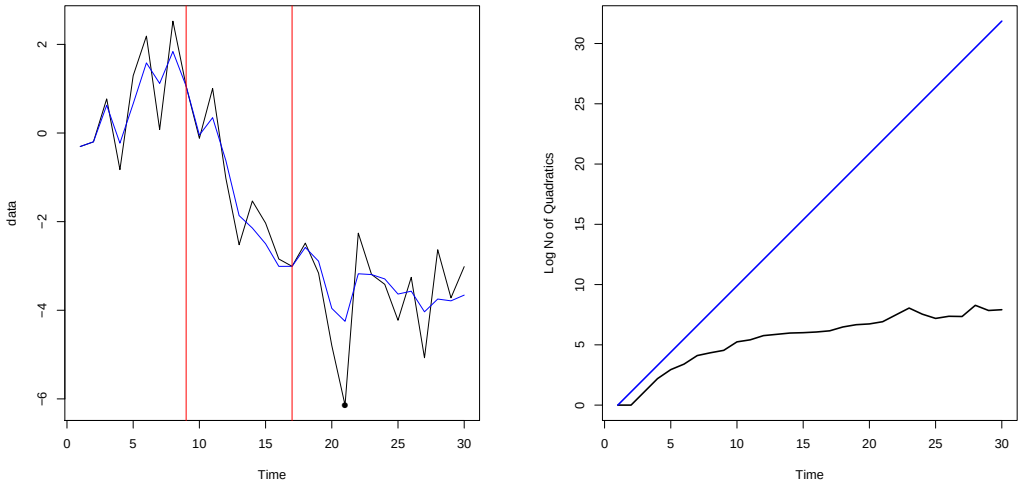


Figure 6.3.1: A time series on the left with changepoints shown in red with outliers highlighted as thicker black circles. On the right a plot of the (logarithm of the) number of quadratics considered at each time step for the inequality pruning method in black and with no pruning in blue.

Pruning heuristic

The inequality pruning considered above can be seen as removing too few quadratics than we would ideally like, but guaranteeing that we retain the optimal quadratic at each time step.

We now present a pruning heuristic called "Conditional quadratics". This heuristic is based on reducing the quadratics in two variables to quadratics in a single variable, then using the functional pruning method in Maidstone et al. [2016a]. This heuristic vastly reduces the number of quadratics stored at each time step however, it is approximate in nature and may remove too many quadratics i.e. the optimal solution at any given time step.

Conditional quadratics

Reducing a two variable quadratic such as $q(\theta, \delta)$ into a single variable quadratic which in this instance we call a conditional quadratic just involves the substitution of a constant in place of one of the variables. We substitute a constant θ_0 for the variable θ so that the quadratic can now be written in terms of the variable δ conditional on the value of θ_0 for the values of the coefficients. The conditional quadratic $q(\delta|\theta = \theta_0)$ can be written as

$$\begin{aligned} q(\delta|\theta = \theta_0) &= c_1\theta_0^2 + c_2\theta_0 + c_3\delta^2 + c_4\delta + c_5\theta_0\delta + c_6 \\ &= c_3\delta^2 + (c_4 + c_5\theta_0)\delta + (c_6 + c_1\theta_0^2 + c_2\theta_0). \end{aligned} \tag{6.3.10}$$

Geometrically this is equivalent to taking a slice through all the quadratic surfaces at a point θ_0, doing this we are left with a collection of single variable quadratics in δ. Now it is much easier to find those quadratics that make up the piece-wise minimum quadratic. The first step involves setting $\delta = -\infty$ and finding which quadratic gives the pointwise minimum here

and so is the first member of the piece-wise minimum. We then find the points of intersection between this quadratic and all of the others, and the minimum of these points gives us the location where the second member of the minimum begins. This is done repeatedly until no other intersections occur. Notice this is an exact method as points of intersection in the single variable case are easily found by finding roots of ordinary quadratic equations so that one source of error is now completely removed from the method.

Pruning using the conditional quadratic gives us the set of quadratics we need to store conditional on the corresponding value of θ. In theory we would need to repeat this for all values of θ over a continuous range. We approximate this by choosing a grid of θ values, finding the set of quadratics we need to keep for each of the θ values on this grid, and then pruning all quadratics that are not required to be kept for any θ on our grid.

This can be seen as approximating the true $F_t(\theta, \delta)$ defined in (6.3.5) by

$$\hat{F}_t(\theta, \delta) = \min_{i \in \hat{S}_t} \left\{ q_t^{(i)}(\theta, \delta) \right\}$$

where $\hat{S}_t \subset S_t$. The approximation error between $\hat{F}_t$ and F_t can be controlled by making the grid of θ values finer. However, this incurs a greater computational cost when performing the conditional quadratic pruning.

In Figure 6.3.2 we can see that the number of quadratics stays relatively constant over time in contrast to the inequality pruning in Figure 6.3.1. We took the grid to be the range of observed values of the series $\theta \in [-7, 3]$ with intervals of width 0.1. For this data the heuristic gives the same (optimal) solution as the exact inequality pruning, however we cannot guarantee our method always gives the same (optimal) solution.

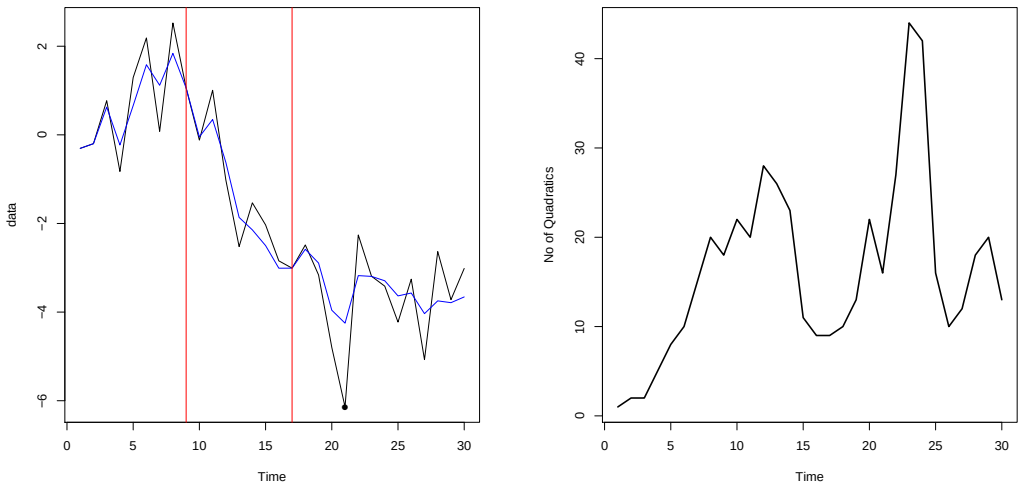


Figure 6.3.2: A time series on the left with changepoints shown in red with outliers highlighted as thicker black circles. On the right a plot of the number of quadratics considered at each time step when conditional quadratic pruning is performed at each step.

Recommendations

For practical use when analsying time series with a length larger than approximately 50 time points we would recommend the use of the conditional quadratic pruning heuristic. There are some tuning parameters to select and be aware of, this is the size and coarseness of the grid of values used for the variable θ. The limits of this grid are found by taking the minimum and maximum values for θ that optimise each of the quadratics considered at time t, denote these as $\hat{\theta}_{\min}$ and $\hat{\theta}_{\max}$. The grid we use for θ is then the interval $\left[\hat{\theta}_{\min} - 1, \hat{\theta}_{\max} + 1\right]$, the coarseness of the grid is taken to have intervals of width 0.1. We have found that this combination provides an efficient pruning method that keeps the number of quadratics considered at each time step relatively constant and the resulting segmentation accurate.

These recommendations as to the limits and coarseness of the grid considered here are conditioned on the data being scaled, so that the residuals have unit variance.

6.4 Simulation study

We call the method introduced in Section 6.3.2 C-ROB (continuous and robust). In this section we evaluate C-ROB on simulated data, comparing performance against three other methods. The first we call C-OLS (continuous and OLS method) introduced in Maidstone et al. [2016a]. The next two methods I-OLS and I-ROB assume independence across segments and use the OLS and robust cost functions in Section 6.3.1.

Our aim is to compare these different methods and evaluate the quality of segmentation in a variety of different situations such as an increasing proportion of outliers, variations in the heaviness of the tails of the outlier distribution and the quality of predictions.

The model we simulate from is a piece-wise continuous linear process in which outliers are present. Firstly, we simulate a changepoint process where the distance between successive changepoints is drawn from a specific geometric distribution (Geom(0.05)). After this process is simulated, values for θ (the positions for the beginning and end of the segment), are drawn from a uniform distribution. This describes the true linear process at each time. We then draw samples from the distribution for residuals which is a mixture between a standard Gaussian random variable for normal points and a heavier tailed t-distribution for the outliers.

6.4.1 Performance of different segmentation methods

We investigate three different situations in this set of simulations and test the performance of the four different segmentation methods.

The simulations in Case 1 examine the impact on performance when the proportion of outliers to normal points in the data increases from one-tenth to around a third of the total points. We

would expect that as this proportion increases the performance of the non-robust methods would degrade. In Case 2 we also vary the proportion of outliers but this time we are interested in the predictive power of the methods when we hold back several data points and extrapolate the final segment.

The third and final case considers how heavy tailed the outlier residuals are. The outlier residuals are drawn from a t-distribution with a varying number of degrees of freedom, from a heavy tailed distribution with two degrees of freedom to lighter tailed distributions with five, ten and then an infinite number of degrees of freedom (i.e. a normal distribution).

We report several different measures of the accuracy of segmentation. These include the Mean Squared Error (MSE) of the estimated process compared to the truth, the proportion of true changepoints detected and the number of false positives. A changepoint is defined as being detected if it is within five time points of the truth.

For greater comparability between the different methods we don't report raw MSE figures but instead we give a 0/1 score depending on which method gives the smallest MSE (1 if that method gives the smallest MSE). In Tables 6.4.1, 6.4.2 and 6.4.3 we then report the proportion of times each of the methods attains the smallest MSE.

Case 1: Effect of p

We would expect that the proportion of outliers in the data would have an effect on the segmentations, especially those obtained via OLS with higher proportions leading to poorer quality segmentations. Outliers come about through adding heavy tailed noise where the

p	Method	Proportion lowest MSE	Proportion detected	Number of False positives
0.1	C-OLS	0.01 (0,0.03)	0.70 (0.68,0.73)	14.6 (13.8,15.5)
	C-ROB	**0.77 (0.71,0.83)**	**0.79 (0.77,0.81)**	**2.06 (1.83,2.30)**
	I-OLS	0.14 (0.10,0.19)	0.48 (0.46,0.51)	11.4 (11.0,11.7)
	I-ROB	0.08 (0.05,0.12)	0.72 (0.70,0.74)	9.97 (9.63,10.4)
0.2	C-OLS	0 (0,0)	0.71 (0.68,0.73)	30.8 (29.7,31.8)
	C-ROB	**0.90 (0.86,0.94)**	**0.74 (0.72,0.76)**	**2.90 (2.63,3.17)**
	I-OLS	0.04 (0.02,0.07)	0.38 (0.36,0.40)	9.44 (9.15,9.73)
	I-ROB	0.06 (0.03,0.10)	0.67 (0.64,0.69)	10.8 (10.4,11.2)
0.3	C-OLS	0 (0,0)	**0.76 (0.75,0.79)**	43.9 (42.6,45.1)
	C-ROB	**0.91 (0.87,0.95)**	0.72 (0.70,0.75)	**3.82 (3.52,4.11)**
	I-OLS	0.04 (0.02,0.07)	0.30 (0.28,0.32)	9.80 (9.51,10.1)
	I-ROB	0.05 (0.02,0.08)	0.57 (0.55,0.60)	7.27 (6.99,7.54)

Table 6.4.1: For all four methods and differing values of p we repeated each experiment 100 times. Three measures were recorded, the MSE between the true and estimated segmentations, the proportion of true changes that were detected and the number of false positives. These values are averaged over the 100 replications and 95% bootstrap confidence intervals are included in brackets.

following mixture distribution for each residual ϵ is used in these studies

$$\epsilon \sim (1-p)N(0,1) + pt(2). \tag{6.4.1}$$

Here p can be thought of as the proportion of outliers in the time series which are drawn from a t-distribution having two degrees of freedom.

In this setting C-OLS performs poorly in the MSE and false positives criteria. This is due to this method enforcing continuity and a segmentation involving OLS which for increasing proportions of outliers in the data estimates segmentations which have too many changepoints as they are excessively affected by outliers.

p	Method	Proportion lowest MSE
0.1	C-OLS	0.24 (0.18,0.30)
	C-ROB	**0.31 (0.25,0.38)**
	I-OLS	0.19 (0.14,0.25)
	I-ROB	0.26 (0.20,0.33)
0.2	C-OLS	0.16 (0.11,0.21)
	C-ROB	**0.43 (0.36,0.51)**
	I-OLS	0.11 (0.07,0.16)
	I-ROB	0.30 (0.24,0.37)
0.3	C-OLS	0.16 (0.11,0.21)
	C-ROB	**0.39 (0.33,0.46)**
	I-OLS	0.17 (0.12,0.22)
	I-ROB	0.28 (0.22,0.34)

Table 6.4.2: For all four methods and differing values of p we repeated each experiment 100 times. We recorded the MSE between the truth and predictions. These values are averaged over the 100 replications and 95% bootstrap confidence intervals are included in brackets.

Case 2: Prediction

The quality of predictions is an important consideration in many applications and in the presence of outliers, predictions can often suffer. We evaluate the predictions given by the differing methods on simulated data sets with varying proportions of outliers.

In this study we segment the data but exclude the last five time points, then we extrapolate the last segment and find the MSE of the predictions compared to the truth. The results are shown in Table 6.4.2.

The results show that as the proportion of outliers in the data increases the predictions given by the C-OLS method degrades. In this study we are effectively just considering the accuracy in identifying the final segment so that we can extrapolate it to form the predictions. Hence, the OLS methods that typically infer a new segment when an outlier is present will be expected to perform poorly as the extrapolated segment will deviate substantially from the simulated path.

df	Method	Proportion lowest MSE	Proportion detected	Number of False positives
2	C-OLS	0.06 (0.01,0.11)	0.70 (0.68,0.73)	14.6 (13.8,15.5)
	C-ROB	**0.94 (0.87,0.99)**	**0.79 (0.77,0.81)**	**2.06 (1.83,2.30)**
5	C-OLS	0.33 (0.27,0.42)	0.71 (0.66,0.75)	6.80 (5.95,8.30)
	C-ROB	**0.67 (0.59,0.74)**	**0.73 (0.70,0.77)**	**3.53 (2.74,4.20)**
10	C-OLS	**0.65 (0.57,0.73)**	**0.76 (0.72,0.80)**	**4.08 (2.86,6.30)**
	C-ROB	0.35 (0.31,0.39)	0.75 (0.73,0.78)	4.75 (3.90,6.83)
∞	C-OLS	**0.83 (0.77,0.89)**	**0.79 (0.75,0.83)**	**1.87 (1.61,2.25)**
	C-ROB	0.17 (0.09,0.25)	0.64 (0.59,0.68)	7.91 (6.23,9.40)

Table 6.4.3: For the two methods and differing values of *df* we repeated each experiment 100 times. Three measures were recorded, the MSE between the true and estimated segmentations, the proportion of true changes that were detected and the number of false positives. These values are averaged over the 100 replications and 95% bootstrap confidence intervals are included in brackets.

Case 3: Outlier distribution

In Cases 1 & 2 the outlier residuals are drawn from a t-distribution with a fixed number of degrees of freedom ($df = 2$). This is a rather heavy tailed distribution so we would expect our C-ROB method to outperform the C-OLS method in this case. However as the tails of the distribution become gradually lighter (with an increasing number of degrees of freedom) we would expect that C-OLS would perform better. This is because it assumes a Normal distribution which is equivalent to a t-distribution with $df = \infty$.

The mixture distribution for each residual ϵ used here is

$$\epsilon \sim (1-p)N(0,1) + pt(df).$$

We vary the number of degrees of freedom of the t-distribution that we use to simulate outliers $t(df)$ from $df = \{2, 5, 10, \infty\}$ and fix $p = 0.1$.

As the outlier distribution gains progressively lighter tails the C-OLS method begins to

improve and outperform our robust method. This is to be expected as the C-OLS method optimally segments data having Normally distributed residuals whilst our C-ROB method is approximate.

In the next section we apply our method to a real data set which has been looked at elsewhere in this thesis.

6.5 Telecommunications event data

This data was first introduced in Section 1.1 and was subsequently analysed using our MRC methodology in Section 5.5.1. In the MRC work we modelled each time series using a robust piece-wise linear regression changepoint model, which in the terminology of this chapter is a change in regression model or I-ROB.

Modelling this data using the (robust) change in slope (C-ROB) method makes practical sense as it assumes that the number of events changes continuously, rather than having piece-wise jumps from week to week at the changepoint locations. Here we just focus on the fully aggregated series of events shown below and also in Figures 1.1.1 and 5.5.1.

It could be useful to combine the method developed here with the MRC method in Chapter 5. This would be relatively simple to do as we would just need to substitute the I-ROB method of segmentation for each time series with our C-ROB method. This new method would have a much higher computational cost.

Comparing the segmentation on the left hand side of Figure 6.5.1 (C-ROB) to the (non-continuous) robust segmentation (I-ROB) in Figure 5.5.1, we can see that the shapes of the two segmentations are roughly similar. However, our method does give more changepoints

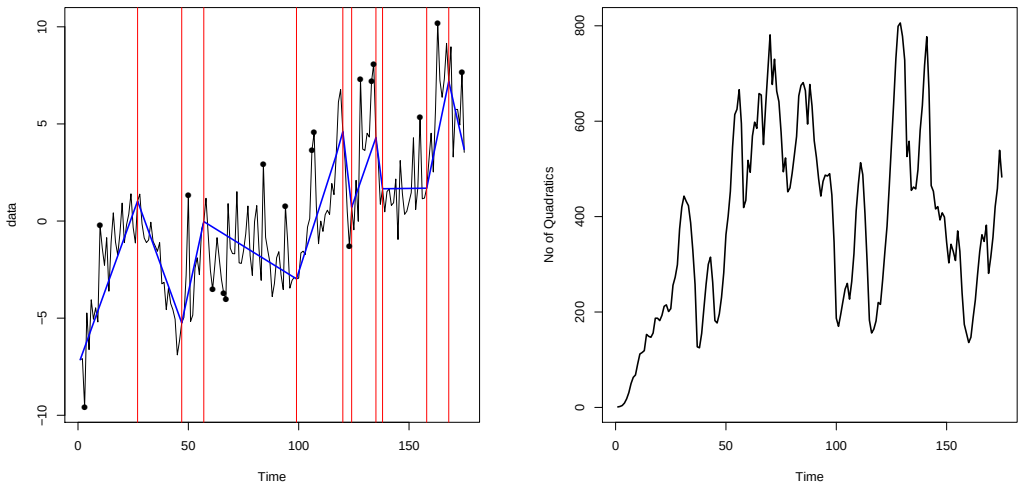


Figure 6.5.1: The telecommunications event data segmented into continuous piece-wise linear segments on the left hand side and the number of quadratics we have to store at each time step on the right hand side.

because we have enforced continuity between segments. In the plot on the right hand side of Figure 6.5.1 we can see that the number of quadratics considered at each time step does not increase uniformly over the length of the data.

6.6 Discussion

In this paper we have developed novel methodology to detect changepoints in data that is made up of piece-wise linear segments which are continuous and contain outliers.

The resulting algorithm, C-ROB, was compared in simulation studies and was shown to outperform other methods in cases where the residuals were indeed heavier tailed but didn't perform as well as other methods when the residuals were Gaussian for which an exact method has been developed (C-OLS).

An advantage of our method is that it gives us the location of both changepoints and outliers in a time series without the need for any pre-processing, however, due to its approximate

nature it under-performs C-OLS when the model assumed (heavier tailed residuals) does not hold.

An R package to run the methods discussed here is available.

Chapter 7

Conclusions and future work

In this thesis we have presented novel methodology for the detection of changepoints in a number of different settings.

Chapters 4 and 5 focused on multivariate time series where changepoints were modelled so that they may occur in only a subset of the variables. The vast majority of methods currently available assume that changes occur in all variables at the same time and so potentially miss more subtle changes that occur only in a subset of series.

These methods required us to develop techniques that used the multivariate nature of the observations to borrow information across the multiple series when only subtle changes were present in a (potentially small) subset of the series. For both methods we showed that this approach increases the accuracy in detecting changes over and above simpler methods based on analysing each of the channels in the multivariate time series independently. The models and methods developed are more complex, however, the algorithms for inference are efficient and the computational cost of our approaches is comparable to the simpler methods.

The BARD approach in Chapter 4 allows for more accurate detection for the location of

copy number variants in DNA and for other models where there are characteristic regions of baseline and abnormal behaviour observed in multivariate data. The MRC algorithm described in Chapter 5 leads to improved forecasts when considering panel data where there may be several different regimes for the final segment of data in different series of the panel.

In Chapter 6 we considered a more complex data model in the univariate setting. This included a constraint on a piece-wise linear process that was constrained to be continuous at the changepoint locations. We also wanted our estimation to automatically be robust to outliers rather than many methods in the literature that require a pre-processing step.

7.1 Future work

The methods developed in this thesis can be extended further. In this section several directions which related research could take in the future are discussed.

Firstly we look at an extension to the BARD method, introduced in Chapter 4. We consider the situation when the model assumed for the data becomes more complex. Specifically the computational challenges that arise when higher dimensional parameters are considered.

We then go on to discuss how dependence can be modelled in the work presented in Chapter 5. Currently, it is assumed that all of the variables in the panel are independent. This assumption leads to simple cost functions that are adapted from the univariate case by summing across the different variables. However, since we model changes occurring at common time points across the different variables this assumption is unlikely to hold and substantial cross-correlation between variables is likely to be present.

7.1.1 Higher dimensional parameters in the BARD method

In Chapter 4 we focused on a specific model of a change in mean from some baseline level. The data model for baseline behaviour is given by $\mathcal{D}$ and for abnormal behaviour is $\mathcal{P}_\theta$

$$\begin{aligned} \mathcal{D} &\sim N(0, \sigma^2) \\ \mathcal{P}_\theta &\sim N(\theta, \sigma^2). \end{aligned} \tag{7.1.1}$$

Here, σ is assumed to be constant across time and can be estimated from the data. Thus, the only difference is in the mean of the different segments θ (which has dimension one).

For more details on the model see Section 4.2.2.

Our method could easily be adapted to any model which specifies some normal and abnormal behaviour as defined in (7.1.1). The only restrictions we place on this is the ability to calculate marginal likelihoods for both types of segment. Typically the marginal likelihood for the normal behaviour is simple to calculate. However, the marginal likelihood for an abnormal segment is more challenging to compute.

If we have an abnormal segment with data $\mathcal{Y}_{t:s}$, with segment parameter θ, the likelihood of the data associated with the kth dimension is

$$p_k \prod_{i=t}^{s} f_{\mathcal{P}}(y_{i,k}|\theta) + (1 - p_k) \prod_{i=t}^{s} f_{\mathcal{D}}(y_{i,k}).$$

Then by independence over dimension

$$p(\mathbf{y}_{t:s}|\theta) = \prod_{k=1}^{d} \left(p_k \prod_{i=t}^{s} f_{\mathcal{P}}(y_{i,k}|\theta) + (1 - p_k) \prod_{i=t}^{s} f_{\mathcal{D}}(y_{i,k}) \right).$$

We place a prior on θ, $\pi(\theta)$ and then find the marginal likelihood $P_A(t,s)$ by integrating

$$P_A(t,s) = \int p(\mathbf{y}_{t:s}|\theta)\pi(\theta)\,\mathrm{d}\theta.$$

The main computational bottleneck is in the calculation of $P_A(\cdot,\cdot)$ as this involves integration over a prior for the parameter(s) which cannot be done analytically, and for higher dimensional parameters would be computationally intensive.

To see this, consider a trivial extension to our model allow different but known variances for each time series. To allow each abnormal segment to have its own variance as well as mean is possible, but would involve extra computation, as a 2-dimensional integral (7.1.2) would be needed to calculate the marginal likelihoods for abnormal segments

$$P_A(t,s) = \int\int p(\mathbf{y}_{t:s}|\theta,\sigma)\pi(\theta,\sigma)\,\mathrm{d}\theta\,\mathrm{d}\sigma. \tag{7.1.2}$$

In Chapter 4 we used simple numerical techniques to calculate $P_A(\cdot,\cdot)$. However, if we were to extend this method in the manner set out above we would require different techniques to calculate higher dimensional integrals. One possibility would be to use a Laplace approximation, however care would need to be taken that this approximation is valid for the cases we consider. If this were to hold then such an approximation would allow us to extend the BARD method to deal with higher dimensional parameters relatively simply while retaining an efficient computational cost.

7.1.2 Modelling dependence

When we set out the model for the MRC problem in Chapter 5 we assumed that there was no temporal dependence in the observations (autocorrelation) and that all of the variables in the panel are independent. This allowed us to formulate the problem in a univariate setting and extend it simply to the multivariate setting, see Section 5.2.2 for more details. These assumptions, however are questionable when analysing real data.

Some theoretical justification for ignoring temporal dependence can be seen in the simulation study with results that show, for example, that detecting changes in mean using a least squares criteria is robust to the presence of temporal dependence in the residuals [Lavielle and Moulines, 2000]. We showed empirically that our method can still detect the most recent changes even in the presence of AR(1) structure.

This was done by simulating an MRC process with a piecewise constant mean function as before but instead of adding IID normally distributed 'noise' we simulated an AR(1) noise process, Z_t, with standard normal errors e_t

$$Z_t = \phi Z_{t-1} + e_t.$$

This process was simulated for a range of values of ϕ which represented mild to moderate autocorrelation

Furthermore, our general approach can easily be extended to allow for modelling of the error structure of the residuals, by using cost functions for the data within each segment that are based on models which allow for autocorrelation.

A central assumption in both Chapters 4 and 5 is that of independence between time series.

This assumption allows us to write the likelihood easily in product form and perform much of the (possibly) high-dimensional inference efficiently. However, the purpose behind us analysing multiple time series together in that the changes occur in different series at common time points infers that this independence assumption is unlikely to hold and substantial cross-correlation between variables is likely to be present.

Several authors within the non-stationary time series community [see for example Ombao et al., 2005, Park et al., 2014] have worked on this problem. However, in the changepoint setting this remains an area for future research.

Appendix A

Lemmas for Proof of Theorem 4.4.1

Throughout this and the following section, we will assume the data is generated from the model detailed in Section 4.4.

We define part of the ratio in (4.4.2) as $X_k(\mu)$

$$X_k(\mu) = \exp\left\{\mu \sum_{u=t}^{s} \left(Y_{k,u} - \frac{\mu}{2}\right)\right\}.$$

The random variable $X_k(\mu)$ is log-normally distributed with different parameters depending on whether the sequence is normal or abnormal for that segment. In the normal segment case it is log-normal with parameters $-\mu^2(s-t+1)/2$ and $\mu^2(s-t+1)$,

$$\text{with } \mathbb{E}X_k(\mu) = 1.$$

For this case we will further define the mth central moment of $X_k(\mu)$ to be $C_m(\mu)$

$$C_m(\mu) = \mathbb{E}\left[(X_k(\mu) - \mathbb{E}X_k(\mu))^m\right].$$

Finally we denote the log of the product over the d terms in 4.4.2 as $S_d(\mu)$, taking the logarithm makes this become a sum over all the time-series

$$S_d(\mu) = \sum_{k=1}^{d} \log(1 + p_d(X_k(\mu) - 1)).$$

We now go on to prove several lemmas about $S_d(\mu)$ for both normal segments which will aid us in proving Theorems 4.4.1.

Lemma A.0.1 (Normal segment moment bounds)**.** *Assume we have a normal segment then*

$$\begin{aligned} \mathbb{E}S_d(\mu) &\leq -\frac{1}{2}C_2(\mu)dp_d^2 + \frac{1}{3}C_3(\mu)dp_d^3 \\ \mathbb{E}\left[(S_d(\mu) - \mathbb{E}S_d(\mu))^{2k}\right] &\leq K_k(\mu)d^k p_d^{2k} \end{aligned} \tag{A.0.1}$$

where $C_m(\mu)$ is the mth central moment of $X_m(\mu)$, and $K_k(\mu) > 0$ does not depend on d.

Proof. Writing out the expectation of $S_d(\mu)$ gives

$$\mathbb{E}S_d(\mu) = \sum_{k=1}^{d} \mathbb{E}\left[\log(1 + p_d(X_k(\mu) - 1))\right] \tag{A.0.2}$$

then we use the inequality $\log(1+x) \leq x - \frac{x^2}{2} + \frac{x^3}{3}$ for $x > 0$. So

$$\begin{aligned}\sum_{k=1}^{d} \mathbb{E}\left[\log(1 + p_d(X_k(\mu) - 1))\right] &\leq \sum_{k=1}^{d} \mathbb{E}\left[p_d(X_k(\mu) - 1) - \frac{p_d^2(X_k(\mu)-1)^2}{2} + \frac{p_d^3(X_k(\mu)-1)^3}{3}\right] \\ &= \sum_{k=1}^{d} -p_d^2 \frac{\mathbb{E}\left[(X_k(\mu)-1)^2\right]}{2} + p_d^3 \frac{\mathbb{E}\left[(X_k(\mu)-1)^3\right]}{3} \\ &= -\frac{1}{2} C_2(\mu) d p_d^2 + \frac{1}{3} C_3(\mu) d p_d^3.\end{aligned}$$

Now to derive the second inequality we consider $S_d(\mu) - \mathbb{E}S_d(\mu)$

$$S_d(\mu) - \mathbb{E}S_d(\mu) = \sum_{i=1}^{d} \left[Z_i(\mu) - \mathbb{E}Z_i(\mu)\right] = \sum_{i=1}^{d} \bar{Z}_i(\mu),$$

where $\bar{Z}_i(\mu) = Z_i(\mu) - \mathbb{E}Z_i(\mu)$. Writing this in terms of the centered random variables $\bar{Z}_i(\mu)$ is advantageous as when we consider raising the sum to the $2k$th power any term including a unit power of $\bar{Z}_i(\mu)$ vanishes by independence as $\mathbb{E}\bar{Z}_i(\mu) = 0$. Define

$$\mathcal{I}_{d,k} = \left\{ (j_1, \ldots, j_d) : j_i \in \{0, 2, 3, \ldots, 2k\} \text{ for } i = 1, \ldots, d \text{ and } \sum_{i=1}^{d} j_i = 2k \right\},$$

the set of non-negative integer vectors of length d, whose entries sum to $2k$, and that have no-entry that is equal to 1. For $\mathbf{j} \in \mathcal{I}_{d,k}$, let $n_{\mathbf{j}}$ be the number of terms in the expansion of

$(\sum_{i=1}^{d} \bar{Z}_i(\mu))^{2k}$ which have powers j_i for $\bar{Z}_i(\mu)$). Thus

$$\begin{aligned}
\mathbb{E}\left[(S_d(\mu) - \mathbb{E}S_d(\mu))^{2k}\right] &= \mathbb{E}\left[\left(\sum_{i=1}^{d} \bar{Z}_i(\mu)\right)^{2k}\right] \\
&= \sum_{\mathbf{j}\in\mathcal{I}_{d,k}} n_{\mathbf{j}} \prod_{i=1}^{d} \mathbb{E}\left(\bar{Z}_i(\mu)^{j_i}\right) \\
&\leq \mathbb{E}\left(\bar{Z}_1(\mu)^{2k}\right) \sum_{\mathbf{j}\in\mathcal{I}_{d,k}} n_{\mathbf{j}}.
\end{aligned}$$

Using $|\log(1+x)| \leq |x| + x^2/2$, we can bound $\mathbb{E}(\bar{Z}_1^{2k})$ by $A_k(\mu)p_d^{2k}$, where $A_k(\mu)$ will depend only on the the first $2k$ moments of $X_k(\mu)$, but not on p_d. Finally note that each term in $\mathcal{I}_{d,k}$ can only involve vectors with at most k non-zero components. For a term with l non-zero-components there will be $O(d^l)$ possible choices for which components are non-zero. Hence we have that

$$\sum_{\mathbf{j}\in\mathcal{I}_{d,k}} n_{\mathbf{j}} \leq B_k d^k,$$

for some constant B_k that does not depend on d. Thus we have the required result, with $K_k(\mu) = A_k(\mu)B_k$. □

Lemma A.0.2 (Probability bound)**.** *Fix μ and assume $p_d \to 0$ as $d \to \infty$. For a normal segment we have that there exists $D_k(\mu) > 0$ such that for sufficiently large d*

$$\Pr\left(S_d(\mu) \geq -\frac{1}{4}C_2(\mu)dp_d^2\right) \leq \frac{D_k(\mu)}{d^k p_d^{2k}}. \tag{A.0.3}$$

Proof. We first bound the probability by the absolute value of the centered random variable

and then use Markov's inequality with an even power of the form $2k$

$$\begin{aligned}\Pr\left(S_d(\mu) \geq -\frac{1}{4}C_2(\mu)dp_d^2\right) &\leq \Pr\left(|S_d(\mu) - \mathbb{E}S_d(\mu)| \geq \frac{1}{4}C_2(\mu)dp_d^2 - \frac{1}{3}C_3(\mu)dp_d^3\right) \\ &\leq \frac{\mathbb{E}\left[\left(S_d(\mu) - \mathbb{E}S_d(\mu)\right)^{2k}\right]}{(\frac{1}{4}C_2(\mu)dp_d^2 - \frac{1}{3}C_3(\mu)dp_d^3)^{2k}}.\end{aligned}$$

For d sufficiently large that $2C_3(\mu)p_d < C_2(\mu)$, we have

$$\frac{1}{4}C_2(\mu)dp_d^2 - \frac{1}{3}C_3(\mu)dp_d^3 > \frac{1}{12}C_2(\mu)dp_d^2.$$

Now using the result from Lemma A.0.1 we can replace the $2k$th centered moment by the bound we obtained above. Thus for sufficiently large d,

$$\Pr\left(S_d(\mu) \geq -\frac{1}{4}C_2(\mu)dp_d^2\right) \leq \frac{K_k(\mu)d^k p_d^{2k}}{(\frac{1}{12}C_2(\mu)dp_d^2)^{2k}}$$

So the result holds with $D_k(\mu) = K_k(\mu)[C_2(\mu)/12]^{-2k}$. □

Lemma A.0.3 (Lower bound for the second derivative of $S_d(\mu)$)**.** *We have that*

$$\frac{d^2 S_d(\mu)}{d\mu^2} \geq -d(s-t+1)$$

Proof. Firstly note that

$$\frac{\mathrm{d}X_k(\mu)}{\mathrm{d}\mu} = \left(\sum_{u=t}^{s} y_{k,u} - \mu(s-t+1)\right) X_k(\mu).$$

Now differentiating $S_d(\mu)$ twice

$$\begin{aligned}
\frac{\mathrm{d}S_d(\mu)}{\mathrm{d}\mu} &= \sum_{k=1}^{d} \frac{p_d \left(\sum_{u=t}^{s} y_{k,u} - \mu(s-t+1)\right) X_k(\mu)}{1 + p_d(X_k(\mu) - 1)} \\
\frac{\mathrm{d}^2 S_d(\mu)}{\mathrm{d}\mu^2} &= \sum_{k=1}^{d} \frac{-p_d(s-t+1)X_k(\mu) + p_d \left(\sum_{u=t}^{s} y_{k,u} - \mu(s-t+1)\right)^2 X_k(\mu)}{1 + p_d(X_k(\mu) - 1)} \\
&\quad - \left(\sum_{u=t}^{s} y_{k,u} - \mu(s-t+1)\right)^2 \left(\frac{p_d X_k(\mu)}{1 + p_d(X_k(\mu) - 1)}\right)^2
\end{aligned}$$

Let

$$Q_k = \frac{p_d X_k(\mu)}{1 + p_d(X_k(\mu) - 1)}$$

and $0 \leq Q_k \leq 1$ as $1 - p_d > 0$ (or $p_d < 1$). Thus the second derivative

$$\begin{aligned}
\frac{\mathrm{d}^2 S_d(\mu)}{\mathrm{d}\mu^2} &= \sum_{k=1}^{d} \left[-(s-t+1)Q_k + \left(\sum_{u=t}^{s} y_{k,u} - \mu(s-t+1)\right)^2 (Q_k - Q_k^2) \right] \\
&\geq \sum_{k=1}^{d} -(s-t+1)Q_k \geq -d(s-t+1)
\end{aligned}$$

has the required lower bound. □

Lemma A.0.4 (Detection of normal segments)**.** *Let $\pi(\mu)$ be a density function with support $[a, b]$ with $a > 0$ and $b < \infty$, and assume $1/p_d = O(d^{\frac{1}{2}-\epsilon})$ for some $\epsilon > 0$. For a normal segment $[t, s]$,*

$$\int \left\{ \prod_{k=1}^{d} \frac{P_{A,k}(t, s; \mu)}{P_{N,k}(t, s)} \right\} \pi(\mu) d\mu \to 0 \tag{A.0.4}$$

in probability as $d \to \infty$.

Proof. Define $C_2 = \min_{\mu\in[a,b]} C_2(\mu)$, and for a given d, M_d to be the smallest integer that is greater than

$$\frac{(b-a)\sqrt{s-t+1}}{p_d\sqrt{C_2}}.$$

Define $\Delta_d = (b-a)/M_d$. Now we can partition $[a,b]$ into M_d intervals of the form $[\mu_{i-1},\mu_i]$ for $i = 1,\ldots,M_d$, where $\mu_i = a + i\Delta_d$. Then the left-hand side of (A.0.4) can be rewritten as

$$\sum_{i=1}^{M_d}\int_{\mu_{i-1}}^{\mu_i}\left\{\prod_{k=1}^{d}[1+p_d(X_k(\mu)-1)]\right\}\pi(\mu)\mathrm{d}\mu.$$

Remember that $S_d(\mu) = \sum_{k=1}^{d}\log[1+p_d(X_k(\mu)-1)]$. Let E_d be the event that

$$S_d(\mu) < -\frac{1}{4}C_2dp_d^2, \text{for all } \mu = \mu_i,\ i = 0,\ldots,M_d.$$

If this event occurs then

$$\max_{\mu\in[a,b]} S_d(\mu) < -\frac{1}{4}C_2dp_d^2 + \Delta_d^2 d(s-t+1)/8,$$

as using Lemma A.0.3 we can bound $S_d(\mu)$ on each interval $[\mu_i,\mu_{i+1}]$ by a quadratic with second derivative $-d(s-t+1)$ and which takes values $-\frac{1}{4}C_2dp_d^2$ at the end-points.

Now by definition of Δ_d,

$$-\frac{1}{4}C_2dp_d^2 + \Delta_d^2 d(s-t+1)/8 < -\frac{1}{4}C_2dp_d^2 + \frac{1}{8}C_2dp_d^2 \to -\infty$$

as $d\to\infty$ because $dp_d^2\to\infty$ under our assumption on p_d. Thus to prove the Lemma we need only show that event E_d occurs with probability 1 as $d\to\infty$.

We can bound the probability of E_d not occurring using Lemma A.0.2. For any integer $k > 0$ we have that the probability E_d does not occur is

$$\begin{aligned}\sum_{i=1}^{M_d+1} \Pr\left(S_d(\mu_i) \geq -\frac{1}{4}C_2 d p_d^2\right) &\leq \sum_{i=1}^{M_d+1} \Pr\left(S_d(\mu_i) \geq -\frac{1}{4}C_2(\mu_i) d p_d^2\right) \\ &\leq \sum_{i=1}^{M_d+1} \frac{D_k(\mu_i)}{d^k p_d^{2k}} \\ &\leq (M_d+1) \max_{\mu\in[a,b]} \frac{D_k(\mu)}{d^k p_d^{2k}}.\end{aligned}$$

Here $D_k(\mu)$ is defined in Lemma A.0.2. It is finite for any μ, and hence $\max_{\mu\in[a,b]} D_k(\mu)$ is finite.

Now $M_d = O(p_d^{-1})$, so we have that the above probability is $O(d^{-k} p_d^{-2k-1}) = O(d^{1/2-(2k+1)\epsilon})$. So by choosing $k > 1/(4\epsilon)$ this is $O(d^{-\epsilon})$ which tends to 0 as required. □

A.1 Lemmas for Proof of Theorem 4.4.2

We use the same notation as in Section 4.4.1. However, we will now consider an abnormal segment from positions t to s. Let α_d denote the proportion of sequences that are abnormal, and μ_0 the mean. The observations in this segment come from a two component mixture. With probability α_d they are normally distributed with mean μ_0 and variance 1; otherwise they have a standard normal distribution. It is straightforward to show that for such an abnormal segment,

$$\mathbb{E}X_k(\mu) = (1-\alpha_d) + \alpha_d e^{\mu\mu_0(s-t+1)}. \tag{A.1.1}$$

Lemma A.1.1 (Abnormal segments, expectation and variance)**.** *Assume we have an abnormal segment* $[t, s]$ *with the mean of affected dimensions being* μ_0*. Let* $f(\mu)$ *be a density function with support* $A \subset \mathbb{R}$ *then*

$$\mathbb{E}\left[\int_A S_d(\mu) f(\mu) d\mu\right] \geq D_1(\mu) dp_d$$
$$\mathrm{Var}\left(\int_A S_d(\mu) f(\mu) d\mu\right) \leq D_2(\mu) dp_d^2 + o(dp_d^2)$$

with

$$\begin{aligned} D_1(\mu) &= \min_{\mu \in A}\left(\mathbb{E}[X_k(\mu) - 1] - \frac{p_d}{2}\mathbb{E}[(X_k(\mu) - 1)^2]\right) && \text{(A.1.2)} \\ &= \min_{\mu \in A}\left[\alpha_d(e^{\mu\mu_0(s-t+1)} - 1) - \frac{p_d}{2}\left(e^{\mu^2(s-t+1)} - 1\right) - \frac{\alpha_d p_d C(\mu)}{2}\right] && \text{(A.1.3)} \\ C(\mu) &= e^{\mu^2(s-t+1)}(e^{2\mu\mu_0(s-t+1)} - 1) - 2(e^{\mu\mu_0(s-t+1)} - 1) \end{aligned}$$

and

$$D_2(\mu) = \max_{\mu \in A} \mathbb{E}\left[(X_k(\mu) - 1)^2\right].$$

Proof. As $S_d(\mu)$ is the sum of d iid terms we can rewrite the expectation and variance with a single term

$$\mathbb{E}\left[\int_A S_d(\mu) f(\mu) \mathrm{d}\mu\right] = d\mathbb{E}\left[\int_A \log(1 + p_d(X_k(\mu) - 1)) f(\mu) \mathrm{d}\mu\right]$$
$$\mathrm{Var}\left(\int_A S_d(\mu) f(\mu) \mathrm{d}\mu\right) = d\mathrm{Var}\left(\int_A \log(1 + p_d(X_k(\mu) - 1)) f(\mu) \mathrm{d}\mu\right).$$

Now as $\log(1+x) > x - x^2/2$,

$$\mathbb{E}\left[\int_A \log(1+p_d(X_k(\mu)-1))f(\mu)\mathrm{d}\mu\right] \geq \mathbb{E}\left[\int_A \left(p_d(X_k(\mu)-1) - \frac{p_d^2(X_k(\mu)-1)^2}{2}\right) f(\mu)\mathrm{d}\mu\right]$$
$$= p_d \int_A \left(\mathbb{E}[X_k(\mu)-1] - \frac{p_d}{2}\mathbb{E}[(X_k(\mu)-1)^2]\right) f(\mu)\mathrm{d}\mu,$$

which gives (A.1.2). We then obtain (A.1.3) by using (A.1.1) and a similar calculation for the variance of $X_k(\mu)$.

We now consider the variance, which is bounded by the second moment. Using $|\log(1+x)| \leq |x| + x^2/2$ we have

$$\begin{aligned}
\mathrm{Var}\left(\int_A \log(1+p_d(X_k(\mu)-1))f(\mu)\mathrm{d}\mu\right) &\leq \mathbb{E}\left[\left(\int_A \log(1+p_d(X_k(\mu)-1))f(\mu)\mathrm{d}\mu\right)^2\right] \\
&\leq \mathbb{E}\left[\int_A \left\{\log(1+p_d(X_k(\mu)-1))\right\}^2 f(\mu)\mathrm{d}\mu\right] \\
&\leq \mathbb{E}\left[\int_A \left\{p_d^2(X_k(\mu)-1)^2 + p_d^3|X_k(\mu)-1|^3 + \frac{p_d^4}{4}(X_k(\mu)-1)^4\right\} f(\mu)\mathrm{d}\mu\right] \\
&\leq \max_{\mu\in A} \mathbb{E}\left\{p_d^2(X_k(\mu)-1)^2\right\} \int_A f(\mu)\mathrm{d}\mu + o(p_d^2),
\end{aligned}$$

which gives the required bound for the variance. □

Lemma A.1.2 (Detection of abnormal segments)**.** *Assume that we have an abnormal segment $[t,s]$. Let α_d be the probability of a sequence being abnormal and the mean of the abnormal observations be μ_0, with $p_d = o(1)$. Assume that there exists a set A such that for all $\mu \in A$ we have*

$$\lim_{d\to\infty} \alpha_d \left(e^{\mu\mu_0(s-t+1)} - 1\right) - \frac{p_d}{2}\left(e^{\mu^2(s-t+1)} - 1\right) > \delta,$$

and $\int_A \pi(\mu)d\mu > \delta'$*, for some* $\delta, \delta' > 0$*. If* $dp_d^2 \to \infty$ *as* $d \to \infty$ *then*

$$\int \left\{ \prod_{k=1}^{d} \frac{P_{A,k}(t,s;\mu)}{P_{N,k}(t,s)} \right\} \pi(\mu)d\mu \to \infty \tag{A.1.4}$$

in probability as $d \to \infty$*.*

Proof. If we restrict the integral in (A.1.4) to one over $A \subset \mathbb{R}$ we get a lower bound. Then rewriting the ratio in (A.1.4), using (4.4.2), in terms of $X_k(\mu)$ we get

$$\int \left\{ \prod_{k=1}^{d} [1 + p_d(X_k(\mu) - 1)] \right\} \pi(\mu)\mathrm{d}\mu \geq \int_A \left\{ \prod_{k=1}^{d} [1 + p_d(X_k(\mu) - 1)] \right\} \pi(\mu)\mathrm{d}\mu.$$

If we consider the logarithm of the above random variable and use Jensen's inequality we get a lower bound

$$\begin{aligned} \log \left(\int_A \left\{ \prod_{k=1}^{d} [1 + p_d(X_k(\mu) - 1)] \right\} \pi(\mu)\mathrm{d}\mu \right) &\geq \int_A \left\{ \sum_{k=1}^{d} \log(1 + p_d(X_k(\mu) - 1)) \right\} \pi(\mu)\mathrm{d}\mu \\ &= \int_A S_d(\mu)\pi(\mu)\mathrm{d}\mu. \end{aligned}$$

Then if we can show this random variable goes to ∞ as $d \to \infty$ the original random variable has the same limit. Let $T_d = \int_A S_d(\mu)\pi(\mu)\mathrm{d}\mu$. Using Lemma A.1.1, we have

$$\mathrm{E}(T_d) > \log(\delta') + \delta d p_d,$$

and for sufficiently large d there exists a constant C such that

$$\mathrm{Var}(T_d) < C d p_d^2.$$

So by Chebyshev's inequality

$$\Pr(T_d \leq \log(\delta') + \delta d p_d - d p_d^2) \leq \Pr(|T_d - \mathbb{E}T_d| \geq d p_d^2) \leq \frac{\mathrm{Var}(T_d)}{d^2 p_d^4} < \frac{C}{d p_d^2}.$$

Thus $T_d \to \infty$ in probability as $d \to \infty$, which implies (A.1.4). □

Appendix B

Updating the polynomials

The first and second updates given in (6.3.8) represent no change at t and involves a change in variables to represent the change in position from time t to $t+1$ but a constant gradient δ. The form of this quadratic is

$$\begin{aligned} q_t(\theta,\delta) &= c_1\theta^2 + c_2\theta + c_3\delta^2 + c_4\delta + c_5\theta\delta + c_6 \\ q_t(\theta-\delta,\delta) &= c_1(\theta-\delta)^2 + c_2(\theta-\delta) + c_3\delta^2 + c_4\delta + c_5(\theta-\delta)\delta + c_6 \\ &= c_1\theta^2 + c_2\theta + (c_1+c_3-c_5)\delta^2 + (c_4-c_2)\delta + (c_5-2c_1)\theta\delta + c_6. \end{aligned} \tag{B.0.1}$$

The third and fourth updates in (6.3.8) represents a changepoint at t and involves a similar change in position variable as above but a different variable for the gradient, δ'

$$\begin{aligned} q_t(\theta-\delta,\delta') &= c_1(\theta-\delta)^2 + c_2(\theta-\delta) + c_3\delta'^2 + c_4\delta' + c_5(\theta-\delta)\delta' + c_6 \\ &= c_1\theta^2 - 2c_1\theta\delta + c_1\delta^2 + c_2\theta - c_2\delta + c_3\delta'^2 + c_4\delta' + c_5\theta\delta' - c_5\delta\delta' + c_6. \end{aligned} \tag{B.0.2}$$

Equation (B.0.2) needs to be minimised with respect to δ', we can do this analytically which

gives us a value for the minimiser δ' as

$$\hat{\delta}' = \frac{c_5\delta - c_5\theta - c_4}{2c_3}.$$

Substituting this back into (B.0.2) gives us the quadratic

$$\begin{aligned}\min_{\delta'} q_t(\theta - \delta, \delta') = &\left(c_1 - \frac{c_5^2}{4c_3}\right)\theta^2 + \left(c_2 - \frac{c_4c_5}{2c_3}\right)\theta + \left(c_1 - \frac{c_5^2}{4c_3}\right)\delta^2 \\ &+ \left(\frac{c_4c_5}{2c_3} - c_2\right)\delta + \left(\frac{c_5^2}{2c_3} - 2c_1\right)\theta\delta + c_6 - \frac{c_4^2}{4c_3}.\end{aligned} \tag{B.0.3}$$

Bibliography

Hirotugu Akaike. A new look at the statistical model identification. *Automatic Control, IEEE Transactions on*, 19(6):716–723, December 1974. ISSN 0018-9286. doi: 10.1109/tac.1974.1100705. URL http://dx.doi.org/10.1109/tac.1974.1100705.

Helen Armstrong, Christopher Carter, Kevin Wong, and Robert Kohn. Bayesian covariance matrix estimation using a mixture of decomposable graphical models. Discussion Papers 2007-13, School of Economics, The University of New South Wales, 2007. URL http://EconPapers.repec.org/RePEc:swe:wpaper:2007-13.

Ivan E. Auger and Charles E. Lawrence. Algorithms for the optimal identification of segment neighborhoods. *Bulletin of Mathematical Biology*, 51(1):39–54, 1989. ISSN 1522-9602. doi: 10.1007/BF02458835. URL http://dx.doi.org/10.1007/BF02458835.

José Azar, Jean-François Kagy, and Martin C Schmalz. Can changes in the cost of carry explain the dynamics of corporate cash holdings. *Review of Financial Studies, forthcoming*, 2015.

Ibrahim Ethem Bagci, Utz Roedig, Ivan Martinovic, Matthias Schulz, and Matthias Hollick. Using channel state information for tamper detection in the internet of things. In *Proceedings of the 31st Annual Computer Security Applications Conference*, ACSAC

2015, pages 131–140, New York, NY, USA, 2015. ACM. ISBN 978-1-4503-3682-6. doi: 10.1145/2818000.2818028. URL http://doi.acm.org/10.1145/2818000.2818028.

D. Barry and J. A. Hartigan. Product partition models for change point problems. *The Annals of Statistics*, 20(1):260–279, 1992. URL http://www.jstor.org/stable/2242159.

Daniel Barry and J. A. Hartigan. A bayesian analysis for change point problems. *Journal of the American Statistical Association*, 88(421):309–319, 1993. URL http://www.jstor.org/stable/2290726.

Thomas W Bates, Kathleen M Kahle, and René M Stulz. Why do US firms hold so much more cash than they used to? *The Journal of Finance*, 64(5):1985–2021, 2009.

Thomas W Bates, Chinghung Henry Chang, and Jianxin Daniel Chi. Why has the value of cash increased over time? *Journal of Financial and Quantitative Analysis (to appear)*, 2017. URL http://dx.doi.org/10.2139/ssrn.1975491.

Richard Bellman. *Dynamic Programming*. Princeton University Press, Princeton, NJ, USA, 1 edition, 1957.

A. Benson and N. Friel. An adaptive MCMC method for multiple changepoint analysis with applications to large datasets. *ArXiv e-prints*, June 2016.

James O. Berger. *Statistical decision theory and Bayesian analysis*. Springer series in statistics. Springer, New York, NY [u.a.], 2. ed edition, 1985. ISBN 3540960988. URL http://gso.gbv.de/DB=2.1/CMD?ACT=SRCHA&SRT=YOP&IKT=1016&TRM=ppn+027440176&sourceid=fbw_bibsonomy.

James R Brown and Bruce C Petersen. Cash holdings and R&D smoothing. *Journal of Corporate Finance*, 17(3):694–709, 2011.

Hongyuan Cao and Wei Biao Wu. Changepoint estimation: another look at multiple testing problems. *Biometrika*, 102(4):974–980, 2015.

C K Carter and R Kohn. On Gibbs sampling for state space models. *Biometrika*, 81(3): 541–553, 1994.

Rong Chen and Jun S. Liu. Mixture kalman filters. *J. R. Statist. Soc. B*, 62:493–508, 2000.

H. Cho and P. Fryzlewicz. Multiscale and multilevel technique for consistent segmentation of nonstationary time series. *Statistica Sinica*, 22:207–229, 2012. URL `http://dx.doi.org/10.5705/ss.2009.280`.

Haeran Cho. Change-point detection in panel data via double cusum statistic. *Electronic Journal of Statistics*, 10(2):2000–2038, 2016. doi: 10.1214/16-EJS1155. URL `http://dx.doi.org/10.1214/16-EJS1155`.

Haeran Cho and Piotr Fryzlewicz. Multiple-change-point detection for high dimensional time series via sparsified binary segmentation. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 77(2):475–507, 2015. ISSN 1467-9868. doi: 10.1111/rssb.12079. URL `http://dx.doi.org/10.1111/rssb.12079`.

D.R. Cox. *Renewal Theory*. Methuen's monographs on applied probability and statistics. Methuen, 1962. URL `http://books.google.co.uk/books?id=0VxRAAAAMAAJ`.

Richard A Davis, Thomas C M Lee, and Gabriel A Rodriguez-Yam. Structural break estima-

tion for nonstationary time series models. *Journal of the American Statistical Association*, 101(473):223–239, 2006.

J. Ding, Y. Xiang, L. Shen, and V. Tarokh. Multiple Change Point Analysis: Fast Implementation And Strong Consistency. *ArXiv e-prints*, May 2016.

P. Fearnhead. Exact and efficient Bayesian inference for multiple changepoint problems. *Statistics and Computing*, 16(2):203–213, 2006. URL `http://www.springerlink.com/content/51j72n747611011q/`.

P. Fearnhead and Z. Liu. On-line inference for multiple changepoint problems. *Journal of the Royal Statistical Society B*, 69:589–605, 2007. URL `http://onlinelibrary.wiley.com/doi/10.1111/j.1467-9868.2007.00601.x/abstract`.

P. Fearnhead and D. Vasileiou. Bayesian analysis of isochores. *Journal of the American Statistical Association*, 104(485):132–141, 2009. URL `http://pubs.amstat.org/doi/abs/10.1198/jasa.2009.0009?journalCode=jasa`.

Paul Fearnhead and Peter Clifford. On-line inference for hidden markov models via particle filters. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 65(4): 887–899, 2003. ISSN 1467-9868. doi: 10.1111/1467-9868.00421. URL `http://dx.doi.org/10.1111/1467-9868.00421`.

Paul Fearnhead and Zhen Liu. Efficient bayesian analysis of multiple changepoint models with dependence across segments. *Statistics and Computing*, 21(2):217–229, 2011.

Paul Fearnhead and Guillem Rigaill. Changepoint detection in the presence of outliers. *arXiv:1609.07363*, 2016.

Klaus Frick, Axel Munk, and Hannes Sieling. Multiscale change point inference. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 76(3):495–580, 2014. ISSN 1467-9868. doi: 10.1111/rssb.12047. URL http://dx.doi.org/10.1111/rssb.12047.

Piotr Fryzlewicz. Wild binary segmentation for multiple change-point detection. *The Annals of Statistics*, 42(6):2243–2281, 12 2014a. doi: 10.1214/14-AOS1245. URL http://dx.doi.org/10.1214/14-AOS1245.

Piotr Fryzlewicz. Wild binary segmentation for multiple change-point detection. *Ann. Statist.*, 42(6):2243–2281, 12 2014b. doi: 10.1214/14-AOS1245. URL http://dx.doi.org/10.1214/14-AOS1245.

Pedro Galeano, Daniel Pea, and Ruey S. Tsay. Outlier detection in multivariate time series by projection pursuit. *Journal of the American Statistical Association*, 101(474):pp. 654–669, 2006. ISSN 01621459. URL http://www.jstor.org/stable/27590725.

Xiaodan Gao. Corporate cash hoarding: The role of just-in-time adoption. *Management Science (to appear)*, 2017. URL https://ssrn.com/abstract=2895779.

John R Graham and Mark T Leary. The evolution of corporate cash. *SSRN*, 2016. URL http://dx.doi.org/10.2139/ssrn.2805505. doi:10.2139/ssrn.2805505.

Peter J. Green. Reversible jump markov chain monte carlo computation and Bayesian model determination. *Biometrika*, 82(4):711–732, 1995. doi: 10.1093/biomet/82.4.711. URL http://biomet.oxfordjournals.org/content/82/4/711.abstract.

Peter D Grünwald. *The minimum description length principle*. MIT press, 2007.

Kaylea Haynes, Idris A. Eckley, and Paul Fearnhead. Computationally efficient changepoint detection for a range of penalties. *Journal of Computational and Graphical Statistics*, 26: 134–143, 2017a. doi: 10.1080/10618600.2015.1116445. URL http:/dx.doi.org/10.1080/10618600.2015.1116445.

Kaylea Haynes, Paul Fearnhead, and Idris A. Eckley. A computationally efficient nonparametric approach for changepoint detection. *Statistics and Computing*, 27(5):1293–1305, 2017b. ISSN 1573-1375. doi: 10.1007/s11222-016-9687-5. URL http://dx.doi.org/10.1007/s11222-016-9687-5.

Marie Hušková. *Robust Change Point Analysis*, pages 171–190. Springer Berlin Heidelberg, Berlin, Heidelberg, 2013. ISBN 978-3-642-35494-6. doi: 10.1007/978-3-642-35494-6_11. URL http://dx.doi.org/10.1007/978-3-642-35494-6_11.

B. Jackson, J.D. Scargle, D. Barnes, S. Arabhi, A. Alt, P. Gioumousis, E. Gwin, P. Sangtrakulcharoen, L. Tan, and Tun Tao Tsai. An algorithm for optimal partitioning of data on an interval. *Signal Processing Letters, IEEE*, 12(2):105–108, Feb 2005. ISSN 1070-9908. doi: 10.1109/LSP.2001.838216.

N. A. James, A. Kejariwal, and D. S. Matteson. Leveraging cloud data to mitigate user experience from 'Breaking Bad'. In *2016 IEEE International Conference on Big Data (Big Data)*, pages 3499–3508, Dec 2016. doi: 10.1109/BigData.2016.7841013.

Venkata Jandhyala, Stergios Fotopoulos, Ian MacNeill, and Pengyu Liu. Inference for single and multiple change-points in time series. *Journal of Time Series Analysis*, 34:423–446, 2013. ISSN 1467-9892. doi: 10.1111/jtsa12035. URL http://dx.doi.org/10.1111/jtsa12035.

X Jessie Jeng, T Tony Cai, and Hongzhe Li. Simultaneous discovery of rare and common segment variants. *Biometrika*, 100(1):157–172, 2013. ISSN 0006-3444. URL http://www.biomedsearch.com/nih/Simultaneous-Discovery-Rare-Common-Segment/23825436.html.

Jiashun Jin. *Detecting a target in very noisy data from multiple looks*, volume Volume 45 of *Lecture Notes–Monograph Series*, pages 255–286. Institute of Mathematical Statistics, Beachwood, Ohio, USA, 2004. doi: 10.1214/lnms/1196285396. URL http://dx.doi.org/10.1214/lnms/1196285396.

R. Killick, P. Fearnhead, and I. A. Eckley. Optimal detection of changepoints with a linear computational cost. *Journal of the American Statistical Association*, 107(500):1590–1598, 2012. doi: 10.1080/01621459.2012.737745. URL http://dx.doi.org/10.1080/01621459.2012.737745.

Rebecca Killick and Idris Eckley. changepoint: An R package for changepoint analysis. *Journal of Statistical Software, Articles*, 58(3):1–19, 2014. ISSN 1548-7660. doi: 10.18637/jss.v058.i03. URL https://www.jstatsoft.org/v058/i03.

Claudia Kirch, Birte Muhsal, and Hernando Ombao. Detection of changes in multivariate time series with application to EEG data. *Journal of the American Statistical Association*, 110(511):1197–1216, 2015. doi: 10.1080/01621459.2014.957545. URL http://dx.doi.org/10.1080/01621459.2014.957545.

V.G. Kulkarni. *Introduction to Modeling and Analysis of Stochastic Systems*. Springer Texts in Statistics. Springer London, Limited, 2012. ISBN 9781461427353. URL http://books.google.co.uk/books?id=2EeGkQEACAAJ.

M. Lavielle and G. Teyssière. Detection of multiple change-points in multivariate time series. *Lithuanian Mathematical Journal*, 46(3):287–306, 2006. ISSN 0363-1672. doi: 10.1007/s10986-006-0028-9. URL http://dx.doi.org/10.1007/s10986-006-0028-9.

Marc Lavielle. Using penalized contrasts for the change-point problem. *Signal Processing*, 85(8):1501 – 1510, 2005. ISSN 0165-1684. doi: http://dx.doi.org/10.1016/j.sigpro.2005.01.012. URL http://www.sciencedirect.com/science/article/pii/S0165168405000381.

Marc Lavielle. Using penalized contrasts for the change-point problem. *Signal Processing*, 85:1501–1510, August 2005.

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

Richard A. Levine and George Casella. Implementations of the Monte Carlo EM Algorithm. *Journal of Computational and Graphical Statistics*, 10(3):422–439, 2001. ISSN 10618600. doi: 10.2307/1391097. URL http://dx.doi.org/10.2307/1391097.

Jun S. Liu and Rong Chen. Sequential monte carlo methods for dynamic systems. *Journal of the American Statistical Association*, 93:1032–1044, 1998.

Ting Fung Ma and Chun Yip Yau. A pairwise likelihood-based approach for changepoint detection in multivariate time series models. *Biometrika*, 103(2):409–421, 2016.

Edgard M. Maboudou-Tchao and Douglas M. Hawkins. Detection of multiple change-points in multivariate data. *Journal of Applied Statistics*, 40(9):1979–1995, 2013. doi: 10.1080/02664763.2013.800471. URL http://dx.doi.org/10.1080/02664763.2013.800471.

R. Maidstone, P. Fearnhead, and AN. Letchford. Optimal changepoint detection for dependent data. 2016a.

R. Maidstone, P. Fearnhead, and A. Letchford. Detecting changes in slope with an L_0 penalty. *ArXiv e-prints*, January 2017a.

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

Robert Maidstone, Toby Hocking, Guillem Rigaill, and Paul Fearnhead. On optimal multiple changepoint algorithms for large data. *Statistics and Computing*, pages 1–15, 2016b. ISSN 1573-1375. doi: 10.1007/s11222-016-9636-3. URL `http://dx.doi.org/10.1007/s11222-016-9636-3`.

David S. Matteson and Nicholas A. James. A nonparametric approach for multiple change point analysis of multivariate data. *Journal of the American Statistical Association*, 109 (505):334–345, 2014. doi: 10.1080/01621459.2013.849605. URL `http://dx.doi.org/10.1080/01621459.2013.849605`.

Thomas Mikosch and Ctlin Stric. Nonstationarities in financial time series, the long-range dependence, and the igarch effects. *The Review of Economics and Statistics*, 86(1):378–390, 2004. URL `http://EconPapers.repec.org/RePEc:tpr:restat:v:86:y:2004:i:1:p:378-390`.

Joseph J. K. O Ruanaidh. *Numerical Bayesian methods applied to signal processing*. Statistics and computing [series]. Springer, New York, 1996.

Adam B. Olshen, E. S. Venkatraman, Robert Lucito, and Michael Wigler. Circular binary

segmentation for the analysis of array based DNA copy number data. *Biostatistics*, 5 (4):557–572, 2004a. doi: 10.1093/biostatistics/kxh008. URL `http://biostatistics.oxfordjournals.org/content/5/4/557.abstract`.

Adam B. Olshen, E. S. Venkatraman, Robert Lucito, and Michael Wigler. Circular binary segmentation for the analysis of arraybased dna copy number data. *Biostatistics*, 5 (4):557, 2004b. doi: 10.1093/biostatistics/kxh008. URL `+http://dx.doi.org/10.1093/biostatistics/kxh008`.

Hernando Ombao, Rainer Von Sachs, and Wensheng Guo. Slex analysis of multivariate nonstationary time series. *Journal of the American Statistical Association*, 100(470):519–531, 2005.

E. S. Page. Continuous inspection schemes. *Biometrika*, 41(1-2):100–115, 1954a. doi: 10.1093/biomet/41.1-2.100. URL `http://biomet.oxfordjournals.org/content/41/1-2/100.short`.

E. S. Page. Continuous inspection schemes. *Biometrika*, 41(1):100–115, 1954b. URL `http://biomet.oxfordjournals.org/content/41/1-2/100.extract`.

Timothy Park, Idris A Eckley, and Hernando C Ombao. Estimating time-evolving partial coherence between signals via multivariate locally stationary wavelet processes. *IEEE Transactions on Signal Processing*, 62(20):5240–5250, 2014.

Despina Vasileiou Paul Fearnhead. Bayesian analysis of isochores. *Journal of the American Statistical Association*, 104(485):132–141, 2009. ISSN 01621459. URL `http://www.jstor.org/stable/40591905`.

Davide Pettenuzzo and Allan Timmermann. Predictability of stock returns and asset allocation under structural breaks. *Journal of Econometrics*, 164(1):60–78, 2011. URL http://EconPapers.repec.org/RePEc:eee:econom:v:164:y:2011:i:1:p:60-78.

Ben Pickering. *Changepoint Detection for Acoustic Sensing Signals*. PhD thesis, Lancaster University, 2016.

Dalila Pinto, Katayoon Darvishi, Xinghua Shi, Diana Rajan, Diane Rigler, Tom Fitzgerald, Anath C. Lionel, Bhooma Thiruvahindrapuram, Jeffrey R. MacDonald, Ryan Mills, Aparna Prasad, Kristin Noonan, Susan Gribble, Elena Prigmore, Patricia K. Donahoe, Richard S. Smith, Ji Hyeon Park, Matthew E. Hurles, Nigel P. Carter, Charles Lee, Stephen W. Scherer, and Lars Feuk. Comprehensive assessment of array-based platforms and calling algorithms for detection of copy number variants. *Nature Biotechnology*, 29(6): 512–521, 2011.

Philip Preuss, Ruprecht Puchstein, and Holger Dette. Detection of multiple structural breaks in multivariate time series. *Journal of the American Statistical Association*, 110(510): 654–668, 2015. doi: 10.1080/01621459.2014.920613. URL http://dx.doi.org/10.1080/01621459.2014.920613.

Guangzhi Qu, S. Hariri, and M. Yousif. Multivariate statistical analysis for network attacks detection. In *Computer Systems and Applications, 2005. The 3rd ACS/IEEE International Conference on*, pages 9–, 2005. doi: 10.1109/AICCSA.2005.1387011.

J. Reese. Solution methods for the p-median problem: An annotated bibliography. *Networks*, 48(3):125–142, 2006. ISSN 1097-0037. doi: 10.1002/net.20128. URL http://dx.doi.org/10.1002/net.20128.

Guillem Rigaill. A pruned dynamic programming algorithm to recover the best segmentations with 1 to kmax change-points. *Journal de la Socit Franaise de Statistique*, 156(4):180–205, 2015.

Gideon Schwarz. Estimating the dimension of a model. *Annals of Statistics*, 6(2):461–464, 03 1978. doi: 10.1214/aos/1176344136. URL http://dx.doi.org/10.1214/aos/1176344136.

A. J. Scott and M. Knott. A Cluster Analysis Method for Grouping Means in the Analysis of Variance. *Biometrics*, 30(3):507–512, 1974. ISSN 0006341X. doi: 10.2307/2529204. URL http://dx.doi.org/10.2307/2529204.

Ashish Sen and Muni S. Srivastava. On tests for detecting change in mean. *Ann. Statist.*, 3 (1):98–108, 01 1975. doi: 10.1214/aos/1176343001. URL http://dx.doi.org/10.1214/aos/1176343001.

David Siegmund, Benjamin Yakir, and Nancy R. Zhang. Detecting simultaneous variant intervals in aligned sequences. *The Annals of Applied Statistics*, 5(2A):645–668, 06 2011. doi: 10.1214/10-AOAS400. URL http://dx.doi.org/10.1214/10-AOAS400.

Stephan Spiegel, Julia Gaebler, Andreas Lommatzsch, Ernesto De Luca, and Sahin Albayrak. Pattern recognition and classification for multivariate time series. In *Proceedings of the Fifth International Workshop on Knowledge Discovery from Sensor Data*, SensorKDD '11, pages 34–42, New York, NY, USA, 2011. ACM. ISBN 978-1-4503-0832-8. doi: 10.1145/2003653.2003657. URL http://doi.acm.org/10.1145/2003653.2003657.

Michael B Teitz and Polly Bart. Heuristic methods for estimating the generalized vertex median of a weighted graph. *Operations Research*, 16(5):955–961, 1968.

Ruey S. Tsay, Daniel Pea, and Alan E. Pankratz. Outliers in multivariate time series. *Biometrika*, 87(4):789–804, 2000. doi: 10.1093/biomet/87.4.789. URL `http://biomet.oxfordjournals.org/content/87/4/789.abstract`.

E. S. Venkatraman. *Consistency results in multiple change-point problems*. PhD thesis, Stanford University, 1993. URL `http://statistics.stanford.edu/~ckirby/techreports/NSA/SIENSA24.pdf`.

Jean-Philippe Vert and Kevin Bleakley. Fast detection of multiple change-points shared by many signals using group LARS. In J.D. Lafferty, C.K.I. Williams, J. Shawe-Taylor, R.S. Zemel, and A. Culotta, editors, *Advances in Neural Information Processing Systems 23*, pages 2343–2351. Curran Associates, Inc., 2010. URL `http://papers.nips.cc/paper/4157-fast-detection-of-multiple-change-points-shared-by-many-signals-using-group-lars.pdf`.

L. Yu. Vostrikova. Detecting disorder in multidimensional random processes. *Soviet Math. Dokl.*, 24:55–59, 1981.

Tengyao Wang and Richard J Samworth. High-dimensional changepoint estimation via sparse projection. *arXiv:1606.06246*, 2016.

Jeffrey M. Wooldridge. *Econometric analysis of cross section and panel data*. MIT Press, 2010.

John Wyse, Nial Friel, and Havard Rue. Approximate simulation-free Bayesian inference for

multiple changepoint models with dependence within segments. *Bayesian Analysis*, 6(4): 501–528, 2011.

Yao Xie and David Siegmund. Sequential multi-sensor change-point detection. *Annals of Statistics*, 41(2):670–692, 04 2013. doi: 10.1214/13-AOS1094. URL `http://dx.doi.org/10.1214/13-AOS1094`.

Yi-Ching Yao. Approximating the distribution of the maximum likelihood estimate of the change-point in a sequence of independent random variables. *Annals of Statistics*, 15(3): 1321–1328, 09 1987. doi: 10.1214/aos/1176350509. URL `http://dx.doi.org/10.1214/aos/1176350509`.

Yi-Ching Yao. Estimating the number of change-points via schwarz' criterion. *Statistics and Probability Letters*, 6(3):181–189, 1988. URL `http://EconPapers.repec.org/RePEc:eee:stapro:v:6:y:1988:i:3:p:181-189`.

Christopher Yau and Christopher C. Holmes. A decision theoretic approach for segmental classification using Hidden Markov models. *http://arxiv.org/abs/1007.4532*, July 2010. URL `http://arxiv.org/abs/1007.4532`.

Nancy R Zhang and David O Siegmund. A modified Bayes information criterion with applications to the analysis of comparative genomic hybridization data. *Biometrics*, 63:22–32, March 2007a. ISSN 0006-341X. doi: 10.1111/j.1541-0420.2006.00662.x.

Nancy R. Zhang and David O. Siegmund. A modified Bayes information criterion with applications to the analysis of comparative genomic hybridization data. *Biometrics*, 63(1):

22–32, 2007b. URL http://EconPapers.repec.org/RePEc:bla:biomet:v:63:y:2007:i:1:p:22-32.

Nancy R. Zhang, David O. Siegmund, Hanlee Ji, and Jun Z. Li. Detecting simultaneous changepoints in multiple sequences. *Biometrika*, 97(3):631–645, 2010. URL http://ideas.repec.org/a/oup/biomet/v97y2010i3p631-645.html.

NancyR. Zhang. DNA copy number profiling in normal and tumor genomes. In Jianfeng Feng, Wenjiang Fu, and Fengzhu Sun, editors, *Frontiers in Computational and Systems Biology*, volume 15 of *Computational Biology*, pages 259–281. Springer London, 2010. ISBN 978-1-84996-195-0. doi: 10.1007/978-1-84996-196-7_14. URL http://dx.doi.org/10.1007/978-1-84996-196-7_14.

Changliang Zou, Guosheng Yin, Long Feng, and Zhaojun Wang. Nonparametric maximum likelihood approach to multiple change-point problems. *Ann. Statist.*, 42(3):970–1002, 06 2014. doi: 10.1214/14-AOS1210. URL http://dx.doi.org/10.1214/14-AOS1210.