Changepoint Detection for Acoustic Sensing Signals

Benjamin James Pickering, B.Sc. (Hons.), M.Res.

Submitted for the degree of Doctor of Philosophy
at Lancaster University.

December 2015

Changepoint Detection for Acoustic Sensing Signals

Benjamin James Pickering, B.Sc. (Hons.), M.Res.

Submitted for the degree of Doctor of Philosophy at Lancaster University.

December 2015

Abstract

This thesis considers the application of changepoint detection methodology for the analysis of acoustic sensing signals. In the first part, we propose a detection procedure for changes in the second-order structure of a univariate time series. This utilises a penalised likelihood based on Whittle's approximation and allows for a non-linear penalty function. This procedure is subsequently used to detect changes in acoustic sensing data which correspond to external disturbances of the measuring cable.

The second part shifts focus to multivariate time series, and considers the detection of changes which occur in only a subset of the variables. We introduce the concept of *changepoint vectors* which we use to model such changes. A dynamic programming scheme is proposed which obtains the optimal configuration of changepoint vectors for a given multivariate series. Consideration of pruning techniques suggests that these are not practically viable for this setting. We therefore introduce approximations which vastly improve computational speed with negligible detrimental impact on accuracy. This approximated procedure is applied to multivariate acoustic sensing data.

Acknowledgements

First and foremost, I would like to thank the students, staff and management of the STOR-i Centre for Doctoral Training for providing such a stimulating and enjoyable research environment. You have each contributed to this memorable experience. Special thanks are due to my supervisors, Idris Eckley and Rebecca Killick, for their time, guidance and tireless efforts throughout the project.

I am very grateful for the financial support provided by the EPSRC and Shell Research Limited. I would like to thank Phil Jonathan for granting me with such an inspiring insight into industrial research. Thanks also to David Randell, Rakesh Paleja, Tim Park and everyone in the Statistics and Chemometrics team at Shell. Your knowledge, enthusiasm and camaraderie has made my experience of working with industry wholly rewarding.

Particular thanks are owed to the '11–'15 cohort of STOR-i: Dave, Emma, Hugo, James, Judd, Rhian, Rob and Tom. Our many laughs and innumerable discussions have maintained my motivation and preserved my sanity. I am privileged to count you as good friends. Thanks also to Mark Bell, Jamie Fairbrother, Chris Nemeth and Shreena Patel, for your limitless source of wisdom, advice and good humour over the past four years.

Finally, I would like to thank my parents and my brothers, Tom and Matthew. Without your continual support, patience and encouragement this venture would not have been possible.

Declaration

I declare that this thesis is my own work and has not been submitted in any form for the award of a higher degree elsewhere.

Benjamin James Pickering

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The abundance of sensors within contemporary devices and systems means that data is now being collected through time at an unprecedented scale. Consider, for example, an oil well. Modern techniques used to monitor oil flow involve the placement of acoustic sensors at various depths throughout the well. These record vibrations at very high resolutions (up to 10000 observations a second) so that this data can later be used to optimise the production of the well. An example of such data can be seen in Figure 1.1.1. Note in particular how the complex autocorrelated and multivariate structure is punctuated by changes. The times at which such changes occur are known as *changepoints*. The main theme of this thesis is the development of new methods to search for and detect such features.

The area of changepoint analysis has seen a resurgence of interest during the last five years, with many seminal contributions being made, such as the PELT method of Killick et al. (2012) and the WBS method of Fryzlewicz (2014). Historically, much of the work in changepoint detection has focused on the scenario where the observations are univariate and assumed to be independent and identically distributed (i.i.d.). Recent developments have considered more sophisticated models, such as those where the data is multivariate, or where there may be serial dependence between the observations. These models often provide a better reflection of the characteristics found in modern data sets. In particular, data obtained using acoustic sensing cannot be suitably modelled using the traditional univariate i.i.d. framework.

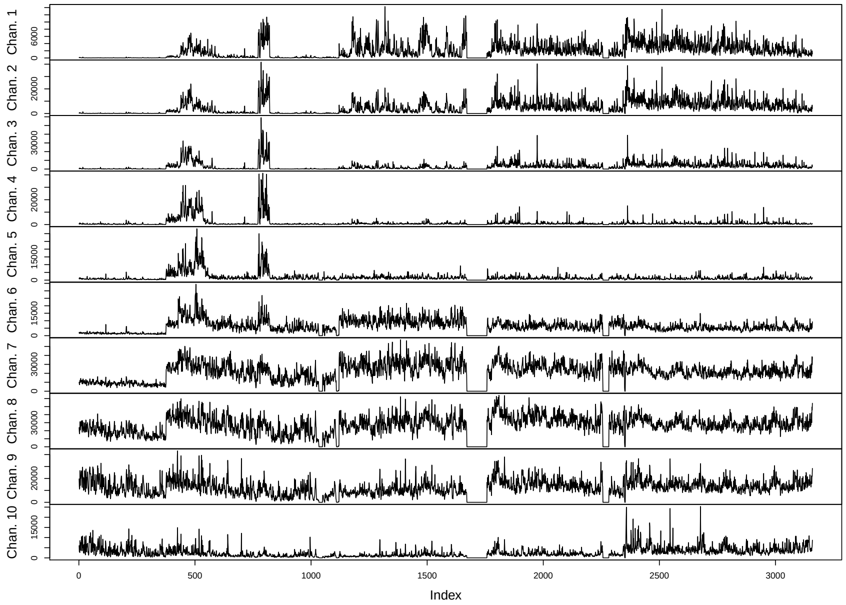


Figure 1.1.1: An example of acoustic sensing data observed at various depths in an oil well.

Within this thesis we consider two separate generalisations of the univariate i.i.d. changepoint problem: (i) the univariate setting with autocorrelation (and the search for changes in this dependence), and (ii) the changepoint problem in the multivariate i.i.d. setting. We begin in Chapter 2 by providing a review of the literature for both univariate and multivariate changepoint detection, examining the problem formulations and different detection methods which have been proposed. In addition, we explore various techniques which have been used to model autocorrelated time series.

A novel method for detecting changes in the dependence structure of a univariate autocorrelated time series is presented in Chapter 3. This method is based on Whittle's likelihood, an approximation to the exact likelihood of autocorrelated observations which allows for a faster computation with only a mild reduction in accuracy. We compare and contrast our approach with other leading procedures through application to univariate acoustic sensing data sets.

The focus of the thesis then shifts to the multivariate changepoint setting in Chapter 4. Here we introduce the concept of *changepoint vectors* which we use to model multivariate changepoints. These allow not only for the specification of the locations of any changes, but also the subsets of the variables in the series which are affected by a given change. A dynamic programming procedure which obtains the optimal configuration of changepoint vectors for a given multivariate series is presented. However, due to the large number of possible configurations, this procedure has a computational complexity of $\mathcal{O}(pn^{2p})$ for a p-variate series of length n. To reduce this computation time, in Chapter 5 we introduce an approximated version of the procedure which considers only the time-points and subsets of variables which are likely to be true changepoints and corresponding affected variable subsets. Application to both simulated time series and multivariate acoustic sensing data demonstrates that this approach vastly improves computational speed with negligible detrimental impact on accuracy. We conclude by presenting avenues for future research in Chapter 6.

Chapter 2

Changepoint Detection and Time Series Models

The term *changepoint* refers to a time-point at which a change occurs in one or more of the statistical properties of a time series. Knowledge of the presence of any changepoints within a time series is critical when forecasting or drawing inferences from the series. Due to this practical significance, the development of methodology capable of detecting such changepoints has received an increasing amount of attention throughout the previous half-century.

Since the first consideration by Page (1954) within the quality control literature, the problem of detecting changepoints has been considered across a wide array of scientific fields. These range from the long-established areas such as finance and economics (Andreou and Ghysels, 2009; Fryzlewicz and Subba Rao, 2014) and climatology (Reeves et al., 2007; Ruggieri et al., 2009), to more modern applications such as geophysical sciences (Velis, 2007; Gallagher et al., 2011), molecular biology (Braun et al., 2000; Xing et al., 2012), genetics (Olshen et al., 2004; Picard et al., 2005), network analysis (Lévy-Leduc and Roueff, 2009; Tartakovsky et al., 2013) and neuroscience (Aston and Kirch, 2012; Cribben et al., 2013).

This chapter examines and discusses various approaches which have been proposed for the detection of changepoints within observed time series. Due to the vast nature of the literature, the focus will largely be on the retrospective, also known as 'offline', changepoint detection problem rather than the sequential or ‘online’ equivalent problem. For an introduction to the sequential changepoint detection problem, we refer the reader to Lai (1995) and Polunchenko and Tartakovsky (2012). The work of Chen and Gupta (2000) also provides a review of changepoint detection methods in general.

The aim of this chapter is to provide an overview of three key areas:

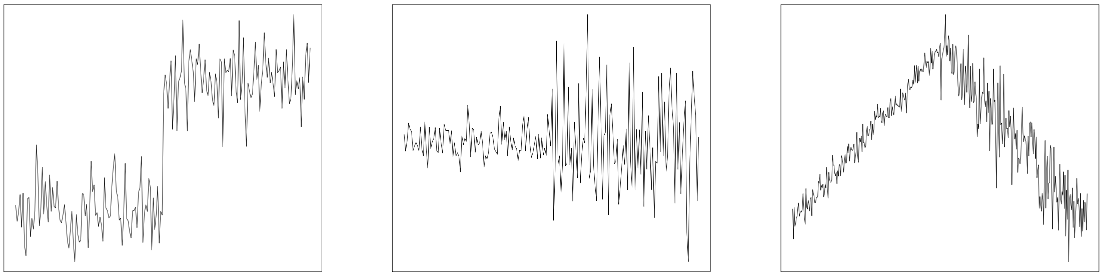
- an introduction to both univariate and multivariate changepoint analysis,
- a review of time series models for autocorrelated observations,
- an insight into how changepoints can be modelled within these dependent time series models.

Together these three components form the foundations of the work presented in this thesis.

The first part of this chapter reviews methods for detecting changes in univariate (i.e. one-dimensional) series, covering a range of different paradigms with a focus on the penalised cost function approach. Discussions of typically-used cost functions, penalties and search algorithms are provided in Sections 2.1.3 and 2.1.4. The second part examines the multivariate changepoint detection problem, with a discourse of the *fully*-multivariate and *subset*-multivariate changepoint models (definitions of which are given in Section 2.2.1). Popular fully-multivariate changepoint detection methods are examined in Sections 2.2.2 and 2.2.3, with a treatment of subset-multivariate methods deferred to Chapter 4. Finally, we present a background to the modelling of autocorrelated time series in Section 2.3 to aid the understanding of methodology introduced in Chapter 3.

2.1 Univariate Changepoint Detection

Within this section we examine many aspects of univariate changepoint detection, including the various different paradigms adopted by methods in the literature. Particular concentration is given to the penalised cost function approach. We begin our review with an introduction to the univariate changepoint model.



2.1.1(a): Change in mean.

2.1.1(b): Change in variance.

2.1.1(c): Change in multiple statistical properties.

Figure 2.1.1: An example of how various univariate changes may arise in time series.

2.1.1 Univariate Changepoint Model

Suppose that $X_{1:n} = \{X_1, X_2, \ldots, X_n\}$ denotes a univariate series of time-ordered observations of length n. The changepoint detection problem aims to identify the possible existence of m locations within the time series at which one or more of its statistical properties change. These locations are denoted by $\boldsymbol{\tau} = (\tau_1, \tau_2, \ldots, \tau_m)$, with $\tau_0 = 0$ and $\tau_{m+1} = n$. The changepoint detection problem generally consists of three main tasks:

- estimating the number of possible changepoints m within the time series,
- identifying the most suitable locations $(\tau_1, \ldots, \tau_m)$ of the m changepoints,
- determining the best-fitting model for each of the $m + 1$ segments.

An additional aim which has received an increasing amount of attention in recent years is the quantification of the uncertainty in estimated changepoint locations via confidence intervals (Hušková and Kirch, 2008; Frick et al., 2014; Nam et al., 2015).

The types of statistical property which may change include, but are not limited to, the mean, variance and regression parameters. More subtle changes such as alterations in the dependence structure of the time series may also occur, as well as changes being exhibited in multiple properties simultaneously. Figure 2.1.1 demonstrates some examples of how univariate changes in these properties may arise.

There exists a range of methodologies which have been designed for the detection of univariate changepoints. Arguably the most comprehensive approach to solving the changepoint detection problem is through the minimisation of a penalised cost function:

$$\sum_{i=1}^{m+1} \left[\mathcal{C}(X_{(\tau_{i-1}+1):\tau_i})\right] + \beta f(m), \tag{2.1.1}$$

where $\mathcal{C}(\cdot)$ denotes a generic cost function which assigns a value to a given sequence of data, β is a constant greater than 0 and $f(m)$ is some increasing function of the number of changepoints, so that $\beta f(m)$ penalises the over-fitting of changepoints. The concept behind the consideration of (2.1.1) is that the best-fitting changepoint model will have the minimum penalised cost across all possible changepoint models. Hence, minimising (2.1.1) allows for the simultaneous acquisition of the optimal number and locations of changepoints and optimal parameter values for each segment. This simultaneous optimisation has meant that the technique has been widely adopted within the literature. Hence, this is the main approach which we describe throughout the thesis.

We continue this section with an examination of three eminent approaches to changepoint detection which are not based on minimising a penalised cost: Likelihood Ratio testing, Bayesian methods, and Hidden Markov Model methods. A careful discussion of how a penalised cost function can be formulated from its constituent components, as well as highlighting some common forms of cost functions and penalties will then follow. Finally, we conclude with a consideration of popular methods which are used to minimise a penalised cost function in the context of changepoint detection.

2.1.2 Changepoint Detection Paradigms

Aside from the minimisation of a penalised cost function, there are three main paradigms which are popular within the changepoint detection literature. These are summarised below.

Likelihood Ratio Testing

The testing of a likelihood ratio is a natural approach to the single changepoint detection problem, since it is essentially the comparison of two nested models: one with a changepoint, and one without. However, it is not possible to form the multiple changepoint problem as a single hypothesis test unless the number of changepoints is known. Therefore, this approach is typically only used for the detection of a possible single changepoint within a time series. As discussed by Eckley et al. (2011), the general pair of hypothesis considered is:

$$\begin{aligned} H_0 &: \text{ No changepoint in the series.} \\ H_1 &: \text{ A single changepoint at location } \tau. \end{aligned}$$

The log-likelihood of the i.i.d. time series $X = \{X_1, X_2, \ldots, X_n\}$ under H_0 is given by

$$l_{H_0}(\theta_0|X) = \log f(X_1, X_2, \ldots, X_n|\theta_0),$$

where f is the probability density function of the distribution of the observations and θ_0 is the parameter vector of the data under H_0. Assuming that the data across the two segments is independent, the log-likelihood of X under H_1 is given by

$$l_{H_1}(\theta_1, \theta_2, \tau|X) = \log f(X_1, \ldots, X_\tau|\theta_1) + \log f(X_{\tau+1}, \ldots, X_n|\theta_2),$$

where τ denotes the changepoint location and θ_1 and θ_2 denote the parameter vectors for the segments before and after the changepoint, respectively. Denote the maximum likelihood estimate of a parameter vector θ by $\hat{\theta}$. The log-likelihood ratio test for a single changepoint within the time series X is therefore given by

$$\lambda = 2\left[\max_{1<\tau<n} l_{H_1}(\hat{\theta}_1, \hat{\theta}_2, \tau|X) - l_{H_0}(\hat{\theta}_0|X)\right].$$

This ratio, λ, is then tested against the pre-specified threshold c. If $\lambda > c$, then the null hypothesis H_0 is rejected and the changepoint is estimated at the location

$\hat{\tau} = \arg\max_{1<\tau<n} l_{H_1}(\hat{\theta}_1, \hat{\theta}_2, \tau|X)$. Otherwise, it is taken that there is no changepoint in the series.

This likelihood ratio testing approach is common in the earlier works which consider the fixed-sample changepoint detection problem. Hinkley (1970) first utilised this approach for the detection of a change in mean within a sequence of Normally distributed observations, with generalisations to other distributional forms coming later (e.g. exponential data (Haccou et al., 1987)) as well as the detection of changes in other properties (e.g. change in variance of Normal data (Chen and Gupta, 1997)). However, it is important to note that this approach identifies at most one change. We postpone a discussion of how this may be extended to a multiple changepoint setting until Section 2.1.4.

Bayesian Methods

The Bayesian paradigm is also commonly adopted within the changepoint detection literature. Typically this involves placing a prior on the number of changepoints within the series, and another prior on the locations. For example, the number of changepoints m may be drawn from a Poisson(λ) distribution, and their corresponding locations can then be independently drawn from a dU$(1, n-1)$ distribution, where dU$(\cdot, \cdot)$ denotes the discrete uniform distribution. While this specification of priors may seem intuitive, Fearnhead (2006) describes how the prior for both the number and locations of changepoints can be jointly specified indirectly via the specification of a prior on the length of a segment, and that such an approach has computational advantages over the specification of two separate priors.

To illustrate the Bayesian approach, we outline the core of the idea for the case where individual priors are placed on the number and locations of changepoints separately. Let θ_k denote the parameter vector for the k^{th} segment of the series $(k = 1, \ldots, m+1)$, and ψ_k denote the hyperparameter for the prior distribution of θ_k. Then the posterior probability of the set of m changepoints at locations

$\boldsymbol{\tau} = (\tau_1, \tau_2, \ldots, \tau_m)$ (with $\tau_0 = 0$ and $\tau_{m+1} = n$) is given by

$$\begin{aligned}
&P(m, \boldsymbol{\tau}, \theta_1, \ldots, \theta_{m+1} | X, \lambda, \psi_1, \ldots, \psi_{k+1}) \\
&\quad \propto P(m|\lambda) P(\boldsymbol{\tau}|m) P(\theta_1, \ldots, \theta_{m+1} | \psi_1, \ldots, \psi_{m+1}) P(X | m, \boldsymbol{\tau}, \theta_1, \ldots, \theta_{m+1}) \\
&\quad = P(m|\lambda) \left(\prod_{k=1}^{m} P(\tau_k) \right) P(\theta_1, \ldots, \theta_{m+1} | \psi_1, \ldots, \psi_{m+1}) \left(\prod_{k=1}^{m+1} \prod_{t=\tau_{k-1}+1}^{\tau_k} P(X_t | \theta_k) \right).
\end{aligned} \tag{2.1.2}$$

Here the $P(\theta_1, \ldots, \theta_{m+1} | \psi_1, \ldots, \psi_{m+1})$ term denotes the joint prior of the parameter vectors and

$$P(X | m, \boldsymbol{\tau}, \theta_1, \ldots, \theta_{m+1}) = \prod_{k=1}^{m+1} \prod_{t=\tau_{k-1}+1}^{\tau_k} P(X_t | \theta_k)$$

represents the likelihood of the given time series.

Popular Bayesian techniques such as MCMC and its variants can be used with (2.1.2) to estimate the true values of $\boldsymbol{\tau}$ and $\theta_1, \ldots, \theta_{m+1}$. In traditional MCMC approaches, the value of m is assumed to be known and is fixed throughout the algorithm. The MCMC algorithm then iteratively updates its estimates of $(\tau_1, \tau_2, \ldots, \tau_m)$ and $(\theta_1, \theta_2, \ldots, \theta_{m+1})$, with the values of both the changepoint locations and the parameter vectors (where possible) each being centred on their corresponding values at the previous iteration.

The assumption of a known m within traditional MCMC algorithms is often prohibitive for application in practical settings. Green (1995) introduces the 'reversible jump MCMC' (RJMCMC) method which mitigates this issue. The RJMCMC algorithm specifies an initial estimate of m, denoted m_0, and then allows for perturbation of this value via the performance of a birth-death step at each iteration. Such a birth-death step occurs after the potential new locations of changepoints currently in the model are proposed. This birth-death step proposes the possible execution of three distinct operations:

- a 'birth' operation, which introduces a changepoint into the model;

- a 'death' operation, with removes changepoint from the model;
- neither a 'birth' nor a 'death' operation, which leaves the number of changepoints in the model unchanged.

During this step, a non-negative number of births, a non-negative number of deaths, or a mixture of both birth and death operations may be proposed (and hence may occur). This includes the situation where an equal number of births and deaths are performed during the same birth-death step, thereby leaving the overall number of changepoints in the model unchanged, but perturbing the locations of some (or potentially all) of these changepoints.

Since a birth operation adds a changepoint into the model, this is equivalent to 'splitting' a single segment into two smaller consecutive segments. Hence, in addition to proposing the location of the new changepoint, it is necessary to remove the parameter vector corresponding to the split segment from the model and propose two new parameter vectors corresponding to the two new segments. Typically, the parameter values for these segments are centred on the parameter values corresponding to the split segment. For example, if the changes are occurring in the mean of the series, and at the $(i+1)^{\text{th}}$ iteration a birth operation has proposed a changepoint which splits the k^{th} segment, then the means values for the new k^{th} and $(k+1)^{\text{th}}$ segments, denoted by $\mu_k^{(i+1)}$ and $\mu_{k+1}^{(i+1)}$, may (for example) be imputed as

$$\begin{aligned} \mu_k^{(i+1)} &= \mu_k^{(i)} + u_k \\ \mu_{k+1}^{(i+1)} &= \mu_k^{(i)} + u_{k+1}, \end{aligned}$$

where $\mu_k^{(i)}$ is the mean of the k^{th} segment at the i^{th} (i.e. previous) iteration, and u_k and u_{k+1} are distinct realisations of some symmetrical random variable centred around 0 (for example, Uniform$(-1, 1)$).

Conversely, if a death operation is being performed at the $(i+1)^{\text{th}}$ iteration, the parameter vectors of the k^{th} and $(k+1)^{\text{th}}$ segments at the i^{th} iteration are 'combined' to form the parameter vector for the k^{th} segment at iteration $(i+1)$. Continuing with the example above where the changes are occurring in the mean, one approach for imputing the new mean value for the k^{th} at iteration $(i+1)$ may be by

$$\mu_k^{(i+1)} = \frac{\mu_k^{(i)} + \mu_{k+1}^{(i)}}{2}.$$

For the case when neither a birth nor death operation are performed, only the locations of the changepoints currently in the model and the corresponding parameter vectors for each of the segment are updated. The decision of what quantity of birth and/or death operations are proposed at each iteration is made randomly, with some probability being assigned to each quantity. Typically, a larger probability is assigned to zero birth and death operations, so that such operations are not performed too frequently. Green shows that this RJMCMC approach can work well for multiple changepoint models.

More recently, Bayesian methods for the detection of multiple changepoints in univariate time series have been proposed by Fearnhead (2006), Fearnhead and Liu (2007), Adams and MacKay (2007) and Wyse et al. (2011).

Hidden Markov Model Methods

Hidden Markov Models (HMMs) have also been used to facilitate the detection of multiple changepoints within time series. A HMM is a Markov model which assumes that the system of interest contains unobserved 'hidden' states. In the changepoint setting, these hidden underlying states are the segment labels. The locations of any changepoints can hence be inferred given these segment labels. The likelihood of a time series $X = \{X_1, X_2, \dots, X_n\}$ modelled as a HMM with hidden segment labels $S = \{S_1, S_2, \dots, S_n\}$ is formulated as the sum of the joint distribution of X and S over the unknown labels S:

$$\sum_S P(X,S) = \sum_S \prod_{i=1}^{n} P(X_i, S_i | S_{i-1}, X_1),$$

where X is assumed to have the first-order Markov property. A HMM can be fitted using either a classical (frequentist) or Bayesian framework, and the distribution of the segment labels S (i.e. the hidden states) given the series X, $P(S|X)$, can be inferred using (for example) the Viterbi algorithm (Viterbi, 1967) or the Expectation-Maximisation algorithm (Dempster et al., 1977).

Luong et al. (2012) provide an illustrative review of HMM methods used for changepoint detection. Specific recent contributions to the changepoint literature which utilise HMMs include Nam et al. (2012), who use finite Markov chain imbedding within a HMM framework to detect changes in fMRI data, and Nam et al. (2015) who use the Locally Stationary Wavelet framework of Nason et al. (2000) within a HMM framework to detect changes in the autocovariance of a time series and quantify the uncertainty in such changepoints.

We now turn out attention to the penalised cost function approach, which is arguably one of the most popular approaches to changepoint detection. We first consider how such a penalised cost function can be formulated.

2.1.3 Formulating Penalised Cost Functions

The penalised cost function approach is one of the most widely-used approaches to the univariate multiple changepoint problem. An important characteristic of this problem is that the addition of a further changepoint into a model will always reduce the model's cost. Therefore, to regulate the trade-off between a reduced cost and a parsimonious model, a penalty value can be added to the cost for each time a changepoint is introduced. This means that the overall 'best' model will provide a good fit using a reasonable amount of changepoints. This is the foundation of the penalised cost function approach.

Recall the form of a penalised cost function for a time series $X = \{X_1, \ldots, X_n\}$ with changepoints $\boldsymbol{\tau} = \{\tau_1, \ldots, \tau_m\}$, originally presented in equation (2.1.1) of Section 2.1.1:

$$\sum_{i=1}^{m+1} \left[\mathcal{C}(X_{(\tau_{i-1}+1):\tau_i}) \right] + \beta f(m).$$

There are two key components to such a function: the cost function $\mathcal{C}(\cdot)$, which provides a measure of fit for a given segment of data, and a penalty term $\beta f(m)$ which is used to penalise (i.e. increase) the cost of a segment for each additional changepoint which is included in the segment. We now discuss some cost functions and penalty types which have been utilised within the changepoint literature.

Cost Functions

Traditionally a property of any cost function to be used as part of a penalised approach is that it should be additive, so that costs for multiple segments can be summed easily as in equation (2.1.1). The types of cost functions which are used for changepoint detection largely fall into two categories: those which are based upon the likelihood of the data, and are hence parametric, and those which are non-parametric and do not assume a model form. Typically, likelihood-based cost functions are a scaled version of the negative log-likelihood:

$$-\log L(\theta | X_{(\tau_{i-1}+1):\tau_i}),$$

where θ is the vector of the model parameters (see, for example, Chen and Gupta (2000) and Eckley et al. (2011)).

The most common non-parametric cost function which has been considered in the literature is the quadratic loss function:

$$\sum_{t=\tau_{i-1}+1}^{\tau_i} (X_t - \hat{\mu}_i)^2,$$

where $\hat{\mu}_i$ is the sample mean of the segment of data between $\tau_{i-1} + 1$ and τ_i. For illustrative examples, see Inclan and Tiao (1994) and Rigaill (2010). Of course, any well-defined non-parametric function can be used within the penalised cost function. The differences between using a likelihood-based approach and a non-parametric approach boil down to the traditional arguments. A likelihood-based cost function can be more powerful, but it imposes the assumption that the observed data follows a certain parametric model; such an assumption may not be true in practice. Conversely, a non-parametric approach means that no assumptions regarding the form of the data are necessary, but the power of the method is reduced in comparison to a likelihood-based approach.

From herein, it is assumed that any cost function considered is such that a better-fitting model results in a lower cost, and so the aim is to minimise the value of the penalised cost function.

Penalty Types

The penalty term $\beta f(m)$ from the penalised cost function in equation (2.1.1) can be decomposed into two parts: (i) the function $f(m)$, which is increasing with the number of changepoints m, and (ii) the constant β. The most common approach is to set $f(\cdot)$ as a linear function, so that $f(m) = m$ (see, for example, Killick et al. (2012)). The value of β, on the other hand, has received much wider attention. Popular values used for β within the literature include the Schwarz information criterion (SIC) (Schwarz, 1978), also known as the Bayesian information criterion (BIC), and the Akaike information criterion (AIC) (Akaike, 1974):

$$\text{SIC / BIC} : \beta = v \log(n)$$
$$\text{AIC} : \beta = 2v,$$

where v is the number of additional parameters introduced in the model by adding an additional changepoint. Yao (1988) was the first to use the SIC in the context of estimating changepoints, establishing the consistency of estimation for the number of changepoints in the case of Normally distributed data. The Hannan-Quinn information criterion (HQIC) (Hannan and Quinn, 1979) is another possible penalty value:

$$\text{HQIC} : \beta = 2 \log \log n,$$

although it has received comparatively little attention within the changepoint literature. This is likely due to the value of $\log \log n$ being very small even for large n, a point supported by Burnham and Anderson (2002). Such a small penalty value would likely lead to the over-fitting of changepoints. More complicated penalty forms have also been considered within the literature. Often, these correspond to setting a more sophisticated function for $f(m)$ and $\beta = 1$. For example, Lebarbier (2005) presents a penalty which is quadratic in m and Zhang and Siegmund (2007) have proposed a modified version of the SIC, both of which incorporate the length of the current segment of interest.

Another possible penalty choice, arising from information theory, is the Minimum Description Length (MDL). First proposed by Rissanen (1989), the essence of the MDL is based on the principle that the best-fitting model is the one that gives the best compression of the data; in other words, the one which requires the lowest computational cost to encode the data. The main premise on which the calculation of the MDL is based is that for an unbounded integer I, roughly $\log_2 I$ bits are required for it to be encoded. A more complex model implies a larger encoding cost, and therefore a larger penalty. For a model with parameter set $\boldsymbol{\theta}$, the MDL can be summarised as

$$\text{MDL}(\boldsymbol{\theta}) = \text{cost}(\text{encoding } \boldsymbol{\theta}) + \text{cost}(\text{assessing quality of fit of model } \boldsymbol{\theta}).$$

As demonstrated by Rissanen (1989), the cost of assessing the quality of fit of a certain model is equal to the negative log-likelihood of that model. As such, the penalty term for the MDL is equal to the cost of encoding $\boldsymbol{\theta}$, and this penalty value is only applicable to likelihood-based cost functions. The MDL has been used in the changepoint setting by Davis et al. (2006) and Li and Lund (2012) for the detection of changes in autocorrelation.

2.1.4 Searching for Multiple Changepoints

An important aspect of changepoint detection is the search for multiple changepoints. In this section, we maintain our focus on the penalised cost approach and consider how multiple changepoints can be obtained within this framework.

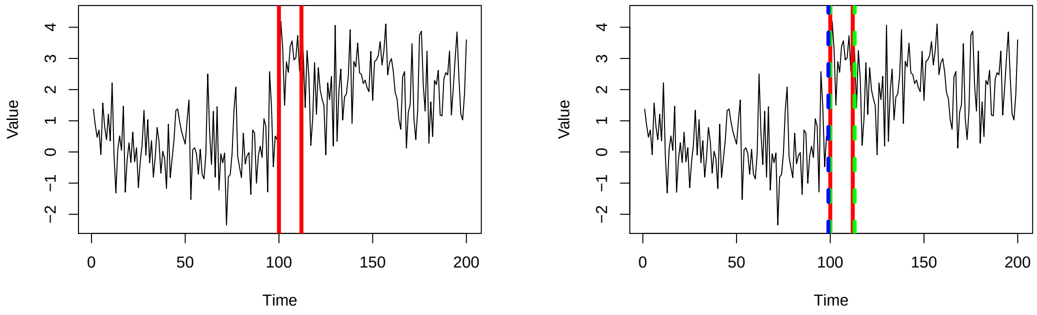
Once a penalised cost function has been formed, the optimal changepoint locations are obtained by minimising the function over all possible number and locations of changepoints, and all possible parameter values (conditional on the changepoints). A wide range of optimisation and heuristic methods can be used to obtain this minimum. The most widely used search methods within the literature fall into three main categories: binary segmentation (and its related variants), dynamic programming methods, and those based on genetic algorithms. We briefly introduce these univariate search approaches below.

Binary Segmentation and Related Variants

Binary segmentation is a generic search method which allows for the estimation of both the number and location of changepoints. It operates by recursively applying any single changepoint detection method, thereby allowing for the detection of multiple changepoints. Initially, the single changepoint method is applied to the the entire dataset. If a changepoint is detected then this location is fixed as an estimated changepoint, and the single changepoint method is applied again to the segments of data either side of the estimated changepoint. Such a process is repeated until no more changepoints are detected in any of the data segments. For the penalised cost approach, the single changepoint method used is the likelihood ratio test (described in Section 2.1.2) or a non-parametric equivalent. If, at a given stage of the procedure, m_0 and m_1 represent the total number of changepoints under the null and alternative hypotheses, respectively, then the threshold c for the likelihood ratio test is $c = -\beta\Big(f(m_1) - f(m_0)\Big)$. This procedure is able to run very quickly, and consequently binary segmentation can obtain a segmentation of a dataset with a typical computational cost of $\mathcal{O}(n \log n)$ (Eckley et al., 2011). However, the local nature of the estimation (with changepoint locations being fixed mid-way through the procedure) means that binary segmentation is an approximate search, and cannot guarantee to produce the optimal changepoint locations.

First implemented by Scott and Knott (1974), binary segmentation has been used to detect changes in independent Normal observations by Venkatraman (1993) and Chen and Gupta (1997). Cho and Fryzlewicz (2012) and Killick et al. (2013) use the method to detect changes in the second order structure of univariate time series, based on a wavelet approach. Venkatraman (1993) and Cho and Fryzlewicz (2012) prove consistency of the procedure for unknown changepoints with additive and multiplicative errors, respectively.

A consequence of binary segmentation's approximate nature is that if a series contains changepoints which are relatively close together, then standard binary segmentation may not be able to detect both changepoints. Such a problem has been noted by Killick et al. (2013) and Fryzlewicz (2014). To demonstrate this, we consider a sequence of 200 Normally distributed observations containing changes in mean at times 100 and 115, presented in Figure 2.1.2(a). The changepoints detected by performing binary segmentation and an exact univariate detection method PELT (discussed in more detail below) are shown in Figure 2.1.2(b). It can be seen that binary segmentation detects the changepoint at time 100, but misses the changepoint after the short segment at time 115. In contrast, the exact method detects both changepoints (with only a small amount of error in location). In an effort to alleviate



2.1.2(a): A sequence of 200 Normally distributed observations containing changes in mean at times 100 and 115, shown by the red lines.

2.1.2(b): The changepoint locations estimated by binary segmentation (blue dashed) and PELT (green dashed), along with the true changepoints (red solid).

Figure 2.1.2: An example demonstrating the weakness of binary segmentation in cases of small segments.

this small segment issue, Fryzlewicz (2014) have proposed a modified version of the method known as 'wild binary segmentation' (WBS). This maximises the test statistic calculated on random intervals, thereby using more information to inform about possible changepoint locations. This sacrifices computation time for an increase in accuracy.

The 'circular binary segmentation' (CBS) algorithm of Olshen et al. (2004) adapts standard binary segmentation to allow for the detection of 1 or 2 changes at each stage of the algorithm. This change is motivated by the detection of variations in DNA copy number, which typically appear as pairs of changepoints.

Dynamic Programming Based Approaches

The concept of dynamic programming is to provide the globally optimal solution of any problem which can be formulated as a 'shortest path' problem. Changepoint detection can be viewed as a problem of this type. In this context, a dynamic programming method works by algorithmically finding the lowest cost from the beginning of the series to each time-point as if it were the end of the series. Once the algorithm reaches the end of the data, every possible changepoint segmentation has been considered, and so the globally optimal configuration of changepoints can be output. This therefore means that dynamic programming is an exact search procedure. This realisation has led to the development of numerous changepoint detection methods which utilise dynamic programming techniques.

Perhaps the earliest example of dynamic programming in the changepoint setting is the Segment Neighbourhood Search approach of Auger and Lawrence (1989). This assumes a maximum number of changepoints M, and for each $m = 1, \ldots, M$ performs a dynamic program to obtain the configuration of m changepoints which best partitions the series. This provides the user with a wide range of possible segmentations. Each individual program requires $\mathcal{O}(n^2)$ calculations, so the total order of computation of segment neighbourhood search is $\mathcal{O}(Mn^2)$. The drawback of this procedure is that the true maximum number of changepoints may not often be known in practice. Therefore, it is difficult to guarantee that the globally optimal set of changepoint locations has been obtained. Further, since a range of segmentations are returned, it may be difficult to determine which segmentation is the best overall. This may be done via the addition of a penalty for each additional changepoint, or by the consideration of an elbow plot that demonstrates which model provides the biggest relative reduction in cost (Lavielle and Teyssiere, 2006).

Jackson et al. (2005) improve upon segment neighbourhood search with their seminal Optimal Partitioning (OP) methodology. This produces the optimal segmentation of a series in a single pass and requires no assumption on the maximum number of changepoints. However, it can only be applied to linear penalty functions, i.e. $f(m) = m$. Similar to segment neighbourhood search, OP works by recursively calculating the minimum cost $F(t)$ up to each time-point $t = 1, 2, \dots, n$ using the formula

$$F(t) = \min_{0 \leq \tau < t} \left\{ F(\tau) + \mathcal{C}(X_{(\tau+1):t}) + \beta \right\},$$

where β is the changepoint penalty and $\mathcal{C}(\cdot)$ is the cost function. Using t^* to denote the optimal changepoint prior to t, we have

$$t^* = \arg \min_{0 \leq \tau < t} \left\{ F(\tau) + \mathcal{C}(X_{(\tau+1):t}) + \beta \right\}.$$

Setting $\tau_0 = 0$ and $\tau_{m+1} = n$, then the i^{th} element of the optimal configuration of changepoints $\boldsymbol{\tau}$ is denoted by τ_i, with $\tau_i = \tau^*_{i+1}$ for $i = 0, \dots, m$. Therefore, we have

$$\boldsymbol{\tau} = (\tau_0 = \tau^*_1, \tau_1 = \tau^*_2, \dots, \tau_m = \tau^*_{m+1}, \tau_{m+1}) = (0, \tau_1, \dots, \tau_m, n).$$

This configuration is optimal over all possible number and locations of changepoints. Since OP can be performed with one pass of the data, it requires $\mathcal{O}(n^2)$ calculations. However, for larger values of n, even this reduced order of computation can become practically infeasible.

To reduce this computation time whilst maintaining an exact search, Rigaill (2010), Killick et al. (2012) and Maidstone et al. (2014) each utilise the concept of pruning to remove unnecessary calculations from these dynamic programming procedures. Each consider a different combination of search method and pruning type. Rigaill's pDPA ('pruned Dynamic Programming Algorithm') implements functional pruning within the SNS method. This reduces the range of values to be considered for the parameter of interest.

The PELT ('Pruned Exact Linear Time') method of Killick et al. (2012) adapts optimal partitioning to include an inequality-based pruning step. This allows the method to run in $\mathcal{O}(n^2/m)$ time, where m is the estimated number of changepoints. Hence, if the number of changepoints in the series is $\mathcal{O}(n)$ then under certain conditions the method is able to run in $\mathcal{O}(n^2/n) = \mathcal{O}(n)$ time. The pivotal theorem introduced by Killick et al. (2012) states that if there exists some non-negative constant K such that the following holds for some time-point r:

$$F(r) + \mathcal{C}(X_{(r+1):s}) + K > F(s), \tag{2.1.3}$$

then at a future time $t > s$, r can never be the optimal last changepoint prior to t. Typically, the vast majority of cost functions used in practice satisfy this condition. Killick et al. (2012) provide full details on the value of K, but if $\mathcal{C}(\cdot)$ is the negative log-likelihood then $K = 0$. If condition (2.1.3) holds, then r does not need to be considered in the calculations for a future time greater than t within the remainder of the dynamic program. To illustrate the full form of PELT, pseudocode adapted from Killick et al. (2012) is presented in Algorithm 1.

Maidstone et al. (2014) present two self-explanatory methods: FPOP ('Functional Pruning in Optimal Partitioning') and SNIP ('Segment Neighbourhood with Inequality Pruning'). The authors show that FPOP has strong performance whilst SNIP performs poorly. FPOP works in a similar manner to PELT, with a functional pruning step performed in place of inequality pruning. It is also shown that FPOP will always prune more than PELT. This means that performances of FPOP can result in computation times which are competitive with (or even faster than) binary segmentation. In contrast to PELT, this computation time increases with the number of

Algorithm 1: PELT (Pruned Exact Linear Time)

```
Input     : A set of observations $(X_1, X_2, \ldots, X_n)$, a function $\mathcal{C}(\cdot)$ which
            assigns a cost to a contiguous set of data and satisfies condition
            (2.1.3) for some non-negative constant $K$, and a penalty constant $\beta$
            which is independent of the number and location of changepoints.
Initialise: Let $n$ be the length of the observation sequence. Set $F(0) = -\beta$,
            $cp = \emptyset$, $R_1 = \{0\}$.
begin
    for $\tau^* = 1, \ldots, n$ do
        Calculate $F(\tau^*) = \min_{\tau \in R_{\tau^*}} \left[F(\tau) + \mathcal{C}(X_{(\tau+1):\tau^*}) + \beta\right]$
        Set $\tau' = \arg\min_{\tau \in R_{\tau^*}} \left[F(\tau) + \mathcal{C}(X_{(\tau+1):\tau^*}) + \beta\right]$
        Set $cp(\tau^*) = \tau'$
        Set $R_{\tau^*+1} = \left\{\tau^* \cap \{\tau \in R_{\tau^*} : F(\tau) + \mathcal{C}(X_{(\tau+1):\tau^*}) + K < F(\tau^*)\}\right\}$
    Set $\tau_{m+1} = n$
    for $k = m+1, m, \ldots, 1$ do
        Set $\tau_{k-1} = cp(\tau_k)$

Output    : The vector $(\tau_0, \tau_1, \ldots, \tau_m, \tau_{m+1})$ which contains the optimal
            changepoints within the time series (including the start- and
            end-points of the data).
```

changepoints in the series.

However, FPOP is less-widely applicable than PELT, as functional pruning requires a stronger condition than inequality-based pruning. Further, functional pruning methods can only be used to detect changes in a single parameter, whereas PELT can detect changes in multiple parameters simultaneously.

The removal of unnecessary calculations means that under certain conditions, PELT and FPOP each require only $\mathcal{O}(n)$ calculations. Similarly, under certain (but different) conditions pDPA only requires $\mathcal{O}(Kn)$ calculations to obtain the optimal segmentation containing k changepoints for each $k = 1, \ldots, K$. Killick et al. (2012) prove this for PELT, whereas Rigaill (2010) and Maidstone et al. (2014) demonstrate the run-times empirically for pDPA and FPOP, respectively. If such conditions are not met, then the order of computation is not necessarily linear in n. In the worst cases, no pruning is performed, in which case PELT and FPOP are equivalent to optimal partitioning and pDPA is equivalent to segment neighbourhood search, with the respective computational costs therefore being $\mathcal{O}(n^2)$ and $\mathcal{O}(Kn^2)$.

SMUCE ('Simultaneous Multiscale Changepoint Estimator') is a novel method which has been recently presented by Frick et al. (2014). SMUCE enables the estimation of the number and location of changes in regression for univariate time series where piecewise-constant distributions of the observations arise from the exponential family. The authors provide theory which demonstrates that the proposed approach maximises the probability of correctly estimating the correct number of changepoints. In addition to the estimation of the number and locations of changepoints, SMUCE is able to estimate confidence bands for the step function representing the underlying signal as well as provide confidence intervals for the estimated changepoint locations $\{\tau_k\}$.

SMUCE uses dynamic programming to minimise a multiscale test statistic representing the likelihood across a range of possible step functions. Inequality-based pruning is also performed, allowing for an improvement in the computational performance. The main disadvantage of SMUCE is that, similar to FPOP, it only allows for the detection of changes in a single parameter within a single performance. This can therefore limit its practical applicability.

Genetic Algorithms

A genetic algorithm is an approximate search procedure which allows for the synchronous estimation of the number and location of changepoints, as well as the model parameters for each segment. Such an algorithm functions by encoding every possible solution as a 'gene' (or 'chromosome'). Throughout the procedure, a set of solutions is held in memory, known as the 'population'. These solutions are then allowed to evolve over time through the application of a series of operations designed to randomly make changes to their characteristics whilst retaining the best-performing solutions at each stage. The underlying principle is that such a procedure exhibits natural selection, with the continual evolution resulting in a solution which is close (or equal) to the global optimum.

Genetic algorithms have seen successful application in changepoint detection. In particular, Davis et al. (2006) utilise a bespoke genetic algorithm to minimise the minimum description length (MDL) to obtain the optimal piecewise autoregressive models for a univariate time series. They call their procedure Auto-PARM (Automatic Piecewise AutoRegressive Models). More recently, Polushina and Sofronov (2011) introduce a genetic algorithm approach to detect multiple changepoints in DNA sequences, and Li and Lund (2012) follow the example of Davis et al. (2006) and use a genetic algorithm with the MDL to detect multiple changepoints in the mean of climatic time series.

The advantage of a genetic algorithm is that the procedure can rapidly obtain multiple changepoint segmentations of high quality. However, since this is an approximate search, there is no guarantee that the process will obtain the optimal configuration of changepoints. Furthermore, the random nature of the algorithm means that repeated runs may not consistently produce the same solution.

2.2 Multivariate Changepoint Detection

The problem of detecting changepoints in multivariate time series is conceptually similar to that of the univariate setting. However, a key difference is that the changes sought can occur in the multidimensional parameters of the series. We formalise this difference by considering the multivariate changepoint model.

2.2.1 Multivariate Changepoint Model

Suppose that $\boldsymbol{X}_{1:n} = \{\boldsymbol{X}_1, \boldsymbol{X}_2, \ldots, \boldsymbol{X}_n\}$ denotes a multivariate time series containing observations from p variables, such that $\boldsymbol{X}_t = (X_t^1, X_t^2, \ldots, X_t^p)$ for $t = 1, \ldots, n$. In addition, suppose that the series contains m distinct changepoints, the locations of which are denoted by $\boldsymbol{\tau} = \{\tau_1, \tau_2, \ldots, \tau_m\}$, where $\tau_i < \tau_j$ for $i < j$. The definitions $\tau_0 = 0$ and $\tau_{m+1} = n$ are made as before.

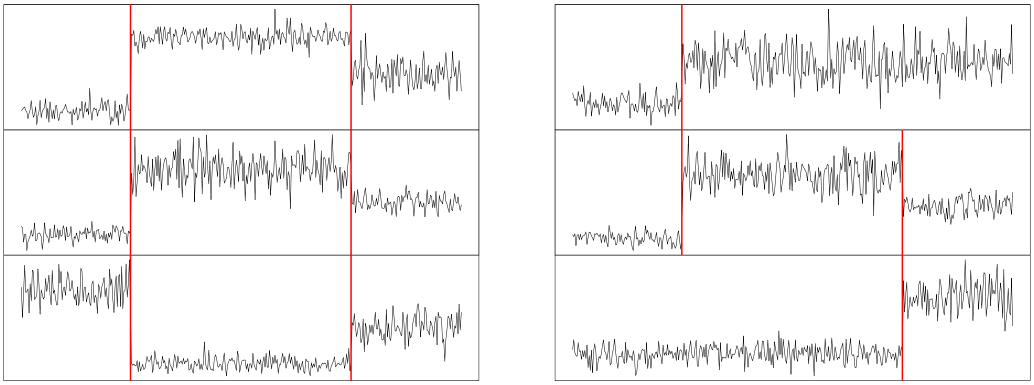
At any given changepoint location either some or all of the variables may alter. This gives rise to two different settings for the multivariate changepoint problem: the *fully-multivariate* changepoint model and the *subset-multivariate* changepoint model.

For the i^{th} changepoint τ_i, denote the subset of variables affected by the change by $\mathcal{S}_i$. Under the fully-multivariate changepoint model, the value of $\mathcal{S}_i$ is fixed as $\mathcal{S}_i = \{1, \ldots, p\}$ for each $i = 1, \ldots, m$. Conversely, under the subset-multivariate changepoint model, $\mathcal{S}_i$ is able to be any possible subset of the observed variables, so that $\mathcal{S}_i \subseteq \{1, \ldots, p\}$ for each $i = 1, \ldots, m$. Therefore, while the fully-multivariate changepoint problem aims to find only the optimal set of changepoint locations $\boldsymbol{\tau} = \{\tau_1, \ldots, \tau_m\}$, the objective of the subset-multivariate changepoint problem is to obtain *both* the optimal values of $\boldsymbol{\tau} = \{\tau_1, \ldots, \tau_m\}$ *as well as* the optimal associated subsets of affected variables, $\boldsymbol{\mathcal{S}} = \{\mathcal{S}_1, \ldots, \mathcal{S}_m\}$. Note that $\mathcal{S}_0$ and $\mathcal{S}_{m+1}$ are fixed such that $\mathcal{S}_0 = \mathcal{S}_{m+1} = \{1, \ldots, p\}$. Using the same nomenclature as above, we refer to changepoints which occur in all variables as *fully-multivariate changepoints*, and those which occur in only a subset of the variables as *subset-multivariate changepoints*. Due to its formulation, the subset-multivariate changepoint model is capable of detecting both subset-multivariate and fully-multivariate changepoints (where the 'subset' for a fully-multivariate changepoint is the improper subset).

To illustrate the difference between fully- and subset-multivariate changepoints, we display an example of each in Figure 2.2.1. In Figure 2.2.1(a), the two changepoints occur in all variables and are hence fully-multivariate. Conversely, in Figure 2.2.1(b) the two changepoints occur in (different) subsets of the variables.

The vast majority of multivariate detection methods assume that any changepoints present in a series are fully-multivariate. See, for example, Lavielle and Teyssiere (2006) or Matteson and James (2014). Such an assumption is often implicit, and is likely made in many cases due to the difficulty of explicitly identifying changes which are restricted to only a subset of variables. However, assuming the fully-multivariate changepoint model in scenarios where subset-multivariate changepoints may be present could lead to fallacious inference.

Due to these differences between the fully-multivariate and subset-multivariate changepoint models, we provide a separate treatment to detection methods which assume each. We forthwith consider fully-multivariate changepoint detection methods, and postpone examination of subset-multivariate changepoint methods to Chapter 4.



2.2.1(a): Fully-multivariate changes.

2.2.1(b): Subset-multivariate changes.

Figure 2.2.1: Two examples highlighting the differences between fully-multivariate changes and subset-multivariate changes.

Contributions in the area of fully-multivariate changepoint detection can be divided into methods which identify at most one changepoint and multiple changepoint methods. We begin by considering the former (Section 2.2.2) before describing recent multiple changepoint contributions (Section 2.2.3).

2.2.2 At Most One Changepoint (AMOC) Methods

One of the earliest contributions in the AMOC setting is provided by Srivastava and Worsley (1986), who consider the detection of a single change in the mean vector of a series of multivariate Normal observations. A likelihood-ratio testing approach is used to search for such a change, and such a likelihood-ratio statistic corresponds to the maximum Hotelling T^2 statistic. An approximation to this statistic is given which provides a theoretically-supported value for the threshold of the test.

Since it is assumed that the change is in the mean, the effect of the variance is neutralised by standardising the observations on either side of the potential changepoint being considered. Such an approach means that the sample mean values either side of the possible changepoint can be fairly compared, and allows for the development of the theory. This use of standardisation within the test statistic remains a core component of many modern multivariate changepoint methods. However, this approach is limited to the detection of a change in mean only, and so if the change in mean was accompanied by a substantial change in variance at another location, then the performance of this approach may deteriorate.

Other methods which take a parametric approach are those of Horváth and Hušková (2012) and Batsidis et al. (2013), who consider the detection of a single change in the mean of panel data and in the probability vectors of a sequence of multinomial observations, respectively. The statistics used in each of these methods are based on scaled divergences of the observations before and after the proposed changepoint. These approaches are useful for practitioners wishing to detect a change in these types of data, but otherwise they do not generalise to data arising from other distributions and hence have a relatively limited applicability in general.

Conversely, Aue et al. (2009) take a non-parametric approach for the detection of a single change in the covariance structure of a zero-mean multivariate time series. They propose two separate test statistics which can be used to detect sudden changes and more gradual changes in covariance, respectively. The advantage of such a non-parametric approach is that it does not assume a distributional form, and so it can be applied to largely any type of time-ordered discrete data.

2.2.3 Multiple Changepoint Methods

We now turn to consider methods which are capable of detecting multiple changes in multivariate series. Recent contributions in the literature can be categorised into those which utilise binary segmentation, dynamic programming methods and alternative techniques. We examine methods from each category in turn.

Binary Segmentation Methods

Due to its fast computational performance, binary segmentation has been adapted to the case of multivariate changepoint detection. However, as in the univariate setting, it remains an approximate search method.

Commonly, multivariate binary segmentation changepoint methods appear within the literature as extensions to single multivariate changepoint detection methods. Srivastava and Worsley (1986) and Aue et al. (2009) are archetypal examples of such cases. Srivastava and Worsley (1986) justify the application of such a binary segmentation process to their likelihood-ratio testing procedure using the result of Vostrikova (1981), who shows that such a procedure consistently estimates all of the changepoints in a multivariate time series in the case of a known covariance matrix Σ. Aue et al. (2009) develop asymptotic theory for the utilisation of each of their proposed non-parametric test statistics within a binary segmentation mechanism, which justifies the usage of such a procedure for the detection of multiple changes in the variance-covariance structure of a multivariate time series.

Modern methods utilising binary segmentation often structure it as the core of their approach, rather than being an extension of a single changepoint detection method. The work of Matteson and James (2014) is a remarkable example of such methodologies.

The method proposed by Matteson and James (2014), termed ‘E-Divisive’, provides a non-parametric procedure for the estimation of the number and locations of any changepoints in a set of multivariate observations, subject to the condition that the observations are piecewise i.i.d. and the α^{th} absolute moment exists for all $\alpha \in (0, 2)$. This means that the method is unable to detect changes in the second-order structure (i.e. auto-covariance and cross-covariance) of the series. The approach taken is based upon the concept of hierarchical clustering, and combines the calculation of Euclidean distances between the multivariate observations with the use of a binary segmentation technique. The premise is that the most likely changepoint location will maximise the ‘distance’ between the two sub-segments.

An attempt is made to mitigate the weakness of binary segmentation where a slight misspecification of the estimated changepoint location can have a compounding effect as the binary segmentation algorithm proceeds. This is done by perturbing the end-point of the sub-segment being considered. This reduces the effect of the misspecified changepoint location which may erroneously influence the cost of the segment by introducing ‘noise’ at the end of the segment. Perturbing the end-point means that this noise is disregarded.

Since the underlying distribution of the observations is unknown, a permutation test is used to generate an approximate p-value to test the significance of the resulting changepoint configuration. While this is intuitive, it is not theoretically justified and producing an exact p-value in this manner is computationally intractable. In addition, the non-parametric nature of the method means that it suffers from the trade-off of wide applicability against the loss of power compared with parametric methods.

Dynamic Programming Methods

The exact nature of the search provided by dynamic programming has meant that such methods remain popular in the multivariate literature. As in the univariate setting, such dynamic program formulations require the problem to be structured as the minimisation of a penalised cost function. For the multivariate problem, these are typically of the form

$$cost(\boldsymbol{X}, \boldsymbol{\tau}) + \text{pen}(\boldsymbol{\tau}),$$

where $cost(\boldsymbol{X}, \boldsymbol{\tau})$ provides a cost for a multivariate time series $\boldsymbol{X}$ segmented by the changepoint configuration $\boldsymbol{\tau}$, and $\text{pen}(\boldsymbol{\tau})$ is a penalisation function which adds a penalty to the cost, the value of which depends on the changepoint configuration $\boldsymbol{\tau}$ being considered. Typically, a larger number of changepoints leads to a larger penalty value. Commonly, this penalisation function can be decomposed such that

$$\text{pen}(\boldsymbol{\tau}) = \beta f(m),$$

where β and $f(m)$ are exactly as in the univariate case. Prominent examples of works which have utilised such a dynamic programming approach include that of Lavielle and Teyssiere (2006), Maboudou and Hawkins (2009), Lung-Yut-Fong et al. (2011b) and James and Matteson (2015).

Lavielle and Teyssiere (2006) use dynamic programming within a penalised cost function framework to detect multiple changes in the covariance structure of a multivariate time series. Such series may be i.i.d., weakly or strongly dependent. Two separate cost functions are considered for detecting changes in the covariance matrix only, and changes in the mean vector and/or covariance matrix. These costs are proportional to the negative log-likelihood for each case.

The penalised cost function is minimised in a similar manner to that of segment neighbourhood search (discussed in Section 2.1.4) by calculating the optimal changepoint locations for a fixed number of changepoints m for each $m = 1, \dots, M$, where M is a pre-defined upper bound on the total number of changepoints. The cost assuming no changepoints in the series is also calculated. Performing this dynamic program for a fixed m works in exactly the same manner as the univariate setting. Assuming the calculation of the multivariate cost for a single changepoint configuration requires $\mathcal{O}(p)$ calculations (where p is the number of variables in the series), then a single program requires $\mathcal{O}(pn^2)$ calculations. Hence, the overall order of computation of the algorithm is $\mathcal{O}(Mpn^2)$. This computational cost means that, when M is large, the method performs slowly even for series of relatively modest length.

An important contribution of this method is a procedure for adaptively choosing the value of the penalisation parameter β for a given cost function and given penalty function. Such a data-driven approach is favourable as it removes the requirement of the practitioner having to choose the value of the penalisation parameter β.

James and Matteson (2015) utilise the dynamic programming approach of Lavielle and Teyssiere (2006), but instead they use it to maximise the non-parametric test statistic used in E-Divisive (Matteson and James, 2014). To improve the computation time of this statistic, they instead use an approximated statistic which only incorporates the data around the possible changepoint in consideration, rather than the whole time series. Therefore, this approach (known as E-CP3O) is an exact search with an approximate test statistic, whereas E-Divisive has an approximate search with an exact test statistic. As such, despite its use of an exact search, E-CP3O is an approximate method and therefore cannot guarantee to produce the optimal changepoint locations in a multivariate series.

Maboudou and Hawkins (2009) use a penalised cost function within a dynamic programming algorithm to detect changes in the mean vector and covariance matrix of multivariate Normal observations. Due to this modelling assumption, the cost function used is twice the negative log-likelihood of the multivariate Normal data across all segments. The penalty term is taken to be the SIC (Schwarz Information Criterion) for multivariate Normal data proposed by Chen and Gupta (2000), so that within the penalised cost function,

$$\beta f(m) = \frac{p(p+3)\log(n)}{2} m.$$

In the same manner as Lavielle and Teyssiere (2006), Maboudou and Hawkins (2009) use a segment neighbourhood search approach to minimise the penalised likelihood for each $m = 1, \ldots, M$. Hence, the computational cost of this approach is also $\mathcal{O}(Mpn^2)$. This approach can therefore perform slowly in practice. In addition, while the choice of the SIC penalty is theoretically supported, it does not have the adaptive nature of that of Lavielle and Teyssiere (2006).

The previous three methods all assume that the observations follow a multivariate Normal distribution. While such an assumption allows for ease of modelling, it is not necessarily always true in practice. Lung-Yut-Fong et al. (2011b) avoid this issue with their MultiRank procedure by utilising a non-parametric rank statistic within a segment neighbourhood search framework to detect any general statistical change in the series. This means it can be applied to a much wider class of processes, however it loses the power of the parametric methods in detecting changes, and hence the magnitude of change needs to be comparatively much larger before it is detected.

As for the univariate problem, the benefit of dynamic programming approaches is that they are exact searches and hence guarantee to obtain the optimal configuration of changepoints for the given penalised cost function. This is provided a high enough maximum number of changepoints M is used for the segment neighbourhood search based procedures. However, this comes with the price of a high computational cost compared to the fast approximate search procedures based on binary segmentation.

Other Multiple Changepoint Methods

While techniques based on dynamic programming and binary segmentation form a large body of the multivariate changepoint detection literature, there are also various methods available which utilise alternative methods for detecting multivariate changes. Examples of these methods which have received considerable attention are the SLEX method proposed by Ombao et al. (2005) and the work of Vert and Bleakley (2010) who utilise the group LASSO to minimise the considered penalised cost function.

Ombao et al. (2005) utilise the SLEX (smooth localised complex exponentials) collection of bases for the segmentation of multivariate time series. These series are segmented such that each segment within the series is 'stationary', which means that the auto- and cross-correlation is constant within a single segment and piecewise constant across the whole series. Therefore, the method is designed strictly for the detection of changes in auto- and cross-correlation. The optimal changepoints are found by minimising a penalised cost function across all possible changepoint locations and the SLEX collection of bases.

A major disadvantage of this SLEX approach is that the bases within the SLEX library are all of dyadic length. Hence, the method is only capable of detecting changes which occur at dyadic time-points within the series. For practical application, such an assumption is highly restrictive, and so such an approach is likely to be unsuitable for usage in a wide range of scenarios.

Vert and Bleakley (2010) consider the problem of detecting multiple changes in the mean vector of a multivariate data series. However, rather than considering the optimisation of a cost function which is penalised by the (non-convex) number of changepoints, they consider penalising by the total variation instead (which is convex). This penalisation takes the form of the l_1 norm of increments of the different segments of the data series. Structuring the problem in this manner allows it to be formulated as a group LASSO and can hence be solved approximately using a group LARS procedure (Yuan and Lin, 2006). This approximate solution can be obtained in $\mathcal{O}(mnp)$ calculations, where m is the number of changepoints and n and p are the length and dimension of the series, respectively. However, since this approach is approximate it cannot guarantee to provide the globally-optimal set of changepoint locations.

2.3 Modelling Dependent Time Series

In this section, we break from considering changepoint detection methods and instead turn to consider common techniques used within the literature to model time series that contain dependence between their observations. The focus will be on univariate time series only, and hence the dependence that will be studied in this section will be autoregressive in nature (i.e. autocorrelation and autocovariance). We will also briefly explore methods which have been proposed for the detection of changes in such dependence structure.

Note that only methodology relating to stationary time series will be examined in this section. For a review of non-stationary time series methods based upon the wavelet paradigm, see Nason (2008).

2.3.1 Stationary Time Series Models

We begin our exploration of stationary time series by examining the concept of stationarity and introducing the autocovariance function. We then move on to consider some popular stationary time series models utilised in the time series literature.

Stationarity and the Autocovariance Function

Stationarity is one of the core concepts of time series analysis. A *stationary* time series is one whose dependence structure does not vary over time. This implies that if a time series $X = \{X_1, X_2, \ldots, X_n\}$ is stationary, then the relationships between the values within the subset of observations

$$\{X_{t_1}, X_{t_2}, \ldots, X_{t_i}\}$$

is the same as the time-shifted subset of observations

$$\{X_{t_1+h}, X_{t_2+h}, \ldots, X_{t_i+h}\}.$$

This can be expressed in terms of probability as follows:

$$P(X_{t_1} \leq c_1, \ldots, X_{t_i} \leq c_i) = P(X_{t_1+h} \leq c_1, \ldots, X_{t_i+h} \leq c_i),$$

for all $i = 1, 2, \ldots$, all time points $t_1, t_2, \ldots, t_i$, all time shifts $h = 0, \pm 1, \pm 2, \ldots$, and all constants $c_1, c_2, \ldots, c_i$ (Shumway and Stoffer, 2000).

This form of stationarity is often referred to as *strict stationarity*, as its requirements are often too strong for certain applications. In practice, a series is said to be stationary if and only if it is *weakly stationary*. A weakly stationary series has finite variance and satisfies the following two conditions:

1. The mean value of the time series, defined for probability mass function p_t by

   $$\mu_t = \mathbb{E}(X_t) = \sum_{t=-\infty}^{\infty} X_t p_t(X_t),$$

   is constant and does not depend on time t; and

2. the autocovariance function $\gamma(s, t)$, defined in Equation (2.3.1) below, depends only on the difference of s and t, $|s - t|$.

A strictly stationary time series is necessarily weakly stationary, however the converse is not true. Note that herein through this thesis, the use of the terms 'stationarity' and 'stationary time series' will be referring to weak stationarity and weakly stationary time series, respectively. A time series which is not stationary is referred to as 'non-stationary'.

As hinted by the definition of weak stationarity, the autocovariance function of a time series plays an important role in characterising its features. For two time-points s and t on a time series X (which is not necessarily stationary), the autocovariance function is denoted by $\gamma(s,t)$ and defined as

$$\gamma(s,t) = E[(X_s - \mu_s)(X_t - \mu_t)]. \tag{2.3.1}$$

If X is stationary, then the autocovariance function can be simplified to a function of the difference between the time-points of interest, h:

$$\gamma_h = \gamma(t, t+h) = E[(X_{t+h} - \mu)(X_t - \mu)],$$

where μ is the time-invariant mean of the process. This value quantifies the amount of dependence between two observations within a time series which are separated by h time-steps. Due to the second-moment nature of the definition of the autocovariance, the dependence of a time series is also referred to as the *second-order structure* of the series. For a stationary time series, the autocovariance is independent of the location of the time-points within the series.

Another measure of dependence which is closely related to the autocovariance function is the *autocorrelation function* (ACF) of a time series. For two observations which are h time-points apart, this is denoted by ρ_h for some lag h and defined as

$$\rho_h = \frac{\gamma(t, t+h)}{\sqrt{\gamma(t+h, t+h)\gamma(t,t)}} = \frac{\gamma_h}{\gamma_0}.$$

Hence, the ACF is essentially the normalised autocovariance function. Note that $-1 \leq \rho_h \leq 1$ for all h, with $\rho_h = 1$ and $\rho_h = -1$ implying perfect positive and negative autocorrelations, respectively, and $\rho_h = 0$ implying complete independence between observations.

Popular Time Series Models

There exist many models which have been proposed to characterise the dependence structure of such stationary series. Being able to quantify the dependence of a time series using such models is often of interest, since it allows the modeller to gain an understanding of how the observed values of a given process may fluctuate over some time window. Such stationary time series models can be broadly classified into three main categories:

- moving-average (MA) processes,
- autoregressive (AR) processes, and
- generalised autoregressive conditional heteroskedasticity (GARCH) processes.

A q^{th}-order moving average process is a process where each observation is a weighted aggregation of the q previous innovation terms, plus the innovation term for the current time-point. The t^{th} observation X_t of such a process is given by

$$X_t = \mu + \epsilon_t + \phi_1 \epsilon_{t-1} + \phi_2 \epsilon_{t-2} + \ldots + \phi_q \epsilon_{t-q}, \tag{2.3.2}$$

where $\epsilon_t \sim N(0, \sigma^2)$ for some σ^2 for all $1 \leq t \leq n$. Such a process is denoted by MA(q). The set of coefficients of the innovation terms $\{\phi_1, \ldots, \phi_q\}$ are referred to as the MA coefficients of the process, and the value of μ represents the mean value of the series. Typically, μ is assumed to be zero; if it is non-zero, it can be estimated via traditional methods and subtracted from the original series. Such moving average processes were first introduced by Yule (1909). Later, Yule (1927) also presented autoregressive processes.

In a similar manner to MA(q) processes, a zero-mean autoregressive process of order p is denoted AR(p), the t^{th} observation X_t of which is given by

$$X_t = \varphi_1 X_{t-1} + \varphi_2 X_{t-2} + \ldots + \varphi_p X_{t-p} + \epsilon_t, \tag{2.3.3}$$

where $\epsilon_t \sim N(0, \sigma^2)$ for some σ^2 for all $1 \leq t \leq n$. Here $\{\varphi_1, \ldots, \varphi_p\}$ is referred to as the set of AR coefficients of the process. Note that the value of X_t given by Equation (2.3.3) is now dependent upon the weighted values of the previous *observations* rather than the innovations.

Whittle (1951) combines the concepts of MA and AR processes to form ‘autoregressive moving average’ (ARMA) processes. These are denoted by ARMA(p, q), where p and q are the AR and MA orders, respectively. The t^{th} observation of such a process is given by

$$X_t = \varphi_1 X_{t-1} + \varphi_2 X_{t-2} + \ldots + \varphi_p X_{t-p} + \epsilon_t + \phi_1 \epsilon_{t-1} + \phi_2 \epsilon_{t-2} + \ldots + \phi_q \epsilon_{t-q}, \quad (2.3.4)$$

where $\epsilon_t \sim N(0, \sigma^2)$ for some σ^2 for all $1 \leq t \leq n$.

The stationary nature, or otherwise, of an AR process is determined solely by the value of the AR coefficients of the process. The relationship is defined through the equation

$$z^p - \varphi_1 z^{p-1} - \varphi_2 z^{p-2} - \ldots - \varphi_{p-1} z - \varphi_p = 0. \quad (2.3.5)$$

If the roots of Equation (2.3.5), denoted $z_1, \ldots, z_p$, each lie within the unit circle so that $|z_i| < 1$ for each $i = 1, \ldots, p$, then the given AR process is stationary. Note that an MA process is always stationary since it consists of the sum of stationary white noise terms (Cowpertwait and Metcalfe, 2009). Therefore, since an ARMA process is essentially the sum of an AR process and an MA process (which is always stationary), then an ARMA is also stationary whenever the AR part of the process is stationary, i.e. when the roots of (2.3.5) lie within the unit circle.

Example To gain an understanding of these processes, we consider the following ARMA$(2, 2)$ process:

$$X_t = 0.7X_{t-1} - 0.5X_{t-2} + \epsilon_t + 0.4\epsilon_{t-1} - 0.4\epsilon_{t-2}.$$

Figure 2.3.1 presents a series of 1000 observations from such a process.

ARMA processes themselves can be generalised to autoregressive integrated moving average processes, denoted by ARIMA(p, d, q). The key difference between ARIMA and ARMA processes is the d^{th} differences of an ARIMA process are modelled as an ARMA process. Hence, an ARIMA$(p, 0, q)$ process is equivalent to an ARMA(p, q) process. The term ‘integrated’ refers to this prior differencing, which is performed to

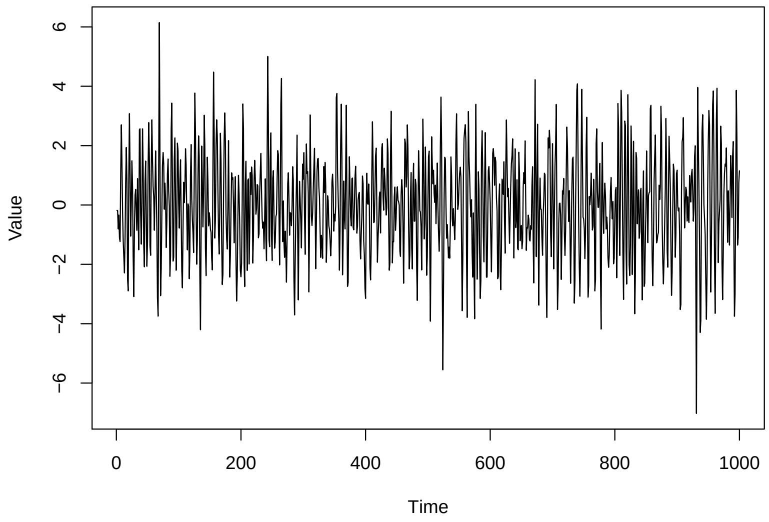


Figure 2.3.1: A series of 1000 observations of an ARMA$(2, 2)$ process.

ensure that the model is stationary.

Linear dependent time series models with independent Gaussian noise such as the models considered above are generalised by *discrete Gaussian processes* (dGP's). A dGP is a process for which every subset of observations is modelled as a multivariate Normal distribution. Hence, if a dGP has a parameter vector $\boldsymbol{\theta}$ and autocovariance matrix $\Gamma_{\boldsymbol{\theta}}$, then the likelihood of the process can be written as

$$L(\boldsymbol{\theta}|X) = \frac{1}{\sqrt{(2\pi)^n|\Gamma_{\boldsymbol{\theta}}|}} \exp\left(-\frac{1}{2}X^T\Gamma_{\boldsymbol{\theta}}^{-1}X\right),$$

where

$$\Gamma_{\boldsymbol{\theta}} = \begin{bmatrix} \gamma_{0,\boldsymbol{\theta}} & \gamma_{1,\boldsymbol{\theta}} & \cdots & \gamma_{n-1,\boldsymbol{\theta}} \\ \gamma_{1,\boldsymbol{\theta}} & \gamma_{0,\boldsymbol{\theta}} & \cdots & \gamma_{n-2,\boldsymbol{\theta}} \\ \gamma_{2,\boldsymbol{\theta}} & \gamma_{1,\boldsymbol{\theta}} & \ddots & \vdots \\ \vdots & & \ddots & \gamma_{1,\boldsymbol{\theta}} \\ \gamma_{n-1,\boldsymbol{\theta}} & \gamma_{n-2,\boldsymbol{\theta}} & \cdots & \gamma_{0,\boldsymbol{\theta}} \end{bmatrix}.$$

Another form of time series model popular within the literature is the *generalised* *autoregressive conditional heteroskedasticity* (GARCH) process, first introduced by Bollerslev (1986). The GARCH model differs from the models considered in this section in that the GARCH process models each observation as a non-linear function of previous observations. Such non-linear models are not of interest in this thesis, and hence these models will not be considered in any greater detail.

2.3.2 Spectral Density and the Periodogram

We now consider two quantities which are widely used within time series modelling and will prove to be useful in this thesis: the *spectral density* and *periodogram* of a time series.

Intuitively, the spectral density of a time series quantifies the amount of 'power' in the underlying signal at a given frequency. More formally, the spectral density $f(\omega)$ of a stationary process X is defined as the Fourier transform of the autocovariances of the process, $\{\gamma_h\}_{h\in(-\infty,\infty)}$:

$$d(\omega) = \sum_{h=-\infty}^{\infty} \gamma_h e^{i2\pi h\omega}, \quad \omega \in [0, 1], \tag{2.3.6}$$

where ω denotes a Fourier frequency (Shumway and Stoffer, 2000).

Hence, the autocovariances themselves have the following representation as the inverse Fourier transform of f:

$$\gamma_h = \int_0^1 f(\omega)e^{-i2\pi h\omega}\, d\omega, \quad h = 0, \pm1, \pm2, \ldots. \tag{2.3.7}$$

In practice, it can sometimes be difficult to obtain the exact value of the spectral density due to the sum over every single possible h. Fortunately, the spectral density can be approximated using a quantity known as the *periodogram* of the process (Shumway and Stoffer, 2000). The periodogram of a stationary process X is given by

$$I(\omega|X) = |b(\omega|X)|^2, \tag{2.3.8}$$

where

$$b(\omega|X) = n^{-1/2} \sum_{j=0}^{n-1} X_j e^{-i2\pi\omega j} \tag{2.3.9}$$

is the discrete Fourier transform of X. Hence,

$$\begin{aligned}
I(\omega) &= |b(\omega)|^2 \\
&= \left| n^{-1/2} \sum_{j=0}^{n-1} X_j e^{-i2\pi\omega j} \right|^2 \\
&= n^{-1} \left(\sum_{j=0}^{n-1} X_j e^{-i2\pi\omega j} \right) \left(\sum_{r=0}^{n-1} X_r e^{i2\pi\omega r} \right) \quad \text{since } z\overline{z} = |z|^2 \text{ and } \overline{e^{ix}} = e^{-ix} \\
&= n^{-1} \sum_{j=0}^{n-1} \sum_{r=0}^{n-1} X_j X_r e^{-i2\pi\omega j} e^{i2\pi\omega r} \\
&= n^{-1} \sum_{j=0}^{n-1} \sum_{r=0}^{n-1} X_j X_r e^{i2\pi\omega(r-j)}.
\end{aligned}$$

Therefore, the periodogram can be used as a data-based estimate of the spectral density. However, note that the periodogram is biased and unsmoothed, and so a bias correction and smoothing procedure should be applied if it is being used to directly estimate the spectrum (Cowpertwait and Metcalfe, 2009).

2.3.3 Changes in Dependence Structure

As discussed in Section 2.1, traditional univariate changepoint models typically assume that the observations of a time series occur independently over time. However, modern changepoint detection methodology has been providing more consideration to cases where there is dependence between observations. Indeed, changepoint methods have been developed which can not only incorporate dependence into the model, but are actively aiming to detect changes within the second-order structure of the series.

Popular approaches adopted by such methods include: (i) a time-domain treatment involving the traditional likelihood of the series, (ii) utilising an approximation to the traditional likelihood called ‘Whittle’s likelihood’ which allows for a frequencydomain analysis, or (iii) considering non-parametric statistics. We consider methods which employ each of these approaches in turn.

A likelihood-based approach is arguably the most common approach to detecting changes in second-order structure. Davis et al. (2006), Gombay (2008), Killick et al. (2010) and Fryzlewicz and Subba Rao (2014) all propose procedures based on calculating the traditional likelihood of dependent time series. The Auto-PARM method of Davis et al. (2006) uses the traditional likelihood-based minimum description length (MDL) of an AR(p) process as a penalised cost function, and use a genetic algorithm to estimate the number and locations of changes in the autoregressive structure. Similarly, Gombay (2008) considers the detection of changes in any combination of parameters of a p-order autoregressive process via a hypothesis testing procedure, where the test statistics are based on the likelihood of the process. Killick et al. (2013) also utilise the traditional likelihood, but instead model the observations as a Locally Stationary Wavelet (LSW) process (Nason et al., 2000), referring to this as the Wavelet Likelihood. They use this likelihood as a test statistic in a binary segmentation framework, and use a graphical data-driven method to determine the number of changepoints (rather than a specific penalty). Fryzlewicz and Subba Rao (2014) also use binary segmentation with a likelihood-based framework, but instead detect multiple changes in ARCH and GARCH processes.

Whittle's likelihood approximates the traditional likelihood in terms of the spectral density of the series. Therefore, this quantity has allowed for the detection of changes in the second-order structure of univariate time series. We provide an in-depth examination of Whittle's likelihood and its application to changepoint detection in Chapter 3. Notable works within the changepoint literature which employ Whittle's likelihood include those of Lavielle and Ludeña (2000), Hsu and Kuan (2001), Yamaguchi (2011) and Yau and Davis (2012).

Lavielle and Ludeña (2000) utilises Whittle's likelihood in a penalised cost function framework to detect changes in the spectral density of a time series. However, the penalty function assumed in their model is required to be linear in the number of changepoints. While this is theoretically interesting, this requirement does not allow for the usage of popular non-linear penalties such as the Minimum Description Length (see Chapter 3 for more details). Hsu and Kuan (2001), Yamaguchi (2011) and Yau and Davis (2012) all consider the context of changes where long-memory may be present. In the case of Hsu and Kuan and Yau and Davis, interest lies in distinguishing whether a given series follows a long-memory model or whether it is a short-memory process with an abrupt change in the dependence structure. The problem considered by Yamaguchi (2011) is the estimation of a changepoint in the long-memory parameter of an Autoregressive Fractionally Integrated Moving Average (ARFIMA) process. Such a process is a generalisation of an ARMA process which allows for fractional differencing, see Hosking (1981) for more details. In each case, Whittle's likelihood approximation is used to evaluate the suitability of a given model.

Giraitis et al. (1996), Ombao et al. (2001) and Cho and Fryzlewicz (2012) detect second-order changepoints using non-parametric approaches. Giraitis et al. (1996) use Kolmogorov-Smirnov-type statistics to test for changes in the distribution of dependent data. Ombao et al. (2001) propose a new set of bases which can be used to decompose a time series, with this decomposition then being used in a non-parametric test statistic to detect second-order changes. However, this suffers from its requirement that changes must occur at dyadic time-points. In a similar manner to Killick et al. (2013), Cho and Fryzlewicz (2012) model observations using the Locally Stationary Wavelet framework, but instead search for changes in the mean of the wavelet coefficients using a non-parametric test statistic in a binary segmentation procedure. These changes in mean in the wavelet coefficients correspond to changes in the second-order structure of the original series.

Killick et al. (2013) demonstrate that their approach (termed 'WL') out-performs the method of Cho and Fryzlewicz (2012) in terms of quality of solutions. This may be due to the assumption made by Cho and Fryzlewicz that the variance of the summary statistic is constant across different segments, which can be difficult to establish in practice. They also show that while the Auto-PARM method of Davis et al. (2006) estimates the correct number of changepoints more often than WL, the changepoint locations estimated by WL are more accurate than those estimated by Auto-PARM.

As has been seen, there exists a range of methods for detecting second-order changes. While there are a number of methods which utilise Whittle's likelihood, the majority of these consider long-memory models, and the method available for short-memory models is impractical. Therefore, in Chapter 3 we propose methodology which employs Whittle's likelihood to detect changes in short-memory time series models which can be easily implemented in practice. This practicality is demonstrated through application to a substantive dataset arising from acoustic sensing observations.

Chapter 3

Detecting Changes in Second-Order Structure: An Application to Acoustic Sensing Data

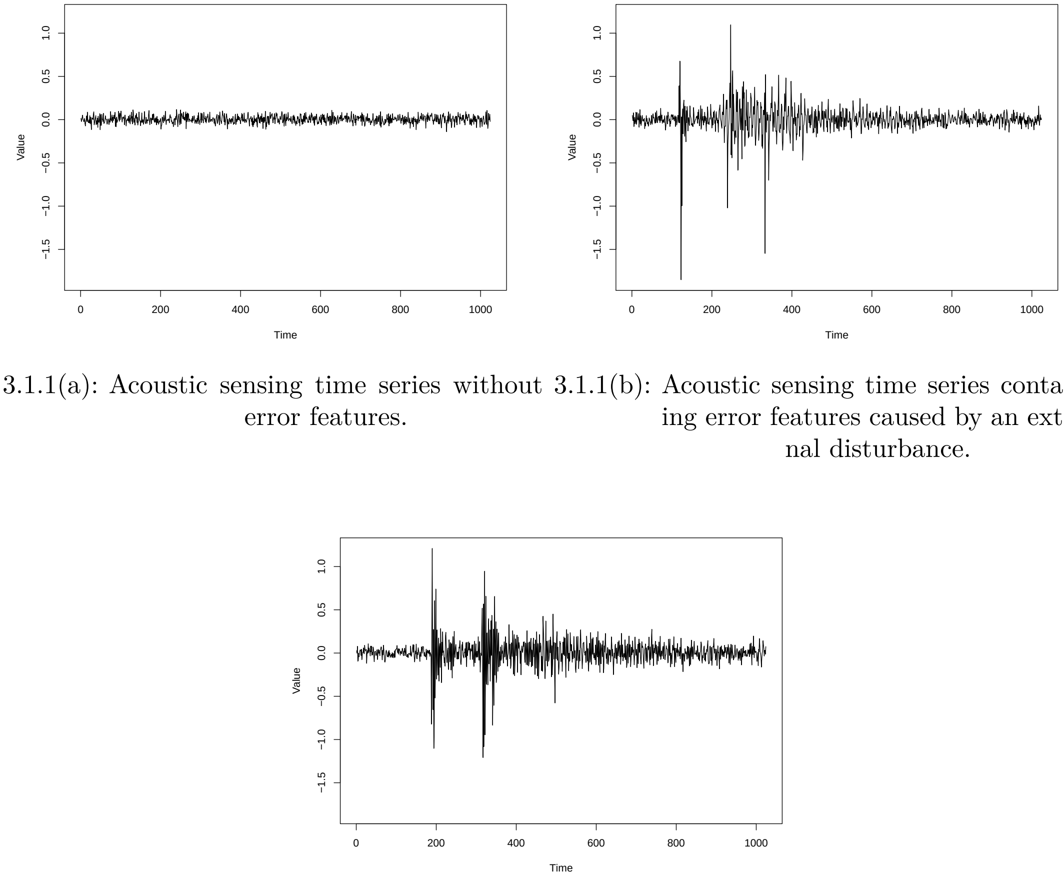
3.1 Introduction to Acoustic Sensing Data

In the previous chapter, we highlighted the development of various approaches to detecting changes within piecewise second-order stationary time series. Simulation studies reported by Davis et al. (2006), Cho and Fryzlewicz (2012) and Killick et al. (2013) have shown that many of these approaches have broadly good performance across a wide range of different scenarios. However, it is well-known that several of these methods are also computationally intensive. Consider, for example, the wavelet-likelihood approach of Killick et al. (2013) which (as we shall see later) is $\mathcal{O}\Big(n^4(\log n)^2\Big)$. Such significant computation can prove prohibitive for even moderately long time series or applications where many time series need to be processed on a regular basis. Acoustic sensing signals, such as those becoming commonly obtained in the oil and gas industry, provide an example of such an application. Within this chapter we therefore seek to explore which approach, of the various available in the literature, provides the best combination of changepoint detection accuracy and speed, and investigate the potential for their application to acoustic sensing data.

Acoustic sensing is the practice of measuring and quantifying the vibrations which are travelling through some medium, typically the ground. Within oil exploration and production, such vibrations are measured by lining the well with a fibre-optic cable. When vibrations occur in the medium they pass through the fibre-optic cable, inducing a change in the intensity of the reflection of the pulses of light being passed through the cable. These pulses of light are produced at a very high rate, often as high as 10kHz, allowing for the ‘real-time’ monitoring of these vibrations to identify features of interest in the well (e.g. the composition of the oil and gas, or areas where the gradient of the piping changes), or mapping of the geology of the local environment. The characteristics of such vibrations means that the observations are generally dependent in time. For further discussion of acoustic sensing in the oil and gas industry see, for example, Van der Horst et al. (2014) and Silkina (2014).

In addition to physical features being visible within these vibration measurements, there occasionally exists error features within the series. Such errors may be due to an external disturbance of the fibre-optic cable or some other (unknown) factor. We are advised by engineers that such error features manifest as sudden changes in the second-order structure of the time series. Typically, error features induced by these disturbances occur at all observed locations of the well. The magnitude of the disturbances relative to the true features is such that it is only necessary for a single channel to be analysed in order to detect the disturbance. Figure 3.1.1 presents three examples of acoustic sensing time series from one particular type of well. Figure 3.1.1(a) shows data without any error effects, as demonstrated by the visibly stationary nature of the series. Conversely, Figures 3.1.1(b) and 3.1.1(c) both demonstrate instances of disturbance, which are clearly illustrated by the abrupt increases in vibration, followed by a period of increased activity, before returning to a low level of vibrations.

The aim of this chapter is to introduce changepoint detection methodology that



3.1.1(a): Acoustic sensing time series without error features.

3.1.1(b): Acoustic sensing time series containing error features caused by an external disturbance.

3.1.1(c): Acoustic sensing time series containing error features caused by an external disturbance.

Figure 3.1.1: Three examples of acoustic sensing time series obtained from one particular type of well. The error features are present in the second and third series.

is capable of identifying second-order changes, such as the error features described, through the utilisation of Whittle's likelihood approximation. This is a popular tool for analysing time series in the stationary context. We demonstrate that our method is pragmatically appropriate and draw comparisons with other leading second-order changepoint methods. The presented methodology is applied to substantive acoustic sensing data where it is shown that the locations of detected changepoints correspond with occurrences of error features.

The remainder of this chapter is structured as follows: Section 3.2 introduces

Whittle's likelihood approximation and examines how it can be used in a penalised likelihood framework for detecting changes in second-order structure. Section 3.3 compares the performance of a second-order changepoint detection method using Whittle's (approximate) likelihood against different approaches which use exact likelihood based formulations. A selection of acoustic sensing data is analysed using the proposed Whittle likelihood based method in Section 3.4, and concluding remarks are presented in Section 3.5.

3.2 Whittle's Likelihood and its Application to Changepoints

The changepoint detection methodology proposed in this chapter is based on a quantity known as *Whittle's likelihood.* Within this section we introduce Whittle's likelihood, observe how it is related to the traditional likelihood, and show how it can be utilised for the purposes of changepoint detection. Our explanation of Whittle's likelihood given below generally follows those of Hurvich (2002) and Gray (2005).

3.2.1 Whittle's Likelihood Approximation

Suppose that we observe a sequence of univariate observations $X_{1:n} = \{X_t\}_{t=1}^{n}$. This series is assumed to follow a discrete Gaussian Process (dGP) which is zero-mean and second-order stationary, and has a set of unknown model parameters denoted by $\boldsymbol{\theta}$ with an associated set of autocovariances $\boldsymbol{\gamma}_{\boldsymbol{\theta}} = \{\gamma_{h,\boldsymbol{\theta}}\}_{h=0,\dots,p}$. Traditionally, obtaining the best-fitting set of model parameters is performed through maximum likelihood estimation (see, for example, Section 2.2 of Shumway and Stoffer (2000); Chapter 7 of Box et al. (2011)).

Given the time-dependent (i.e. non-i.i.d.) nature of the observations $X_{1:n}$, the joint density of the observations is determined directly through the autocovariances of the process. Using the fact that every subset of dGP observations are multivariate

Normally distributed, the likelihood of the series is given by

$$L(\boldsymbol{\theta}|X_{1:n}) \equiv L(\boldsymbol{\gamma_\theta}|X_{1:n}) = \frac{1}{\sqrt{(2\pi)^n|\Gamma_{\boldsymbol{\theta}}|}} \exp\left(-\frac{1}{2}X_{1:n}^T\Gamma_{\boldsymbol{\theta}}^{-1}X_{1:n}\right), \tag{3.2.1}$$

where $\Gamma_{\boldsymbol{\theta}}$ is the autocovariance matrix of the process.

As discussed in Section 2.3.2, the autocovariances can be expressed as the inverse discrete Fourier transform of the spectral density:

$$\gamma_{h,\boldsymbol{\theta}} = \sum_{j=0}^{n-1} d_{\boldsymbol{\theta}}(\omega_j)e^{-i2\pi h\omega_j}, \quad h = 0, 1, 2, \ldots. \tag{3.2.2}$$

Therefore, expressing the likelihood directly in terms of the autocovariances allows it to be easily evaluated under any time series model with a known spectrum (as long as the model lies in the class of discrete Gaussian Processes). This is advantageous as it provides a more holistic measure of fit by assessing the fit of the given model spectrum to the periodogram of the data, rather than assessing the fit of the parameters to the individual points, as is the case in traditional time-domain maximum likelihood estimation. Using (3.2.2), the autocovariance matrix $\Gamma_{\boldsymbol{\theta}}$ can be rewritten in terms of the spectral density $d_{\boldsymbol{\theta}}$. To emphasise the dependence of this covariance matrix on the spectral density, rather than $\boldsymbol{\theta}$ directly, we use $\Gamma_{d_{\boldsymbol{\theta}}}$ to denote this re-expressed matrix. This leads to the expression of the likelihood of $X_{1:n}$ in terms of the spectral density:

$$L(d_{\boldsymbol{\theta}}|X_{1:n}) = \frac{1}{\sqrt{(2\pi)^n|\Gamma_{d_{\boldsymbol{\theta}}}|}} \exp\left(-\frac{1}{2}X_{1:n}^T\Gamma_{d_{\boldsymbol{\theta}}}^{-1}X_{1:n}\right), \tag{3.2.3}$$

with the negative log-likelihood given by

$$-\log(L(d_{\boldsymbol{\theta}}|X_{1:n})) = \frac{n}{2}\log(2\pi) + \frac{1}{2}\log|\Gamma_{d,\boldsymbol{\theta}}| + X_{1:n}^T\Gamma_{d,\boldsymbol{\theta}}^{-1}X_{1:n}. \tag{3.2.4}$$

The best-fitting spectrum $\hat{d}_{\boldsymbol{\theta}}$ for the process $X_{1:n}$ can now be found by maximising (3.2.3) (or equivalently minimising (3.2.4)) over all $d_{\boldsymbol{\theta}} \in \mathcal{F}$, where $\mathcal{F}$ is the set of all possible spectrums.

Using traditional matrix methods, the inversion of $\Gamma_{d_{\boldsymbol{\theta}}}$ can be computed using $O(n^3)$ operations. For increasingly large n, the calculation of these operations can become prohibitively expensive. Therefore, it is often preferable to consider a quantity which is an approximately equivalent to the exact likelihood but has a reduced computation time. Whittle's likelihood approximation (Whittle, 1951) represents such a quantity.

Definition 3.2.1. *For a given time series $X_{1:n}$ and set of model parameters $\boldsymbol{\theta}$ with corresponding spectral density $d_{\boldsymbol{\theta}}$, Whittle's likelihood approximation of the negative log-likelihood (3.2.4) is denoted by $W(d_{\boldsymbol{\theta}}|X_{1:n})$ and defined as*

$$W(d_{\boldsymbol{\theta}}|X_{1:n}) := \frac{n}{2}\log(2\pi) + \frac{1}{2}\sum_{j=0}^{n-1}\left(\log(d_{\boldsymbol{\theta}}(\omega_j)) + \frac{I(\omega_j|X_{1:n})}{d_{\boldsymbol{\theta}}(\omega_j)}\right). \tag{3.2.5}$$

The $I(\cdot)$ term denotes the periodogram of the series and $\{\omega_j = j/n\}_{j=1,\dots,n}$ denotes the discrete Fourier frequencies.

Arguably, the greatest computational benefit which arises from approximating with the Whittle likelihood is that it does not require the inversion of the covariance matrix, but instead requires the calculation of the periodogram of the data. This can be calculated in $\mathcal{O}(n\log n)$ time through the use of the Fast Fourier Transform. Hence, the use of Whittle's likelihood in place of the exact likelihood may be more appealing in scenarios where the number of data points can increase rapidly. However, this comes at the expense of being an approximation to the likelihood, rather than the exact value.

In a similar manner to the exact negative log-likelihood shown in (3.2.4), Whittle's likelihood can be minimised over all possible $d_{\boldsymbol{\theta}} \in \mathcal{F}$ to obtain the best-fitting model spectrum for $W(\cdot|X_{1:n})$, denoted $\hat{d}_{\boldsymbol{\theta},W}$. Choudhuri et al. (2004) show that any estimator based on Whittle's likelihood has the same consistency and rate of convergence as the equivalent estimator based on the exact likelihood. Since it is well-known that the MLE under the exact likelihood is consistent and has a rate of convergence of $\mathcal{O}(n^{-1/2})$ (Wald, 1949), the Whittle MLE is therefore also consistent and has a rate of convergence of $\mathcal{O}(n^{-1/2})$. Hence, its use in the maximum likelihood setting is theoretically justified.

Due to its reduction in computational complexity over the traditional calculation of the likelihood, we wish to utilise Whittle's likelihood for the detection of changes in second-order structure, and consequently the detection of error effects in acoustic sensing data. In the next section we will consider a framework for detecting changes in second-order structure, leading to our proposed method which uses Whittle's likelihood as part of such a detection procedure.

3.2.2 Detecting Changes in Second-Order Structure using Whittle's Likelihood

In the previous section we described how Whittle's likelihood can be used to approximate the negative log-likelihood of a discrete Gaussian process with a reduced computation time. Our aim in this section is to explore how Whittle's likelihood can be utilised within a penalised cost function approach to detect changes in second-order structure. As such, this work is similar in spirit to that of Lavielle and Ludeña (2000). However, our approach differs in that it can be used with any penalty function that is non-linear in the number of changepoints m. The method of Lavielle and Ludeña (2000) requires the penalty to be linear in m.

We introduce our approach below prior to comparing it against an exact likelihood equivalent and the contemporary methods of Davis et al. (2006) and Killick et al. (2013) in Section 3.3. As discussed in Section 2.3.3, these methods represent the forefront of second-order changepoint detection. This comparison is done through a simulation study and application to an acoustic sensing dataset to identify the various benefits and side-effects which occur from using this approximation approach.

We tackle the problem of detecting second-order changes in a time series $\{X_1, X_2, \ldots, X_n\}$ using a model selection framework. The aim is to select the best-fitting m changepoints $\boldsymbol{\tau} = (\tau_1, \tau_2, \ldots, \tau_m)$ such that the spectral density of the time series d is given by

$$d = d_{\boldsymbol{\theta}_k} \quad \text{for} \quad \tau_{k-1} + 1 \leq X_t \leq \tau_k, \tag{3.2.6}$$

where $k = 1, \dots m+1$, $\tau_0 = 0$, $\tau_{m+1} = n$, $d_{\boldsymbol{\theta}_k} \neq d_{\boldsymbol{\theta}_{k+1}}$ and m is unknown. Here $d_{\boldsymbol{\theta}_k}$ represents the best-fitting spectral density for the k^{th} segment, where $\boldsymbol{\theta}_k$ is the corresponding set of parameter values.

To obtain these best-fitting values of m, $\boldsymbol{\tau}$ and $d_{\boldsymbol{\theta}_1}, \dots, d_{\boldsymbol{\theta}_{m+1}}$, we consider the minimisation of the following penalised cost function:

$$\sum_{k=1}^{m+1} W(d_{\boldsymbol{\theta}_k} | X_{(\tau_{k-1}+1):\tau_k}) + \beta f(m), \tag{3.2.7}$$

where $W(\cdot|\cdot)$ is Whittle's likelihood as described in equation (3.2.5), β is a constant and $f(m)$ is the penalty function. The only restriction on this function is that it is a concave function of m. There is no requirement that it is linear in m, and so any non-linear concave function can be used. In particular, this allows for the use of popular non-linear penalties such as the minimum description length (Rissanen, 1989) and Lebarbier's penalty (Lebarbier, 2005).

The adoption of Whittle's likelihood approximation in the penalised cost function (3.2.7) means that this approach reaps the benefits over using traditional likelihood. In particular, its ability to be calculated in $\mathcal{O}(n \log n)$ time, compared to the $\mathcal{O}(n^3)$ time required for the traditional likelihood.

Due to the exact nature of its search, we wish to use a dynamic programming procedure to minimise the penalised cost function (3.2.7). However, because of the potentially non-linear nature of the penalty function $f(m)$, the now well-established optimal partitioning or PELT methods cannot be used due to their reliance on the linearity of the penalty function. Equally, we do not wish to use segment neighbourhood search due to its requirement of specifying a maximum number of changepoints. Instead we use a modified version of PELT described by Killick et al. (2012, Section 4.3.1) called *iterative PELT* which can accommodate concave penalty functions.

Iterative PELT works by iteratively performing PELT with a different number of assumed changepoints for each run. Once the number of changepoints output by PELT matches the value of m used in the penalty then the algorithm terminates. An important consequence of this modification is that the algorithm can no longer guarantee to produce the optimal solution. However, the resulting configuration of changepoints still represents a very high quality solution which is obtained without specification of the bounds on the number of changepoints.

For comparison purposes, we note that a penalised cost function that is equivalent to (3.2.7) can be formulated using the exact likelihood:

$$\sum_{k=1}^{m+1} \left[-\log(L(d_{\boldsymbol{\theta}_k} | X_{(\tau_{k-1}+1):\tau_k})) \right] + \beta f(m). \tag{3.2.8}$$

This can also be minimised in the same manner using iterative PELT. We refer to these two approaches as WHIP (Whittle Iterative PELT) and EXIP (Exact Iterative PELT) respectively. We illustrate both algorithms in Algorithm 2.

Algorithm 2: WHIP / EXIP

```
1. Fix the number of changepoints as $m_0$.
2. For WHIP  Use PELT to obtain the following minimum:
       $\min_{m,\boldsymbol{\tau},d_{\boldsymbol{\theta}_1},\ldots,d_{\boldsymbol{\theta}_{m+1}}} \left\{ \sum_{k=1}^{m+1} W(d_{\boldsymbol{\theta}_k} | X_{(\tau_{k-1}+1):\tau_k}) + \beta f(m_0) \right\}$
   For EXIP  Use PELT to obtain the following minimum:
       $\min_{m,\boldsymbol{\tau},d_{\boldsymbol{\theta}_1},\ldots,d_{\boldsymbol{\theta}_{m+1}}} \left\{ \sum_{k=1}^{m+1} \left[ -\log(L(d_{\boldsymbol{\theta}_k} | X_{(\tau_{k-1}+1):\tau_k})) \right] + \beta f(m_0) \right\}$
3.   • If PELT outputs $m = m_0$ changepoints, stop and output the changepoint
       locations $\hat{\boldsymbol{\tau}}_{1:m_0}$.
     • Else, set $m_0 = m$ and repeat from step 1.
```

To investigate the performance of WHIP and assess the level of approximation made by Whittle's likelihood, we compare WHIP with EXIP and two approaches representing the current state of the art in second-order changepoint detection: the Wavelet Likelihood (WL) method of Killick et al. (2013) and the Auto-PARM (AP) method of Davis et al. (2006). These two methods are described in Section 2.3.3. Comparisons are made through a simulation competition and a study of the methods' theoretical computational complexities.

3.3 Comparison of Second-Order Changepoint Methods

We compare the four methods of WHIP, EXIP, WL and AP on the accuracy of their predicted changepoints, in terms of the number of estimated changepoints and their predicted locations. The details of this comparison are given in Section 3.3.1. A comparison of the computational complexity of the four methods is detailed in Section 3.3.2. Comparing the methods on these two aspects allows for a holistic understanding of the effectiveness of WHIP, and its relative performance against the cutting-edge of second-order changepoint detection methods.

3.3.1 Accuracy of Estimation

To assess the accuracy of estimation of the WHIP, EXIP, WL and AP methods, their performance in a range of scenarios is investigated. Simulations from thirteen different models, of which all but one were considered by Killick et al. (2013), are used to assess the quality of the changepoints estimated by the methods in terms of both the number of changepoints detected and their locations. The benefits of using both of these measures as criteria for assessing the quality of a changepoint detection method are noted by both Killick et al. (2013) and Eckley et al. (2011).

We note that the WL method uses the exact likelihood formulated using wavelet coefficients, and AP uses an approximation to the likelihood for AR processes based on the Yule-Walker estimate for the variance of the innovations (Davis et al., 2006).

The details of the models considered are discussed in Section 3.3.1 and the results obtained are summarised and discussed in Section 3.3.1.

Model Details

The models considered include autoregressive (AR) processes and moving-average (MA) processes, which represent a range of different types of behaviour which occur in time series models. All of the models examined are second-order stationary within their segments, and assume a white noise term $\epsilon_t \sim N(0, 1)$ for all t, unless otherwise stated. For each model, 100 replications are considered. Full details of the models are outlined below.

Models 1–6: AR(1) processes containing no changepoints Each of these models are discrete Gaussian processes $\{X_t\}_{t=1}^n$ of the form

$$X_t = aX_{t-1} + \epsilon_t \quad \text{for} \quad 1 \leq t \leq 1024, \tag{3.3.1}$$

where the value of the AR(1) coefficient a is equal to one of $(-0.7, -0.4, -0.1, 0.1, 0.4,$ $0.7)$, depending on the model being considered. This range of values is considered to investigate the false-positive rate of the algorithm across a range of different types of autocorrelation.

Model 7: Piecewise AR process with two clearly observable changes The data for this model are simulated from

$$X_t = \begin{cases} 0.9X_{t-1} + \epsilon_t & \text{if } 1 \leq t \leq 512, \\ 1.68X_{t-1} - 0.81X_{t-2} + \epsilon_t & \text{if } 513 \leq t \leq 768, \\ 1.32X_{t-1} - 0.81X_{t-2} + \epsilon_t & \text{if } 769 \leq t \leq 1024. \end{cases} \tag{3.3.2}$$

In this case, the AR coefficients are relatively large in magnitude whilst keeping the model stationary within each segment.

Model 8: Piecewise AR process with two less clearly observable changes The data for this model are simulated from

$$X_t = \begin{cases} 0.4X_{t-1} + \epsilon_t & \text{if } 1 \leq t \leq 400, \\ -0.6X_{t-1} + \epsilon_t & \text{if } 401 \leq t \leq 612, \\ 0.5X_{t-1} + \epsilon_t & \text{if } 613 \leq t \leq 1024. \end{cases} \tag{3.3.3}$$

The magnitude of the AR coefficients are smaller compared to those in Model 7, and the changepoint locations are no longer at dyadic time points.

Model 9: Piecewise AR process with one change, one short segment The data for this model are simulated from

$$X_t = \begin{cases} 0.75X_{t-1} + \epsilon_t & \text{if } 1 \leq t \leq 50, \\ -0.5X_{t-1} + \epsilon_t & \text{if } 51 \leq t \leq 1024. \end{cases} \tag{3.3.4}$$

In this model, the single changepoint occurs after a relatively short period of time, leading to a short initial segment followed by a longer segment.

Model 10: Piecewise AR process with two changes and high autocorrelation The data for this model are simulated from

$$X_t = \begin{cases} 1.399X_{t-1} - 0.4X_{t-2} + \epsilon_t, & \epsilon_t \sim N(0, 0.8^2) & \text{if } 1 \leq t \leq 400, \\ 0.999X_{t-1} + \epsilon_t, & \epsilon_t \sim N(0, 1.2^2) & \text{if } 401 \leq t \leq 750, \\ 0.699X_{t-1} + 0.3X_{t-2} + \epsilon_t, & \epsilon_t \sim N(0, 1) & \text{if } 751 \leq t \leq 1024. \end{cases} \tag{3.3.5}$$

The magnitude of the AR coefficients in this model are very large, with each of the segments being only on the verge of stationarity. Note that the variance of the white noise term is also changing between the segments in this example, a feature not replicated in any of the other models.

Model 11: Piecewise ARMA(1,1) process with three changes The data for this model are simulated from

$$X_t = \begin{cases} 0.7X_{t-1} + \epsilon_t + 0.6\epsilon_{t-1} & \text{if } 1 \leq t \leq 125, \\ 0.3X_{t-1} + \epsilon_t + 0.3\epsilon_{t-1} & \text{if } 126 \leq t \leq 352, \\ 0.9X_{t-1} + \epsilon_t & \text{if } 353 \leq t \leq 704, \\ 0.1X_{t-1} + \epsilon_t - 0.5\epsilon_{t-1} & \text{if } 705 \leq t \leq 1024. \end{cases} \tag{3.3.6}$$

This is the only model considered which contains both autoregressive and moving-average terms. Such a feature is of interest since the AP method is designed only for fitting AR models, not MA. It also contains the most changepoints of all the models considered.

Model 12: Piecewise MA process with a clearly observable change The data for this model are simulated from

$$X_t = \begin{cases} \epsilon_t + 0.8\epsilon_{t-1} & \text{if } 1 \leq t \leq 128, \\ \epsilon_t + 1.68\epsilon_{t-1} - 0.81\epsilon_{t-2} & \text{if } 129 \leq t \leq 256. \end{cases} \tag{3.3.7}$$

Similarly, this model is the only one which contains exclusively moving-average terms. As with Model 11, this is of interest due to AP being constructed for AR-only models. The time series in this case are also shorter in length compared to the previous models.

Model 13: Piecewise MA process with a less clearly observable change The data for this model are simulated from

$$X_t = \begin{cases} \epsilon_t + 0.1\epsilon_{t-1} - 0.2\epsilon_{t-2} & \text{if } 1 \leq t \leq 180 \\ \epsilon_t - 0.7\epsilon_{t-1} - 0.2\epsilon_{t-2} & \text{if } 181 \leq t \leq 256. \end{cases} \tag{3.3.8}$$

This model also contains exclusively moving-average terms, but the change is now only in a single coefficient and is smaller in magnitude compared to Model 12. This model is also of a shorter length compared to Models 1–11.

For each of the models considered, the WHIP and EXIP methods are applied as described in Algorithm 2, with the possible spectral densities considered limited to those of ARMA processes. The spectral density of an ARMA(p, q) process with AR coefficients $\varphi_1, \dots, \varphi_p$, MA coefficients $\phi_1, \dots, \phi_q$ and innovation variance σ^2 is given by

$$d(\omega) = \sigma^2 \left| \frac{1 + \sum_{k=1}^{q} \phi_k \exp(-2\pi i \omega k)}{1 - \sum_{j=1}^{p} \varphi_j \exp(-2\pi i \omega j)} \right|^2 . \tag{3.3.9}$$

The penalty function $f(m)$ is set such that the penalised cost functions are equivalent to Minimum Description Length (MDL) of the data (see Rissanen (1989) for full details). For a piecewise ARMA process containing m changepoints with orders (p_k, q_k) and length n_k for the k^{th} segment, this penalty function is given by

$$h(m) = \log(m) + (m+1)\log(n) + \sum_{k=1}^{m+1} \left[\log(p_k) + \log(q_k) + \frac{p_k + q_k + 1}{2} \log n_k \right] . \tag{3.3.10}$$

Similarly, the AP method uses the MDL as its penalised cost function, except the authors only consider AR models (with no MA component). This choice of penalty function for WHIP and EXIP is motivated by the positive results shown from its use in Davis et al. (2006).

In addition, some realistic constraints are incorporated into WHIP and EXIP to aide computation time.

1. Maximum values are imposed for the autoregressive order p_j and moving-average order q_j for each segment. For all thirteen models the maximum AR order is set to 5. For Models 1–10 which are known to contain only autoregressive terms, the maximum MA order is fixed as 0. For Models 11, 12 and 13, which contain MA terms, the maximum MA order is set to 3.

2. The minimum distance between any two changepoints is fixed at 20 time points.

3. For the iterative PELT algorithm, the maximum number of iterations for the algorithm is set to 10. If the algorithm has not converged by this stage, then

the penalised cost for the best segmentation of the data at each iteration is calculated and the segmentation corresponding to the lowest of these is output.

Note that throughout all of these simulations, each run of WHIP and EXIP converged before reaching the maximum number of iterations. Also note that code for the WL algorithm is not available, and so no results for WL are provided for Model 13.

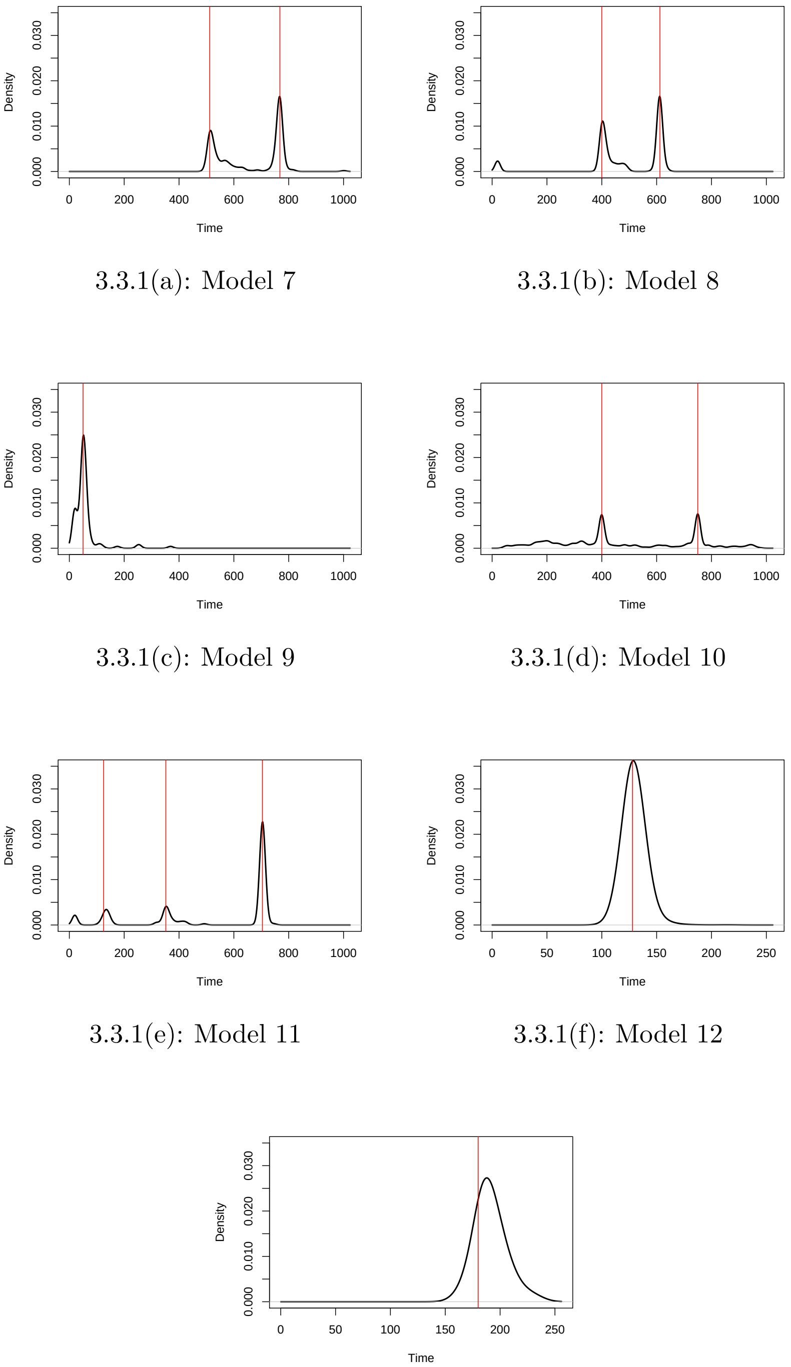
Results and Discussion

Tables 3.3.1, 3.3.2 and 3.3.3 present the estimated number of changepoints across all replications for each model. The true number of changepoints for a given model are highlighted in bold. The densities of the locations of detected changepoints found by WHIP are shown in Figures 3.3.1(a) – 3.3.1(g) for models where there is at least one true changepoint (i.e. models 7–13).

	Model 1 a = -0.7				Model 2 a = -0.4			
Nº cpts	WHIP	EXIP	WL	AP	WHIP	EXIP	WL	AP
0	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**
1	0	0	0	0	0	0	0	0
≥2	0	0	0	0	0	0	0	0
	Model 3 a = -0.1				Model 4 a = 0.1			
Nº cpts	WHIP	EXIP	WL	AP	WHIP	EXIP	WL	AP
0	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**
1	0	0	0	0	0	0	0	0
≥2	0	0	0	0	0	0	0	0
	Model 5 a = 0.4				Model 6 a = 0.7			
Nº cpts	WHIP	EXIP	WL	AP	WHIP	EXIP	WL	AP
0	**100**	**100**	**100**	**100**	**99**	**100**	**91**	**100**
1	0	0	0	0	1	0	9	0
≥2	0	0	0	0	0	0	0	0

Table 3.3.1: Percentage of repetitions which identified a certain number of changepoints for Models 1–6. True number of changepoints for each model shown in bold.

Table 3.3.1 contains the results for Models 1–6, which each contain no changepoints



3.3.1(a): Model 7

3.3.1(b): Model 8

3.3.1(c): Model 9

3.3.1(d): Model 10

3.3.1(e): Model 11

3.3.1(f): Model 12

3.3.1(g): Model 13

Figure 3.3.1: Plots showing the densities of changepoint locations detected by WHIP for Models 7–13. True changepoint locations are shown by red vertical lines.

	Model 7				Model 8			
N^{o} cpts	WHIP	EXIP	WL	AP	WHIP	EXIP	WL	AP
0	0	0	0	0	0	0	0	0
1	1	0	0	0	11	8	0	0
2	**86**	**99**	**98**	**94**	**89**	**92**	**94**	**100**
3	12	1	2	6	0	0	6	0
4	1	0	0	0	0	0	0	0
$\geq$5	0	0	0	0	0	0	0	0

	Model 9				
N^{o} cpts	WHIP	EXIP	WL	AP	
0	2	0	4	0	
1	**98**	**100**	**94**	**100**	
2	0	0	2	0	
3	0	0	0	0	
4	0	0	0	0	
$\geq$5	0	0	0	0	

Table 3.3.2: Percentage of repetitions which identified a certain number of changepoints for Models 7–9. True number of changepoints for each model shown in bold.

and can hence be viewed as assessments for the false-positive case. It can be seen that WHIP has at most a 1% false-positive rate, and achieves a 0% false-positive rate for half of these models. These results are on par with the performances of EXIP and AP, which each have a 0% false-positive rate in each of the models. WL performs slightly worse, giving a 9% false-positive rate when there is reasonably large positive autocorrelation.

The results for Models 7–9 and 10–13, presented in Tables 3.3.2 and 3.3.3 respectively, show that the performances of WHIP, EXIP, AP and WL are generally comparable for most cases. For each of these models, the percentages of cases where each method identified the correct number of changepoints are within at least 13% of the equivalent percentages for the other methods. Interestingly, for Model 9 (where there is a short segment) and Model 12 (where there is a clearly observable change in an MA process) WHIP out-performs WL. This result occurs even though WHIP uses an approximate cost function whereas the WL method uses an exact formulation of the likelihood. Clearly the binary segmentation search method of WL contributes to

	Model 10				Model 11			
N^o cpts	WHIP	EXIP	WL	AP	WHIP	EXIP	WL	AP
0	0	0	0	0	0	0	0	0
1	0	10	26	9	68	55	20	51
2	**8**	**72**	**45**	**33**	18	23	22	33
3	28	16	26	32	**13**	**22**	**35**	**16**
4	32	2	3	15	1	0	22	0
5	32	0	0	12	0	0	1	0

	Model 12				Model 13		
N^o cpts	WHIP	EXIP	WL	AP	WHIP	EXIP	AP
0	0	0	0	0	69	68	66
1	**100**	**100**	**99**	**100**	**31**	**32**	**34**
2	0	0	1	0	0	0	0
3	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0

Table 3.3.3: Percentage of repetitions which identified a certain number of changepoints for models 10–13. True number of changepoints for each model shown in bold.

this reduction in quality.

Models 10 and 11 cause substantial difficulty for WHIP in detecting the correct number of changepoints compared to the other methods. This suggests that in situations where the autocorrelation in the series is very high (as in Model 10) or the change in dependence is small and at a higher order (as in Model 11), then the quality of the approximation made by Whittle's likelihood is reduced, thereby making it more difficult for WHIP to accurately detect changes. High autocorrelation causes the series to appear non-stationary in certain areas (since the series is only on the edge of stationarity), which then causes WHIP to attempt to induce stationarity in the data by segmenting it into more stationary segments. This leads to an overestimation of the number of changepoints, demonstrated in Table 3.3.3 for Model 10.

WL and AP are also affected in a similar manner. EXIP, on the other hand, does not suffer as much from this drawback since it directly calculates the exact likelihood of the process (based on the autocovariances). Hence, no approximations are made and there are relatively fewer parameters to estimate (ARMA model parameters in

EXIP versus the multiple wavelet coefficients of WL).

Conversely, smaller changes in dependence at higher orders (as in Model 11), are more difficult to detect using parametric methods such as WHIP, EXIP and AP. In these cases such methods are more likely to give an underestimation of changepoints, as observed for Model 11. Non- or semi-parametric methods such as WL are less affected since they do not assume a parametric form of the data, and consider a wide range of frequencies or scales instead of a small number of parameters.

Across all models, the results of WHIP at best match those of EXIP and AP. This is to be expected for EXIP, since EXIP is precisely the same algorithm as WHIP with the likelihood used in place of Whittle's likelihood. Therefore, theoretically the results of WHIP can at best match those of EXIP. However, for AP this result is not as theoretically obvious since AP also approximates the exact likelihood (albeit in a different way) and uses an approximate search method in the form of a genetic algorithm to estimate changepoint locations. Killick et al. (2012) have demonstrated that iterative PELT with the MDL for AR models as the penalised cost function outperforms AP, and so this suggests that the reduction in quality of the changepoint estimates due to the approximation made by Whittle's likelihood is greater than the increase in quality due to the use of iterative PELT.

The density of changepoints detected by WHIP for each model, shown in Figure 3.3.1, demonstrate that in general WHIP detects changes at their correct locations. The only model where the detected locations appear to diverge from the true locations is Model 13, shown in Figure 3.3.1(g). This difficulty in detecting the correct location is likely due to the change being small in magnitude. This difficulty is also reflected in the detection percentages, where WHIP, EXIP and AP all only detect the change in just over 30% of cases.

Overall, we have seen that WHIP is an improvement over WL, and in some aspects comparable with AP. Therefore, WHIP represents a pragmatically appropriate method for utilisation in the context of detecting changes in the second-order structure of acoustic sensing time series. Note that we do not compare the running times of the different algorithms, as in this case they are each implemented in different programming languages which have varying architectures and efficiencies. Hence, these discrepancies can result in vast differences in running times which are not necessarily a result of the methods themselves (but rather their implementations), and so comparing such values would not provide a fair comparison. Instead, we examine the computational complexities of the algorithms, which allow for unprejudiced comparison of the computational speed of the methods.

3.3.2 Computational Complexity

Changepoint detection in practical applications often favours methods which can be executed with a faster computational speed. Therefore, the computational complexity of each of the four approaches examined forms an important consideration when assessing their overall performance. The complexities of the WHIP, EXIP, WL and AP algorithms are each considered in turn.

The order of computation of the four methods can be summarised as follows:

$$\begin{aligned}
&\text{Complexity(method)}\\
&\quad = \mathcal{O}\Big(\mathcal{O}(\text{calculating penalised cost function})\\
&\qquad \times \mathcal{O}(\text{optimising penalised cost function} \mid \text{number and location of changepoints})\\
&\qquad \times \mathcal{O}(\text{optimising number and location of changepoints})\Big).
\end{aligned}$$

For WHIP and EXIP, the complexity of the optimisations is exactly the same. The only difference is the order of computation for the penalised cost calculation. We use the limited-memory BFGS algorithm (Liu and Nocedal, 1989) to perform the non-linear optimisation of the penalised cost function given the number and location of changepoints. This requires a computation time which is linear in the number of model parameters (Byrd et al., 1995), not including the changepoint locations, and therefore does not depend on the length of the data. In practice, we place a maximum possible order on the ARMA models which are considered. Denoting the maximum AR and MA orders considered by R and Q respectively, then optimising the penalised cost function given the number and locations of changepoints requires

$\mathcal{O}(R+Q)$ operations (and is hence independent of n).

For these two methods, the cost of optimising the number and locations of changepoints is equal to the cost of performing iterative PELT. Since the maximum number of iterations is bounded, this reduces to the computational cost of PELT. Killick et al. (2012) show that under certain conditions this is equal to Ln, where L is some constant. This makes use of the assumption that the number of changepoints increases linearly as the length of the time series increases, which is not particularly restrictive in this instance. As discussed in Section 3.2, Whittle's likelihood can be calculated in $\mathcal{O}(n \log n)$ time, whereas the exact likelihood used in EXIP can be calculated in $\mathcal{O}(n^3)$ time. Hence, the computational complexity of WHIP is

$$\begin{aligned}\text{Complexity(WHIP)} &= \mathcal{O}\Big(n \log n \times (R+Q) \times Ln\Big) \\ &= \mathcal{O}(n^2 \log n),\end{aligned}$$

since R and Q are constant and do not depend on n. In a similar manner, the computational complexity of EXIP is

$$\begin{aligned}\text{Complexity(EXIP)} &= \mathcal{O}\Big(n^3 \times (R+Q) \times Ln\Big) \\ &= \mathcal{O}(n^4).\end{aligned}$$

For Auto-PARM, the optimisation of both the penalised cost function and the number and location of changepoints is performed simultaneously through a genetic algorithm (GA). These optimisations depend on two components: the size of the population of solutions (P) and the number of generations considered (G). Stark and Spall (2002) describe how the total number of evaluations of the penalised cost function required for a given P and G is $\Big(P + (G-1)(P-1)\Big)$. It is not clear how these values depend on n, therefore we do not simplify them further. Since Auto-PARM uses the approximation to the likelihood based on the Yule-Walker estimate of the innovations variance, its penalised cost can be calculated in $\mathcal{O}(n)$ time using the innovations algorithm (see Sections 5.2 and 8.7 of Brockwell and Davis (2009)).

Therefore, the computational complexity of Auto-PARM is

$$\text{Complexity(Auto-PARM)} = \mathcal{O}\Big(n \times \Big(P + (G-1) \times (P-1)\Big)\Big).$$

For WL, the optimal value of the wavelet likelihood given the number and locations of changepoints can be obtained using a closed-form expression which requires $\mathcal{O}(n^3 \log n)$ operations to calculate (see Section 3.1.2 of Killick et al. (2013) for more details). Optimising the number and locations of changepoints is performed using the binary segmentation algorithm, which requires $\mathcal{O}(n \log n)$ operations (Vostrikova, 1981). Hence, the overall computational complexity of the WL procedure is

$$\begin{aligned} \text{Complexity(WL)} &= \mathcal{O}(n^3 \log n \times n \log n) \\ &= \mathcal{O}\Big(n^4 (\log n)^2\Big). \end{aligned}$$

Therefore, the complexity of WHIP is lower than both EXIP and WL, at least. Hence, motivated by this along with its relatively strong performance in its accuracy of estimation, we consider the application of WHIP (as well as other methods) for the analysis of acoustic sensing data.

3.4 Analysis of Acoustic Sensing Data

As discussed in Section 3.1, acoustic sensing is used within the oil industry for the monitoring of vibrations in wells. Interest lies in the detection of error features within such acoustic sensing time series, since these reflect the locations in time where the fibre-optic cable may have been disturbed externally (for example, at the surface of the well). Recall that engineers advise that such features manifest as sudden changes in the second-order structure of the time series, and that the influence of these disturbances is large enough that it is only necessary to analyse a single channel to detect them.

It is important to remove such error effects to allow for the effective analysis of acoustic sensing data. Ideally, this would be done without human intervention or use of 'expert knowledge'. Therefore, changepoint detection methodology offers a useful approach for this context.

The aim of this section is to use changepoint methodology to identify the location of error features within the examples of acoustic sensing data presented in Figures 3.1.1(b) ('Series 1') and 3.1.1(c) ('Series 2') above, via the detection of changes in the second-order structure of the series. To achieve this aim, the WHIP, EXIP and Auto-PARM methods are each independently applied to these acoustic sensing series. The changepoint locations detected by each of these methods for the series in Figures 3.1.1(b) and 3.1.1(c) are presented in Figures 3.4.1(a) and 3.4.1(b), respectively.

Examination of the results for Series 1 (Figure 3.4.1(a)) and Series 2 (Figure 3.4.1(b)) shows that all three methods provide sensible segmentations of the data. The detected changepoints illustrate that these error features caused by disturbances are categorised by a 'double burst' effect. There is an initial short powerful burst of vibration activity, with another longer less-powerful burst shortly after, followed by a period of activity before returning to its normal state.

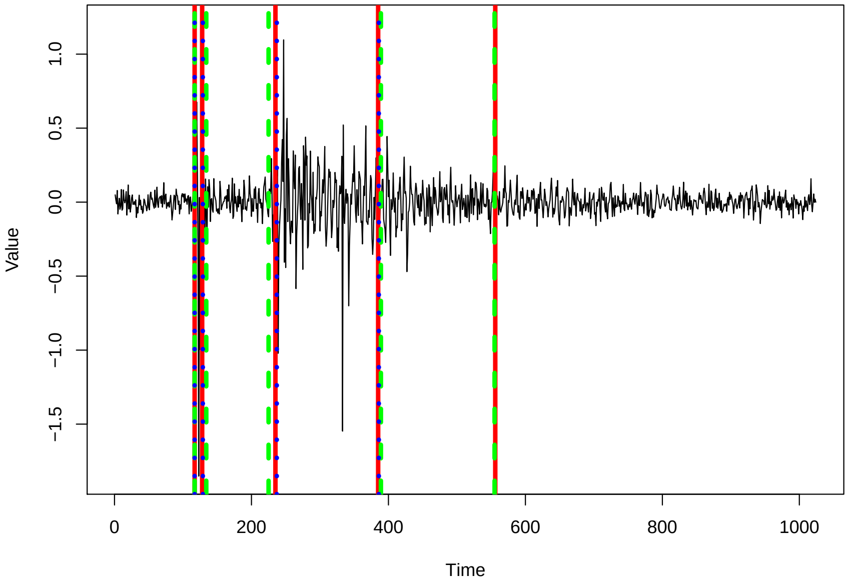
The changepoints estimated by WHIP and EXIP are very similar in both their number and location. The main exception is in Series 1, where the final changepoint detected by WHIP (and Auto-PARM) is not detected by EXIP. The estimates of AP are also similar to WHIP, with a small amount of variation in their location (particularly in Series 2). This is likely due to the stochastic nature of the genetic algorithm used in AP. Hence, since WHIP has a lower computational complexity than the exact likelihood approach of EXIP, and performs similarly to a method representing the cutting-edge of second-order changepoint detection, the WHIP method represents a sensible choice for the analysis of data possibly containing changes in the second-order structure.

3.5 Concluding Remarks

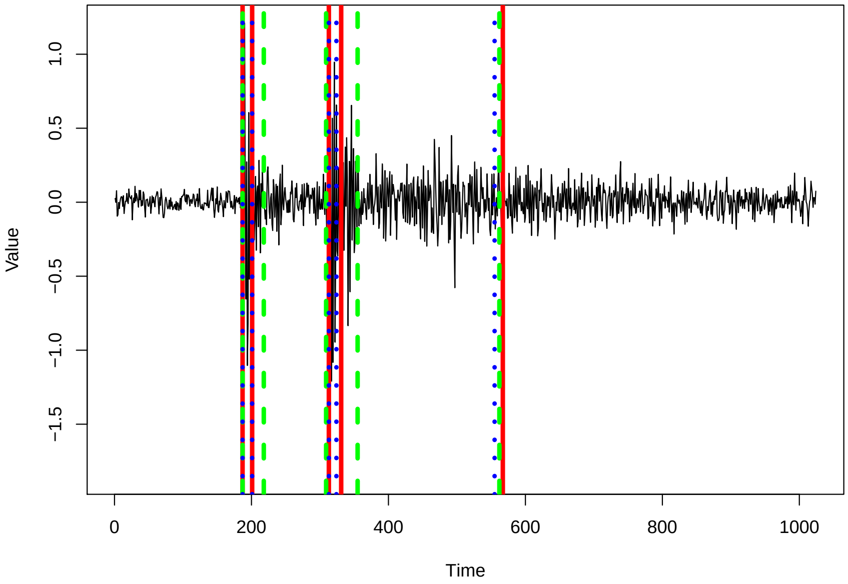
The detection of changes in dependence structure is an important aspect of analysing many real-world time series, in particular acoustic sensing data. We consider a penalised cost function approach for estimating the number and locations of second-order changepoints. Due to its reduced computational complexity and its success in modelling stationary time series, we utilise Whittle's likelihood approximation within the penalised cost. The use of a concave penalty function in our formulation is novel in this context and allows for the use of popular non-linear penalties such as the minimum description length. An iterative version of the PELT algorithm (Killick et al., 2012) allows us to obtain a high-quality solution to the optimisation problem.

To establish the difference in performance between our approach (WHIP) and similar likelihood-based methods, a simulated competition is performed. Comparisons and contrasts are draw between WHIP, an exact likelihood equivalent (EXIP), and two methods which represent the cutting-edge: Auto-PARM (Davis et al., 2006) and the Wavelet Likelihood approach (Killick et al., 2013). The results of this study demonstrate that WHIP is generally an improvement over the Wavelet Likelihood method, and reasonably comparable with EXIP and Auto-PARM.

Given these results, we apply WHIP, along with EXIP and Auto-PARM to two examples of acoustic sensing time series. This application illustrates how WHIP can be used to identify error features within the series, which correspond to time-points where the fibre-optic cable has been disturbed externally. As before, the three methods perform similarly. Therefore, given its computational benefit over EXIP, WHIP represents a pragmatically appropriate method for the detection of changes in the second-order structure of a time series.



3.4.1(a): Series 1 estimates



3.4.1(b): Series 2 estimates

Figure 3.4.1: Estimated locations of changes in spectral density for the two acoustic sensing time series from Figures 3.1.1(b) and 3.1.1(c) (Series 1 and Series 2), respectively. WHIP estimates are solid red, EXIP estimates are dotted blue, and Auto-PARM estimates are dashed green.

Chapter 4

Multivariate Changepoint Detection with Subsets

4.1 Introduction

Historically much of the research on changepoint analysis has focused on the univariate setting. However, increasingly data found in contemporary scientific fields are multivariate in nature, with each observation in a sequence containing the values of multiple variables which have been observed simultaneously. Such a shift has resulted in escalating interest in the problem of detecting changes which occur within multiple observed variables. The locations in time of such changes are referred to as *changepoints*. Areas in which such multivariate changepoints are important range from finance (Cho and Fryzlewicz, 2015) and geology (Srivastava and Worsley, 1986) to network analysis (Lung-Yut-Fong et al., 2011a) and genetics (Zhang et al., 2010; Jeng et al., 2013).

The multivariate changepoints which may be observed within such time series can be categorised as either *fully-multivariate* or *subset-multivariate*. Fully-multivariate changepoints refer to those changes in structure which occur simultaneously in *all* variables. Conversely, subset-multivariate changepoints refer to those which occur in only *a subset* of the observed variables. Such a situation is not uncommon in practice. Consider, for example, the finance setting. 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.

Traditionally, multivariate changepoint detection methods typically assume that all changes within a series are fully-multivariate. Popular approaches within such methods include the minimisation of a penalised cost function via dynamic programming (Lavielle and Teyssiere, 2006; Lung-Yut-Fong et al., 2011b), and the utilisation of binary segmentation techniques (Aue et al., 2009; Matteson and James, 2014). However, since these methods adopt the fully-multivariate changepoint model, they are making the assumption that any observed changes occur in all variables. This means that they are not able to accurately capture the 'subset' nature of multivariate changes often observed in practice.

More recently, increasing attention has been focused on the detection of subset-multivariate changes. A selection of methods which detect such changes are examined in Section 4.2.2. As will be discussed, many of the methods which have been proposed do not explicitly output the subsets of variables affected by the changes. They merely take into account the fact that not all of the variables may be changing. In addition, at the time of writing, *all* subset-multivariate changepoint detection methods are approximate in nature. That is, they cannot guarantee to produce the optimal segmentation of a multivariate time series under the subset-multivariate changepoint model.

The work presented in this chapter considers a novel approach to subset-multivariate changepoint detection in the general context. In particular, an exact search method is introduced which identifies both the locations of changes and the corresponding subsets of affected variables within a multivariate time series. These subsets are explicitly output in addition to the changepoint locations. The method is based upon a dynamic programming approach. Due to its exact nature, the resulting segmentation of the time series given by these changepoint locations and corresponding subsets is guaranteed to be optimal with respect to the goodness-of-fit criterion used. However, as we shall see later, obtaining such results under this model is an NP-hard problem. Hence, the methodology presented has limited ability to scale to scenarios with higher dimensions.

The remainder of this chapter is structured as follows. Section 4.2 introduces the multivariate changepoint detection problem in general, and formally details the difference between the fully-multivariate and subset-multivariate models. Section 4.3 outlines the presented formulation of the subset-multivariate changepoint detection problem and discusses how the problem can be tackled using a penalised cost function approach. Section 4.4 provides full details of the algorithm developed for the detection of subset-multivariate changepoints. Section 4.5 presents a simulation study used to demonstrate the characteristics of this methodology, with a discussion of the results given in Section 4.5.1. An analysis of the annual river flows of four rivers in Quebec is performed using the method is presented in Section 4.6 to demonstrate its potential for practical usage. The possibility of using inequality-based pruning within the proposed algorithm to improve computation time is then considered in Section 4.7.

4.2 The Multivariate Changepoint Detection Problem

The multivariate changepoint detection problem can be summarised as the search for potentially multiple changes in the statistical properties of a multivariate time-ordered data sequence. Such changes often manifest as shifts in the values of the mean or variance parameters of the observed variables, though more subtle changes such as alterations in the auto- or cross-correlation structure of the time series may also occur. The set of affected variables may differ for each change within the series.

More formally, suppose that $\boldsymbol{X}_{1:n} = \{\boldsymbol{X}_1, \boldsymbol{X}_2, \ldots, \boldsymbol{X}_n\}$ denotes a multivariate time series containing observations from p variables, such that $\boldsymbol{X}_t = (X_t^1, X_t^2, \ldots, X_t^p)$ for $t = 1, \ldots, n$. Suppose further that the series contains m distinct changepoints, the locations of which are denoted by $\boldsymbol{\tau} = \{\tau_1, \tau_2, \ldots, \tau_m\}$, where $\tau_i < \tau_j$ for $i < j$. For notational convenience, the definitions $\tau_0 = 0$ and $\tau_{m+1} = n$ are made. Each of these m changepoints has a corresponding subset of variables which are affected by the change. For the i^{th} changepoint τ_i, this subset is denoted by $\mathcal{S}_i$. Under the *fully*-multivariate changepoint model, the value of $\mathcal{S}_i$ is fixed as $\mathcal{S}_i = \{1, \dots, p\}$ for each $i = 1, \dots, m$. Conversely, under the *subset*-multivariate changepoint model, $\mathcal{S}_i$ could contain any possible subset of the observed variables, so that $\mathcal{S}_i \subseteq \{1, \dots, p\}$ for each $i = 1, \dots, m$. Therefore, while the *fully*-multivariate changepoint problem aims to find only the optimal set of changepoint locations $\boldsymbol{\tau} = \{\tau_1, \dots, \tau_m\}$, the objective of the *subset*-multivariate changepoint problem is to obtain the optimal values of $\boldsymbol{\tau} = \{\tau_1, \dots, \tau_m\}$ as well as the optimal associated subsets of affected variables, $\boldsymbol{\mathcal{S}} = \{\mathcal{S}_1, \dots, \mathcal{S}_m\}$. Note that $\mathcal{S}_0$ and $\mathcal{S}_{m+1}$ are fixed such that $\mathcal{S}_0 = \mathcal{S}_{m+1} = \{1, \dots, p\}$. We emphasise that fully-multivariate changepoints (i.e. the changepoints for which $\mathcal{S}_i = \{1, \dots, p\}$) are simply special cases of subset-multivariate changepoints, and can hence be detected under the subset-multivariate changepoint model.

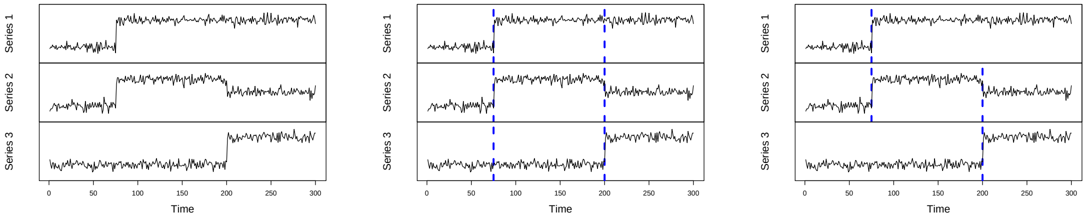
To obtain the optimal changepoint locations and associated affected variable subsets under the subset-multivariate changepoint model, we consider the minimisation of a penalised cost function of the form:

$$cost(\boldsymbol{X}_{1:n}, \boldsymbol{\tau}, \boldsymbol{\mathcal{S}}) + pen(\boldsymbol{\tau}, \boldsymbol{\mathcal{S}}). \tag{4.2.1}$$

The concept of minimising a penalised cost function for changepoint detection has been used with success in the univariate setting and in the fully-multivariate context, see Chapter 2 for more details. Here $cost(\boldsymbol{X}_{1:n}, \boldsymbol{\tau}, \boldsymbol{\mathcal{S}})$ provides a cost for a multivariate time series $\boldsymbol{X}_{1:n}$ segmented by the changepoint configuration specified by $\boldsymbol{\tau}$ and $\boldsymbol{\mathcal{S}}$. The $pen(\boldsymbol{\tau}, \boldsymbol{\mathcal{S}})$ term is a penalty function selected to prevent the over-estimation of the number of changepoints and size of the affected variable subsets. A lower value of the cost function means that the corresponding $\boldsymbol{\tau}$ and $\boldsymbol{\mathcal{S}}$ provide a better fit to the data, while $pen(\boldsymbol{\tau}, \boldsymbol{\mathcal{S}})$ increases with each additional changepoint included in the model, and each additional variable added to the affected subset for a given changepoint. We note that our use of the word *optimal* is in the sense that they minimise our penalised cost function.

4.2.1 Fully-Multivariate vs Subset-Multivariate

To emphasise the difference between fully-multivariate and subset-multivariate changepoint models, consider the example of a multivariate time series presented in Figure 4.2.1(a). This series $\boldsymbol{X}_{1:300}$ contains $n = 300$ observations of $p = 3$ variables (numbered in ascending order from top to bottom in Figure 4.2.1(a)). There are two changes in the mean vector within the series at times $\tau_1 = 75$ and $\tau_2 = 200$, with the corresponding subsets of affected variables being $\mathcal{S}_1 = \{1, 2\}$ and $\mathcal{S}_2 = \{2, 3\}$. The data are i.i.d. within the segments, each segment is independent of the others, and the variables have zero cross-correlation.



4.2.1(a): Series containing two multivariate changes.

4.2.1(b): Fully-multivariate model.

4.2.1(c): Subset-multivariate model.

Figure 4.2.1: An example of a multivariate time series with changes in subsets of variables. Here the changes are in mean, but in practice they can be in any statistical property. The plots shows the changepoints placed under the fully-multivariate and subset-multivariate models, respectively.

Under the fully-multivariate changepoint model, such as that proposed by Matteson and James (2014), a detection method would place two changepoints in the series across all variables, as demonstrated in Figure 4.2.1(b). It is clear that this does not accurately reflect the true nature of the changes, since variable 3 does not change at τ_1 and variable 1 does not change at τ_2. Rather it would be desirable to have a detection method which adopts the subset-multivariate changepoint model. Under the subset-multivariate model, a detection method identifies and utilises information regarding the subset-multivariate nature of the changepoints in order to assist in their detection.

More generally, the use of a fully-multivariate detection method in scenarios where subset-multivariate changepoints are present may lead to a reduction in the quality of estimation. This is due to the inherent overestimation of the number of affected variables for a given subset-multivariate changepoint when a fully-multivariate method is used. If this overestimation is large (which will be the case when the true number of affected variables is small), then this may in turn lead to a poor estimation of the changepoint location(s) in the series. This is due to the fully-multivariate method attempting to 'correct' for its overestimation of the subset size (which it cannot control) by shifting the changepoint locations or adding additional changepoints (which it is able to control). It is this attempt at compensation for the intrinsic fully-multivariate assumption which is likely to lead to poor segmentations.

If the fully-multivariate method is based on a penalised cost approach, then a larger penalty value could be used in an effort to potentially reduce this overestimation of the number of changepoints (which has been induced by the intrinsic fully-multivariate assumption). However, this could potentially result in an underestimation of the number of changepoints. This is because true subset-multivariate changepoints are only affecting a subset of the variables and therefore likely to have less impact on the value of the fully-multivariate test statistic. Hence, if a larger penalty a used, this would increase the threshold for which the test statistic value would need to exceed, therefore making it even more difficult for the subset-multivariate changes to be detected (compared to the true fully-multivariate changes, where each variable is contributing to the test statistic).

Ideally, a subset-multivariate detection method would be able to place changepoints in only the correct set of affected variables, as shown in Figure 4.2.1(c). However, as we discuss in Section 4.2.2 below, the majority of subset-multivariate changepoint detection methods available in the literature do not possess such a feature.

4.2.2 Current Subset-Multivariate Approaches

Recent methods tackling the multivariate changepoint problem have been proposed which consider the detection of subset-multivariate changes. Such methods can be subdivided into two categories: those which do not output the set of affected variables within the series for each detected changepoint, and those which do output such subsets. Examples from the former include Zhang et al. (2010), Siegmund et al. (2011), Xie and Siegmund (2013), Jeng et al. (2013), Bardwell and Fearnhead (2014) and Cho and Fryzlewicz (2015).

Zhang et al. (2010), Siegmund et al. (2011) and Jeng et al. (2013) all introduce methods within the genomics literature for the detection of multiple intervals of altered mean within multivariate DNA copy number profiles. Often the DNA variations will occur in only a proportion of the samples, so subset-multivariate changepoint detection techniques are necessary. These methods all search for pairs of changepoints which correspond to the altered-mean intervals. The observations are modelled as multivariate Normal with diagonal covariance matrices. The test statistics used by the methods are based on the scaled Normal log-likelihood under the assumption of a segment of altered mean. Each of these methods promote the utilisation of modified binary segmentation procedures. However, they differ in the nature of the changes they detect. Zhang et al. (2010) detects changes which have a relatively large number of affected variables (referred to as 'common' changes), whereas Siegmund et al. (2011) detects those changes which have a relatively small number of affected variables (referred to as 'rare' changes). Jeng et al. (2013) detects both rare and common changes.

Other methods for detecting changes in DNA copy number profiles are proposed by Xie and Siegmund (2013) and Bardwell and Fearnhead (2014), but in contrast Xie and Siegmund (2013) use the test statistic of Siegmund et al. (2011) to detect rare changes in sequentially-observed data and Bardwell and Fearnhead (2014) adopt a Bayesian approach which utilises a hidden state model. In other literature, Cho and Fryzlewicz (2015) use a binary segmentation approach for the detection of changes in the auto- and cross-covariance of multivariate time series. They use a wavelet-based test statistic which aggregates information across variables with thresholding, reducing the effect of variables not affected by the change.

In contrast to the approaches considered above, Maboudou-Tchao and Hawkins

(2013) and Preuß et al. (2015) each present methods which explicitly output *both* the locations of subset-multivariate changepoints *and* their corresponding sets of affected variables. The method of Maboudou-Tchao and Hawkins (2013) works by first performing the dynamic program from Maboudou and Hawkins (2009) (discussed in Chapter 2) under the fully-multivariate changepoint model, and then performing variable-specific hypothesis tests for each estimated changepoint to determine its affected variable subset.

Similar to Cho and Fryzlewicz (2015), Preuß et al. (2015) deviate from the setting of i.i.d. data and detect multiple changes in autocovariance through the consideration of raw periodograms. A three-step procedure is used: testing for the structural breaks, identifying the variables affected by each changepoint, and the localisation of the changes.

As discussed, each of the methods considered are approximate in their nature and hence cannot guarantee to provide the optimal configuration of changepoints and affected variable subsets. In addition, only a small number of the available methods explicitly output the set of affected variables for the changepoints. Motivated by this, the aim of our work in this chapter is to develop methodology which obtains exactly the optimal changepoint locations and corresponding subsets of affected variables, and explicitly output both these locations and subsets. The problem we consider is similar to the i.i.d. setting considered by Maboudou-Tchao and Hawkins (2013), rather than the scenario examined by Preuß et al. (2015) where auto- and cross-correlation may be present.

4.3 Modelling Subset-Multivariate Changepoints

We now consider how the subset-multivariate changepoint problem can be formulated with a view to producing an optimal solution for the piecewise i.i.d. setting. To begin, we introduce *changepoint vectors*, a quantity that will prove useful as it permits us to specify the most recent changepoints locations in each variable of a series at a given time-point. There then follows a discussion of how subset-multivariate changepoints can be modelled via the penalised cost paradigm using these changepoint vectors as a building block.

4.3.1 Changepoint Vectors

Traditionally, changepoints in the multivariate setting are modelled as the time-points at which all variables change. However, this approach can suffer in scenarios where only some of the variables are changing. We therefore propose an alternative approach using the concept of *changepoint vectors*. This idea is introduced to allow for the segmentation of a time series under the subset-multivariate model. This in turn allows for the costing of a multivariate time series under this model.

Let c_t^j denote the location of the most recently observed changepoint in variable j prior to, and including, time t. Hence, if a changepoint occurs at time u in variable j, we have $c_u^j = u$. The *changepoint vector* corresponding to time t is defined as the vector of these most recently observed changepoints for all variables at time t. For a p-variate series of length n, this is denoted by $c_t = (c_t^1, c_t^2, \ldots, c_t^p)$. If there are m known changes in this series at $\tau_1, \tau_2, \ldots, \tau_m$ (with $\tau_0 = 0$ and $\tau_{m+1} = n$) that have affected variables subsets $\mathcal{S}_1, \ldots, \mathcal{S}_m$, then for each $k = 0, \ldots, m+1$ we have $c_{\tau_k}^j = \tau_k$ for all $j \in \mathcal{S}_k$. Also note that the changepoint vectors are only updated when a changepoint occurs, so that $c_t = c_{t-1}$ for all $\tau_k + 1 \leq t < \tau_{k+1}$ $(k = 1, \ldots, m)$. For notational simplicity, we define $c_0 = (0, 0, \ldots, 0)$ and $c_n = (n, n, \ldots, n)$.

Consideration will be given to various sets of these changepoint vectors throughout the proposed methodology. The most important of these is the set C_t, which denotes the set of all possible previous changepoint vectors c_t up to and including a given time t. For example, if $p = 2$ and $t = 2$, then we have

$$C_t = \{(0,0), (0,1), (1,0), (1,1), (0,2), (2,0), (1,2), (2,1), (2,2)\}.$$

We fix $C_0 = \{c_0\}$ and $C_n = \{C_{n-1}, c_n\}$. As we shall see later, this construct will be pivotal for the segmentation of a multivariate time series under the subset-multivariate model. Further, for a given t, we define $\bar{C}_t$ to be the set of all possible c_t such that

$c_t^j = t$ for at least one j, so that at least one variable is changing at time t. For the same example of $p = 2$ and $t = 2$, we have

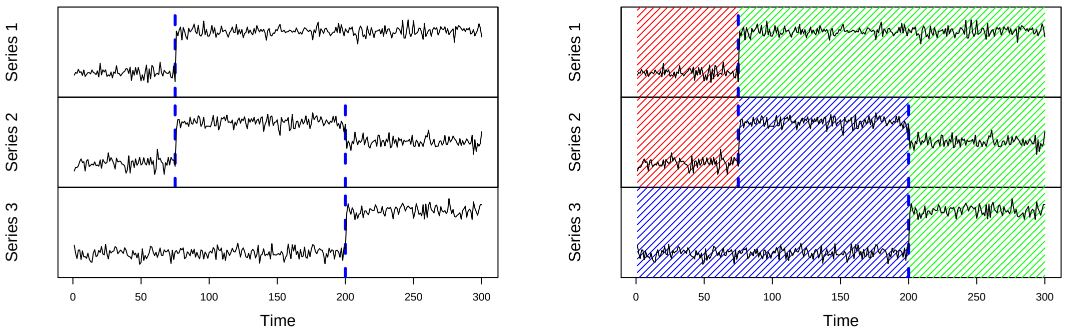
$$\bar{C}_t = \{(0, 2), (2, 0), (1, 2), (2, 1), (2, 2)\}.$$

We note that $\bar{C}_0 = \{c_0\}$ and $\bar{C}_n = \{c_n\}$. Also, since for each $c_s \in \bar{C}_s$ we have $s \in c_s$, and so for some $t \neq s$ by the definition of $\bar{C}_s$ and $\bar{C}_t$ we must have $c_t \notin \bar{C}_s$ for each $c_t \in \bar{C}_t$. Therefore, we have $\bar{C}_s \cap \bar{C}_t = \emptyset$ for each s, t such that $s \neq t$. Consequently, we can write

$$C_t = \{\bar{C}_0, \bar{C}_1, \ldots, \bar{C}_{t-1}, \bar{C}_t\}.$$

Finally, suppose we have a p-variate series with r changepoints before some time t, at locations $\tau_1, \ldots, \tau_r$. Then for some given changepoint vector $c_t \in \bar{C}_t$, we define $\boldsymbol{c}(c_t) = (c_{\tau_0}, c_{\tau_1}, \ldots, c_{\tau_r}, c_t)'$, where $\tau_0 = 0$. This means that $\boldsymbol{c}(c_t)$ represents a $(r+2)\times p$ matrix containing the unique changepoint vectors occurring prior to and including c_t. This is conceptually similar to the set of true changepoint locations in the traditional fully-multivariate or univariate changepoint models. Where clear, we simply use $\boldsymbol{c} = \boldsymbol{c}(c_n)$ to denote the set of all unique true changepoint vectors in a series. Hence, $\boldsymbol{c}$ contains the equivalent information about the changes in the series as $(\boldsymbol{\tau}, \boldsymbol{\mathcal{S}})$.

Example To illustrate the outlined notation, we refer back to the example time series $\boldsymbol{X}_{1:300}$ presented in Figure 4.2.1(a). The changepoint locations and corresponding subsets of affected variables are known. These are highlighted once again in Figure 4.3.1(a). Since we have $\tau_1 = 75$, $\tau_2 = 200$ and $\mathcal{S}_1 = \{1, 2\}$ and $\mathcal{S} = \{2, 3\}$, then the changepoint vectors corresponding to τ_1 and τ_2 are given by $c_{\tau_1} = (\tau_1, \tau_1, \tau_0) = (75, 75, 0)$ and $c_{\tau_2} = (\tau_1, \tau_2, \tau_2) = (75, 200, 200)$. Note that, by convention, $c_{\tau_0} = c_0 = (0, 0, 0)$ and $c_{\tau_3} = c_{300} = (300, 300, 300)$. Also, for $1 \leq r < \tau_1$, $\tau_1 \leq s < \tau_2$ and $\tau_2 \leq t < 300$, we have $c_r = (0, 0, 0)$, $c_s = (75, 75, 0)$ and $c_t = (75, 200, 200)$. Figure 4.3.1(b) presents a visualisation of the segmentation provided by these changepoint vectors. Each different shading represents a different segment.



4.3.1(a): Multivariate time series with known changepoints and affected variable subsets.

4.3.1(b): One possible visualisation of the segmentation of the example time series using the changepoint vector concept.

Figure 4.3.1: An example of a subset-multivariate time series and its segmentation using the changepoint vector concept.

Note that this concept of changepoint vectors is only one possibility for segmenting a time series under the subset-multivariate model. Other segmentations are conceivable and equally valid. The proposed segmentation is preferred because the right-hand side of each segment is 'flat', meaning that the segment can then be thought of as 'closed-off' for any following time-points.

4.3.2 Formulating a Penalised Cost Function

We now consider how a cost can be assigned to a multivariate time series under the subset-multivariate model. In particular we focus on a scenario where the number and locations of changepoints and affected variable subsets are unknown.

Suppose we have a p-variate series $\boldsymbol{X}$ which contains an unknown number of changepoints (potentially zero), and the locations of these possible changepoints and the subsets of variables in which they occur are unknown. As before, suppose that the variables are uncorrelated and that the observations within the segments are i.i.d. and independent of the those in other segments. We define $\mathcal{D}_j(\cdot)$ as a generic additive cost function for each variable $j = 1, \ldots, p$ which assigns a cost to a set of contiguous i.i.d. univariate observations, and use $\mathbb{I}(\cdot)$ to denote the indicator function.

Since the changepoints' number, locations and affected variables are unknown, we might typically calculate the cost of this multivariate series under a range of different potential segmentations. This permits us to decide which segmentations are most suitable for the data. As the introduction of a changepoint into the model generally provides a reduction in cost, it is possible to over-fit to the data. Hence, to avoid the over-fitting of changepoints, it is necessary to penalise the addition of a changepoint in the model through the addition of a penalty term.

We begin the presentation of this penalised approach by defining the size of a given subset of variables. Define q_{τ_k} to be the total number of elements of the changepoint vector c_{τ_k} which are equal to τ_k, so that

$$q_{\tau_k} := \sum_{j=1}^{p} \mathbb{I}(c^j_{\tau_k} = \tau_k).$$

Then q_{τ_k} can be interpreted as the number of variables changing at τ_k, and are hence affected by the change at τ_k. We can therefore define the penalised cost of $\boldsymbol{X}$ for the case of unknown changepoint vectors $\boldsymbol{c} = (c_{\tau_0}, c_{\tau_1}, \ldots, c_{\tau_m}, c_{\tau_{m+1}})$ by

$$cost(\boldsymbol{X}, \boldsymbol{c}) + pen(\boldsymbol{c}) = \sum_{k=1}^{m+1} \left(\sum_{j=1}^{p} \left[\mathbb{I}(c^j_{\tau_k} = \tau_k) \mathcal{D}_j(X^j_{(c^j_{\tau_{k-1}}+1):c^j_{\tau_k}}) \right] + \alpha g(q_{\tau_k}) \right) + \beta f(m) \tag{4.3.1}$$

Here the $\alpha g(q_{\tau_k})$ term is a penalty to guard against over-fitting the number of variables affected by the k^{th} changepoint, and the $\beta f(m)$ penalty term is to guard against over-fitting the number of changepoints in the series. We assume that these two aspects behave independently of one another. The functions g and f are increasing functions of their respective parameters, and both α and β are positive constants which are referred to as the penalty constants. We adopt this approach to explicitly allow for a greater degree of control regarding how the addition of changepoints is penalised within the model.

The choice of the functions to use for g and f is itself an open question. However, as is common in the literature, we take $g(q_{\tau_k}) = q_{\tau_k}$ and $f(m) = m$. Informally, this value of f means that for every changepoint included in the model an extra β is added to the cost function. In the case of no true changepoints, a single β is present due to the ‘changepoint’ at the end of the data. Similarly, this value of g means that for a given changepoint τ_k, an additional α is added to the cost function for each additional variable which is said to contain the change at τ_k (in addition to the original β which is added for initially detecting change). Equation (4.3.1) becomes

$$\begin{aligned}
&cost(\boldsymbol{X}, \boldsymbol{c}) + pen(\boldsymbol{c}) \\
&= \sum_{k=1}^{m+1} \left(\sum_{j=1}^{p} \left[\mathbb{I}(c^j_{\tau_k} = \tau_k) \mathcal{D}_j (X^j_{(c^j_{\tau_{k-1}}+1):c^j_{\tau_k}}) \right] + \alpha q_{\tau_k} \right) + \beta(m+1), \\
&= \sum_{k=1}^{m+1} \left(\sum_{j=1}^{p} \left[\mathbb{I}(c^j_{\tau_k} = \tau_k) \mathcal{D}_j (X^j_{(c^j_{\tau_{k-1}}+1):c^j_{\tau_k}}) \right] + \alpha \sum_{j=1}^{p} \mathbb{I}(c^j_{\tau_k} = \tau_k) \right) + \beta(m+1), \\
&= \sum_{k=1}^{m+1} \left\{ \sum_{j=1}^{p} \left[\mathbb{I}(c^j_{\tau_k} = \tau_k) \left(\mathcal{D}_j (X^j_{(c^j_{\tau_{k-1}}+1):c^j_{\tau_k}}) + \alpha \right) \right] + \beta \right\}. \qquad (4.3.2)
\end{aligned}$$

The optimal changepoint vectors for the given multivariate time series $\boldsymbol{X}$ are those which minimise $cost(\boldsymbol{X}, \boldsymbol{c}) + pen(\boldsymbol{c})$. Therefore, the detection of changepoints (and corresponding subsets) in the subset-multivariate changepoint model corresponds to the minimisation of (4.3.2). In the next section we introduce methodology which is capable of performing this minimisation exactly.

4.4 Detecting Subset-Multivariate Changepoints

Suppose we wish to identify the subset-multivariate changepoint model for a given multivariate time series which is optimal with respect to the cost function $\mathcal{D}_j$ being used. We therefore need to minimise the penalised cost function (4.3.2) over all possible changepoints and all possible subsets of variables for each changepoint. With a view to utilising the dynamic programming techniques which have been successfully applied in other multivariate changepoint detection methods (Lavielle and Teyssiere (2006), Maboudou-Tchao and Hawkins (2013)), we propose a method which we call Subset Multivariate Optimal Partitioning (SMOP).

The aim of the proposed method is to relate the optimal (i.e. minimum) penalised cost of the series up to the current changepoint vector, to the optimal penalised cost of the series up to the most recent distinctly-different changepoint vector. To this end, consider a p-variate dataset $\boldsymbol{X}_{c_u} = (X^1_{1:c^1_u}, X^2_{1:c^2_u}, \ldots, X^p_{1:c^p_u})'$, where $c_u \in \bar{C}_u$ is the vector of most recent changepoints in each variable up to (and including) time u. This implies that the individual series for each of the variables may have differing lengths. We assume that each variable is independent of the others (i.e. there is zero cross-correlation), the observations are i.i.d. within each segment and the observations in one segment are independent of those in all the other segments. Define $F(c_u)$ to be the minimisation of the penalised cost (4.3.2) for $\boldsymbol{X}_{c_u}$. Also define $\mathcal{H}_{c_u}$ to be the set

$$\mathcal{H}_{c_u} = \left\{ \boldsymbol{c}(c_u) = (c_{\tau_0}, c_{\tau_1}, \ldots, c_{\tau_m}, c_{\tau_{m+1}} = c_u)' : \begin{array}{l} 0 = \tau_0 < \tau_1 < \ldots < \tau_m < \tau_{m+1} = u; \\ c_{\tau_k} \in \bar{C}_{\tau_k} \quad \forall\ 1 \leq k \leq m+1; \\ c^j_{\tau_i} \leq c^j_{\tau_k} \quad \forall\ i < k,\ j \in \{1, \ldots, p\} \end{array} \right\}.$$

By construction, $\mathcal{H}_{c_u}$ is the set of all possible matrices of the most recently observed changepoints for each variable at each distinct τ_k up to and including $\tau_{m+1} = u$. In addition, we make the following definitions for changepoint vectors $c \in C_n$, $c_r \in C_r$ and $c_s \in C_s$ (with $r < s \leq n$ and $c^j_r \leq c^j_s$ for all $j = 1, \ldots, p$):

- $\mathcal{L}(c)$ is the set of all previous changepoint locations occurring in any variable prior to *and including* the corresponding $c \in C_n$; and

- $M(c) = |\mathcal{L}(c)|$ is the number of changepoint locations occurring in any variable up to and including those in $c \in C_n$; and

- $m(c_r, c_s) = |c_s \setminus \mathcal{L}(c_r)|$, so that $m(c_r, c_s)$ represents the number of additional changepoints which have occurred between c_r and c_s (including the changes occurring at c_s, but *not* those at c_r).

Proposition 4.4.1 now demonstrates how the minimum cost of $\boldsymbol{X}_{c_u}$ can be calculated in terms of the minimum cost of $\boldsymbol{X}_{c_t}$, where $t < u$ and $c^j_t \leq c^j_u$ for all $j = 1, \ldots, p$.

Proposition 4.4.1. *For a given changepoint vector $c_u \in C_n$ (where $u = \max(c_u)$), we have*

$$F(c_u) = \min_{0 \leq t < u} \left\{ \min_{c_t \in \{\bar{C}_t \,:\, c_t^j \leq c_u^j \ \forall\ j\}} \left[F(c_t) + \sum_{j=1}^{p} \left[\mathbb{I}(c_t^j \neq c_u^j) \left(\mathcal{D}_j(X^j_{(c_t^j+1):c_u^j}) + \alpha \right) \right] + m(c_t, c_u)\beta \right] \right\}. \tag{4.4.1}$$

Proof. See Appendix A.1 for a full proof. □

Hence, finding the minimum value of the penalised cost function (4.3.2) for the whole time series $\boldsymbol{X} = (\boldsymbol{X}_1, \boldsymbol{X}_2, \ldots, \boldsymbol{X}_n)$ over all possible changepoints and all possible subsets is equivalent to finding $F(c_n)$, recalling that $c_n = (n, n, \ldots, n)$. This is obtained by recursively calculating $F(c_u)$ for every possible $c_u \in \bar{C}_u$ in turn for each $u = 1, 2, \ldots, n$.

Suppose we have some time-point $t \in [1, n-1]$, a corresponding changepoint vector $c_t \in \bar{C}_t$, and some previous changepoint vector $c \in C_t$ such that $c^j \leq c^j_{\tau^*}$ for all variables $j \in [1, p]$. Then we define $h_{c_t}(c)$ as

$$h_{c_t}(c) = F(c) + \sum_{j=1}^{p} \left[\mathbb{I}(c^j \neq c_t^j) \left(\mathcal{D}_j(X^j_{(c^j+1):c_t^j}) + \alpha \right) \right] + m(c, c_t)\beta. \tag{4.4.2}$$

Intuitively, $h_{c_t}(c)$ denotes the minimum cost to c_t under the assumption that c is the vector of the optimal most-recent changepoints prior to c_t. In order to calculate the minimum penalised cost of the whole series, it is necessary to calculate $h_{c_t}(c)$ for every $c \in C_t$ for every $t \in [1, n-1]$ and $c_t \in \bar{C}_t$. It is readily shown that if $p > 1$, then for a given $t \in [1, n-1]$ there are $(t+1)^p - t^p$ elements of $\bar{C}_t$ and $(t+1)^p$ elements of C_t. Therefore, this is an $\mathcal{O}\left(\sum_{t=1}^{n-1} [((t+1)^p - t^p) \times (t)^p \times p]\right) = \mathcal{O}\left(pn^{2p}\right)$ calculation.

We introduce an algorithm for solving this recursion which takes a similar approach to the Optimal Partitioning method of Jackson et al. (2005). We refer to our algorithm as Subset Multivariate Optimal Partitioning (SMOP). To describe this algorithm, we first define the following set for a given $\tau^* \in \{1, \ldots, n\}$, $c_{\tau^*} \in \bar{C}_{\tau^*}$ and $\tau \in$

$\{1, \ldots, \tau^* - 1\}$:

$$C_\tau(c_{\tau^*}) = \left\{ c \in C_\tau : \; c^j < c^j_{\tau^*} \;\; \forall \; j \in [1, p] \; \right\}, \tag{4.4.3}$$

so that $C_\tau(c_{\tau^*})$ contains all changepoint vectors in C_τ which are 'before' c_{τ^*}. Steps for the implementation of SMOP are given in Algorithm 3.

```
Algorithm 3: Subset Multivariate Optimal Partitioning (SMOP)
Input     : A multivariate time series $\boldsymbol{X} = (\boldsymbol{X}_1, \boldsymbol{X}_2, \ldots, \boldsymbol{X}_n)$ containing $p$
            variables, a univariate cost function $\mathcal{D}_j(\cdot)$ for each variable $j$, and
            penalty constants $\alpha$ and $\beta$.
Initialise: Set $F(c_0) = 0$, $\mathcal{L}(c_0) = \emptyset$ and $\boldsymbol{c}(c_0) = \emptyset$.
begin
    for $\tau^* \in \{1, \ldots, n\}$ do
        for $c_{\tau^*} \in \bar{C}_{\tau^*}$ do
            for $c \in C_{\tau^*-1}(c_{\tau^*})$ do
                Set $h_{c_{\tau^*}}(c) = F(c) + \sum_{j=1}^{p} \left[ \mathbb{I}(c^j \neq c^j_{\tau^*}) \left( \mathcal{D}_j(X^j_{(c^j+1):c^j_{\tau^*}}) + \alpha \right) \right]$
                                 $+ m(c, c_{\tau^*})\beta$
            Set $F(c_{\tau^*}) = \min_{c \in C_{\tau^*-1}(c_{\tau^*})} \{h_{c_{\tau^*}}(c)\}$
            Set $c' = \arg\min_{c \in C_{\tau^*-1}(c_{\tau^*})} \{h_{c_{\tau^*}}(c)\}$
            Set $\mathcal{L}(c_{\tau^*}) = \mathcal{L}(c') \cup \{c^1_{\tau^*}, c^2_{\tau^*}, \ldots, c^p_{\tau^*}\}$
            Set $\boldsymbol{c}(c_{\tau^*}) = \left( \boldsymbol{c}(c'), c_{\tau^*} \right)$
Output    : The sequence of most-recent changepoint vectors recorded in
            $\boldsymbol{c}\Big( (n, n, \ldots, n) \Big)$.
```

The strength of the SMOP algorithm is its ability to obtain exactly the subset-multivariate segmentation of a series which is optimal with respect to the cost function and penalty values used, in terms of both the locations in time at which any changes occur and the subset of variables which are affected. This is possible as no assumptions are made regarding whether or not certain variables (or a certain number or proportion of variables) contain a change. However, due to the exploding size of the C_t and $\bar{C}_t$ sets, particularly for large t, execution of the method becomes increasingly computationally intensive even for relatively small n.

We note that if one wishes to consider only fully-multivariate changepoints within the SMOP algorithm, so that only changepoint vectors of the form $(\tau^*, \tau^*, \ldots, \tau^*)$ are considered, then the SMOP algorithm becomes equivalent to performing the PELT algorithm of Killick et al. (2012) using a multivariate cost function (instead of a univariate) with a penalty of $p\alpha + \beta$ (where α and β are the penalty constants used within SMOP). Hence, in such a case the computational burden of SMOP would be equivalent to that of PELT.

4.5 Simulation Study

We examine SMOP through the execution of a simulation study, the aim of which is to demonstrate the characteristics of the method and its performance in a range of different scenarios. Six different scenarios are considered, each of which has been constructed to reflect a certain situation that illustrates interesting features of SMOP or allows for interesting comparisons with other leading changepoint detection methods. We note that due to the computational intensity of the approach, in general we consider time series of length $n = 100$ containing $p = 3$ variables.

In each scenario we assume that the individual variables are piecewise Normally distributed (except in indicated cases), i.i.d. within their segments, that each segment is independent of the others and that there is zero cross-correlation between the variables. All changes are either in mean, variance, or both, with the appropriate cost functions being used in each case. For each changepoint, only a certain subset of variables in the series change. A total of 100 replications are simulated for each scenario. Full details of the scenarios considered and their corresponding results are outlined in Section 4.5.1 below.

For each application of SMOP, the number and locations of the detected changepoints are recorded along with the corresponding subsets of affected variables. To assess the performance of SMOP on each scenario, we consider three different metrics:

- the average number of changepoints estimated;
- the average V-measure (Rosenberg and Hirschberg, 2007) of the segmentations produced;
- the density of estimated changepoints at each time-point in each variable.

The *V-measure*, proposed by Rosenberg and Hirschberg (2007), is a quality-of-fit measure which rates the quality of a given segmentation (compared to the true segmentation) on the $[0, 1]$ scale. This rating depends on how successful the segmentation is in satisfying the criteria of homogeneity and completeness. These criteria assess how well a segmentation assigns those, and only those, observations from a certain true segment to a single estimated segment. A larger value indicates higher accuracy, with a value of 1 indicating a perfect segmentation.

More specifically, V-measure can be calculated in the following manner. Suppose that the true segmentation of a time series is denoted by $R_{\text{true}} = \{r^1_{\text{true}}, r^2_{\text{true}}, \ldots, r^{N_{\text{true}}}_{\text{true}}\}$, so that r^i_{true} denotes the i^{th} true segment and N_{est} denotes the number of true segments. Similarly, suppose $R_{\text{est}} = \{r^1_{\text{est}}, r^2_{\text{est}}, \ldots, r^{N_{\text{est}}}_{\text{est}}\}$ denotes some estimated segmentation of the same series, with r^i_{est} and N_{est} defined as equivalent for the true segment. Define the set $A = \{a_{ij} : i = 1, \ldots, N_{\text{true}},\ \ j = 1, \ldots, N_{\text{est}}\}$ where a_{ij} is the number of observations which lie in the true segment r^i_{true} and the estimated segment r^j_{est}. Then homogeneity $\mathcal{U}$ can be defined as

$$\mathcal{U} = \begin{cases} 1 & \text{if } H(R_{\text{true}}) = 0 \\ 1 - \frac{H(R_{\text{true}}|R_{\text{est}})}{H(R_{\text{true}})} & \text{otherwise} \end{cases}, \tag{4.5.1}$$

where

$$\begin{aligned} H(R_{\text{true}}|R_{\text{est}}) &= -\sum_{j=1}^{N_{\text{est}}} \sum_{i=1}^{N_{\text{est}}} \frac{a_{ij}}{n} \log \frac{a_{ij}}{\sum_{i=1}^{N_{\text{true}}} a_{ij}} \\ H(R_{\text{true}}) &= -\sum_{i=1}^{N_{\text{true}}} \frac{\sum_{j=1}^{N_{\text{est}}} a_{ij}}{N_{\text{true}}} \log \frac{\sum_{j=1}^{N_{\text{est}}} a_{ij}}{N_{\text{true}}}. \end{aligned}$$

Similarly, completeness $\mathcal{W}$ can be defined as

$$\mathcal{W} = \begin{cases} 1 & \text{if } H(R_{\text{est}}) = 0 \\ 1 - \frac{H(R_{\text{est}}|R_{\text{true}})}{H(R_{\text{est}})} & \text{otherwise} \end{cases}, \tag{4.5.2}$$

where

$$\begin{aligned} H(R_{\text{est}}|R_{\text{true}}) &= -\sum_{i=1}^{N_{\text{true}}} \sum_{j=1}^{N_{\text{est}}} \frac{a_{ij}}{n} \log \frac{a_{ij}}{\sum_{j=1}^{N_{\text{est}}} a_{ij}} \\ H(R_{\text{est}}) &= -\sum_{j=1}^{N_{\text{est}}} \frac{\sum_{i=1}^{N_{\text{true}}} a_{ij}}{N_{\text{true}}} \log \frac{\sum_{j=1}^{N_{\text{true}}} a_{ij}}{N_{\text{true}}}. \end{aligned}$$

V-measure $\mathcal{V}$ is then calculated as the harmonic mean of homogeneity and completeness:

$$\mathcal{V} = \frac{\mathcal{U} \times \mathcal{W}}{\mathcal{U} + \mathcal{W}}.$$

The consideration of V-measure is useful since it takes into account both the number and locations of changepoints, and the corresponding variables which are affected. The measure is increasingly being used within the changepoint literature, see for example Li et al. (2014).

Use of these measures provides a systematic measure of the quality of the segmentations estimated by the method. In the next section we detail the seven different scenarios examined and summarise the results of application of our procedure to each.

4.5.1 Scenario Details and Results

Details of the six scenarios considered are given below. For each scenario we present an example time series, and illustrate the different segments under the subset-multivariate changepoint model (a different colour indicates a different segment). The changepoint locations are also highlighted: a red line indicates a change in mean, blue is a change in variance and green is a change in both mean and variance.

The SMOP algorithm is applied to each of the seven scenarios. In each case, we use penalty values of $\alpha = 20$ and $\beta = 40$, as these values demonstrated promising results in initial testing. Other choices of α and β are equally valid. Across all scenarios, we set the minimum distance between two consecutive changepoints to be two time-points.

To illustrate the advantage of using SMOP to detect subset-multivariate changepoints, we also consider the application of a repeated-univariate approach and a fully-multivariate approach. For the repeated-univariate approach, we apply the univariate detection method PELT (Killick et al., 2012) independently to each variable in a series. For the fully-multivariate approach, we apply the E-Divisive method of Matteson and James (2014). Chapter 2 discusses both PELT and E-Divisive in more detail.

For univariate PELT, we set the penalty to be our variable-specific penalty $\alpha + \frac{1}{p}\beta$, where p is the number of variables in the series. This particular penalty is chosen to be comparable with the penalisation within SMOP. For E-Divisive, the minimum distance between any two changepoints is set to two. Otherwise, all parameters for both methods are set to their default values. Both methods are implemented using `R` (R Development Core Team, 2011). PELT is implemented using the `changepoint` package (Killick et al., 2015) and E-Divisive is implemented using the `ecp` package (James and Matteson, 2014).

For each simulation scenario, we record the average number of estimated changepoints and corresponding affected variable subsets, together with the average V-measure of the resulting segmentations. These results are displayed in Table 4.5.1 for each model. The values in parentheses denote the standard errors of the corresponding averages.

Below we describe each scenario in turn, also providing a brief discussion of the results which we obtain after applying SMOP, the repeated-PELT approach and the fully-multivariate E-Divisive method.

Scenario 1: Univariate Series A univariate series (i.e. $p = 1$) with a single change in mean at the mid-point of the series, see Figure 4.5.1. The data for this scenario is

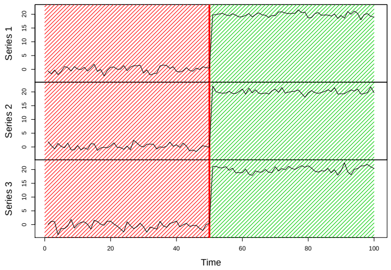


Figure 4.5.2: An example of a replication of the data from Scenario 2.

simulated using the following model:

$$\boldsymbol{X}_{1:50} \sim \mathcal{N}(0,1), \qquad \boldsymbol{X}_{51:100} \sim \mathcal{N}(20,1).$$

The scenario is included to show that SMOP can be applied in a univariate context, and in this case the method works in the same manner as PELT (Killick et al., 2012). Note Table 4.5.1 where both SMOP and PELT produce the correct segmentation of the series for all replications.

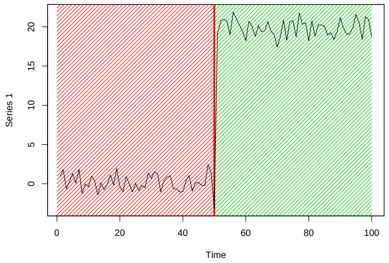


Figure 4.5.1: An example of a replication of the data from Scenario 1.

Scenario 2: Fully-Multivariate Series A single change which occurs in all variables at the same time, as in Figure 4.5.2. The data are simulated using:

$$\boldsymbol{X}_{1:50} \sim \mathcal{N}\left(\begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}\right), \qquad \boldsymbol{X}_{51:100} \sim \mathcal{N}\left(\begin{bmatrix} 20 \\ 20 \\ 20 \end{bmatrix}, \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}\right).$$

The results of this scenario (Table 4.5.1) demonstrate that SMOP performs as expected for the traditional fully-multivariate changepoint scenario, with the segmentations produced by the method being comparable to those of the fully-multivariate

E-Divisive method.

Scenario 3: Changes with different affected variable subsets Changes in both mean and variance (separately and together) which occur with differing affected variable subsets. In this case the series consists of $n = 500$ observations from $p = 2$ variables. Figure 4.5.3 shows an example series for this scenario. The model is given by:

$$\boldsymbol{X}_{1:100} \sim \mathcal{N}\left(\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}\right), \qquad \boldsymbol{X}_{101:150} \sim \mathcal{N}\left(\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 100 & 0 \\ 0 & 1 \end{bmatrix}\right),$$

$$\boldsymbol{X}_{151:200} \sim \mathcal{N}\left(\begin{bmatrix} 0 \\ 20 \end{bmatrix}, \begin{bmatrix} 100 & 0 \\ 0 & 1 \end{bmatrix}\right), \qquad \boldsymbol{X}_{201:300} \sim \mathcal{N}\left(\begin{bmatrix} 20 \\ 20 \end{bmatrix}, \begin{bmatrix} 100 & 0 \\ 0 & 1 \end{bmatrix}\right),$$

$$\boldsymbol{X}_{301:320} \sim \mathcal{N}\left(\begin{bmatrix} 20 \\ 20 \end{bmatrix}, \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}\right), \qquad \boldsymbol{X}_{321:420} \sim \mathcal{N}\left(\begin{bmatrix} 20 \\ 20 \end{bmatrix}, \begin{bmatrix} 1 & 0 \\ 0 & 100 \end{bmatrix}\right),$$

$$\boldsymbol{X}_{421:500} \sim \mathcal{N}\left(\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 100 & 0 \\ 0 & 1 \end{bmatrix}\right).$$

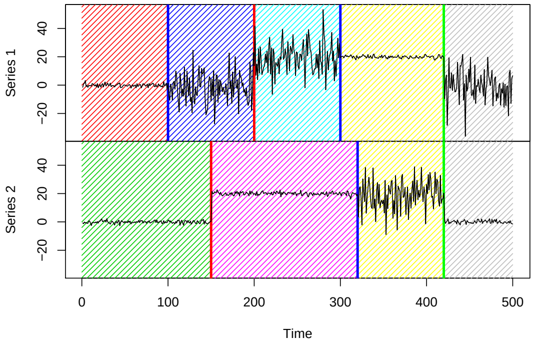


Figure 4.5.3: An example of a replication of the data from Scenario 3.

This scenario investigates the performance of SMOP for series where subset-multivariate changes are present, with differing affected variable subsets for each change. Table 4.5.1 shows that SMOP provides an excellent segmentation across all replications. This highlights the additional benefit of the subset-multivariate approach adopted by SMOP, which not only detects univariate and fully-multivariate changepoints, but also allows for changes occurring in subsets of the variables within the series. Conversely, E-Divisive performs poorly due to its assumption that all changes occur in all variables, and repeated-univariate PELT has reduced accuracy due to the lack of multivariate consideration. This is because repeated-univariate PELT only iteratively applies PELT to single variables independently, and hence does not contain a penalisation component which takes into account multiple variables simultaneously.

Scenario 4: Changes of different magnitude This scenario considers series containing three variables. In the first variable, no change occurs. In the second and third variables, single changes in variance occur with relatively small and relatively large magnitudes, respectively. An example of this scenario is given in Figure 4.5.4. Specifically, the data are simulated using:

$$\boldsymbol{X}_{1:50} \sim \mathcal{N}\left(\begin{bmatrix}0\\0\\0\end{bmatrix}, \begin{bmatrix}1&0&0\\0&1&0\\0&0&1\end{bmatrix}\right), \quad \boldsymbol{X}_{51:100} \sim \mathcal{N}\left(\begin{bmatrix}0\\0\\0\end{bmatrix}, \begin{bmatrix}1&0&0\\0&15&0\\0&0&100\end{bmatrix}\right).$$

This scenario is considered to highlight the advantage of using SMOP to detect subset-multivariate changepoints over the repeated application of univariate methods or the application of a fully-multivariate method.

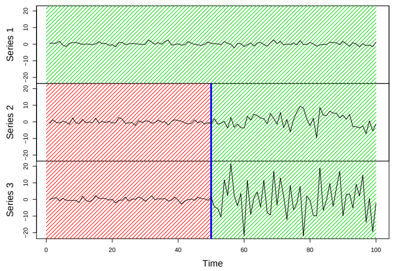
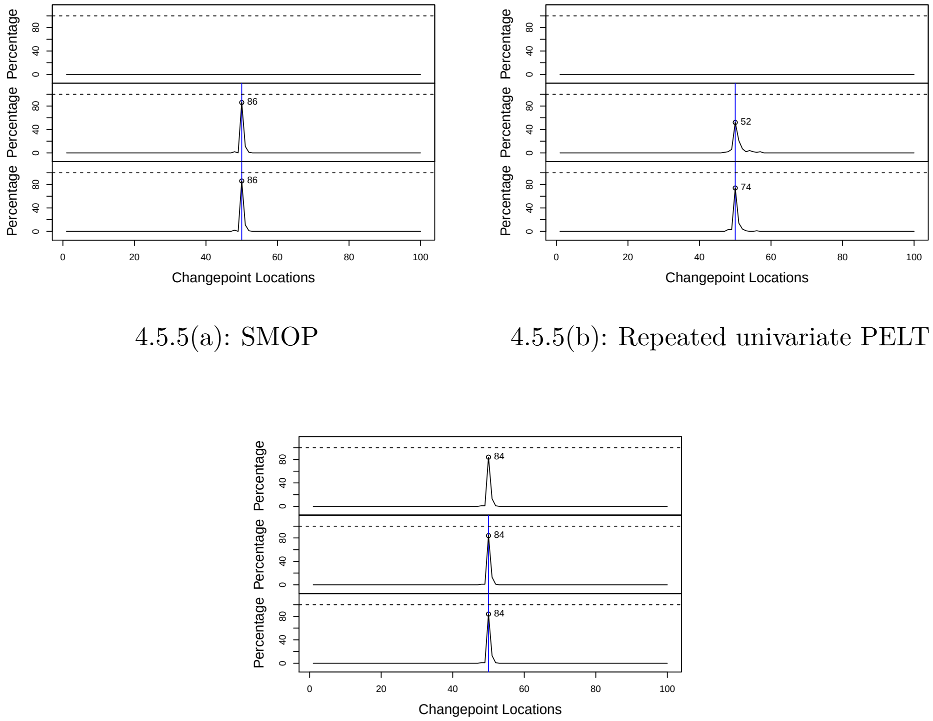


Figure 4.5.4: An example of a replication of the data from Scenario 4.

As expected, Table 4.5.1 and Figure 4.5.5 shows that SMOP provides the best segmentations on average out of the three methods. Repeated application of PELT generally over-estimates the number and changepoints and has less certainty in their locations, because such an approach is unable to utilise the multivariate nature of the changes. In particular, there is less certainty for the more subtle change in variable 2. However, this approach does not detect spurious changes in variable 1, where no change in occurring. This is not true for the fully-multivariate E-Divisive method, which estimates many such spurious changes due to its assumption of changes occurring in all variables. In other words, whilst this approach does capitalise on multivariate structure, it can lead to poor segmentations, particularly in scenarios where only



4.5.5(a): SMOP

4.5.5(b): Repeated univariate PELT

4.5.5(c): Fully-Multivariate PELT

Figure 4.5.5: Figures showing the frequency of changepoints estimated at each time-point by three different methods applied to Scenario 4.

a small number of the variables are changing. SMOP is able to harness multivariate power without a fully-multivariate assumption.

Scenario 5: Changes in different properties at a single time-point This scenario is simulated using:

$$\boldsymbol{X}_{1:50} \sim \mathcal{N}\left(\begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}\right), \qquad \boldsymbol{X}_{51:100} \sim \mathcal{N}\left(\begin{bmatrix} 20 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 100 \end{bmatrix}\right).$$

This case is considered to investigate the situation where different properties are changing in different variables.

The superior performance of SMOP in this scenario, exhibited in Table 4.5.1, highlights its ability to detect changes occurring in multiple different properties in different variables at the same time. Such a feature is especially useful for practical situations where the variables are related but may react differently to changes.

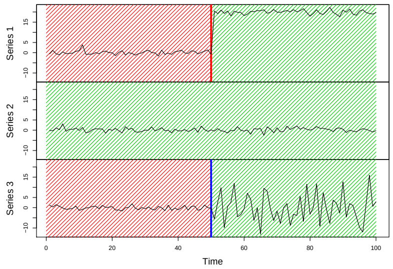


Figure 4.5.6: An example of a replication of the data from Scenario 5.

Scenario 6: Variables with differing distributional forms Here we consider a situation where the variables within the series have different distributional forms. In this case, two of the variables (1 and 3) follow a Normal distribution, and the second variable follows a Gamma(k, θ) distribution. This is reflected in the cost function used. The data for this model is simulated as follows:

$$X^1_{1:30} \sim \mathcal{N}(0,1), \qquad X^1_{31:100} \sim \mathcal{N}(10,20),$$
$$X^2_{1:30} \sim \text{Gamma}(1,1), \quad X^2_{31:70} \sim \text{Gamma}(10,1), \quad X^2_{71:100} \sim \text{Gamma}(1,1),$$
$$X^3_{1:70} \sim \mathcal{N}(0,1), \qquad X^3_{71:100} \sim \mathcal{N}(10,20).$$

Figure 4.5.7 illustrates a realisation of a time series produced under this scenario. Note that, following Chen and Gupta (2000), we fix the scale parameter of the Gamma distributions, denoted by θ, as $\theta = 1$ as this is necessary to perform changepoint detection for a Gamma distribution. Further, we note that the mean and variance of a Gamma distribution are given by $k\theta$ and $k\theta^2$, respectively. Hence, since both the mean and variance terms contain the shape parameter k, then any distributional changes in a Gamma distribution must be in both

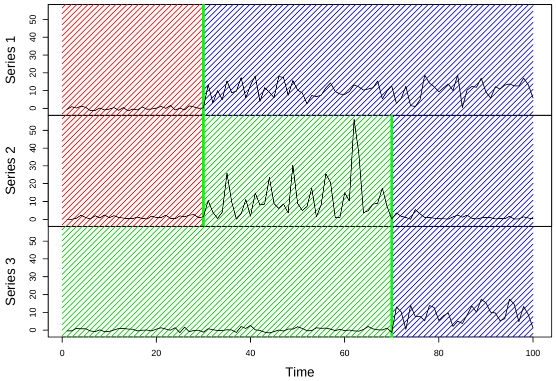


Figure 4.5.7: An example of a replication of the data from Scenario 6.

mean and variance.

The results for this scenario in Table 4.5.1 illustrate that whilst SMOP is able to perform reasonably well in this scenario, other methods cannot cope so easily.

The application of SMOP in these six scenarios demonstrates that it can be applied to a wide range of situations, and is not limited to any particular distribution. The results also reiterate the advantage of using SMOP for detecting subset-multivariate changepoints over application of repeated-univariate or fully-multivariate methods.

Metric	Scenario	SMOP	PELT	E-Divisive
Average V-Measure	1	1 (0)	1 (0)	0.99 (0.00457)
	2	1 (0)	1 (0)	0.988 (0.00503)
	3	0.996 (0.000199)	0.993 (0.000533)	0.762 (0.000532)
	4	0.991 (0.00231)	0.893 (0.0103)	0.464 (0.00302)
	5	1 (0)	0.944 (0.00842)	0.469 (0.00297)
	6	0.919 (0.0117)	0.912 (0.00719)	0.565 (0.000883)
Average Number of Changepoints	1	1 (0)	1 (0)	1.06 (0.00278)
	2	1 (0)	1 (0)	1.09 (0.00404)
	3	6 (0)	6.07 (0.00256)	5.01 (0.00522)
	4	1 (0)	1.51 (0.00502)	1.06 (0.00239)
	5	1 (0)	1.31 (0.00465)	1.12 (0.00383)
	6	2.01 (0.001)	2.93 (0.00807)	2.02 (0.002)

Table 4.5.1: The average V-measure of the segmentations and the average number of changepoints estimated by SMOP for each model. The values in parentheses denote the standard error of the corresponding average V-measure and average number of changepoints.

Given the positive results of SMOP demonstrated in this simulation study, its application is now considered to a dataset consisting of annual river flows to search for any possible changes in these flows.

4.6 Analysis of Quebec River Flows

The SMOP algorithm is now applied to a dataset containing the annual January to June steamflow amounts for four rivers in Quebec (Baleine, Churchill Falls, Manicouagan and Romaine) from 1972 to 1994. The flow measurements have been recorded in litres per kilometre-squared per second (L/km^2s). This dataset has previously been analysed by Perreault et al. (2000) and was originally published by the Centre d'Expertise Hydrique Quebec. The dataset has been made available in the `bcp` package (Erdman and Emerson, 2007), from which the data has been obtained. A plot of this data is shown in Figure **??**.

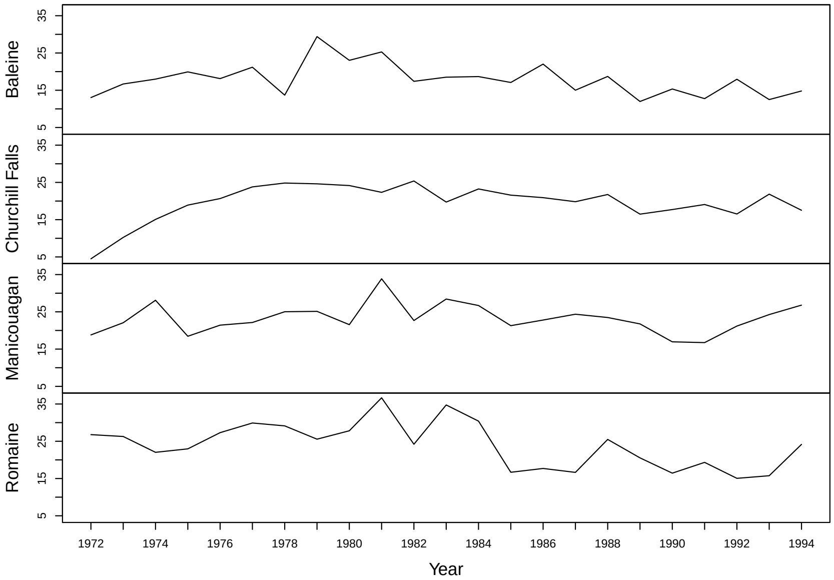


Figure 4.6.1: The annual January to June streamflow amounts for four rivers in Quebec from 1972 to 1994, measured in $L/(km^2s)$.

Interest lies in detecting changes in the streamflow of the rivers. Whilst Perreault et al. (2000) search only for shifts in the mean level, visual inspection of the data suggests that changes may be occurring in the mean and/or variance of the flow. Therefore, we consider changes in both properties. Inspection of the series for Churchill Falls may lead to the interpretation that it could be non-stationary near the beginning. If this is believed to be the case, then a non-stationary analysis of this univariate series could be performed, for example using the Locally Stationary Wavelet process (see Nason et al. (2000) for more details). The low-frequency components could then be filtered out to remove this behaviour and leave the information regarding the mean and variance relatively unaffected. However, in this instance we take the view that this apparent behaviour is simply due to the stochastic nature of the observations, and that the series will be segmented appropriately by a changepoint detection procedure.

Since it is feasible that some rivers may be affected by a change whilst others may not, it is prudent to search for subset-multivariate (rather than strictly fully-multivariate) changes. Therefore, the SMOP algorithm is applied to the data in an effort to detect such changes. To draw further comparisons with the repeated-univariate and fully-multivariate approaches, we apply the univariate PELT algorithm independently to each channel, as well as performing fully-multivariate PELT on the series. For each of the three methods we use a cost function which assumes a Normal likelihood with changes occurring in both mean and variance. For SMOP, we set penalty values $\alpha = 2\log n$ and $\beta = 2\log p \log n$. For these values of α and β, repeated-univariate PELT is applied with a variable-specific penalty of $\alpha + \frac{1}{p}\beta$, and fully-multivariate PELT is applied with penalty $p\alpha + \beta$. These penalty choices are made for similar reasons to those discussed in Section 4.5.

The results of applying SMOP, repeated-univariate PELT and fully-multivariate PELT to these Quebec river flows are presented in Figures 4.6.2, 4.6.3(a) and 4.6.3(b) respectively.

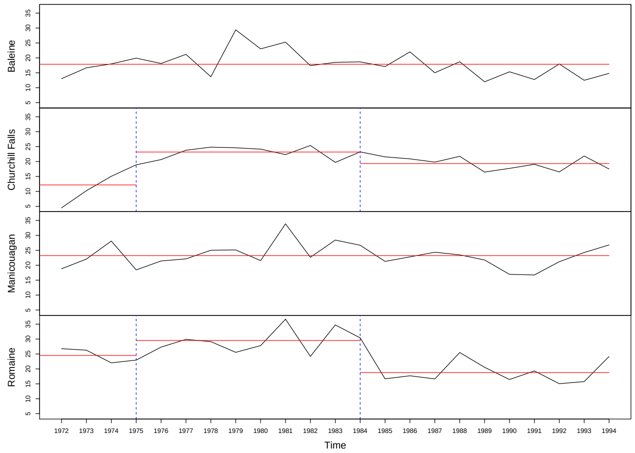
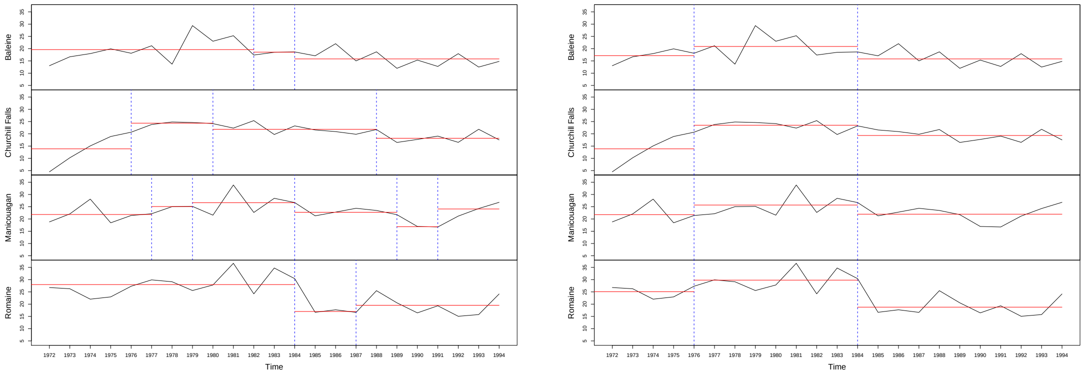


Figure 4.6.2: The results of applying SMOP to the Quebec river flows. The blue vertical lines represent changepoint locations, and the red horizontal lines represent the corresponding means of those segments.

We see from Figure 4.6.2 that SMOP estimates two changepoints in the series, at the years 1975 and 1984. These two changes affect Churchill Falls and Romaine, and



4.6.3(a): Repeated-univariate PELT results.

4.6.3(b): Fully-multivariate PELT results.

Figure 4.6.3: The results of applying repeated-univariate PELT and fully-multivariate PELT to the Quebec river flows. The blue vertical lines represent changepoint locations, and the red horizontal lines represent the corresponding means of those segments.

no changepoints are estimated in the river flows of Baleine and Manicouagan. We note that the detected locations correspond to the findings of Perreault et al. (2000), who search for a single changepoint and estimate one at 1984. The multiple changepoint approach of SMOP allows the detection of the additional changepoint.

Comparatively, as can be seen in Figure 4.6.3(a), repeated-univariate PELT also detects a change at 1984, but it detects the change in Baleine, Manicouagan and Romaine, and not Churchill Falls. In addition, the method does not detect a change at 1975 in Churchill Falls or Romaine, and instead detects additional changepoints at varying locations in the flows of the four rivers. These differing locations of changes in the rivers compared to those detected by SMOP is due to the lack of a multivariate consideration, and so multivariate power cannot be harnessed across the four series. Hence, the changes are detected independently.

Similar to SMOP, fully-multivariate PELT detects a changepoint at 1984, but due to the fully-multivariate assumption the change is detected across all rivers. A changepoint is also detected at 1976 across all rivers. This is near to the 1975 changepoint detected by SMOP, but has likely been placed slightly different by fully-multivariate PELT due to the necessity of estimating the changepoints in all variables.

Therefore, the results of performing SMOP, repeated-univariate PELT and fully-multivariate PELT reflect the results of Scenario 5 from the simulation study in Section 4.5. Repeated-univariate PELT seems to overestimate the number of changepoints (which can lead to poor estimation of the true change locations), and fully-multivariate PELT generally estimates the correct locations but overestimates the number of affected variables (which, if severe, could begin to affect the location estimates).

Given the positive results of SMOP in this applied context, the next section gives consideration to techniques which have the potential to reduce the computational cost of the procedure.

4.7 Pruning Changepoint Vectors

While the simulation study in Section 4.5 and the practical application to Quebec River data in Section 4.6 demonstrates the good performance of SMOP in terms of accuracy, the method is limited by the fact that its computational cost is non-polynomial. Specifically, as discussed in Section 4.4, for p-variate series (with $p > 1$) of length n SMOP has a computational complexity of $\mathcal{O}(pn^{2p})$. Therefore, it would be desirable if we could reduce the computational cost of the SMOP algorithm without sacrificing its exactness. To this end, there are two possible avenues of exploration:

1. Utilise pruning techniques which remove only the changepoint vectors that are guaranteed to not lie in the optimal solution under the subset-multivariate changepoint model. This is the approach taken by Killick et al. (2012) in the univariate setting. Here the search remains exact.

2. Use approximation techniques which reduce the amount of changepoint vectors considered by the algorithm. These are likely to result in a significant improvement in speed, but at the expense of the search no longer being exact.

In this section we focus on the former and postpone treatment of the latter to Chapter 5. Our aim is to utilise the concept of inequality-based pruning introduced by Killick et al. (2012) in an attempt to reduce the number of changepoint vectors required to be considered within the calculations of the method, whilst still retaining the optimality of the final set of changepoint locations and affected variables subsets produced.

We propose two types of inequality-based pruning in an effort to achieve this: retrospective pruning, which prunes changepoint vectors which have been considered previously but no longer need to be considered for future time-points $t > \tau^*$; and subset pruning, which prunes the changepoint vectors which do not need to be considered at the current time-point τ^* being investigated.

4.7.1 Retrospective Pruning

As discussed in Section 4.3.2, a key assumption within SMOP is that the addition of a changepoint into a model will reduce the cost of the model. In order to make this assumption more formal, we generalise our notation so that $\boldsymbol{X}_{c_r:c_t}$ denotes the sequence of multivariate data between the changepoint vectors c_r and c_t, including c_t but *not* including c_r. The cost for the multivariate data segment $\boldsymbol{X}_{c_r:c_t}$ is then defined by

$$cost(\boldsymbol{X}_{c_r:c_t}, \boldsymbol{c}) = \sum_{j=1}^{p} \left[\mathbb{I}(c_r^j \neq c_t^j) \mathcal{D}_j(X^j_{(c_r^j+1):c_t^j}) \right]. \tag{4.7.1}$$

Henceforth, for ease of notation, we will drop the dependence of $cost(\cdot)$ on $\boldsymbol{c}$, although it is obviously still implicit.

For the time-points $u < v < w$, suppose we have the three changepoint vectors $c_u \in \bar{C}_u$, $c_v \in \bar{C}_v$ and $c_w \in \bar{C}_w$ such that $c_u^j \leq c_v^j \leq c_w^j$ for each $j \in [1, p]$, and $c_u^j < c_v^j$ and $c_v^j < c_w^j$ for at least one $j \in [1, p]$. Analogous to Killick et al. (2012) in the univariate setting, we assume that there exists a constant K such that for all c_u, c_v and c_w as described we have

$$cost(\boldsymbol{X}_{c_u:c_v}) + cost(\boldsymbol{X}_{c_v:c_w}) + K \leq cost(\boldsymbol{X}_{c_u:c_w}). \tag{4.7.2}$$

We wish to establish whether it is possible to identify circumstances within which elements of $\bar{C}_\tau$ can be 'pruned' from consideration when finding the optimal last changepoint vector prior to some changepoint vector c_{τ^*}, for a given τ^*. Indeed, such circumstances exists and this is demonstrated in Proposition 4.7.1.

Proposition 4.7.1. *Suppose that assumption (4.7.2) holds and that there exists another constant k such that*

$$k = K - (\alpha + \beta)p. \tag{4.7.3}$$

Suppose further that

$$F(c_u) + cost(\boldsymbol{X}_{c_u:c_v}) + k \geq F(c_v) \tag{4.7.4}$$

holds for some $c_u \in \bar{C}_u$ and $c_v \in \bar{C}_v$ for time-points $u < v$ with $c_u^j \leq c_v^j$ for all j. Then at a changepoint vector c_w for some future time w (such that $c_w^j \geq c_v^j \geq c_u^j\ \forall\ j$), c_u can never be the optimal last changepoint vector prior to c_w.

Proof. See Appendix A.2 for a full proof. □

Proposition 4.7.1 implies that if equation (4.7.4) holds, then for some changepoint vector c_w (as described), the best segmentation with the most recent changepoint vector prior to c_w occurring at c_v will be better than any segmentation that has its most recent changepoint vector (prior to c_w) at c_u.

Many commonly used cost functions will satisfy assumption (4.7.2). For example, if the cost function is the negative log-likelihood, then we can take $K = 0$. To make use of calculations already performed in the SMOP algorithm, in practice we prune the c_u which satisfy the following equivalent condition:

$$\begin{aligned} F(c_u) + cost(\boldsymbol{X}_{c_u:c_v}) + \alpha \sum_{j=1}^{p} \mathbb{I}(c_u^j \neq c_v^j) + m(c_u, c_v)\beta + k \\ \geq F(c_v) + \alpha \sum_{j=1}^{p} \mathbb{I}(c_u^j \neq c_v^j) + m(c_u, c_v)\beta. \end{aligned} \tag{4.7.5}$$

Such a pruning condition is important as it allows certain candidate changepoint vectors to be discarded, thereby removing computations which are not required in order to obtain the final set of optimal changepoint vectors. Since this pruning removes changepoint vectors which have previously been considered, then we refer to this type of pruning as *retrospective pruning*.

4.7.2 Subset Pruning

We have seen how retrospective pruning can be used to remove previous changepoint vectors from future considerations. However, supposing we are at some current time-point τ^* within the algorithm, this method of pruning does not prune any of the $c_{\tau^*} \in \bar{C}_{\tau^*}$ which each have to be considered at τ^*. Pruning these vectors would reduce the amount of vectors $c_{\tau^*} \in \bar{C}_{\tau^*}$ for which $h_{c_{\tau^*}}(c)$ has to be calculated for each $c \in C_{\tau^*-1}(c_{\tau^*})$. Within this section we introduce further theory which allows for the pruning of such vectors at each time-point τ^*, which we refer to herein as *subset pruning.*

Before continuing, we define some new notation in order to accommodate this theory. We use $f_j(t)$ to denote the minimum cost from time 0 up to time t in variable j, including the α penalties but not the β penalties. We exclude these because $f_j(t)$ represents a univariate cost, whereas β represents a multivariate penalty. Also, recall that for some changepoint vector $c \in C_n$, $M(c)$ is the number of changepoint locations occurring in any variable up to and including those in c. Hence, for some changepoint vector $(t_1, t_2, \ldots, t_p)$, we can decompose $F(\cdot)$ as follows:

$$F\Big((t_1, t_2, \ldots, t_p)\Big) = \sum_{j=1}^{p} f_j(t_j) + \beta M\Big((t_1, t_2, \ldots, t_p)\Big).$$

Further, for a given $J \in \{1, \ldots, p\}$, we use $\bar{C}^J_{\tau^*}$ to denote the distinct subsets of $\bar{C}_{\tau^*}$ such that $\bar{C}^J_{\tau^*}$ contains only the $c_{\tau^*} \in \bar{C}_{\tau^*}$ which have J variables changing at time τ^*, so that $\sum_{j=1}^{p} \mathbb{I}(c^j_{\tau^*} = \tau^*) = J$. This can be expressed by

$$\bar{C}^J_{\tau^*} = \left\{ c_{\tau^*} \in \bar{C}_{\tau^*} : \sum_{j=1}^{p} \mathbb{I}(c^j_{\tau^*} = \tau^*) = J \right\}. \tag{4.7.6}$$

Note that $\bar{C}^p_{\tau^*} = \{(\tau^*, \tau^*, \ldots, \tau^*)\}$. For ease of notation, we define P to be the set of all variables, so that $P = \{1, \ldots, p\}$.

The motivation behind subset pruning is the consideration of the following scenario. Suppose that we have some p-variate series $\boldsymbol{X}$ of length n, time-points w and τ^* such that $\tau^* < w$, and some $c_w \in \bar{C}_w$. Suppose further that we make the assumption that the minimum cost to c_w from the changepoint vector $(\tau^*, \tau^*, \ldots, \tau^*)$ is lower than the minimum cost from all changepoint vectors $c_J \in \bar{C}^J_{\tau^*}$, for some $J \in P$ with $J < p$. Given this, our aim is to determine whether or not the minimum cost from

$(\tau^*, \tau^*, \ldots, \tau^*)$ to c_w is lower that the minimum cost from all $c_i \in \bar{C}^i_{\tau^*}$, for $i < J$, to c_w. If such a property holds true, then this would allow for the pruning of different subsets of affected variables, depending on the number of variables they contain which are changing at τ^*.

We will see in the following proposition that this characteristic does indeed hold under certain conditions. Before examining this result, it is necessary to introduce some further notation. For a given time-point τ^* and changepoint vector c_{τ^*}, define $\mathcal{P}_{\tau^*}(c_{\tau^*})$ to be the set of variable indices of c_{τ^*} such that $c^j_{\tau^*} = \tau^*$, so that $|\mathcal{P}_{\tau^*}(c_J)| = J$ for each $c_J \in \bar{C}^J_{\tau^*}$. That is,

$$\mathcal{P}_{\tau^*}(c_{\tau^*}) = \left\{ j \in P : c^j_{\tau^*} = \tau^* \right\}. \tag{4.7.7}$$

Finally, for a given $c_{\tau^*} \in \bar{C}^{J^*}_{\tau^*}$, for $J < J^*$ define the following set:

$$E^J_{\tau^*}(c_{\tau^*}) = \left\{ c \in \bar{C}^J_{\tau^*} : c^j \leq c^j_{\tau^*} \ \ \forall \ j \in P \right\}, \tag{4.7.8}$$

so that $E^J_{\tau^*}(c_{\tau^*})$ is the set of previous time-point vectors which are 'viable' for being changepoint vectors prior to c_{τ^*}. Proposition 4.7.2 establishes that, under certain conditions regarding the changepoint vectors with one variable changing at some time-point τ^*, then we can prune the changepoint vectors which have i variables changing at τ^*.

Proposition 4.7.2. *Suppose that for some $J \in \{1, \ldots, p\}$ and each $c_J \in \bar{C}^J_{\tau^*}$, we have for every $c_{J-1} \in \left\{ E^{J-1}_{\tau^*}(c_J) \ : c^j_{J-1} = c^j_J \quad \forall \ j \in P \setminus \mathcal{P}_{\tau^*}(c_J) \right\}$ that*

$$h_{c_w}(c_J) < h_{c_w}(c_{J-1}) \tag{4.7.9}$$

for some future vector $c_w \in \bar{C}_w$, where $w > \tau^*$.

Suppose further that we have changepoint vectors $\{c_{J-1,j^*_1}, c_{J-1,j^*_2}, \ldots, c_{J-1,j^*_i}\} \in E^{J-1}_{\tau^*}(c_J)$ such that for each $x = 1, \ldots, i$, we have $c^{j^*_x}_{J-1,j^*_x} = t_{j^*_x}$ and $c^{j^*_x}_J = \tau^*$ (with $t_{j^*_x} < \tau^*$), and $c^j_{J-1,j^*_x} = c^j_J$ for all $j \in \{P \setminus \mathcal{P}_{\tau^*}(c_J)\}$.

Then if it holds that $(i-1)M(c_J) \geq \sum_{x=1}^{i} M(c_{J-1,j_x^*})$, *we have*

$$h_{c_w}(c_J) < h_{c_w}(c_{J-i}) \tag{4.7.10}$$

for every $c_{J-i} \in \{E_{\tau^*}^{J-i}(c_J) : c_{J-i}^{j} = c_J^{j} \ \forall \ j \in P \setminus \mathcal{P}_{\tau^*}(c_J)\}$, $i = 2, \ldots, J-1$.

Proof. See Appendix A.3 for a full proof. □

Proposition 4.7.2 implies that we do not need to calculate any of the $h_{c_w}(c_{J-i})$ for any c_{J-i}. Hence, these c_{J-i} can be 'pruned' from our considerations for c_w. Otherwise, it is not necessarily true that $h_{c_w}(c_J) < h_{c_w}(c_{J-i})$, and so we are not able to use such an inequality for pruning purposes.

4.7.3 Practical Applicability of Pruning

The practical applicability of the pruning techniques presented in Sections 4.7.1 and 4.7.2 is not as straightforward as it may first appear. In fact, as we establish below, the computational complexity introduced by the implementation of pruning outweighs the benefits provided.

Retrospective Pruning While retrospective pruning allows for a potential reduction in the number of previous changepoint vectors to be considered at each iteration of the algorithm, both additional calculations and additional storage are required to perform the pruning in practice. The cost of these additional calculations and storage outweigh the benefits of retrospective pruning.

This effect is due to two main factors. Firstly, for the storage costs, practically implementing the retrospective pruning requires the creation of a boolean matrix which holds the information about which previous changepoint vectors are pruned for each possible changepoint vector. This matrix requires $\mathcal{O}\Big((n^p+1)\times(n^p+1)\Big) = \mathcal{O}(n^{2p})$ storage, which is potentially much larger than the $\mathcal{O}\Big((p+n)n^p\Big)$ storage required without pruning.

Secondly, for the computation costs, for every iteration of SMOP (that is, each $c_{\tau^*} \in \bar{C}_{\tau^*}$ for each $\tau^* \in \{1, \ldots, n\}$) each changepoint vector being considered has different 'valid' prior changepoint vectors. Therefore, when using retrospective pruning, in addition to checking which prior vectors are valid it is necessary to perform an additional check to determine which of these vectors have been pruned previously. Such checking requires an additional $\mathcal{O}(pn^{2p-1})$ calculations. Therefore, since the computational complexity of SMOP without any pruning is $\mathcal{O}(pn^{2p})$ (for $n > p$), then even for moderate values of p the use of retrospective pruning has minimal effect on reducing the number of calculations in practice. Combined with the vastly increased storage required to prune, this implies that these additional computation and storage costs outweigh the advantages of retrospective pruning.

Subset Pruning For subset pruning to be applicable to the $c_{J-i} \in \Big\{E_{\tau^*}^{J-i}(c_J) : c_{J-i}^j = c_J^j \;\forall\; j \in P \setminus \mathcal{P}_{\tau^*}(c_J)\Big\}$, the condition that

$$(i-1)M(c_J) \geq \sum_{x=1}^{i} M(c_{J-1,j_x^*})$$

is required to be true for the c_J and c_{J-1,j_x^*} as described in Section 4.7.2. In practice, this condition needs to be performed for each set of changepoint vectors $\{c_{J-1,j_1^*}, c_{J-1,j_2^*}, \ldots, c_{J-1,j_i^*}\}$ which correspond to each of the c_{J-i}. Determining this set of corresponding vectors for each c_{J-i} is itself time consuming, and this needs to be done for all c_{J-i} for each $i = 2, 3, \ldots, J-1$. This can result in a very large number of additional considerations, particularly as c_J moves towards the end of the multivariate series. In terms of the comparison itself, it is not intuitively clear as to how often pruning will be performed, and the condition is highly dependent on the c_{J-i} vector of interest (and hence the corresponding $\{c_{J-1,j_x^*}\}_{x=1}^{i}$ vectors). Due to this obscurity and the seemingly very large number of additional calculations likely required to perform subset pruning, for potentially little or no improvement, we do not implement it in the SMOP algorithm.

4.8 Concluding Remarks

We have considered the problem of detecting changes which occur in subsets of the observed variables within a multivariate time series. Our aim has been to obtain both the changepoint locations and the corresponding subsets of affected variables. To this end, we have formalised the concept of 'changepoint vectors', which encapsulate information regarding both of these entities. A novel exact search method has been proposed which obtains the optimal changepoint vectors for a given multivariate time series, via the minimisation of a penalised cost function. No other method in the changepoint detection literature currently provides such an exact search under this model. Simulation results demonstrate the advantages of using SMOP over other possible approaches to this problem, and illustrate how it can be applied to a wide range of scenarios.

Producing such optimal estimates is an NP-hard problem. In an attempt to reduce the computational complexity of the SMOP algorithm, we have tried to emulate the success of Killick et al. (2012) in the univariate context and utilize inequality-based pruning techniques to reduce the amount of changepoint vectors which need to be considered in the procedure. However, we have demonstrated that such pruning is not practically viable and so do not implement it in the algorithm. Therefore, in Chapter 5 we will focus our attention on the use of approximation techniques which allow the algorithm to consider only those changepoint vectors which are likely to be present in the optimal set of changepoint vectors.

Chapter 5

Approximate Segmentation of Multivariate Time Series

5.1 Introduction and Motivation

The Subset Multivariate Optimal Partitioning (SMOP) algorithm, proposed in Chapter 4, is a multivariate changepoint detection procedure which obtains the locations of changes and identifies the corresponding affected variable subsets. This is achieved through the optimisation of a penalised cost function using an exact search. While the exact nature of this search method is of theoretical interest, the large volume of calculations required by SMOP means that it is computationally too expensive to be used in many practical applications. This is particularly true in cases where datasets contain a large number of observations from many different variables, such as in the analysis of electroencephalograms (EEG) (Kirch et al., 2015) and the detection of DDoS attacks in network traffic data (Lung-Yut-Fong et al., 2011a). These scenarios require a method which is capable of segmenting a multivariate time series within reasonable computational time. Therefore, in this chapter we focus on introducing an approximation of the SMOP algorithm that substantially reduces the search space within the dynamic program, and seek to consider the impact which this has on the accuracy of the resulting estimates.

As examined in Chapter 2, we are not the first to use approximations when estimating the changepoint locations under the subset-multivariate model. In particular, the binary segmentation approach of Jeng et al. (2013) uses a global test statistic which only accepts contributions from variables whose variable-specific statistics exceed some threshold. Not only does this reduce the spurious influence from unaffected variables, but also shortens the computation time by removing unnecessary calculations. Similarly, Maboudou-Tchao and Hawkins (2013) reduce the search space of their dynamic program. They initially assume that any considered change affects all variables, then use post-processing hypothesis tests to identify which variables are actually affected for each estimated changepoint. The work presented in this chapter demonstrates how equivalent ideas can be used in this setting.

We propose two stages of pre-processing which allow for a substantial reduction in the size of the search space considered by SMOP. The first of these involves a reduction in the number of time-points considered as possible changepoints. This is performed by preliminarily identifying 'likely' changepoint locations in each variable, and considering only these such time-points as possible changepoint locations within the SMOP algorithm. The second stage reduces the number of affected variable subsets to be considered for each possible changepoint.

The remainder of this chapter is organised as follows. Section 5.2 provides detail on how both the number of potential changepoints and affected variable subsets to be considered within SMOP can be reduced, and presents a new version of the SMOP algorithm which includes these approximation steps. Section 5.3 summarises the results of a simulation study which illustrates the behaviour of this approximate SMOP algorithm, and compares its performance when using each of the two different approximation mechanisms for obtaining possible affected variable subsets. The scalability of the algorithm for datasets of increasing size is also investigated. The performance of this computationally tractable algorithm is also demonstrated on an acoustic sensing data set in Section 5.4. This approximate SMOP is then compared and contrasted with the original SMOP method through an application of both to the annual flow measurements of four rivers in Quebec (this dataset was first considered in Section 4.6). Finally, the possibility of incorporating additional computation-saving logic into the algorithm when certain structure is present in the changepoints is discussed, along with the benefits and shortcomings of such an implementation.

5.2 Search Space Reduction

In order to improve the practical applicability of SMOP, it is necessary to reduce the size of the search space considered. This can be achieved by considering the changepoint vectors which are likely to be optimal under the subset-multivariate model. This translates into considering only candidate changepoint locations and affected variable subsets which are in some sense plausible.

5.2.1 Reducing Possible Changepoint Locations and Affected Variables

To reduce the number of possible changepoint locations considered by the SMOP algorithm, and the number of corresponding subsets of affected variables, we aim to consider only changepoints and affected variable subsets which are likely to appear in the final segmentation provided by SMOP. To obtain these values, we apply the univariate changepoint detection method PELT (Killick et al., 2012) to each individual variable of the series. Since PELT is an exact (univariate) search method, the changepoint locations it estimates in a given variable have a good possibility of being estimated as changepoints in that variable by SMOP under the subset-multivariate changepoint model. The penalty used within PELT is set to α, the variable-specific penalty used in the multivariate penalised cost function (4.3.1). This choice is made because α represents the minimum reduction in the cost function necessary to have a chance of being detected as a changepoint in a given variable by SMOP.

More formally, suppose that PELT with penalty α has been applied to the p distinct univariate series constituting the multivariate series $\boldsymbol{X}_{1:n} = \{\boldsymbol{X}_1, \boldsymbol{X}_2, \ldots, \boldsymbol{X}_n\}$, defined in Section 4.2. For each variable j (where ‘variable’ refers to a single channel of the multivariate series), denote the set of changepoint locations estimated by PELT in that variable by $\boldsymbol{\tau}^j = (\tau_1^j, \tau_2^j, \ldots, \tau_{m_j}^j)$, where m_j represents the number of changepoints detected in j. Then the set of possible changepoint locations considered by SMOP is given by

$$\boldsymbol{\tau} = \bigcup_{j=1}^{p} \boldsymbol{\tau}^{j}.$$

Forming the set of possible changepoint locations in this manner means that the repeated application of PELT results in the removal of time-points which are unlikely to be estimated as changepoint locations by SMOP. In comparison, the original exhaustive version of SMOP considers all time-points as possible changepoint locations. Therefore, in many practical settings, the collection of possible changepoint locations $\boldsymbol{\tau}$ considered is likely to be vastly reduced in size compared to the number of locations considered by exhaustive SMOP.

The repeated application of PELT can also be used to potentially reduce the number of variables considered when forming the set of possible affected variable subsets for a given possible changepoint. As described in Proposition 5.2.1, if PELT does not detect any changepoints in variable j^* using penalty α, then no changepoints would be detected in j^* using SMOP.

Proposition 5.2.1. *Suppose PELT is performed on a variable j^* in a multivariate time series $\boldsymbol{X}_{1:n}$ using a penalty of α, where α is the variable-specific penalty used in the penalised cost function (4.3.2) in Chapter 4. Then if this results in no changepoints being detected in j^*, then no changepoints will be present in j^* in the optimal configuration of changepoints detected under the subset-multivariate changepoint model using SMOP.*

Proof. See Appendix B.1 for proof. □

Intuitively, the result of Proposition 5.2.1 holds because α is the univariate penalty in SMOP, and so if a change in a given variable does not improve the likelihood by more than α, then it will not be detected by SMOP. Since performing PELT with penalty α has a similar outcome, then we can use this fact to inform the possible changepoint locations to be considered by SMOP.

An obvious consequence of Proposition 5.2.1 is that if no changepoints are detected in some variable j^*, then j^* does not need to be considered further for the remainder of the method. In some scenarios this could lead to a great reduction in the number of variables which need to be considered in the algorithm. In particular, this is useful for high-dimensional time series which contain many 'noisy' variables that do not contribute to any change in the series.

In the next section we propose two procedures which allow for a further reduction the number of possible affected variable subsets considered for a given possible changepoint within SMOP.

5.2.2 Further Subset Reduction

To further reduce the search space of changepoint vectors considered by SMOP, we introduce two additional procedures which reduce the number of possible subsets of affected variables for each of the candidate changepoint locations in $\boldsymbol{\tau}$. These are based on a windowing argument, and are referred to as 'hard subset restriction' and 'soft subset restriction' respectively. Before considering each procedure in turn, we introduce some notation. Let $s_\tau = (s_\tau^1, s_\tau^2, \ldots, s_\tau^p)$ denote a possible subset of affected variables for a given changepoint location τ, with s_τ^j being a binary indicator denoting whether or not variable j is affected by the (potential) change occurring at τ. Let $\mathcal{S}_\tau$ denote the set of all such possible affected variable subsets for a given τ, i.e. $\mathcal{S}_\tau = \left\{s_{\tau,(1)}, s_{\tau,(2)}, \ldots, s_{\tau,(|\mathcal{S}_\tau|)}\right\}$ (where $s_{\tau,(i)}$ denotes the i^{th} element of $\mathcal{S}_\tau$).

Hard Subset Restriction

The intuition behind this procedure is that if two potential changepoints in different variables (detected using independent applications of univariate PELT) are 'close' in time, as defined by some window size w, then it is likely that these possible locations both correspond to the same underlying change. Hence, it is reasonable to assume that both variables can be classified as 'affected' for the changepoints under consideration. We therefore wish to use this information to reduce the number of possible affected variable subsets considered for the potential changepoints, thereby reducing the computation required by the method.

More formally, this procedure restricts the set of all potentially affected variable subsets $\mathcal{S}_\tau$ to a single subset for a given changepoint τ, denoted S_τ. The approach can be implemented as follows. For a given variable j^* and each corresponding possible changepoint location $\tau^{j^*} \in \boldsymbol{\tau}^{j^*}$, we specify a window around τ^{j^*}, denoted by $[\tau^{j^*} - w, \tau^{j^*} + w]$, where w is referred to as the window size. The affected variable subset for τ^{j^*} is given by $S_{\tau^{j^*}}$, with $S^{j^*}_{\tau^{j^*}} = 1$. Then, if for any $j \in \{1, \dots, p \setminus j^*\}$ there exists a $\tau^j \in \boldsymbol{\tau}^j$ such that $\tau^j \in [\tau^{j^*} - w, \tau^{j^*} + w]$, we set $S^j_{\tau^{j^*}} = 1$. Otherwise, we set $S^j_{\tau^{j^*}} = 0$. This procedure is repeated for each $\tau^{j^*} \in \boldsymbol{\tau}^{j^*}$ for all $j^* = \{1, \dots, p\}$. This is presented in algorithmic form in Algorithm 4, where $\mathbb{I}(\cdot)$ denotes the indicator function.

Algorithm 4: Hard Subset Restriction

```
Input     : A set of variables $j^* = 1, \dots, p$ corresponding to a multivariate time
            series $\boldsymbol{X}$, a set of possible changepoint locations $\boldsymbol{\tau}^{j^*}$ for each
$j^* = 1, \dots, p$,
            and a window size $w$.
Initialise: Set $\boldsymbol{\tau} = \bigcup_{j=1}^{p} \boldsymbol{\tau}^j$, and $S_\tau$ = NULL for all $\tau \in \boldsymbol{\tau}$.
begin
    for $j^* \in \{1, \dots, p\}$ do
        for $\tau^{j^*} \in \boldsymbol{\tau}^{j^*}$ do
            for $j \in \{1, \dots, p\}$ do
                Set $S^j_{\tau^{j^*}} = \mathbb{I}(\exists\ \tau^j \in \boldsymbol{\tau}^j$ s.t. $\tau^j \in [\tau^{j^*} - w, \tau^{j^*} + w])$
            Set $\mathcal{S}_{\tau^{j^*}} = \{S_{\tau^{j^*}}\}$
Output    : The set of affected variable subsets $\mathcal{S}_\tau$ for each $\tau \in \boldsymbol{\tau}$.
```

Herein, we use $hard(\boldsymbol{\tau}^1, \dots, \boldsymbol{\tau}^p, w)$ to denote the resulting set of affected variable subsets produced by applying hard subset restriction to the sets of changepoint locations $(\boldsymbol{\tau}^1, \dots, \boldsymbol{\tau}^p)$ with window size w. This procedure is referred to as ‘hard’ subset restriction due to the enforced ‘cut-off’ nature of the windowing: if a given variable does not contain a changepoint within the given window for τ^{j^*}, then it is not considered to be affected by the possible change at τ^{j^*}.

It is important to note that the choice of the window size w is context dependent.

Informally, its value can be thought of as a tolerance for the slight misestimation of multivariate changepoint locations within PELT. A larger value means that estimated changepoints across different variables that are 'close' (in time) are more likely to be treated as the same changepoint across those variables. For example, for data observed at high frequency it may be prudent to use a larger w.

In contrast to hard subset restriction, the second procedure considers additional permutations of affected variables within its restriction.

Soft Subset Restriction

Soft subset restriction allows for more than one possible affected variable subset for a given changepoint τ. The procedure works by initially performing hard subset restriction to obtain the single affected variable subset for each $\tau \in \boldsymbol{\tau}$. Denote this specific affected variable subset for a given τ by S_τ. Next, the set $J_\tau = \{j = 1, \dots, p : S_\tau^j = 0, \boldsymbol{\tau}^j \neq \emptyset\}$ is defined for each τ. Note that this excludes the variables with $\boldsymbol{\tau}^j \neq \emptyset$ since Proposition 5.2.1 demonstrates that no changepoints will be present in these variables in the optimal configuration obtained by SMOP. Then, the remaining elements of $\mathcal{S}_\tau$ for each τ are generated by fixing $s_\tau^j = 1$ for the $j \in \{1, \dots, p : S_\tau^j = 1\}$ and permuting the values of s_τ^j for all $j \in J_\tau$. Each permutation represents a different affected variable subset for τ. Hence, this gives a total of $2^{|J_\tau|}$ elements of $\mathcal{S}_\tau$ for each $\tau \in \boldsymbol{\tau}$. Algorithm 5 presents this procedure in algorithmic form. We use B_k to denote the set of all binary permutations of length k, so if $k = 2$ then $B_k = \{(0,0), (0,1), (1,0), (1,1)\}$.

Similar to hard restriction, we use $soft(\boldsymbol{\tau}^1, \dots, \boldsymbol{\tau}^p, w)$ to denote the set of affected variable subsets produced by applying soft subset restriction to $(\boldsymbol{\tau}^1, \dots, \boldsymbol{\tau}^p)$ with window size w. Since soft subset restriction considers more affected variable subsets for each $\tau \in \boldsymbol{\tau}$ than hard subset restriction, this procedure leads to a comparatively larger search space for SMOP, and hence has a relatively longer computation time. However, the advantage of this procedure is that it considers additional permutations of variables which might be affected by a given changepoint. Therefore, given that soft subset restriction is essentially a relaxation of hard subset restriction, soft restriction

Algorithm 5: Soft Subset Restriction

```
Input     : A set of variables $j^* = 1, \ldots, p$ corresponding to a multivariate time
            series $\boldsymbol{X}$, a set of possible changepoint locations $\boldsymbol{\tau}^{j^*}$ for each
$j^* = 1, \ldots, p$,
            and a window size $w$.
Initialise: Set $\boldsymbol{\tau} = \bigcup_{j=1}^{p} \boldsymbol{\tau}^j$.
begin
    Set $\{S_\tau\}_{\tau \in \boldsymbol{\tau}} = hard(\boldsymbol{\tau}^1, \ldots, \boldsymbol{\tau}^p, w)$
    for $\tau \in \boldsymbol{\tau}$ do
        Set $\mathcal{S}_\tau = \{S_\tau\}$
        Set $J_\tau = \{j = 1, \ldots, p : S_\tau^j = 0, \boldsymbol{\tau}^j \neq \emptyset\}$
        Set $J_\tau^* = \{j = 1, \ldots, p : S_\tau^j = 1\}$
        for $b \in B_{|J_\tau|}$ do
            Set $s_\tau^{J_\tau^*} = 1$
            Set $s_\tau^{J_\tau} = b$
            Set $\mathcal{S}_\tau = \{\mathcal{S}_\tau, s_\tau\}$
Output    : The set of affected variable subsets $\mathcal{S}_\tau$ for each $\tau \in \boldsymbol{\tau}$.
```

will always result in estimates that are more accurate than (or, at worst, as accurate as) those produced by hard restriction. We formalise this in Proposition 5.2.2.

Proposition 5.2.2. *For a multivariate time series $X_{1:n}$, suppose $\boldsymbol{c}^{soft}$ and $\boldsymbol{c}^{hard}$ denote the optimal configurations of changepoint vectors obtained using the approximate SMOP algorithm with soft and hard subset restriction, respectively. If the corresponding optimal costs are denoted by F^{soft} and F^{hard} respectively, then we have*

$$F^{soft} \leq F^{hard}.$$

Proof. A proof is presented in Appendix B.2. □

In other words, the segmentation produced by soft restriction always has a cost which is lower than, or the same as, the cost of the segmentation produced by hard restriction. The price of this improved accuracy is the increased computation time.

5.2.3 Approximate SMOP

We now turn to consider how the restriction techniques proposed above can be incorporated into the SMOP algorithm. We refer to this version of the algorithm as Approximate SMOP (A-SMOP), and this is presented in Algorithm 6.

Within Algorithm 6, we make use of notation which generalises some of the definitions introduced in Chapter 4. Recall that we define the changepoint vector $c_t = (c_t^1, c_t^2, \ldots, c_t^p)$ to be the vector of most recently observed changepoints in each variable, prior to and including some time-point t. We use $C_{N,\mathcal{S},t}$ to denote the set of all possible prior changepoint vectors c_t defined by the set of time-points N and the set of corresponding sets of affected variable subsets $\mathcal{S} = \{\mathcal{S}_\tau : \tau \in N\}$, up to and including the given time-point t. The natural extensions to the corresponding definitions of $\bar{C}_\tau$ and $C_\tau(c_{\tau^*})$ from Chapter 4 are also made:

$$\begin{aligned}
&\bar{C}_{N,\mathcal{S},t} = \{c_t \in C_{N,\mathcal{S},t} : \exists\ j \text{ s.t. } c_t^j = t\}\\
&C_{N,\mathcal{S},\tau}(c_{\tau^*}) = \left\{ c \in C_{N,\mathcal{S},\tau} :\ c^j \leq c_{\tau^*}^j\ \forall\ j \in [1,p] \right\}.
\end{aligned}$$

Finally, we define α-PELT$(X_{1:n})$ to be the set of changepoints detected by performing PELT with penalty α on the univariate set of observations $X_{1:n}$, including the end-point n.

The trade-off for implementing the proposed techniques within SMOP is that the final segmentation produced is no longer exact. Consequently, the algorithm is no longer guaranteed to identify the changepoint locations and corresponding subsets which are optimal for the penalised cost function. For example, it is feasible that the performance of univariate PELT on the individual channels may not yield the true changepoint locations, and hence the true locations would not be present in the search space considered by SMOP. Similarly, even if the true changepoints are output by PELT, both the hard and soft subset restriction procedures could potentially fail to produce the true affected variable subset for a given true changepoint. Therefore, this modified SMOP algorithm is approximate in nature, and hence leads to the name Approximate SMOP.

5.3 Simulation Study

We now consider the performance of A-SMOP on a range of simulated time series. This assessment is divided into two separate studies. The aim of the first study is to illustrate the differences between the characteristics of hard and soft subset restriction, and demonstrate how implementing the proposed approximations vastly improves the computation time of SMOP whilst only mildly compromising on quality. Comparisons are also drawn with leading repeated-univariate and fully-multivariate approaches. The aim of the second study is to investigate the scalability of A-SMOP with both hard and soft restriction for increasing n, p and m.

Across these two simulation studies, we consider a range of series with differing values of n, p and m. We consider 100 replicate time series for each scenario investigated. These time series are assumed to follow a multivariate Normal distribution with no cross-correlation, unless otherwise stated. The changes may occur in the mean or variance parameters, depending on the scenario. For change in mean examples, the variance is fixed as $\sigma_j^2 = 1$ for each variable j. When the variance is changing, we fix $\mu_j = 0$ for each j. The magnitude and direction of the shifts are randomly chosen so that the parameter values for the i^{th} segment can be generated as follows:

- for clearly observable changes in mean: $\mu_{j,i} = \mu_{j,i-1} \pm \mathcal{N}(2, 0.05)$;
- for less-clearly observable changes in mean: $\mu_{j,i} = \mu_{j,i-1} \pm \mathcal{N}(1.2, 0.05)$;
- for changes in variance: $\sigma_{j,i}^2 = \sigma_{j,i-1}^2 \times \mathcal{N}(5, 0.05)$, or $\sigma_{j,i}^2 = \sigma_{j,i-1}^2 / \mathcal{N}(5, 0.05)$.

Here $\mu_{j,i}$ and $\sigma_{j,i}^2$ denote the mean and variance, respectively, for the i^{th} segment of variable j. In each case we have $\mu_{j,1} = 0$ and $\sigma_{j,1}^2 = 1$. These changes are linearly spaced through the data; the variables which are affected differ for each scenario.

As in the simulation study for SMOP (Section 4.5), to assess the performance of A-SMOP we calculate the average V-measure and the average number of estimated changepoints in all of the scenarios considered. In addition, we measure the average computation time for each case. These metrics are used to provide a holistic picture of the performance of A-SMOP, with the average computation time being particularly important when assessing the scalability of A-SMOP. Unfortunately, due to the computational cost of the exact SMOP algorithm we have not been able to compare these results against the exact approach, and so instead we compare against the truth.

5.3.1 Comparison of Methodology

We consider eleven different scenarios which highlight the differences in the behaviour of A-SMOP when using hard subset restriction compared to soft subset restriction. Unless stated otherwise, each scenario has $n = 500$ observations of $p = 4$ variables containing $m = 4$ changepoints. The following two sub-sections outline the details of this study and present the corresponding results.

Simulation Study Details

The details for each scenario are as follows:

Scenarios 1 and 2 These scenarios contain clearly observable changes in the mean parameter, which affect all of the variables in Scenario 1 and only single variables in Scenario 2. These scenarios are examined to investigate the performances of hard and soft subset restrictions in the 'extremes' of the affected variable subsets. In particular, Scenario 2 represents the case where hard and soft restriction have the greatest difference in terms of required computation.

Scenarios 3 and 4 These contain less-clearly observable changes in mean, affecting three and two of the $p = 4$ variables respectively. These scenarios are designed to represent 'typical' series containing subset-multivariate changes, with differing proportions of variables affected to further investigate the difference in performance of hard and soft subset restriction.

Scenario 5 Here the changes are in variance, affecting two of the variables in the series. This scenario is considered in addition to Scenarios 3 and 4 to show that they have similar performance, therefore demonstrating that the behaviour of algorithm is (in general) not influenced by the property which is changing.

Scenario 6 This scenario is identical to Scenario 5, except the individual series exhibit serial correlation. Rather than being Normally distributed, each individual variable follows an AR(2) process given by

$$X_t = 0.9X_{t-1} - 0.3X_{t-2} + \varepsilon_t,$$

where $\varepsilon_t \sim N(0, \sigma^2)$, with the variance σ^2 being the parameter which is changing. This scenario is examined to draw comparisons with the performance of Scenario 5 and demonstrate that A-SMOP is reasonably robust to the presence of autocorrelation.

Scenarios 7–10 These scenarios all contain more subtle changes in mean which affect two of the four variables, similar to those in Scenario 4. However, each scenario has a different combination of n and m in order to demonstrate how A-SMOP performs in relation to these values, in terms of both accuracy and running times. See the captions within Figure 5.3.1 for details.

Scenario 11 This is identical to Scenario 4, except the series contains cross-correlation between the different variables. The correlation matrix stays constant throughout the series and is given by

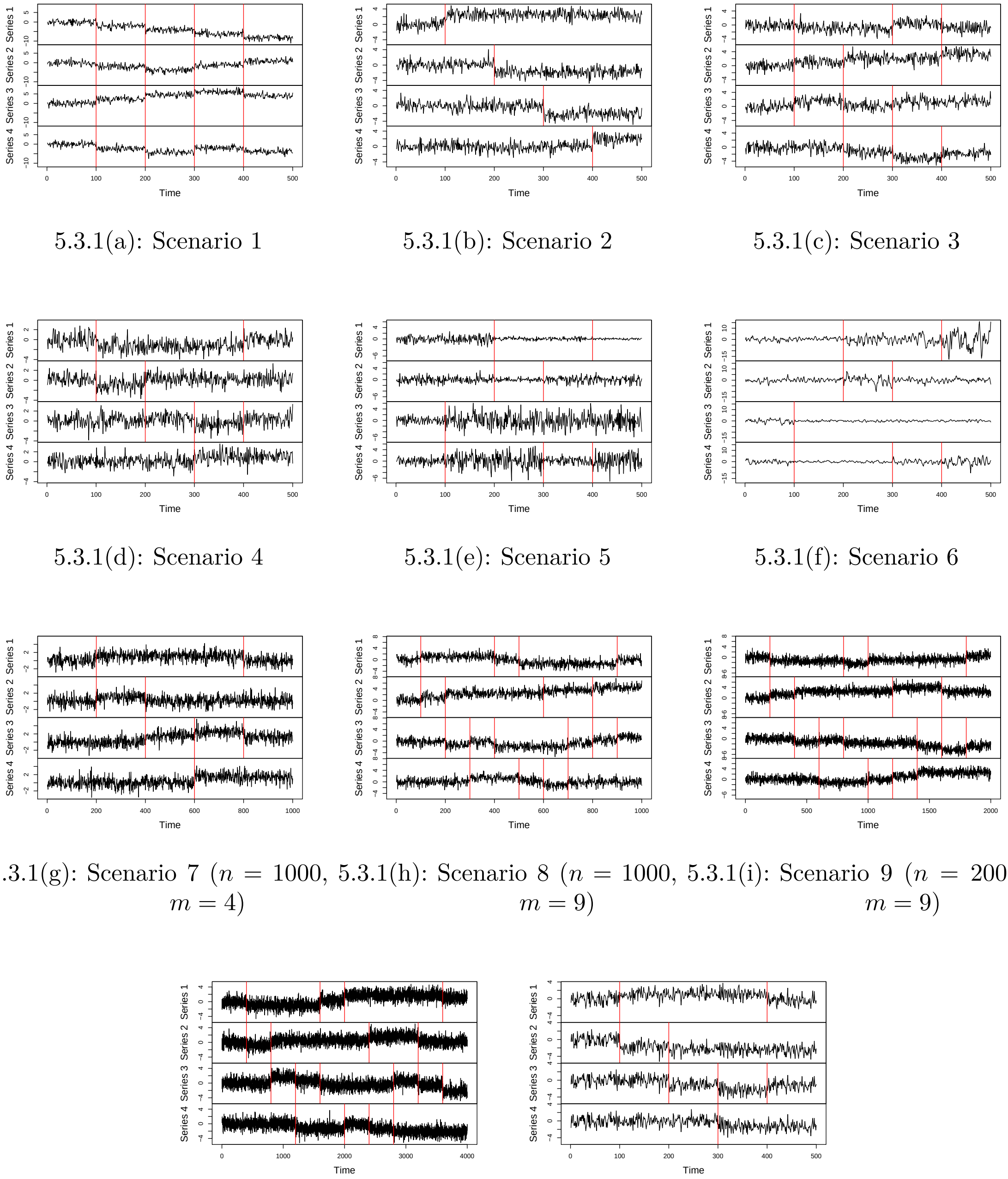
$$\begin{bmatrix} 1.0 & 0.9 & -0.9 & -0.9 \\ 0.9 & 1.0 & -0.9 & -0.9 \\ -0.9 & -0.9 & 1.0 & 0.9 \\ -0.9 & -0.9 & 0.9 & 1.0 \end{bmatrix}.$$

This scenario is investigated to highlight how A-SMOP is robust to the presence of cross-correlation, despite the assumption of independence between the different variables.

Figure 5.3.1 illustrates realisations of the time series arising from each of these scenarios. Throughout our simulations, we set the cost function as twice the negative log-likelihood for changes in mean only or variance only (depending on the scenario in consideration). We set α to be the modified Bayes Information Criterion (mBIC). If the associated univariate cost function is twice the negative log-likelihood of the observations, then the mBIC (proposed by Zhang and Siegmund (2007)) is defined as

$$\text{mBIC} := \sum_{i=1}^{m+1} \log(n_i) + (2m - 1)\log(n),$$

where n_i denotes the number of observations in the i^{th} segment. Hence, this penalty scales with the number of observations and considers the length of the segments as part of its penalisation. Both Zhang and Siegmund (2007) and Hocking et al. (2013) demonstrate good results for the mBIC in general for univariate time series.



5.3.1(a): Scenario 1

5.3.1(b): Scenario 2

5.3.1(c): Scenario 3

5.3.1(d): Scenario 4

5.3.1(e): Scenario 5

5.3.1(f): Scenario 6

5.3.1(g): Scenario 7 ($n = 1000$, $m = 4$)

5.3.1(h): Scenario 8 ($n = 1000$, $m = 9$)

5.3.1(i): Scenario 9 ($n = 2000$, $m = 9$)

5.3.1(j): Scenario 10 ($n = 4000$, $m = 9$)

5.3.1(k): Scenario 11

Figure 5.3.1: Examples of time series arising from the different scenarios considered.

Similarly, the value of β is set as $\beta = 2\log(p)\log(n)$. The usage of the $2\log(n)$ factor in this β value is based on the BIC, and the $\log(p)$ factor has been selected by applying similar logic to the number of variables. This particular value of β was selected after demonstrating good performance in initial simulation trials. Different values of β, such as those with different multiplicative constants or a factor of p instead of $\log(p)$, are equally valid. Note that the original mBIC is not an appropriate choice for β as the mBIC is a univariate penalty, and since the role of β is a multivariate penalty it is prudent that it scales with both the number of observations and the number of variables.

The exception to our choice of α is in Scenario 6. Initial testing revealed that use of the mBIC will likely to lead to additional spurious changepoints being detected. This is due to our assumption of independence between observations within the cost function when in reality serial correlation is present. Therefore, to investigate the effect of using an increased penalty to reduce these spurious changepoints, we also consider an increased value of α which demonstrated good results in testing:

$$\alpha = \sum_{i=1}^{m+1} \log(n_i) + (6m-1)\log(n).$$

This is examined in addition to the case where α is the mBIC. This case with the increased penalty is referred to as Scenario 6a. Across all scenarios, we set the minimum distance between changepoints to be 2, and the window size $w = 5$.

To draw comparisons with a repeated-univariate approach and a fully-multivariate approach, for each scenario we ran univariate PELT on each variable separately, and the fully-multivariate methods E-Divisive (Matteson and James, 2014) and E-CP3O (James and Matteson, 2015). E-Divisive and E-CP3O are both non-parametric changepoint methods. To ensure fair comparability of results, we set the penalty value used within PELT to be $\alpha + (1/p)\beta$. Such a choice means that for the case a fully-multivariate change, repeated PELT has exactly the same penalisation as A-SMOP. PELT is implemented using the `changepoint` package (Killick et al., 2015).

For E-Divisive and E-CP3O, we use the default settings with the exception of the minimum distance between changepoints, which was set to 2 for E-Divisive and to 15 for E-CP3O using the `ecp` package (James and Matteson, 2014). The difference in these minimum distances between changepoints is due to the behaviour of the different test statistics. In E-Divisive, the whole dataset is always used to formulate the test statistic, whereas in E-CP3O the minimum distance value is used in determining the amount of data utilised in the test statistic calculation. For a potential changepoint location τ and minimum changepoint distance d, E-CP3O considers only the observations corresponding to the time-points in $[\tau - d, \tau + d]$. Consequently, the minimum changepoint distance can be set as low as possible for E-Divisive, whereas the choice for E-CP3O has a more complex effect on accuracy, and its value was selected after initial testing to ensure reasonable results.

Results and Discussion

Table 5.3.1 presents the average V-measure and average number of estimated changepoints across all replications for the resulting segmentations produced by A-SMOP for both restrictions in each scenario, as well as those for PELT, E-Divisive and E-CP3O. As can be seen from the table, hard and soft subset restriction provide similar accuracy for scenarios containing clearly observable changes. However, for scenarios with less-clearly observable changes, soft restriction gives better accuracy than hard. Nevertheless, hard restriction still has reasonable performance in such cases. Such behaviour is due to the hard restriction being able to more easily identify the correct affected variable subset when changes are prominent, and less so when changes are subtle. Hence, the fact that soft restriction considers more subsets gives it an accuracy advantage in situations containing subtle changes. Table 5.3.2 shows the mean run times for each scenario. These demonstrate that computationally the soft restriction approach is somewhat more intensive than the hard restriction. In particular we note from the run times of Scenarios 3 and 4 that the relative run time of soft-restricted A-SMOP increases as the number of non-affected variables for a given change increases (provided those variables contain at least one other change).

As expected, the results of Scenario 6 show that A-SMOP overestimates the number of changepoint in the series. The is due to the algorithm overcompensating for the autocorrelation in the series by attempting to fit more independent segments. Interestingly, while the increased α penalty used in Scenario 6a does reduce the number of spurious changepoints estimated, the average V-measure is lower in comparison to Scenario 6. This is likely due to the difficulty of correctly identifying the true changepoint locations in the presence of the autocorrelation. While this is also an issue for Scenario 6, it is somewhat mitigated by the additional estimated changepoints. Nevertheless, the reasonable V-measure values for Scenario 6 suggest that A-SMOP is relatively robust to misspecification with respect to serial dependence, provided an increased penalty is used.

Similarly, the results of Scenario 11 demonstrate that A-SMOP is robust to the presence of cross-correlation between the variables, without any necessary increase in penalty. The results for this scenario are comparable with those of Scenario 4, the equivalent scenario without cross-correlation.

Scenarios 7–10 illustrate two noteworthy points. Firstly, as one might expect, a decrease in the ratio of the number of changepoints (m) to sequence length (n) improves accuracy. Secondly, the relative running time increases sharply with an increase in m, and exhibits a less-sharp increase when n increases. This is due to two distinct but related reasons. The first is that an increase in the true number of changepoints means that α-PELT (defined in Section 5.2.3) will likely detect more changepoints in the initial state of the algorithm. The second is that an increase in n means that there is a greater chance of α-PELT detecting spurious changes, though the number of additional changes detected will likely be relatively lower than those found with an increase in m. Both of these scenarios mean that the search space considered within the SMOP stage of the algorithm is increased in size, and hence the algorithm requires a longer computation time.

The results of PELT in Table 5.3.1 show that while reasonably good segmentations are obtained (as indicated by the V-measures), it generally overestimates the number of changepoints. This is due to its lack of multivariate power: it is unable to determine whether two ‘close’ changepoints occurring in two separate variables actually correspond to the same change. Conversely, while E-Divisive and E-CP3O often estimate the correct number of changes, their assumption of fully-multivariate changes often results in segmentations which are erroneous. This is especially true when a smaller amount of variables are affected by the changes, shown by the lower V-measure values. In addition, Table 5.3.1 shows E-CP3O consistently underestimates the number of changepoints across all scenarios. This is likely due to the test statistic used by E-CP3O only incorporating the data around the possible changepoint in consideration, rather than the whole time series (as discussed in Section 2.2.3). This local consideration of data means that E-CP3O likely has less power in detecting changes compared to the other methods considered, which utilise all of the data available.

We note that hard-restricted A-SMOP is in general both faster and more accurate than E-Divisive and E-CP3O, whilst soft-restricted A-SMOP is always more accurate. Therefore, if a faster runtime is preferred whilst maintaining a good level of accuracy, we would recommend the use of A-SMOP with hard restriction over E-Divisive and E-CP3O for series of moderate length and dimension.

Scenarios similar to 1–5 containing $p = 6$ variables instead of $p = 4$ were also considered. Performance of A-SMOP (using both hard and soft restriction) gave similar results, with the exception of an increased running time for soft subset restriction. This is because an increase in the number of variables leads to an exponential increase in the number of affected variable subsets being considered. This suggests that increasing the number of variables does not compromise the accuracy of the algorithm.

5.3.2 Scalability of A-SMOP

We now consider a range of different scenarios which have increasing numbers of observations (n), variables (p) and changepoints (m), respectively. The aim of this study is to identify how the computation time of A-SMOP scales with increases in such values. The details of the study and the scenarios investigated are first introduced, followed by the corresponding results and discussion.

Simulation Study Details

Three sets of scenarios are considered in the study. For each set, two of the values of n, p and m are fixed, and the third is increased along some scale. The details of these three sets of scenarios are given below.

Increasing Number of Observations For these scenarios, the number of observations n is increased with the number of variables and changepoints fixed as $p = 4$ and $m = 10$. The scenario numbers and the corresponding values of n considered are given in Table 5.3.3.

Scenario	1.1	1.2	1.3	1.4	1.5	1.6	1.7
n	1000	5000	10000	50000	100000	250000	500000

Table 5.3.3: Scenarios with differing values of n considered for assessing the scalability of A-SMOP, with fixed values of $p = 4$ and $m = 10$.

Increasing Number of Variables For these scenarios, the number of variables p is increased while fixing the number of observations $n = 50000$ and number of changepoints $m = 10$. The different values of p considered are shown in Table 5.3.4.

Scenario	2.1	2.2	2.3
p	4	6	8

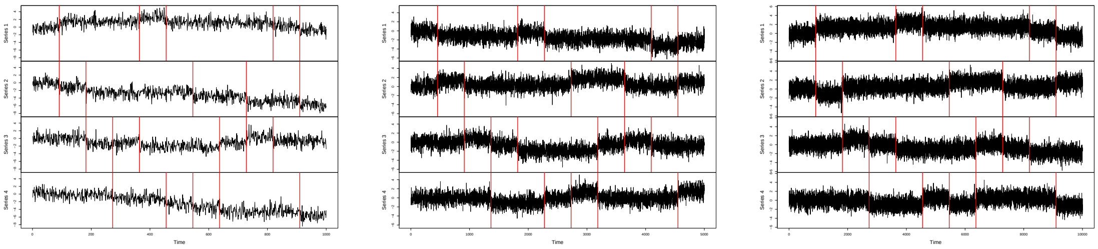
Table 5.3.4: Scenarios with differing values of p considered for assessing the scalability of A-SMOP, with fixed values of $n = 50000$ and $m = 10$.

Increasing Number of Changepoints For these scenarios, the number of changepoints m is increased with fixed $n = 50000$ and $p = 4$. The values of m investigated are given in Table 5.3.5.

Scenario	3.1	3.2	3.3	3.4	3.5
m	10	12	15	20	30

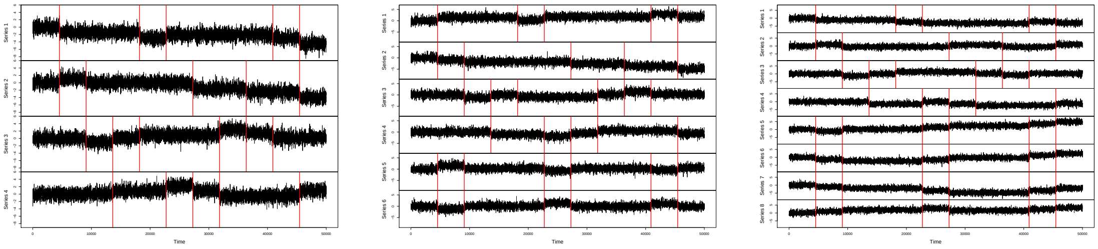
Table 5.3.5: Scenarios with differing values of m considered for assessing the scalability of A-SMOP, with fixed values of $n = 50000$ and $p = 4$.

The subsets of variables affected by the changes generally differ between the scenarios. Figures 5.3.2, 5.3.3 and 5.3.4 present plots for the scenarios with increasing n, p and m, respectively, and these show the subsets of affected variables for each change. These subsets remain constant over all replicates for a given scenario. Note that for those scenarios with increasing n, each scenario has the same set of affected variable subsets for the changes. Hence, for the scenarios with increasing n only the plots for Scenarios 1.1, 1.2 and 1.3 are presented as an illustration.



5.3.2(a): Scenario 1.1 5.3.2(b): Scenario 1.2 5.3.2(c): Scenario 1.3

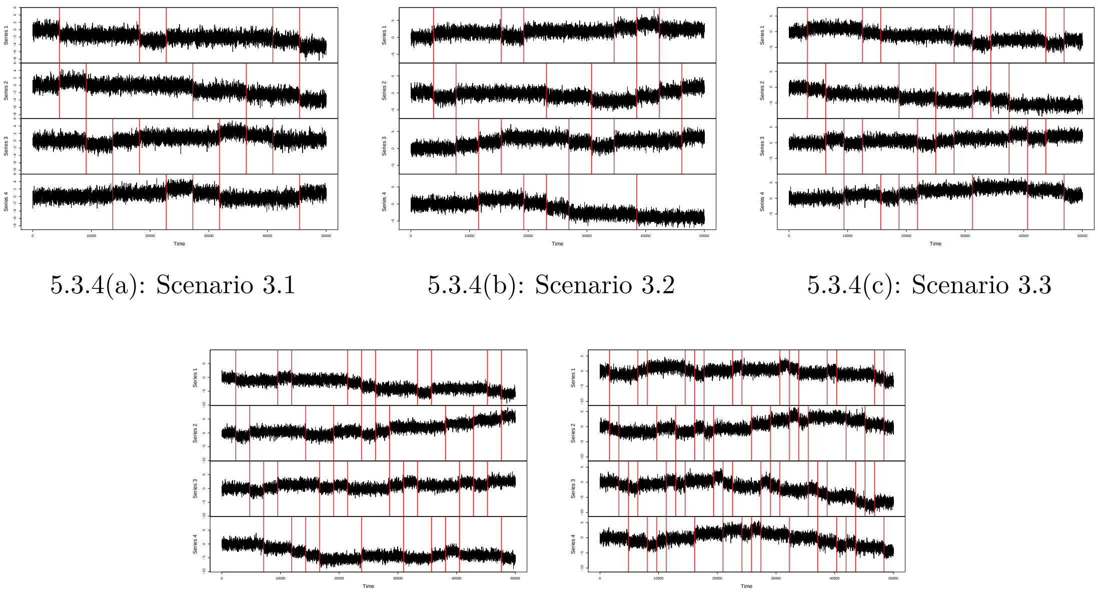
Figure 5.3.2: Example time series of the scenarios with increasing values of n and fixed $p = 4$, $m = 10$.



5.3.3(a): Scenario 2.1 5.3.3(b): Scenario 2.2 5.3.3(c): Scenario 2.3

Figure 5.3.3: Example time series of the scenarios with increasing values of p and fixed $n = 50000$, $m = 10$.

We note that some of these scenarios are equivalent due to the values of n, p and m and the subsets of affected variables considered. Specifically, Scenarios 1.4, 2.1 and 3.1 all consider the case of $n = 50000$, $p = 4$ and $m = 10$, with identical affected variable subsets for the changes across the scenarios. Since the A-SMOP algorithm (using either hard or soft restriction) is deterministic for a given dataset, cost function, penalty values and window size, then application of the method to these three scenarios will always produce identical results. As in the methodology comparison study in Section 5.3, we set penalty values of $\alpha = \sum_{i=1}^{m+1} \log(n_i) + (2m - 1)\log n$ and $\beta = 2 \log p \log n$, the minimum distance between changepoints to be 2, and the window size $w = 5$. To assess the scalability of the A-SMOP algorithm as a whole, we apply A-SMOP using both hard and soft restriction to each scenario. The average computation times of these applications are recorded in each case to analyse the scalability of the algorithm. In addition, the average V-measures and average number of detected changepoints are recorded to observe how the accuracy of the algorithm is affected (if at all) as the scale of the data increases.



5.3.4(a): Scenario 3.1

5.3.4(b): Scenario 3.2

5.3.4(c): Scenario 3.3

5.3.4(d): Scenario 3.3

5.3.4(e): Scenario 3.3

Figure 5.3.4: Example time series of the scenarios with increasing values of m and fixed $n = 50000$, $p = 4$.

Results and Discussion

Tables 5.3.6, 5.3.7 and 5.3.8 present results from the application of A-SMOP using both hard and soft restriction to the scenarios with increasing n, increasing p and increasing m, respectively. The results contain the average V-measures, average number of estimated changepoints and average computation times (in minutes) of the segmentations produced by the algorithm. For some scenarios in these tables, the results for soft-restricted A-SMOP as listed as N/A. This is because in these scenarios soft-restricted A-SMOP has been applied, but either the computation time or memory required is significantly increased such that its performance is not viable given the resources available, and so they are not considered further here.

Metric (Average)	Scenario	1.1	1.2	1.3	1.4	1.5	1.6	1.7
		$n = 1000$	$n = 5000$	$n = 10000$	$n = 50000$	$n = 100000$	$n = 250000$	$n = 500000$
V-Measure	Hard	$0.938_{0.00278}$	$0.972_{0.00158}$	$0.971_{0.00157}$	$0.975_{0.00128}$	$0.977_{0.00155}$	$0.975_{0.00134}$	$0.976_{0.00141}$
	Soft	$0.966_{0.00198}$	$0.991_{0.00103}$	$0.995_{0.000558}$	$0.995_{0.0009}$	$0.997_{0.000787}$	$0.997_{0.000642}$	$0.998_{0.000634}$
Number of Changepoints	Hard	$11.8_{0.111}$	$12.2_{0.133}$	$12.4_{0.14}$	$12.3_{0.117}$	$12.1_{0.138}$	$12.3_{0.127}$	$12.3_{0.131}$
	Soft	$10.1_{0.0403}$	$10.1_{0.0338}$	$10.2_{0.0386}$	$10.2_{0.0386}$	$10.1_{0.0273}$	$10.1_{0.0367}$	$10.1_{0.0239}$
Computation Time (Mins.)	Hard	$0.0339_{0.00153}$	$0.146_{0.00904}$	$0.32_{0.0176}$	$1.72_{0.124}$	$4.36_{0.391}$	$12.4_{0.799}$	$21.1_{1.44}$
	Soft	$0.977_{0.0854}$	$4.55_{0.575}$	$11.2_{1.24}$	$58.3_{6.2}$	146_{18}	$524_{74.3}$	$827_{90.9}$

Table 5.3.6: The average V-measures, average number of changepoints and average computation time (in minutes) of the segmentations produced by A-SMOP using both hard and soft restrictions for the scenarios with increasing values of n. The values $p = 4$ and $m = 10$ are fixed across the scenarios.

Metric (Average)	Scenario	2.1	2.2	2.3
		$p = 4$	$p = 6$	$p = 8$
V-Measure	Hard	$0.975_{0.00128}$	$0.965_{0.00175}$	$0.96_{0.00172}$
	Soft	$0.995_{0.0009}$	N/A	N/A
Number of Changepoints	Hard	$12.3_{0.117}$	$13.3_{0.164}$	$14.1_{0.164}$
	Soft	$10.2_{0.0386}$	N/A	N/A
Computation Time (Mins.)	Hard	$1.72_{0.124}$	$29.9_{3.41}$	$335_{53.1}$
	Soft	$58.3_{6.2}$	N/A	N/A

Table 5.3.7: The average V-measures, average number of changepoints and average computation time (in minutes) of the segmentations produced by A-SMOP using both hard and soft restrictions for the scenarios with increasing values of p. The values $n = 50000$ and $m = 10$ are fixed across the scenarios.

Metric (Average)	Scenario	3.1 $m = 10$	3.2 $m = 12$	3.3 $m = 15$
V-Measure	Hard Soft	$0.975_{0.00128}$ $0.995_{0.0009}$	$0.977_{0.0013}$ $0.998_{0.000454}$	$0.98_{0.000957}$ $0.998_{0.000334}$
Number of Changepoints	Hard Soft	$12.3_{0.117}$ $10.2_{0.0386}$	$14.7_{0.155}$ $12_{0.0171}$	$18.2_{0.152}$ $15.1_{0.0314}$
Computation Time (Secs.)	Hard Soft	$1.72_{0.124}$ $58.3_{6.2}$	$3.38_{0.261}$ $142_{21.2}$	$12.7_{0.769}$ $486_{50.7}$

Metric (Average)	Scenario	3.4 $m = 20$	3.5 $m = 30$	
V-Measure	Hard Soft	$0.978_{0.000887}$ N/A	$0.98_{0.000548}$ N/A	
Number of Changepoints	Hard Soft	$24.8_{0.189}$ N/A	$36.8_{0.193}$ N/A	
Computation Time (Mins.)	Hard Soft	$88.1_{5.98}$ N/A	$1340_{91.2}$ N/A	

Table 5.3.8: The average V-measures, average number of changepoints and average computation time (in minutes) of the segmentations produced by A-SMOP using both hard and soft restrictions for the scenarios with increasing values of m. The values $n = 50000$ and $p = 4$ are fixed across the scenarios.

From Tables 5.3.6, 5.3.7 and 5.3.8 it can be seen that, as might be expected, an increase in the scale of the data leads to an increase in the computation time of the algorithm. As observed in the comparison study in Section 5.3, and as would be expected given the differences in the restrictions, hard restriction always has a vastly reduced computation time compared to soft restriction. As the scale of the data increases, this difference (in real terms) in the computation time between the restrictions widens. In general, it appears from the tables that both hard- and soft-restricted A-SMOP scale at the same rate. This implies that if in some scenario hard restriction has a 100% increase in computation time (for example), then soft restriction also has a 100% increase in computation time. In terms of accuracy, we note that the results here reflect the results on accuracy presented in Section 5.3, which show that soft-restricted A-SMOP is always more accurate than (or, at least, as accurate as) hard-restricted A-SMOP.

For the case of increasing n, Table 5.3.6 shows that even in the largest case considered with $n = 500000$, hard-restricted A-SMOP requires less than 25 minutes for a single replicate. In contrast, for the same scenario soft-restricted A-SMOP requires over 13 hours of computation for a single replicate. Even for Scenario 1.5, where $n = 100000$, soft-restricted A-SMOP requires overs 2 hours, compared to the 4 minutes required when using hard restriction. Therefore, for practical application in situations with large n (especially those with $n > 250000$) it is recommended to use A-SMOP with hard restriction rather than soft-restriction. This is due to the significantly increased computation time of soft-restricted A-SMOP which is likely to make application of the method impractical (or even infeasible for the resources provided for cases of very large n).

Considering the cases with increasing p, it can be seen from Table 5.3.7 that even for $p = 6$, soft-restricted A-SMOP is infeasible for the resources available. Comparatively, hard-restricted A-SMOP is still able to perform with a reasonable computation time of approximately 30 minutes per replicate. However, adding only two variables to the series (so that $p = 8$) increases the computation time of hard-restricted A-SMOP to over 5 hours for a single replicate. Additional scenarios containing larger numbers of variables were also considered (for example, with $p = 10$), however the performance of even hard-restricted A-SMOP was infeasible for these scenarios given the resources available. These results demonstrate that A-SMOP has difficulty in scaling even to moderate values of p for these given values of $n = 50000$ and $m = 10$. This reflects the discussion regarding the computational complexity of the SMOP algorithm (on which A-SMOP is based) in Section 4.7, which illustrates that the computational cost increases exponentially with an increase in p.

Similarly, the results for increasing m in Table 5.3.8 show that for $m \geq 20$ (with fixed $n = 50000$ and $p = 4$), soft-restricted A-SMOP is infeasible for the resources available. For $m = 15$, while soft-restricted A-SMOP is feasible, it requires over 8 hours for a single replicate compared to the 12 minutes required for hard-restricted. When m is increased to 30 in Scenario 3.5, computation time for hard-restricted A-SMOP rises to over 22 hours on average. This significantly increased computation time with increased m is due to the additional changepoint locations being detected by PELT in the initial stages of A-SMOP, which are then fed into SMOP within the latter stage, thereby increasing the number of necessary calculations. Such a large computation time is likely to be impractical for substantive problems, and so any further scenarios with increased m are not considered here.

In general, these results demonstrate that increasing p and m have the greatest influence on the scalability of A-SMOP, with even small increases p in particular significantly increasing the computation time of the algorithm. While it is possible that these computation times could be improved with the implementation of modifications such as parallel programming, more efficient architecture or more powerful computer machinery, the observed results suggest that the scalability of the A-SMOP algorithm has been explored as much as possible for the resources available.

5.4 Application to Acoustic Sensing Data

Given the strong performance of A-SMOP on a range of simulated data, we now consider its application to a dataset arising from acoustic sensing. As discussed in Chapter 3, the use of acoustic sensing technology is becoming increasingly prominent within the oil and gas industry. Such technology uses fibre-optic cables to record the vibrations along pipelines in oil and gas wells. The behaviour of the vibration measurements recorded by these sensors provide information regarding the nature of the flowing oil or gas in the well.

Changes in these vibration measurements often correspond to the presence of certain features within the well. We consider acoustic sensing measurements from an oil and gas extraction well. This dataset represents Fourier-transformed observations from multiple depths within the well over time. The Fourier-transformed data is examined as this is the form of the data analysed by the engineers. Due to the very high measuring frequency of the fibre-optics cables (up to 10kHz), we were provided with data which had been sub-sampled by a factor of 100. Figure 5.4.1 shows an example of 3159 observations from ten consecutive depths in the well where vibrations were recorded.

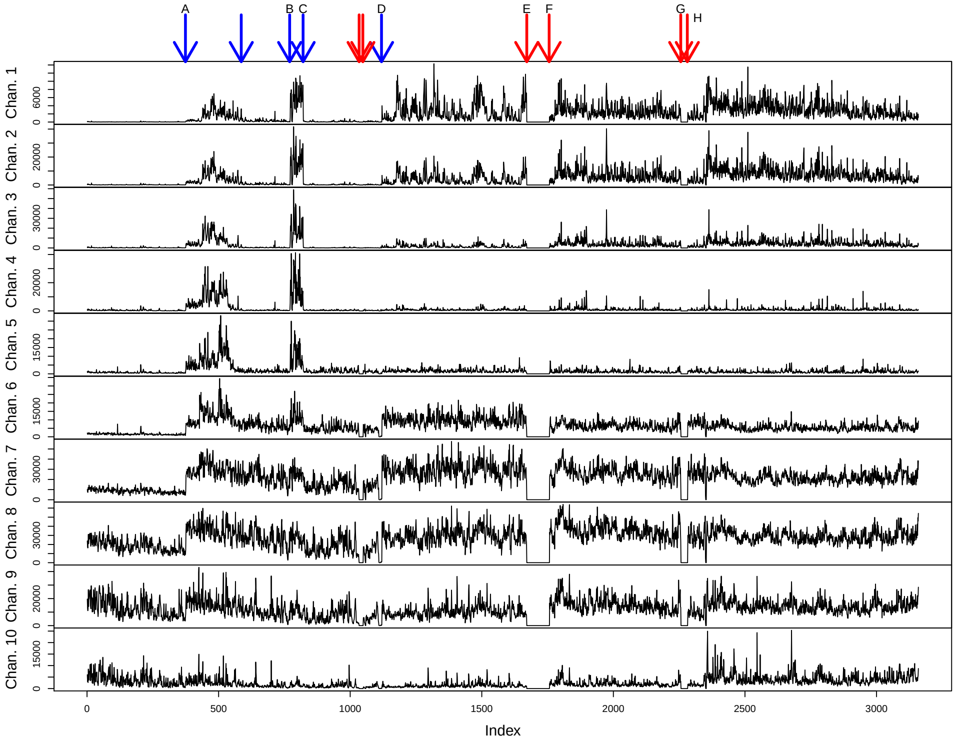


Figure 5.4.1: Example of ten consecutive channels of Fourier-transformed acoustic sensing data. Blue arrows indicate known slug events, red arrows indicate known striping events.

For this dataset, interest lies in the detection of two particular features. The first of these is the presence of ‘slugs’ in the well. These occur when the gas and oil in the multiphase flow separate into different bands, so that the flow becomes alternating single-phase between liquid (oil) and gas. These slug bands are characterised by irregular flows and sudden surges, which correspond to sudden changes in the recorded vibrations. The fast rate of flow and the sub-sampling of the data mean that these appear as changepoints. Since slugs can both form and disperse naturally (although they may persist as they flow up the well), these sudden changes in vibration generally only affect a subset of series which represent different depths. Identifying the presence of slugs is important as they can reduce the pressure in the well and hence cause blockages. Determining which subset of depths is affected can provide information on the size and location of the slug, which can in turn allow for the necessary action to be taken to return to multiphase flow.

The second feature of interest is a type of error feature referred to as ‘striping’. This occurs when there is an error in the measuring equipment which means that no observations are recorded by the equipment over any channels for a certain period. Note that this is different to the error features discussed previously in Chapter 3, which manifest as changes in second-order structure.

The signals provided for analysis are sub-sampled Fourier transformed acoustic signals. For such signals, one can identify both slugs and stripes as changes in variation within the de-trended acoustic sensing signal. To identify such changes, we perform the A-SMOP algorithm using hard subset restriction on the ten channels presented in Figure 5.4.1. The penalty values $\alpha = 12 \log n$ and $\beta = 4 \log p \log n$ have been used, as these demonstrated promising results in initial tests on sub-segments of the data. We also compare the performance of A-SMOP with the segmentations obtained via other multivariate changepoint methods described in Section 2.2.3 (E-Divisive, E-CP3O and repeated application of univariate PELT). The locations estimated by hard-restricted A-SMOP, E-Divisive, E-CP3O and PELT are shown in red, blue, orange and green respectively.

The results in Figures 5.4.2 and 5.4.3 show that A-SMOP clearly identifies both cases of striping (shown by changes occurring in all variables) and changes in vibration (shown by changes occurring in only subsets of variables). For example, it has detected the change in vibration at event A in variables 1–7, and the four striping-related events at E, F, G and H. In comparison, E-Divisive and E-CP3O identify the events A-H (and more), since these generally have a large number of affected variables. However, their fully-multivariate assumptions mean that they are unable to distinguish between those changes which correspond to striping (E, F, G and H), and those which correspond to true vibration changes (A, B, C and D). Repeated-univariate PELT identifies the same changepoint locations with the same variables as A-SMOP (events A–H, and more), but it also estimates additional changepoint locations which do not appear to correspond to any particular features (events i–viii). This is likely due to the lack of a multivariate consideration and the presence of the serial dependence within the data. Increasing the penalty value could reduce these additional changepoints,

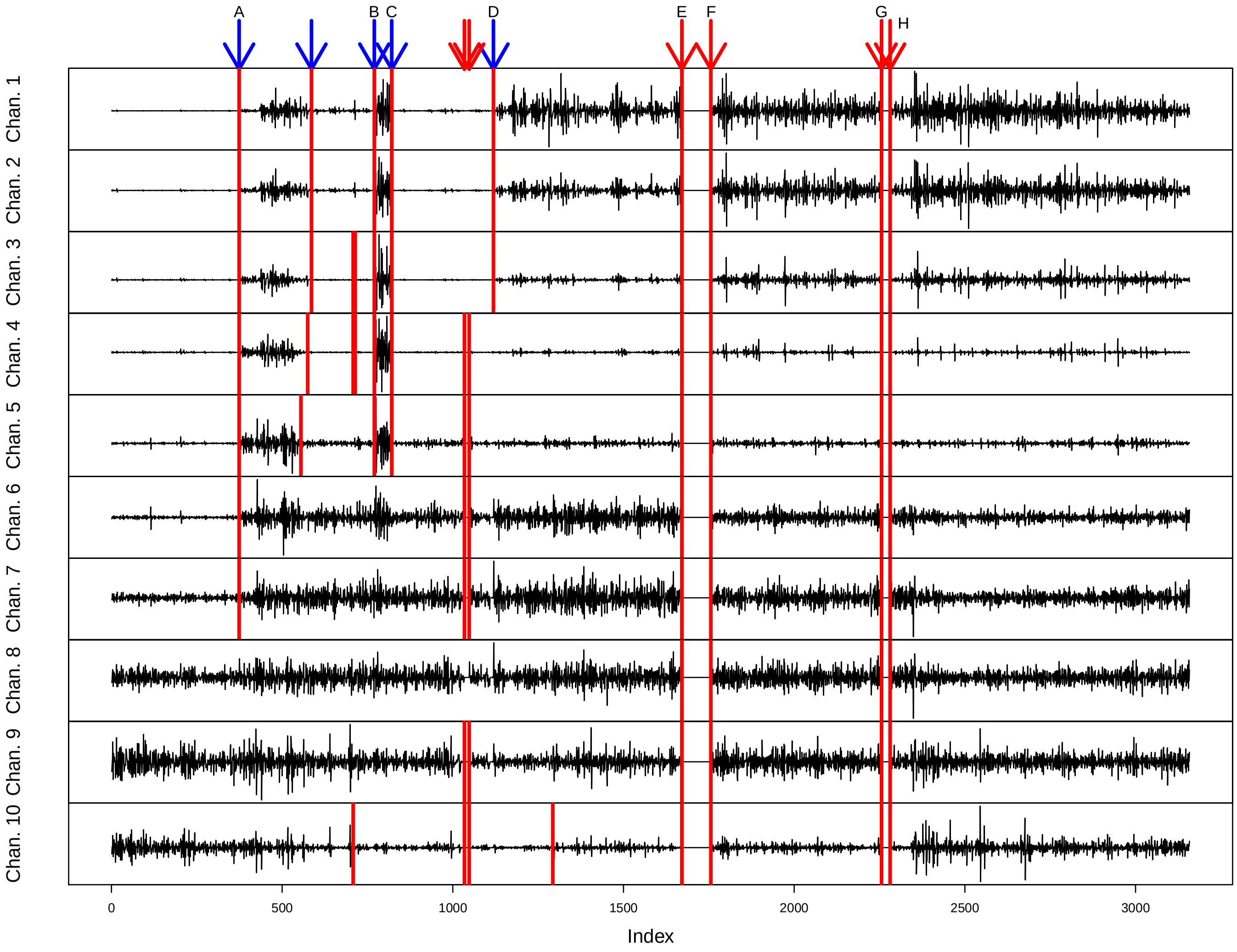


Figure 5.4.2: Changepoints detected by A-SMOP in ten channels of differenced acoustic sensing data. Blue arrows indicate slug events, red arrows indicate striping events. A, B, C and D correspond to specific slug events, and E, F, G and H correspond to specific striping events.

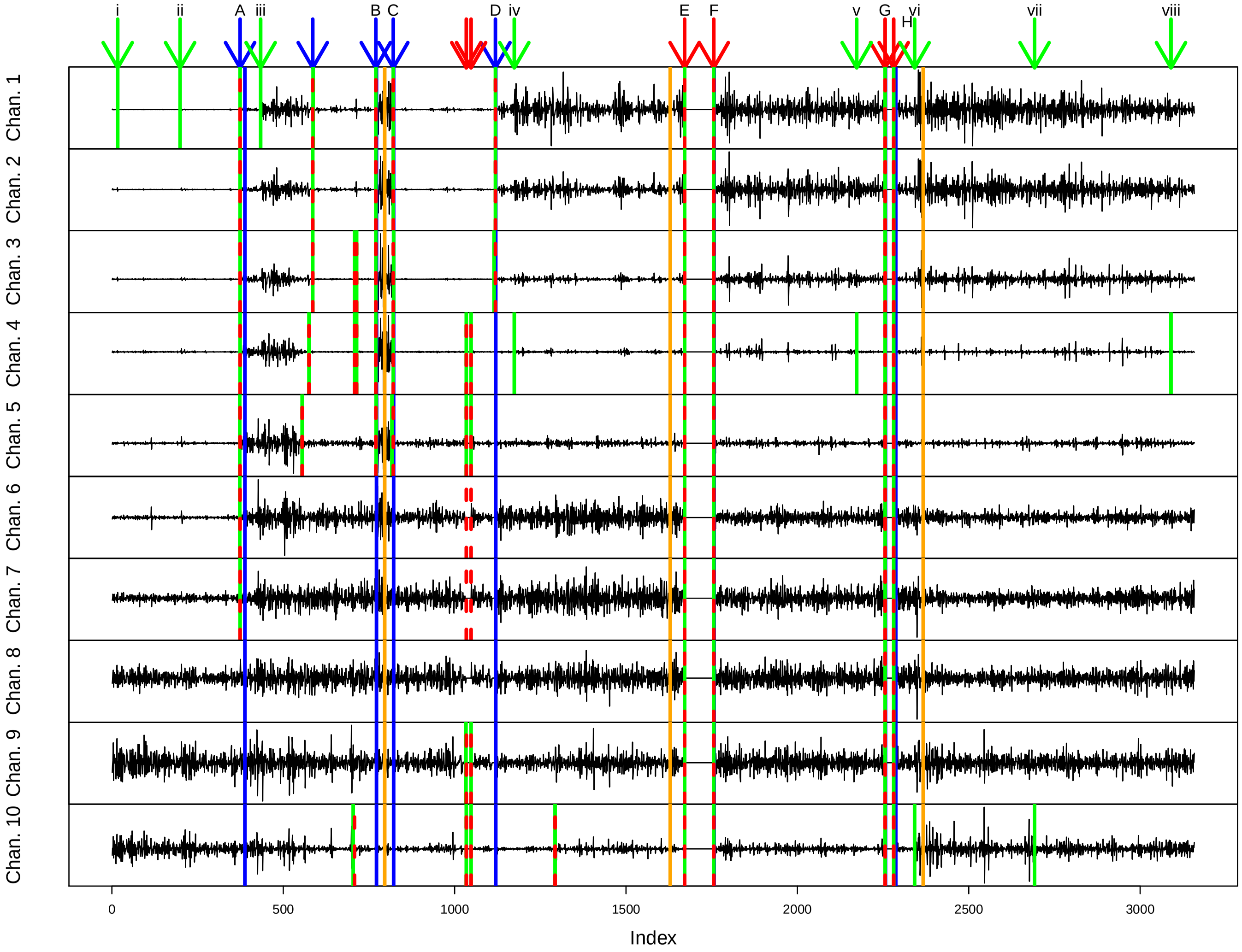


Figure 5.4.3: Changepoints detected by A-SMOP (red), E-Divisive (blue), E-CP3O (orange) and PELT (green) in ten channels of differenced acoustic sensing data. Blue arrows indicate slug events, red arrows indicate striping events. A, B, C and D correspond to specific slug events, and E, F, G and H correspond to specific striping events. The locations i–viii correspond to changepoints detected by PELT which do not appear to correspond to any particular event.

however this may also mean that true changes are not detected. Consequently such a repeated-univariate approach is not as flexible as A-SMOP, which identifies these features without such overestimation.

5.5 SMOP vs A-SMOP: Application to Quebec River Flows

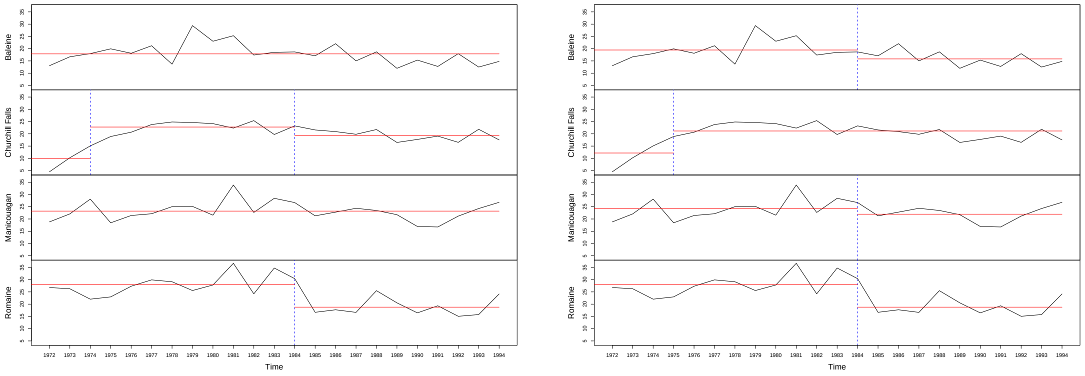
To highlight the differences in performance between SMOP and A-SMOP in a practical setting, we consider application of both methods to the set of annual river flow measurements of four rivers in Quebec. The application of SMOP to this dataset has been considered in Section 4.6. In this section, we will also consider the application of A-SMOP, and in addition to the detected changepoint locations and affected variable subsets we assess the parameter estimates for each segment and the methods' computation times.

For both SMOP and A-SMOP, similar to the analysis in Section 4.6 we use the multivariate Normal likelihood as a cost function and assume changes are occurring in both mean and variance. However, deviating from Section 4.6 and following the penalty adopted in the simulation study in Section 5.3, we set α to be the modified Bayesian information criterion:

$$\alpha = \sum_{i=1}^{m+1} \log(n_i) + (2m-1)\log(n). \tag{5.5.1}$$

As before, we set $\beta = 2 \log n \log p$. For A-SMOP, soft restriction is used with a window size of 3. The results of applying SMOP and A-SMOP to these river flow measurements are presented in Figures 5.5.1(a) and 5.5.1(b), respectively. In particular, these show the detected changepoint locations, the affected variables and the mean values of each segment. In addition, Table 5.5.1 provides the estimated mean and standard deviation parameter values for each segment in each variable.

From Figures 5.5.1(a) and 5.5.1(a), it can be seen that A-SMOP detects changepoint locations in 1975 and 1984, which are similar to the changes detected at 1974



5.5.1(a): SMOP results.

5.5.1(b): A-SMOP results.

Figure 5.5.1: The results of applying SMOP and A-SMOP to the Quebec river flows. The blue vertical lines represent changepoint locations, and the red horizontal lines represent the corresponding means of those segments.

and 1984 by SMOP. However, the affected variables detected by A-SMOP are slightly different to those detected by SMOP (which are guaranteed optimal for the given cost function and penalties). This represents the downside of A-SMOP, even when using soft-restriction, in that it does not guarantee to estimate the optimal set of changepoints and corresponding affected variables. Although, as demonstrated in the plots, these estimates are similar to those of SMOP. This similarity is also reflected in the mean and standard deviation estimates in Table 5.5.1, which are similar generally and are identical in Romaine where the same changepoint is estimated by SMOP and A-SMOP.

This lack of a guarantee of optimality by A-SMOP is also demonstrated in the corresponding likelihood values of the two segmentations: the negative log-likelihood of the A-SMOP segmentation is 545.66, whereas the corresponding value for the SMOP segmentation is 533.09 (note that since this is negative log-likelihood, a lower value indicates a better model fit). However, the benefit of A-SMOP over SMOP is its vastly reduced computation time. For this dataset an application of SMOP required 130 minutes, whereas A-SMOP required 0.05 seconds. Therefore, in this case A-SMOP is over 156,000 times faster than SMOP; such behaviour is typical of the algorithms. Combined with the good quality of segmentations, this emphasises how A-SMOP is the superior choice over SMOP for usage in practical settings.

		SMOP		A-SMOP	
		Mean	Standard Deviation	Mean	Standard Deviation
Baleine (Variable 1)	Segment 1	17.9	4.29	19.5	4.48
	Segment 2	N/A	N/A	15.8	3.16
Churchill Falls (Variable 2)	Segment 1	9.93	5.29	12.2	6.23
	Segment 2	22.8	2.27	21.2	2.82
	Segment 3	19.3	2.16	N/A	N/A
Manicouagan (Variable 3)	Segment 1	23.2	3.94	24.2	4.31
	Segment 2	N/A	N/A	22.0	3.18
Romaine (Variable 4)	Segment 1	28.0	4.26	28.0	4.26
	Segment 2	18.8	3.59	18.8	3.59

Table 5.5.1: The estimated values of the mean and standard deviation parameters for each of the detected segments in each river. Values of N/A reflect cases where there is no such segment in that variable. Note that Segment i in Variable j may be at different locations and a different length between SMOP and A-SMOP

5.6 A-SMOP for Structured Data

The properties of the acoustic sensing dataset considered in Section 5.4 suggests that the changepoints present may be 'structured' in some manner. In particular, it may be that in a given scenario, only adjacent variables in the multivariate series may be affected by a changepoint. Hence, in such a scenario, instead of considering all possible subsets of affected variables for a potential changepoint, the set could be restricted to subsets with only contiguous variables being affected. Restricting the subsets of affected variables in this way has the potential to significantly reduce the computation time of the algorithm.

If this assumption of structured changepoints is made while using hard-restricted A-SMOP, there is not likely to be a significant change in computation time. This is because hard-restricted A-SMOP considers only a single affected variable subset per potential changepoint, and so no improvement can be made in terms of the number of subsets considered, although the structured changepoint assumption may change the actual subset considered. However, if the structured changepoint assumption is applied while using soft-restricted A-SMOP, then this is likely to lead to improvements in computation speed. This is because such an assumption drastically reduces the number of different affected variable subsets to consider within the soft-restricted algorithm, since only the subset with contiguous affected variables are considered. The magnitude of the computation speed improvements depends on the number of variables which are affected by the true change.

In general, if the structured changepoint assumption is valid for the problem being considered, then imposing this assumption would not alter the final subset of affected variables which would have otherwise been detected if the assumption had not been imposed. While imposing the assumption of structured changepoints may potentially reduce the computation time, there are some important issues which need to be considered before making such an assumption. One such issue is that additional information would be required regarding the ‘closeness’ of the multiple variables. This may be possible in cases such as the acoustic sensing dataset considered in Section 5.4 (since the different variables are different depths in the oil well), but in general such information is not likely to be available or indeed exist. For example, for financial or stock market data specifying a measure of ‘closeness’ could be challenging due to their non-physical nature. However, even for variables where such a measure is possible (for example, geographical locations), it may not be sensible to impose a linear ordering. For example, three-dimensional spatial locations do not necessarily have a natural two-dimensional ordering.

In addition, the SMOP and A-SMOP algorithms make the assumption of independence between the multiple variables, but incorporating information regarding the structure of the changepoints implies that the variables are dependent in some manner. Therefore, this could imply that it is necessary to also model this dependence within the penalised cost function approach (in addition to restricting the subsets of affected variables which are considered). This represents a considerable modification to the SMOP and A-SMOP approaches. Hence, for this reason and the difficulty of specifying a closeness measure for the variables in practice, we do not implement a restricted approach for structured changepoints.

5.7 Conclusion

Within this Chapter, we have demonstrated how the computation time of the SMOP algorithm (introduced in Chapter 4) can be vastly reduced via two stages of approximation. The first stage reduces the number of possible changepoint locations to be considered, whilst the second stage reduces the amount of possible affected variable subsets considered for each possible changepoint. Two procedures are proposed for the second stage: hard restriction, and soft restriction. The resulting algorithm implementing these approximations is termed Approximate SMOP (A-SMOP).

Empirical results from the simulation study demonstrate that hard restriction favours a shorter computation time at the expense of some accuracy, whilst soft restriction provides greater accuracy but requires slightly longer computation time. More generally, simulation results show that the reduction in accuracy is dependent on the relative magnitude of the changes: the larger the magnitude, the smaller the reduction in accuracy. Similarly, a smaller number of true changes, larger magnitudes of shifts, and larger subsets of affected variables (if using soft restriction) provide a greater reduction in computation time. Further simulations have been considered to investigate the scalability of the algorithm.

A comparison of A-SMOP with PELT, E-Divisive and E-CP3O demonstrates that the subset-multivariate approach taken by A-SMOP represents an intermediate between a fully-multivariate approach and a repeated univariate approach. The multivariate power of detecting changes across multiple variables is harnessed, whilst the univariate benefits of not assuming fully-common changes and ignoring ‘noisy’ variables are also exploited. These advantages come with only a mild increase in computational cost, with the possibility of limiting this increase at the expense of some accuracy. In addition, the use of PELT within A-SMOP means that the benefits of any future improvements made to PELT can also be reaped by A-SMOP. Comparisons have also been drawn between SMOP and A-SMOP to assess the differences between the two algorithms in practice.

Finally, we have considered the possibility of utilising information regarding the structure of changepoints within the algorithm. While this could potential result in reduced computation time, the associated practical difficulties mean that such a feature is not implemented in the algorithm.

The methodology developed in this chapter and in Chapter 4 has been implemented in the `changepointmv` package in `R`. This package is available at `http://www.lancaster.ac.uk/~pickerin/software.html`. Details of this package, including its structure, methods and examples, are contained in Appendix C.

Algorithm 6: Approximate Subset Multivariate Optimal Partitioning

```
Input      : A multivariate time series $\boldsymbol{X} = (\boldsymbol{X}_1, \boldsymbol{X}_2, \ldots, \boldsymbol{X}_n)$ containing $p$
             variables, a univariate cost function $\mathcal{D}_j(\cdot)$ for each variable $j$, and
             penalty values $\alpha$ and $\beta$.
Initialise: Set $F(c_0) = 0$, $\mathcal{L}(c_0) = \emptyset$, and $\boldsymbol{c}(c_0) = \emptyset$.
begin
    for $j^* \in \{1, \ldots, p\}$ do
        Set $\boldsymbol{\tau}^{j^*} = \alpha$-PELT$(X^{j^*}_{1:n})$
    Set $\boldsymbol{\tau} = \bigcup_{j=1}^{p} \boldsymbol{\tau}^j$
    if Using Hard Subset Restriction then
        Set $\mathcal{S} := \{\mathcal{S}_\tau\}_{\tau \in \boldsymbol{\tau}} = hard(\boldsymbol{\tau}^1, \ldots, \boldsymbol{\tau}^p, w)$
    else if Using Soft Subset Restriction then
        Set $\mathcal{S} := \{\mathcal{S}_\tau\}_{\tau \in \boldsymbol{\tau}} = soft(\boldsymbol{\tau}^1, \ldots, \boldsymbol{\tau}^p, w)$
    for $\tau^* \in \boldsymbol{\tau}$ do
        for $c_{\tau^*} \in \bar{C}_{\boldsymbol{\tau}, \mathcal{S}, \tau^*}$ do
            for $c \in C_{\boldsymbol{\tau}, \mathcal{S}, \tau^*-1}(c_{\tau^*})$ do
                Set $h_{c_{\tau^*}}(c) = F(c) + \sum_{j=1}^{p} \left[ \mathbb{I}(c^j \neq c^j_{\tau^*}) \left( \mathcal{D}_j(X^j_{(c^j+1):c^j_{\tau^*}}) + \alpha \right) \right]$
                                $+ m(c, c_{\tau^*})\beta$
            Set $F(c_{\tau^*}) = \min_{c \in C_{\boldsymbol{\tau}, \mathcal{S}, \tau^*-1}(c_{\tau^*})} \{h_{c_{\tau^*}}(c)\}$
            Set $c' = \arg\min_{c \in C_{\boldsymbol{\tau}, \mathcal{S}, \tau^*-1}(c_{\tau^*})} \{h_{c_{\tau^*}}(c)\}$
            Set $\mathcal{L}(c_{\tau^*}) = \mathcal{L}(c') \cup \{c^1_{\tau^*}, c^2_{\tau^*}, \ldots, c^p_{\tau^*}\}$
    Set $\boldsymbol{c}(c_{\tau^*}) = \left( \boldsymbol{c}(c'), c_{\tau^*} \right)$
Output     : The sequence of most-recent changepoint vectors recorded in
             $\boldsymbol{c}\left((n, n, \ldots, n)\right)$.
```

Quality Measure	Estimates	Scenario 1	Scenario 2	Scenario 3	Scenario 4	Scenario 5	Scenario 6
Average V-Measure	Hard	$0.994_{0.00127}$	$\mathbf{0.994_{0.000563}}$	$0.913_{0.00557}$	$0.931_{0.00555}$	$0.905_{0.0149}$	$0.855_{0.00654}$
	Soft	$\mathbf{0.998_{0.000542}}$	$\mathbf{0.994_{0.000563}}$	$\mathbf{0.974_{0.00305}}$	$\mathbf{0.959_{0.00447}}$	$\mathbf{0.943_{0.014}}$	$\mathbf{0.893_{0.00764}}$
	PELT	$0.842_{0.00421}$	$\mathbf{0.994_{0.000563}}$	$0.801_{0.0035}$	$0.88_{0.00318}$	$0.86_{0.00249}$	$0.824_{0.00317}$
	E-Divisive	$\mathbf{0.998_{0.000624}}$	$0.307_{0.000577}$	$0.706_{0.00108}$	$0.475_{0.00101}$	$0.302_{0.00955}$	$0.347_{0.00452}$
	E-CP3O	$0.997_{0.000447}$	$0.304_{0.00125}$	$0.677_{0.007}$	$0.456_{0.00454}$	$0.357_{0.00621}$	$0.408_{0.00398}$
Average # of cpts	True	4	4	4	4	4	4
	Hard	$4.09_{0.00288}$	$\mathbf{4_{0}}$	$5.13_{0.00917}$	$4.85_{0.00783}$	$5.08_{0.0121}$	$6.34_{0.0117}$
	Soft	$4.01_{0.001}$	$\mathbf{4_{0}}$	$\mathbf{4.03_{0.00171}}$	$\mathbf{4.11_{0.00314}}$	$\mathbf{4_{0.00636}}$	$\mathbf{4.48_{0.00659}}$
	PELT	$9.14_{0.0178}$	$\mathbf{4_{0}}$	$9.94_{0.0123}$	$7.19_{0.00775}$	$7.63_{0.0063}$	$8.2_{0.00853}$
	E-Divisive	$4.03_{0.00171}$	$4.14_{0.00472}$	$4.07_{0.00293}$	$4.19_{0.00506}$	$1.36_{0.00772}$	$12.51_{0.0633}$
	E-CP3O	$\mathbf{4_{0}}$	$3.74_{0.00799}$	$3.62_{0.00993}$	$3.46_{0.0113}$	$1.83_{0.00985}$	$3.35_{0.0111}$

		Scenario 6a	Scenario 7	Scenario 8	Scenario 9	Scenario 10	Scenario 11
Average V-Measure	Hard	$0.742_{0.0155}$	$0.961_{0.00301}$	$0.942_{0.00271}$	$0.961_{0.00207}$	$0.97_{0.00164}$	$0.937_{0.00531}$
	Soft	$\mathbf{0.806_{0.016}}$	$\mathbf{0.981_{0.00243}}$	$\mathbf{0.962_{0.00238}}$	$\mathbf{0.984_{0.00135}}$	$\mathbf{0.99_{0.001}}$	$\mathbf{0.957_{0.00452}}$
	PELT	$0.734_{0.012}$	$0.892_{0.00236}$	$0.897_{0.00157}$	$0.908_{0.0013}$	$0.913_{0.00119}$	$0.886_{0.00344}$
	E-Divisive	N/A*	$0.482_{0.000495}$	$0.583_{0.000509}$	$0.586_{0.000422}$	$0.589_{0.000221}$	$0.481_{0.000795}$
	E-CP3O	N/A*	$0.434_{0.00523}$	$0.489_{0.00666}$	$0.486_{0.00677}$	$0.49_{0.00657}$	$0.475_{0.0026}$
Average # of cpts	True	4	4	9	9	9	4
	Hard	$4.86_{0.0132}$	$4.85_{0.0077}$	$11.02_{0.0134}$	$11.16_{0.0124}$	$11.08_{0.0133}$	$4.67_{0.00753}$
	Soft	$\mathbf{4.09_{0.00793}}$	$\mathbf{4.11_{0.00345}}$	$9.21_{0.00409}$	$\mathbf{9.13_{0.00367}}$	$9.14_{0.00349}$	$4.16_{0.00368}$
	PELT	$6.32_{0.0138}$	$7.28_{0.00766}$	$16.4_{0.0119}$	$16.36_{0.0111}$	$16.49_{0.0111}$	$6.79_{0.00935}$
	E-Divisive	N/A*	$\mathbf{4.11_{0.00314}}$	$\mathbf{9.16_{0.00395}}$	$9.23_{0.00529}$	$\mathbf{9.12_{0.00433}}$	$\mathbf{4.11_{0.00399}}$
	E-CP3O	N/A*	$2.66_{0.0143}$	$2.48_{0.0111}$	$2.4_{0.0105}$	$2.44_{0.0108}$	$3.86_{0.0062}$

Table 5.3.1: Average V-Measure and Estimated Number of Changepoints (‘cpts’) for A-SMOP with both hard and soft subset restriction, E-Divisive and PELT for each scenario. The best values for each scenario are highlighted in bold. *Note that Scenario 6a is repeat of Scenario 6 except with a larger penalty value. Since E-Divisive and E-CP3O do not utilise a penalty, we do not include them in the results of this scenario.

Run-times (secs)	Scenario 1	Scenario 2	Scenario 3	Scenario 4
Hard Restriction	$0.0877_{0.00126}$	$0.0876_{0.00142}$	$0.115_{0.0021}$	$0.103_{0.00148}$
Soft Restriction	$0.0898_{0.00153}$	$0.159_{0.00459}$	$0.302_{0.0308}$	$0.358_{0.0304}$
PELT	$\mathbf{0.0789_{0.000984}}$	$\mathbf{0.0779_{0.00121}}$	$\mathbf{0.0856_{0.00155}}$	$\mathbf{0.0809_{0.00114}}$
E-Divisive	$18.8_{0.32}$	$18.5_{0.334}$	$18.7_{0.322}$	$18.6_{0.282}$
E-CP3O	$0.251_{0.00269}$	$0.285_{0.00199}$	$0.283_{0.00216}$	$0.291_{0.0025}$

	Scenario 5	Scenario 6	Scenario 6a	Scenario 7
Hard Restriction	$0.304_{0.129}$	$0.171_{0.0056}$	$0.145_{0.00234}$	$0.264_{0.00226}$
Soft Restriction	$0.738_{0.188}$	$7.68_{1.32}$	$0.651_{0.0517}$	$1.04_{0.0791}$
PELT	$\mathbf{0.0881_{0.00122}}$	$\mathbf{0.114_{0.00228}}$	$\mathbf{0.117_{0.00192}}$	$\mathbf{0.16_{0.00203}}$
E-Divisive	$12.8_{0.342}$	$39.5_{1.53}$	N/A*	$66.2_{0.725}$
E-CP3O	$0.355_{0.00734}$	$0.345_{0.00619}$	N/A*	$1.61_{0.00863}$

	Scenario 8	Scenario 9	Scenario 10	Scenario 11
Hard Restriction	$1.8_{0.0831}$	$2.98_{0.145}$	$6.07_{0.314}$	$0.119_{0.00203}$
Soft Restriction	$77.8_{11.7}$	$123_{10.8}$	$225_{21.2}$	$0.297_{0.0181}$
PELT	$\mathbf{0.161_{0.00206}}$	$\mathbf{0.321_{0.00366}}$	$\mathbf{0.634_{0.00568}}$	$\mathbf{0.0945_{0.00109}}$
E-Divisive	$87.6_{1.12}$	$360_{4.09}$	$1450_{12.1}$	$10.2_{0.0678}$
E-CP3O	$1.47_{0.00842}$	$6.88_{0.0562}$	$33_{0.257}$	$0.286_{0.00259}$

Table 5.3.2: Mean running times for A-SMOP with both hard and soft subset restriction, PELT, E-Divisive and E-CP3O for each scenario. The best values for each scenario are highlighted in bold. *Note that Scenario 6a is repeat of Scenario 6 except with a larger penalty value. Since E-Divisive and E-CP3O do not utilise a penalty, we do not include them in the results of this scenario.

Chapter 6

Conclusions and Future Directions

This thesis has presented novel methodology for the detection of changepoints. Two important settings have been considered: (i) autocorrelated univariate time series where changes are occurring in the second-order structure, and (ii) multivariate time series where changes may occur in only a subset of the variables. These two issues represent key aspects of the analysis of acoustic sensing signals that have received comparatively little attention in the changepoint literature. Our goal has been to develop methods which address these, with the emphasis on providing a solution which is accurate whilst maintaining a reasonable computational cost.

Our proposed approach to the second-order univariate changepoint problem was presented in Chapter 3. The procedure, referred to as WHIP, minimises a penalised cost function based on Whittle's likelihood (Whittle, 1951). A key contribution is that the method allows for use of a non-linear penalty in the penalised likelihood. We demonstrate that WHIP allows for a reduced computational complexity over the exact likelihood approach with only slight impact on accuracy. Moreover, our empirical studies demonstrate that our method is comparable with other leading second-order changepoint techniques. Given this, we use WHIP to search for changes in acoustic sensing data which correspond to the occurrence of external disturbances of the measuring cable.

For the multivariate changepoint detection problem, the vast majority of methods currently available assume that changes occur in all variables at the same time.

However, in many practical applications, and for acoustic sensing data in particular, it is often true that only a certain subset of variables are affected by a given change. In Chapter 4 we introduced the concept of *changepoint vectors* which allows for the explicit modelling of both changepoints and their corresponding sets of affected variables. To obtain the optimal configuration of such changepoint vectors for a given multivariate time series, we propose the SMOP algorithm. This uses dynamic programming to minimise a penalised likelihood with two separate penalties. These permit independent control to avoid detecting (i) too many changepoints and (ii) too many variables in a given affected subset. To our knowledge, no current approach in the literature provides both the changepoint locations and their sets of affected variables in the general setting considered here.

The exact approach taken by SMOP requires the evaluation of all possible changepoints and subsets, leading to a computational complexity of $\mathcal{O}(pn^{2p})$. Unfortunately we demonstrated that the use of pruning techniques, akin to those used by Killick et al. (2012), does not reduce this complexity in practice. To tackle this computational burden, in Chapter 5 we proposed an approximation of the SMOP algorithm, A-SMOP. This considers only 'likely' changepoint locations and affected variable subsets, obtained through pre-processing steps. This reduces the computational cost of the method whilst retaining a high-quality (though no longer guaranteed optimal) solution. Studies on both simulated time series and a substantive data set from the acoustic sensing context illustrate the strong performance of A-SMOP against leading competitors for both speed and accuracy.

The multivariate changepoint detection methodology developed in this thesis has been made available in the `changepointmv` package in `R`. This package is available at `http://www.lancaster.ac.uk/~pickerin/software.html`. Details of this package are provided in Appendix C.

6.1 Future Work

We now turn to consider possible avenues for future research. In particular we consider further developments which build from the A-SMOP method. These comprise of heuristic pruning, the inclusion of cross-correlation into the subset-multivariate changepoint model, and incorporation of the CROPS algorithm (Haynes et al., 2014) within the initial stages of A-SMOP to possibly increase the accuracy of the method.

6.1.1 Heuristic Pruning

We begin by considering a heuristic approach to pruning. Suppose that a performance of PELT using penalty $\alpha+\beta$ (herein referred to as $(\alpha+\beta)$-PELT) on a given variable results in some changepoints being detected. We refer to these changepoints as $(\alpha+\beta)$-changepoints. Then inclusion of each of these changes improves the likelihood by more than $\alpha+\beta$ in a single variable. This means that these changes are likely to be detected as a changepoint under the subset-multivariate model by A-SMOP. We therefore wish to use this information to reduce the number of calculations performed by A-SMOP. This could be done as follows. During the A-SMOP algorithm (given in Algorithm 6), suppose we are considering some changepoint vector $c_{\tau^*} \in \bar{C}_{\boldsymbol{\tau},\mathcal{S},\tau^*}$. Suppose further there is a $(\alpha+\beta)$-changepoint $\tau^{j^*}_{\alpha+\beta}$ in some variable $j^* \in \{1,\dots,p\}$, such that $\tau^{j^*}_{\alpha+\beta} < c^{j^*}_{\tau^*}$. Then if for some other changepoint vector $c \in C_{\boldsymbol{\tau},\mathcal{S},\tau^*-1}(c_{\tau^*})$ we have $c^{j^*} < \tau^{j^*}_{\alpha+\beta}$, then we do not need to consider c as the most recent changepoint vector prior to c_{τ^*}. Here we make use of the fact that $\tau^{j^*}_{\alpha+\beta}$ is likely to be a changepoint which is detected within variable j^* by A-SMOP, and hence do not consider any changepoint vectors where the most recent changepoint in variable j^* would be before $\tau^{j^*}_{\alpha+\beta}$.

We refer to this concept as *heuristic pruning*. It is heuristic in the sense that there is no rigorous theoretical argument which justifies the pruning, as there is for example in the pruning used in PELT. Rather, it is based on logical arguments. Heuristic pruning could be implemented within the A-SMOP algorithm by first performing $(\alpha+\beta)$-PELT, in addition to α-PELT, on each variable in the series. This would produce a set of $(\alpha+\beta)$-changepoints for each variable. The pruning would then occur within the dynamic programming stage of the A-SMOP algorithm as described.

The benefit of such pruning would be a reduction in computational time. Such a reduction could be vast, particularly in scenarios where there are frequent changes of relatively large magnitude in many variables. However, the accuracy of the method could be reduced, as the pruning has the potential to discard a possible changepoint location which represents an optimal changepoint. Nevertheless, heuristic pruning may represent a credible extension to A-SMOP for situations where a solution with a higher level of approximation is acceptable in exchange for a reduced computation time.

6.1.2 Modelling Cross-Correlation

Another possible extension to A-SMOP is modelling of the inter-variable correlation within the subset-multivariate changepoint framework. Currently, it is assumed that all variables are independent (i.e. zero cross-correlation), and this allows for the easy summation of individual costs from all variables, as shown in equation (4.3.2) in Chapter 4. However, since we are considering changes which may occur at common time-points across multiple variables, it is reasonable to assume that there may be instances where cross-correlation between these variables is present. One possible approach to including such correlation in the subset-multivariate changepoint model would be to consider the minimisation of the following penalised cost function:

$$\begin{aligned}
&\mathit{cost}(X_{1:n}, \boldsymbol{\tau}, \boldsymbol{\mathcal{J}}) + \mathit{pen}(\boldsymbol{\tau}, \boldsymbol{\mathcal{J}}) \\
&\quad = \sum_{k=1}^{m+1} \sum_{j=1}^{p} \left[\mathbb{I}(c^j_{\tau_k} = \tau_k) \mathcal{D}_j \left(X^j_{(c^j_{\tau_{k-1}}+1):c^j_{\tau_k}}\right) \right] + D(X^1_{1:n}, X^2_{1:n}, \dots, X^p_{1:n}) \\
&\qquad + \left(\alpha \sum_{k=1}^{m+1} \sum_{j=1}^{p} \mathbb{I}(c^j_{\tau_k} = \tau_k) + (m+1)\beta \right), \qquad (6.1.1)
\end{aligned}$$

where $D(X^1_{1:n}, X^2_{1:n}, \dots, X^p_{1:n})$ represents a function quantifying the multivariate dependence structure between the variables. If $\mathcal{D}_j(\cdot)$ is taken to be a likelihood for each $j \in \{1, \dots, p\}$, then $D(\cdot)$ represents the multivariate copula density for the variables.

This function (6.1.1) could then be minimised in exactly the same way as equation (4.3.2) within A-SMOP.

6.1.3 Increasing Changepoint Accuracy using CROPS

A third possible addition could be the use of the Changepoints for a Range Of Penalties (CROPS) algorithm, proposed by Haynes et al. (2014), within the initial stage of A-SMOP to identify potential changepoint locations. The CROPS algorithm works by running PELT on a given time series for a range of penalty values. These values are chosen such that each penalty considered provides a different number of changepoint locations.

In this case, a range of penalties could be considered up to $\alpha+\beta$, and would include α, where α and β are the penalty values discussed in Chapter 4. Then if the CROPS algorithm (instead of traditional PELT) is applied to each variable using this range of penalty values, the changepoint locations which are detected could be used within the formulation of the search space for A-SMOP. This would produce a larger set of possible changepoint locations compared to running PELT with a single penalty α on each variable. Consequently, using CROPS instead would provide greater accuracy at the expense of computation time, since the search space considered by A-SMOP when using CROPS would be larger compared to when using PELT.

Appendix A

Proofs for ‘Multivariate Changepoint Detection with Subsets’

A.1 Proof of Proposition 4.4.1

Proof. Since $F(c_u)$ is, by definition, the minimisation of the penalised cost function (4.3.2) for the series $\boldsymbol{X}_{c_u}$, we have:

$$\begin{aligned}
F(c_u) &= \min_{\boldsymbol{c} \in \mathcal{H}_{c_u}} \left\{ \sum_{k=1}^{m+1} \left(\sum_{j=1}^{p} \left[\mathbb{I}(c^j_{\tau_k} = \tau_k) \left(\mathcal{D}_j(X^j_{(c^j_{\tau_{k-1}}+1):c^j_{\tau_k}}) + \alpha \right) \right] + \beta \right) \right\} \\
&= \min_{c_t \in \{C_t \,:\, c^j_t \leq c^j_u \;\forall\, j\}} \left\{ \min_{\boldsymbol{c} \in \mathcal{H}_{c_t}} \sum_{k=1}^{m+1} \left(\sum_{j=1}^{p} \left[\mathbb{I}(c^j_{\tau_k} = \tau_k) \left(\mathcal{D}_j(X^j_{(c^j_{\tau_{k-1}}+1):c^j_{\tau_k}}) + \alpha \right) \right] + \beta \right) \right. \\
&\qquad \left. + \sum_{j=1}^{p} \left[\mathbb{I}(c^j_t \neq c^j_u) \left(\mathcal{D}_j(X^j_{(c^j_t+1):c^j_u}) + \alpha \right) \right] + |c_u \setminus \mathcal{L}(c_t)|\, \beta \right\} \\
&= \min_{0 \leq t < u} \left\{ \min_{c_t \in \{\bar{C}_t \,:\, c^j_t \leq c^j_u \;\forall\, j\}} \left[\min_{\boldsymbol{c} \in \mathcal{H}_{c_t}} \sum_{k=1}^{m+1} \left(\sum_{j=1}^{p} \left[\mathbb{I}(c^j_{\tau_k} = \tau_k) \left(\mathcal{D}_j(X^j_{(c^j_{\tau_{k-1}}+1):c^j_{\tau_k}}) + \alpha \right) \right] \right. \right. \right. \\
&\qquad \left. \left. \left. + \ \beta \right) + \sum_{j=1}^{p} \left[\mathbb{I}(c^j_t \neq c^j_u) \left(\mathcal{D}_j(X^j_{(c^j_t+1):c^j_u}) + \alpha \right) \right] + |c_u \setminus \mathcal{L}(c_t)|\, \beta \right] \right\} \\
&= \min_{0 \leq t < u} \left\{ \min_{c_t \in \{\bar{C}_t \,:\, c^j_t \leq c^j_u \;\forall\, j\}} \left[F(c_t) + \sum_{j=1}^{p} \left[\mathbb{I}(c^j_t \neq c^j_u) \left(\mathcal{D}_j(X^j_{(c^j_t+1):c^j_u}) + \alpha \right) \right] \right. \right.
\end{aligned}$$

$$
\begin{aligned}
&\qquad + |c_u \setminus \mathcal{L}(c_t)|\, \beta \Bigg]\Bigg\} \\
&= \min_{0 \le t < u} \left\{ \min_{c_t \in \{\bar{C}_t \,:\, c_t^j \le c_u^j \ \forall\ j\}} \left[F(c_t) + \sum_{j=1}^{p} \left[\mathbb{I}(c_t^j \neq c_u^j) \left(\mathcal{D}_j(X^j_{(c_t^j+1):c_u^j}) + \alpha \right) \right] \right.\right. \\
&\qquad \left.\left. + m(c_t, c_u)\beta \right] \right\},
\end{aligned}
$$

by definition of $m(\cdot,\cdot)$. Hence the result. □

A.2 Proof of Proposition 4.7.1

Proof. Recall that for some changepoint vectors c_r and c_s, $m(c_r, c_s) = |c_s \setminus \mathcal{L}(c_r)|$, so that $m(c_r, c_s)$ represents the number of additional changepoints which have occurred between c_r and c_s (including the changes occurring at c_s, but not those at c_r). We note that since both $cost(\cdot)$ and $\mathbb{I}(\cdot)$ are always non-negative, $m(\cdot,\cdot) > 0$ and $\alpha, \beta > 0$ by definition, then we can add $cost(\boldsymbol{X}_{c_v:c_w}) + m(c_u, c_w)\beta + m(c_v, c_w)\beta + \alpha \sum_{j=1}^{p} \mathbb{I}(c_u^j \neq c_w^j) + \alpha \sum_{j=1}^{p} \mathbb{I}(c_v^j \neq c_w^j)$ to both sides of equation (4.7.4) to obtain the following:

$$
\begin{aligned}
&F(c_u) + cost(\boldsymbol{X}_{c_u:c_v}) + k + cost(\boldsymbol{X}_{c_v:c_w}) + m(c_u, c_w)\beta + m(c_v, c_w)\beta \\
&\qquad + \alpha \sum_{j=1}^{p} \mathbb{I}(c_u^j \neq c_w^j) + \alpha \sum_{j=1}^{p} \mathbb{I}(c_v^j \neq c_w^j) \\
&\quad \ge F(c_v) + cost(\boldsymbol{X}_{c_v:c_w}) + m(c_u, c_w)\beta + m(c_v, c_w)\beta + \alpha \sum_{j=1}^{p} \mathbb{I}(c_u^j \neq c_w^j) \\
&\qquad + \alpha \sum_{j=1}^{p} \mathbb{I}(c_v^j \neq c_w^j). \qquad \text{(A.2.1)}
\end{aligned}
$$

Now since $p \ge m(c_v, c_w)$, $m(c_u, c_w) \ge 0$, $p \ge \sum_{j=1}^{p} \mathbb{I}(c_v^j \neq c_w^j)$ and $\sum_{j=1}^{p} \mathbb{I}(c_u^j \neq c_w^j) \ge 0$, we therefore have $p\beta \ge m(c_v, c_w)\beta$, $m(c_u, c_w)\beta \ge 0$, $\alpha p \ge \alpha \sum_{j=1}^{p} \mathbb{I}(c_v^j \neq c_w^j)$ and $\alpha \sum_{j=1}^{p} \mathbb{I}(c_u^j \neq c_w^j) \ge 0$. Hence, the inequality (A.2.1) becomes

$$
\begin{aligned}
&F(c_u) + cost(\boldsymbol{X}_{c_u:c_v}) + k + cost(\boldsymbol{X}_{c_v:c_w}) + m(c_u, c_w)\beta + p\beta + \alpha \sum_{j=1}^{p} \mathbb{I}(c_u^j \neq c_w^j) + \alpha p \\
&\quad \ge F(c_v) + cost(\boldsymbol{X}_{c_v:c_w}) + m(c_v, c_w)\beta + \alpha \sum_{j=1}^{p} \mathbb{I}(c_v^j \neq c_w^j). \qquad \text{(A.2.2)}
\end{aligned}
$$

Recalling from Proposition 4.7.1 that $k = K - (\alpha + \beta)p$, then by replacing k on the left-hand side of equation (A.2.2) and cancelling the $\alpha p + \beta p$ terms, we have

$$\begin{aligned}
&F(c_u) + cost(\boldsymbol{X}_{c_u:c_v}) + cost(\boldsymbol{X}_{c_v:c_w}) + K + \alpha \sum_{j=1}^{p} \mathbb{I}(c_u^j \neq c_w^j) + m(c_u, c_w)\beta \\
&\quad \geq F(c_v) + cost(\boldsymbol{X}_{c_v:c_w}) + \alpha \sum_{j=1}^{p} \mathbb{I}(c_v^j \neq c_w^j) + m(c_v, c_w)\beta. \qquad \text{(A.2.3)}
\end{aligned}$$

Therefore, recalling that for some changepoint vectors c_r and c_s (where $c_r^j \leq c_s^j$ for all $j = 1, \ldots, p$) we define $cost(\boldsymbol{X}_{c_r:c_s}) = \sum_{j=1}^{p} \left[\mathbb{I}(c_r^j \neq c_s^j) \mathcal{D}_j (X^j_{(c_r^j+1):c_s^j}) \right]$, by using assumption (4.7.2) we have

$$\begin{aligned}
&F(c_u) + cost(\boldsymbol{X}_{c_u:c_w}) + \alpha \sum_{j=1}^{p} \mathbb{I}(c_u^j \neq c_w^j) + m(c_u, c_w)\beta \\
&\quad \geq F(c_v) + cost(\boldsymbol{X}_{c_v:c_w}) + \alpha \sum_{j=1}^{p} \mathbb{I}(c_v^j \neq c_w^j) + m(c_v, c_w)\beta \qquad \text{(A.2.4)} \\
\Longrightarrow\ &F(c_u) + \sum_{j=1}^{p} \left[\mathbb{I}(c_u^j \neq c_w^j) \mathcal{D}_j (X^j_{(c_u^j+1):c_w^j}) \right] + \alpha \sum_{j=1}^{p} \mathbb{I}(c_u^j \neq c_w^j) + m(c_u, c_w)\beta \\
&\quad \geq F(c_v) + \sum_{j=1}^{p} \left[\mathbb{I}(c_v^j \neq c_w^j) \mathcal{D}_j (X^j_{(c_v^j+1):c_w^j}) \right] + \alpha \sum_{j=1}^{p} \mathbb{I}(c_v^j \neq c_w^j) + m(c_v, c_w)\beta \\
&\qquad \text{(A.2.5)} \\
\Longrightarrow\ &F(c_u) + \sum_{j=1}^{p} \left[\mathbb{I}(c_u^j \neq c_w^j) \left(\mathcal{D}_j (X^j_{(c_u^j+1):c_w^j}) + \alpha \right) \right] + m(c_u, c_w)\beta \\
&\quad \geq F(c_v) + \sum_{j=1}^{p} \left[\mathbb{I}(c_v^j \neq c_w^j) \left(\mathcal{D}_j (X^j_{(c_v^j+1):c_w^j}) + \alpha \right) \right] + m(c_v, c_w)\beta. \qquad \text{(A.2.6)}
\end{aligned}$$

Hence, the minimum cost to c_w with c_u as the most recent changepoint vector will always be greater than (or equal to) the minimum cost to c_w with c_v as the most recent changepoint vector. Thus it follows that c_u cannot be a future minimiser of the sets

$$\begin{aligned}
h_{c_w} = \Bigg\{ &F(c_\tau) + \sum_{j=1}^{p} \left[\mathbb{I}(c_\tau^j \neq c_w) \left(\mathcal{D}_j (X^j_{(c_\tau^j+1):c_w^j}) + \alpha \right) \right] + m(c_\tau, c_w)\beta : \\
&\qquad c_\tau \in \bar{C}_\tau;\ \ \tau = 0, \ldots, w-1 \Bigg\} \qquad \text{(A.2.7)}
\end{aligned}$$

and can be removed from the set of c_τ for each future step.

□

A.3 Proof of Proposition 4.7.2

Proof. Suppose that we have some changepoint vectors $c_J \in \bar{C}^J_{\tau^*}$ and $c_{J-1,j^*} \in E^{J-1}_{\tau^*}(c_J)$ such that $c^j_{J-1,j^*} = c^j_J$ for all $j \in \{P \setminus \mathcal{P}_{\tau^*}(c_J)\}$. We use j^* to denote the series which is such that $c^{j^*}_J = \tau^*$ and $c^{j^*}_{J-1} = t_{j^*}$, where t_{j^*} is some time-point such that $t_{j^*} < \tau^*$. Then we can refer to c_{J-1} by c_{J-1,j^*} in order to highlight which is the discrepant series.

Our proof of this proposition consists of the consideration of two lemmas. The first of these is given by Lemma A.3.1.

Lemma A.3.1. *Assume that for every such $c_J \in \bar{C}^J_{\tau^*}$ and $c_{J-1,j^*} \in \Big\{E^{J-1}_{\tau^*}(c_J)$: $c^j_{J-1,j^*} = c^j_J \quad \forall\ j \in P \setminus \mathcal{P}_{\tau^*}(c_J)\}$ we have for some $c_w \in \bar{C}_w$ $(w > \tau^*)$*

$$h_{c_w}(c_J) < h_{c_w}(c_{J-1,j^*}), \tag{A.3.1}$$

i.e. the minimum penalised cost to c_w with c_J as the most recent changepoint vector is lower than when c_{J-1,j^*} is the most recent changepoint vector. Then the following inequality holds:

$$\begin{aligned} &f_{j^*}(\tau^*) + \mathcal{D}_{j^*}(X^{j^*}_{\tau^*:c^{j^*}_w}) + \alpha\mathbb{I}(\tau^* \neq c^{j^*}_w) \\ &\quad < f_{j^*}(t_{j^*}) + \mathcal{D}_{j^*}(X^{j^*}_{t_{j^*}:c^{j^*}_w}) + \alpha\mathbb{I}(t_{j^*} \neq c^{j^*}_w) + \beta\Big(M(c_{J-1,j^*}) - M(c_J)\Big). \end{aligned} \tag{A.3.2}$$

The proof of Lemma (A.3.1) is given in Section A.3.1. Lemma (A.3.1) implies that the univariate minimum penalised cost in series j^* up to $c^{j^*}_w$ with τ^* as the most recent changepoint is lower than the similar penalised cost when t_{j^*} is the most recent changepoint *plus* the term $\beta\Big(M(c_{J-1,j^*}) - M(c_J)\Big)$. We note that this additional term is always either 0 or β. This is because we always have

$$M(c_{J-1,j^*}) = \begin{cases} M(c_J) & \text{if } c^{j^*}_{J-1,j^*} \in \left\{ c^j_{J-1,j^*} \ : \ j \in P \setminus \{\mathcal{P}_{\tau^*}(c_J)\} \right\} \\ M(c_J)+1 & \text{if } c^{j^*}_{J-1,j^*} \notin \left\{ c^j_{J-1,j^*} \ : \ j \in P \setminus \{\mathcal{P}_{\tau^*}(c_J)\} \right\} \end{cases},$$

and so

$$\beta\Big(M(c_{J-1,j^*}) - M(c_J)\Big) = \begin{cases} 0 & \text{if } c^{j^*}_{J-1,j^*} \in \left\{ c^j_{J-1,j^*} \ : \ j \in P \setminus \{\mathcal{P}_{\tau^*}(c_J)\} \right\} \\ \beta & \text{if } c^{j^*}_{J-1,j^*} \notin \left\{ c^j_{J-1,j^*} \ : \ j \in P \setminus \{\mathcal{P}_{\tau^*}(c_J)\} \right\} \end{cases}.$$

In the second stage of our proof, we consider Lemma (A.3.2)

Lemma A.3.2. *Suppose that we now have some changepoint vector c_{J-i} where $c_{J-i} \in \{E^{J-i}_{\tau^*}(c_J) : c^j_{J-i} = c^j_J \ \forall \ j \in P \setminus \mathcal{P}_{\tau^*}(c_J)\}$, with the corresponding set of i discrepant variables denoted by $\{j^*_1, j^*_2, \ldots, j^*_i\} = \{j^*_x : x = 1, 2, \ldots, i\} = \{\mathcal{P}_{\tau^*}(c_J) \setminus \mathcal{P}_{\tau^*}(c_{J-i})\}$. Hence, for each j^*_x $(x = 1, 2, \ldots, i)$ we have $c^{j^*_x}_J = \tau^*$ and $c^{j^*_x}_{J-i} = t_{j^*_x}$, where $t_{j^*_x}$ is some time-point such that $t_{j^*_x} < \tau^*$. Since the inequality (A.3.2) holds for* all *such c_J and c_{J-1,j^*}, then in particular it holds for the changepoint vectors*

$$c_{J-1,j^*_1}, c_{J-1,j^*_2}, \ldots, c_{J-1,j^*_i} \in \{E^{J-1}_{\tau^*}(c_J) : c^j_{J-1} = c^j_J \ \forall \ j \in P \setminus \mathcal{P}_{\tau^*}(c_J)\}$$

where $c^{j^*_x}_{J-1,j^*_x} = t_{j^*_x}$ for each $x = 1, 2, \ldots, i$ and $c^j_{J-1,j^*_x} = c^j_J$ otherwise. Then for such changepoint vectors, it can be shown that

$$h_{c_w}(c_J) < h_{c_w}(c_{J-i}) + \beta \left[\sum_{x=1}^{i} M(c_{J-1,j^*_x}) - (i-1)M(c_J) \right]. \tag{A.3.3}$$

The proof of Lemma A.3.2 is given in Section A.3.2. The inequality (A.3.3) leads to two cases:

1. If $(i-1)M(c_J) \geq \sum_{x=1}^{i} M(c_{J-1,j^*_x})$, then we have $h_{c_w}(c_J) < h_{c_w}(c_{J-i})$.
2. If $(i-1)M(c_J) < \sum_{x=1}^{i} M(c_{J-1,j^*_x})$, then we cannot say whether or not the statement $h_{c_w}(c_J) < h_{c_w}(c_{J-i})$ is true.

Therefore, assuming that case 1 is true, we have the result of the Proposition. □

The next two sections demonstrate the proofs of the Lemmas A.3.1 and A.3.2, used in stages 1 and 2 of the proof of Proposition 4.7.2, respectively.

A.3.1 Proof of Lemma A.3.1

Proof. Recall that we assume inequality (A.3.1) holds. Then we have

$$\begin{aligned}
& h_{c_w}(c_J) < h_{c_w}(c_{J-1,j^*}) \\
\iff & F(c_J) + cost(\boldsymbol{X}_{c_J:c_w}) + \alpha \sum_{j=1}^{p} \mathbb{I}(c_J^j \neq c_w^j) + m(c_J, c_w)\beta \\
& < F(c_{J-1,j^*}) + cost(\boldsymbol{X}_{c_{J-1,j^*}:c_w}) + \alpha \sum_{j=1}^{p} \mathbb{I}(c_{J-1,j^*}^j \neq c_w^j) + m(c_{J-1,j^*}, c_w)\beta \\
\iff & \sum_{j \in P} f_j(c_J^j) + \beta M(c_J) + \sum_{j \in P} \mathcal{D}_j(X^j_{c_J^j:c_w^j}) + \alpha \sum_{j \in P} \mathbb{I}(c_J^j \neq c_w^j) + m(c_J, c_w)\beta \\
& < \sum_{j \in P} f_j(c_{J-1,j^*}^j) + \beta M(c_{J-1,j^*}) + \sum_{j \in P} \mathcal{D}_j(X^j_{c_{J-1,j^*}^j:c_w^j}) + \alpha \sum_{j \in P} \mathbb{I}(c_{J-1,j^*}^j \neq c_w^j) \\
& \qquad + m(c_{J-1,j^*}, c_w)\beta \\
\iff & \sum_{j \in P} f_j(c_J^j) + \beta M(c_J) + \sum_{j \in P} \mathcal{D}_j(X^j_{c_J^j:c_w^j}) + \alpha \sum_{j \in P} \mathbb{I}(c_J^j \neq c_w^j) \\
& < \sum_{j \in P} f_j(c_{J-1,j^*}^j) + \beta M(c_{J-1,j^*}) + \sum_{j \in P} \mathcal{D}_j(X^j_{c_{J-1,j^*}^j:c_w^j}) + \alpha \sum_{j \in P} \mathbb{I}(c_{J-1,j^*}^j \neq c_w^j),
\end{aligned}$$

since $m(c_J, c_w) \leq m(c_{J-1,j^*}, c_w)$. Hence, by separating the terms in this inequality and recollecting them for different groups of variables (i.e. different sets of j's), and cancelling where necessary, we have

$$\begin{aligned}
& h_{c_w}(c_J) < h_{c_w}(c_{J-1,j^*}) \\
\iff & \sum_{j \in \{P \setminus \mathcal{P}_{\tau^*}(c_J)\}} f_j(c_J^j) + \sum_{j \in \{\mathcal{P}_{\tau^*}(c_J) \setminus j^*\}} f_j(\tau^*) + f_{j^*}(\tau^*) \\
& + \sum_{j \in \{P \setminus \mathcal{P}_{\tau^*}(c_J)\}} \mathcal{D}_j(X^j_{c_J^j:c_w^j}) + \sum_{j \in \{\mathcal{P}_{\tau^*}(c_J) \setminus j^*\}} \mathcal{D}_j(X^j_{\tau^*:c_w^j}) + \mathcal{D}_{j^*}(X^{j^*}_{\tau^*:c_w^{j^*}}) \\
& + \alpha \left[\sum_{j \in \{P \setminus \mathcal{P}_{\tau^*}(c_J)\}} \mathbb{I}(c_J^j \neq c_w^j) + \sum_{j \in \{\mathcal{P}_{\tau^*}(c_J) \setminus j^*\}} \mathbb{I}(\tau^* \neq c_w^j) + \mathbb{I}(\tau^* \neq c_w^{j^*}) \right] + \beta M(c_J)
\end{aligned}$$

$$
\begin{aligned}
&< \sum_{j\in\{P\setminus\mathcal{P}_{\tau^*}(c_J)\}} f_j(c^j_{J-1,j^*}) + \sum_{j\in\{\mathcal{P}_{\tau^*}(c_J)\setminus j^*\}} f_j(\tau^*) + f_{j^*}(t_{j^*})\\
&\quad + \sum_{j\in\{P\setminus\mathcal{P}_{\tau^*}(c_J)\}} \mathcal{D}_j(X^j_{c^j_{J-1,j^*}:c^j_w}) + \sum_{j\in\{\mathcal{P}_{\tau^*}(c_J)\setminus j^*\}} \mathcal{D}_j(X^j_{\tau^*:c^j_w}) + \mathcal{D}_{j^*}(X^{j^*}_{t_{j^*}:c^{j^*}_w})\\
&\quad + \alpha\left[\sum_{j\in\{P\setminus\mathcal{P}_{\tau^*}(c_J)\}} \mathbb{I}(c^j_{J-1,j^*} \neq c^j_w) + \sum_{j\in\{\mathcal{P}_{\tau^*}(c_J)\setminus j^*\}} \mathbb{I}(\tau^* \neq c^j_w) + \mathbb{I}(t_{j^*} \neq c^{j^*}_w)\right]\\
&\quad + \beta M(c_{J-1,j^*})\\
\iff &\sum_{j\in\{P\setminus\mathcal{P}_{\tau^*}(c_J)\}} \left[f_j(c^j_J) + \mathcal{D}_j(X^j_{c^j_J:c^j_w}) + \alpha\mathbb{I}(c^j_J \neq c^j_w)\right]\\
&\quad + \sum_{j\in\{\mathcal{P}_{\tau^*}(c_J)\setminus j^*\}} \left[f_j(\tau^*) + \mathcal{D}_j(X^j_{\tau^*:c^j_w}) + \alpha\mathbb{I}(\tau^* \neq c^j_w)\right]\\
&\quad + \left[f_{j^*}(\tau^*) + \mathcal{D}_{j^*}(X^{j^*}_{\tau^*:c^{j^*}_w}) + \alpha\mathbb{I}(\tau^* \neq c^{j^*}_w)\right] + \beta M(c_J)\\
&< \sum_{j\in\{P\setminus\mathcal{P}_{\tau^*}(c_J)\}} \left[f_j(c^j_{J-1,j^*}) + \mathcal{D}_j(X^j_{c^j_{J-1,j^*}:c^j_w}) + \alpha\mathbb{I}(c^j_{J-1,j^*} \neq c^j_w)\right]\\
&\quad + \sum_{j\in\{\mathcal{P}_{\tau^*}(c_J)\setminus j^*\}} \left[f_j(\tau^*) + \mathcal{D}_j(X^j_{\tau^*:c^j_w}) + \alpha\mathbb{I}(\tau^* \neq c^j_w)\right]\\
&\quad + \left[f_{j^*}(t_{j^*}) + \mathcal{D}_{j^*}(X^{j^*}_{t_{j^*}:c^{j^*}_w}) + \alpha\mathbb{I}(t_{j^*} \neq c^{j^*}_w)\right] + \beta M(c_{J-1,j^*})\\
\iff &\sum_{j\in\{P\setminus\mathcal{P}_{\tau^*}(c_J)\}} \left[f_j(c^j_J) + \mathcal{D}_j(X^j_{c^j_J:c^j_w}) + \alpha\mathbb{I}(c^j_J \neq c^j_w)\right]\\
&\quad + \left[f_{j^*}(\tau^*) + \mathcal{D}_{j^*}(X^{j^*}_{\tau^*:c^{j^*}_w}) + \alpha\mathbb{I}(\tau^* \neq c^{j^*}_w)\right] + \beta M(c_J)\\
&< \sum_{j\in\{P\setminus\mathcal{P}_{\tau^*}(c_J)\}} \left[f_j(c^j_{J-1,j^*}) + \mathcal{D}_j(X^j_{c^j_{J-1,j^*}:c^j_w}) + \alpha\mathbb{I}(c^j_{J-1,j^*} \neq c^j_w)\right]\\
&\quad + \left[f_{j^*}(t_{j^*}) + \mathcal{D}_{j^*}(X^{j^*}_{t_{j^*}:c^{j^*}_w}) + \alpha\mathbb{I}(t_{j^*} \neq c^{j^*}_w)\right] + \beta M(c_{J-1,j^*}).
\end{aligned}
$$

Since we are assuming that (A.3.1) is true for all $c_J \in \bar{C}^J_{\tau^*}$ and $c_{J-1} \in E^{J-1}_{\tau^*}(c_J)$ such that $c^j_{J-1,j^*} = c^j_J$ for all $j \in \{P \setminus \mathcal{P}_{\tau^*}(c_J)\}$, then this implies that we must have:

$$
\begin{aligned}
&f_{j^*}(\tau^*) + \mathcal{D}_{j^*}(X^{j^*}_{\tau^*:c^{j^*}_w}) + \alpha\mathbb{I}(\tau^* \neq c^{j^*}_w)\\
&\qquad < f_{j^*}(t_{j^*}) + \mathcal{D}_{j^*}(X^{j^*}_{t_{j^*}:c^{j^*}_w}) + \alpha\mathbb{I}(t_{j^*} \neq c^{j^*}_w) + \beta M(c_{J-1,j^*}) - \beta M(c_J), \qquad \text{(A.3.4)}
\end{aligned}
$$

and hence the result is proved. □

A.3.2 Proof of Lemma A.3.2

Proof. Recall that inequality (A.3.2) holds in particular for the changepoint vectors c_{J-1,j_1^*}, c_{J-1,j_2^*}, $\ldots$, $c_{J-1,j_i^*} \in \{E_{\tau^*}^{J-1}(c_J) : c_{J-1}^j = c_J^j \ \forall \ j \in P \setminus \mathcal{P}_{\tau^*}(c_J)\}$. Hence, we have

$$\begin{aligned}
&h_{c_w}(c_J)\\
&= \sum_{j \in \{P \setminus \mathcal{P}_{\tau^*}(c_J)\}} \Big[f_j(c_J^j) + \mathcal{D}_j(X^j_{c_J^j : c_w^j}) + \alpha \mathbb{I}(c_J^j \neq c_w^j)\Big]\\
&\quad + \sum_{j \in \{\mathcal{P}_{\tau^*}(c_J) \setminus \{j_1^*, j_2^*, \ldots, j_i^*\}\}} \Big[f_j(\tau^*) + \mathcal{D}_j(X^j_{\tau^* : c_w^j}) + \alpha \mathbb{I}(\tau^* \neq c_w^j)\Big]\\
&\quad + \sum_{j \in \{j_1^*, j_2^*, \ldots, j_i^*\}} \Big[f_j(\tau^*) + \mathcal{D}_j(X^j_{\tau^* : c_w^j}) + \alpha \mathbb{I}(\tau^* \neq c_w^j)\Big] + \beta M(c_J) + \beta m(c_J, c_w)\\
&< \sum_{j \in \{P \setminus \mathcal{P}_{\tau^*}(c_J)\}} \Big[f_j(c_J^j) + \mathcal{D}_j(X^j_{c_J^j : c_w^j}) + \alpha \mathbb{I}(c_J^j \neq c_w^j)\Big]\\
&\quad + \sum_{j \in \{\mathcal{P}_{\tau^*}(c_J) \setminus \{j_1^*, j_2^*, \ldots, j_i^*\}\}} \Big[f_j(\tau^*) + \mathcal{D}_j(X^j_{\tau^* : c_w^j}) + \alpha \mathbb{I}(\tau^* \neq c_w^j)\Big]\\
&\quad + \sum_{j \in \{j_1^*, j_2^*, \ldots, j_i^*\}} \Big[f_j(t_j) + \mathcal{D}_j(X^j_{t_j : c_w^j}) + \alpha \mathbb{I}(t_j \neq c_w^j)\Big] + \sum_{x=1}^{i} \beta M(c_{J-1,j_x^*}) - \sum_{x=1}^{i} \beta M(c_J)\\
&\quad + \beta M(c_J) + \beta m(c_J, c_w),
\end{aligned}$$

using inequality (A.3.2). Now, using the fact that $c_J^j = c_{J-i}^j$ for $j \in \{P \setminus \mathcal{P}_{\tau^*}(c_J)\}$, we have

$$\begin{aligned}
&h_{c_w}(c_J)\\
&< \sum_{j \in \{P \setminus \mathcal{P}_{\tau^*}(c_J)\}} \Big[f_j(c_{J-i}^j) + \mathcal{D}_j(X^j_{c_{J-i}^j : c_w^j}) + \alpha \mathbb{I}(c_{J-i}^j \neq c_w^j)\Big]\\
&\quad + \sum_{j \in \{\mathcal{P}_{\tau^*}(c_J) \setminus \{j_1^*, j_2^*, \ldots, j_i^*\}\}} \Big[f_j(\tau^*) + \mathcal{D}_j(X^j_{\tau^* : c_w^j}) + \alpha \mathbb{I}(\tau^* \neq c_w^j)\Big]\\
&\quad + \sum_{j \in \{j_1^*, j_2^*, \ldots, j_i^*\}} \Big[f_j(t_j) + \mathcal{D}_j(X^j_{t_j : c_w^j}) + \alpha \mathbb{I}(t_j \neq c_w^j)\Big] + \beta m(c_J, c_w) + \sum_{x=1}^{i} \beta M(c_{J-1,j_x^*})\\
&\quad - i\beta M(c_J) + \beta M(c_J)\\
&\leq \sum_{j \in \{P \setminus \mathcal{P}_{\tau^*}(c_J)\}} \Big[f_j(c_{J-i}^j) + \mathcal{D}_j(X^j_{c_{J-i}^j : c_w^j}) + \alpha \mathbb{I}(c_{J-i}^j \neq c_w^j)\Big]
\end{aligned}$$

$$
\begin{aligned}
&+ \sum_{j \in \{\mathcal{P}_{\tau^*}(c_J) \setminus \{j_1^*, j_2^*, \ldots, j_i^*\}\}} \left[f_j(\tau^*) + \mathcal{D}_j(X^j_{\tau^*:c_w^j}) + \alpha \mathbb{I}(\tau^* \neq c_w^j) \right] \\
&+ \sum_{j \in \{j_1^*, j_2^*, \ldots, j_i^*\}} \left[f_j(t_j) + \mathcal{D}_j(X^j_{t_j:c_w^j}) + \alpha \mathbb{I}(t_j \neq c_w^j) \right] + \beta m(c_{J-i}, c_w) + \sum_{x=1}^{i} \beta M(c_{J-1,j_x^*}) \\
&- i\beta M(c_J) + \beta M(c_J) \\
= {}& h_{c_w}(c_{J-i}) + \beta \left[\sum_{x=1}^{i} M(c_{J-1,j_x^*}) - (i-1) M(c_J) \right].
\end{aligned}
$$

The second inequality here is due to the fact that $m(c_J, c_w) \leq m(c_{J-i}, c_w)$, since the i discrepant variables have changes at locations other than τ^*, and hence may introduce additional changepoint locations. Therefore, we have

$$
h_{c_w}(c_J) < h_{c_w}(c_{J-i}) + \beta \left[\sum_{x=1}^{i} M(c_{J-1,j_x^*}) - (i-1) M(c_J) \right],
$$

and hence the result. □

Appendix B

Proofs for 'Approximate Segmentation of Multivariate Time Series'

B.1 Proof of Proposition 5.2.1

Proof. We first define α-PELT as the univariate PELT method with the penalty set as α. Suppose that a performance of SMOP on the p-variate time series $\boldsymbol{X}_{1:n}$ produces an optimal configuration of changepoints, denoted by $(\boldsymbol{\tau}^1, \boldsymbol{\tau}^2, \ldots, \boldsymbol{\tau}^p)$. Here $\boldsymbol{\tau}^j$ represents a vector containing the changepoint locations estimated in the j^{th} variable, so that $\boldsymbol{\tau}^j = (\tau_1^j, \tau_2^j, \ldots, \tau_{m_j}^j)$, with m_j denoting the number of (true) univariate changepoints detected in the j^{th} variable. In particular, for variable j^* we have $\boldsymbol{\tau}^{j^*} = (\tau_1^{j^*}, \ldots, \tau_{m_{j^*}}^{j^*})$. Note that for each $j = 1, \ldots, p$ we set $\tau_0^j = 0$ and $\tau_{m_j+1}^j = n$.

Let m be the total number of *multivariate* changepoints detected, so that

$$m = \left| \bigcup_{j=1}^{p} \boldsymbol{\tau}^j \right| \leq \sum_{j=1}^{p} m_j. \tag{B.1.1}$$

Due to the assumption of zero cross-correlation in our model and our interest in series j^*, the cost of this optimal configuration $(\boldsymbol{\tau}^1, \boldsymbol{\tau}^2, \ldots, \boldsymbol{\tau}^p)$ produced by SMOP can be decomposed into the optimal cost for series j^* plus the optimal cost for all the other series:

$$\begin{aligned}
&\sum_{j=1}^{p}\sum_{i=1}^{m_j+1}\mathcal{D}_j\left(X^j_{(\tau^j_{i-1}+1):\tau^j_i}\right)+\sum_{j=1}^{p}(m_j+1)\alpha+(m+1)\beta\\
&=\sum_{j\in J}\sum_{i=1}^{m_j+1}\mathcal{D}_j\left(X^j_{(\tau^j_{i-1}+1):\tau^j_i}\right)+\sum_{j\in J}(m_j+1)\alpha+(m_J+1)\beta\\
&\quad+\sum_{i=1}^{m_{j^*}+1}\mathcal{D}_{j^*}\left(X^{j^*}_{(\tau^{j^*}_{i-1}+1):\tau^{j^*}_i}\right)+(m_{j^*}+1)\alpha+m^*\beta, \qquad \text{(B.1.2)}
\end{aligned}$$

where $J=\{1,\ldots,p\setminus j^*\}$, $m_J=\left|\bigcup_{j\in J}\boldsymbol{\tau}^j\right|$ and $m^*=\left|\boldsymbol{\tau}_{j^*}\setminus\bigcup_{j\in J}\boldsymbol{\tau}^j\right|$. Note that we are continuing with the convention that the penalty terms are also added for the 'changepoint' at the end of the data.

Since, by assumption, the performance of α-PELT detects no changepoints in variable j^*, then we must have that

$$\mathcal{D}_{j^*}(X^{j^*}_{1:n})+\alpha\leq\sum_{i=1}^{m_{j^*}+1}\mathcal{D}_{j^*}\left(X^{j^*}_{(\tau^{j^*}_{i-1}+1):\tau^{j^*}_i}\right)+(m_{j^*}+1)\alpha \qquad \text{(B.1.3)}$$

for all possible m_{j^*}. Therefore, since $m^*\geq 0$ and $\beta\geq 0$, we must have that

$$\mathcal{D}_{j^*}(X^{j^*}_{1:n})+\alpha\leq\sum_{i=1}^{m_{j^*}+1}\mathcal{D}_{j^*}\left(X^{j^*}_{(\tau^{j^*}_{i-1}+1):\tau^{j^*}_i}\right)+(m_{j^*}+1)\alpha+m^*\beta. \qquad \text{(B.1.4)}$$

Adding the terms $\sum_{j\in J}\sum_{i=1}^{m_j+1}\mathcal{D}_j\left(X^j_{(\tau^j_{i-1}+1):\tau^j_i}\right)+\sum_{j\in J}(m_j+1)\alpha+(m_J+1)\beta$ to both sides of the inequality (B.1.4), we have

$$\begin{aligned}
&\sum_{j\in J}\sum_{i=1}^{m_j+1}\mathcal{D}_j\left(X^j_{(\tau^j_{i-1}+1):\tau^j_i}\right)+\sum_{j\in J}(m_j+1)\alpha+(m_J+1)\beta+\mathcal{D}_{j^*}(X^{j^*}_{1:n})+\alpha\\
&\leq\sum_{j\in J}\sum_{i=1}^{m_j+1}\mathcal{D}_j\left(X^j_{(\tau^j_{i-1}+1):\tau^j_i}\right)+\sum_{j\in J}(m_j+1)\alpha+(m_J+1)\beta\\
&\quad+\sum_{i=1}^{m_{j^*}+1}\mathcal{D}_{j^*}\left(X^{j^*}_{(\tau^{j^*}_{i-1}+1):\tau^{j^*}_i}\right)+(m_{j^*}+1)\alpha+m^*\beta \qquad \text{(B.1.5)}
\end{aligned}$$

The RHS of inequality (B.1.5) is equal to the RHS of equation (B.1.2), which is in turn equal to the cost of the optimal changepoint configuration provided by SMOP.

The LHS of inequality (B.1.5) is equal to the cost of $\boldsymbol{X}_{1:n}$ under the configuration where variable j contains the changepoints $\boldsymbol{\tau}^j$ for each $j \in J$, and variable j^* contains no changepoints. Therefore, since the RHS of (B.1.5) is the optimal (i.e. minimal) cost over all possible changepoint configurations, this tells us that the LHS cannot be less than the RHS, and hence they must be equal. Consequently, we must have $m_{j^*} = 0$, with $\boldsymbol{\tau}^{j^*} = \emptyset$ (and hence $m^* = 0$). This implies that the optimal changepoint configuration of $\boldsymbol{X}_{1:n}$ contains no changepoints in variable j^*.

This result shows that if no changepoints are detected in variable j^* using α-PELT, then no changepoints will be present in variable j^* in the optimal changepoint configuration under the subset-multivariate changepoint model obtained by SMOP.

□

B.2 Proof of Proposition 5.2.2

Proof. Suppose that for each variable $j^* = 1, \dots, p$ within the multivariate times series $\boldsymbol{X}_{1:n}$, we apply α-PELT to obtain a set of changepoint locations $\boldsymbol{\tau}^{j^*}$. Set $\boldsymbol{\tau} = \bigcup_{j^*=1}^{p} \boldsymbol{\tau}^{j^*}$. For some window size w, we can apply both hard subset restriction and soft subset restriction to this set of changepoint locations, resulting in two possible sets of affected variable subsets for each $\tau \in \boldsymbol{\tau}$. For each such τ, denote these respective sets by $\mathcal{S}_{\tau}^{(hard)}$ and $\mathcal{S}_{\tau}^{(soft)}$. Note that by the definition of hard subset restriction (see Algorithm 4), $\mathcal{S}_{\tau}^{(hard)}$ is actually a set containing one element: $\mathcal{S}_{\tau}^{(hard)} = \{s_{\tau,(hard)}\}$. Also denote the sets $\boldsymbol{\mathcal{S}}^{hard} = \{\mathcal{S}_{\tau}^{(hard)}\}_{\tau \in \boldsymbol{\tau}}$ and $\boldsymbol{\mathcal{S}}^{soft} = \{\mathcal{S}_{\tau}^{(soft)}\}_{\tau \in \boldsymbol{\tau}}$.

Now suppose we have some $\tau^* \in \boldsymbol{\tau}$, along with the affected variable subset $s_{\tau^*,(hard)}$ and the set of subsets $\mathcal{S}_{\tau^*}^{(soft)}$. From the definition of soft subset restriction in Algorithm 5, it can be seen that the affected variable subsets $s_{\tau^*,(soft)} \in \mathcal{S}_{\tau^*}^{(soft)}$ are formed as follows. Suppose B_k is the set of binary permutations of length k, $J_{\tau^*} = \{j : s^j_{\tau^*,(hard)} = 0, \boldsymbol{\tau}^j \neq \emptyset\}$ and $J^*_{\tau^*} = \{j : s^j_{\tau^*,(hard)} = 1\}$. Then $\mathcal{S}_{\tau^*}^{(soft)}$ is a set of $|B_{|J_{\tau^*}|}|$ subsets such that $s^j_{\tau^*,(soft)} = 1$ for all $j \in J^*_{\tau^*}$ and $s^j_{\tau^*,(soft)} = b^j$ for all $j \in J_{\tau^*}$, where $b \in B_{|J_{\tau^*}|}$ is a binary permutation with a different $s_{\tau^*,(soft)}$ corresponding to a different b. Since every such $b \in B_{|J_{\tau^*}|}$ is considered, the zero permutation

$b = (0, \ldots, 0)$ is always considered in particular. For the $s_{\tau^*,(soft)}$ in this case, we have

$$s^{j}_{\tau^*,(soft)} = \begin{cases} 1 & \text{if } j \in J^*_{\tau^*} \\ 0 & \text{if } j \in J_{\tau^*} \end{cases} . \tag{B.2.1}$$

Since the RHS of equation (B.2.1) is also equivalent to $s^{j}_{\tau^*,(hard)}$, we therefore have $s^{j}_{\tau^*,(soft)} = s^{j}_{\tau^*,(hard)}$ for the case when b is the zero permutation. As such, we always have $s_{\tau^*,(hard)} \in \mathcal{S}^{(soft)}_{\tau^*}$. This is true for all $\tau^* \in \boldsymbol{\tau}$, and so we have $\boldsymbol{\mathcal{S}}^{hard} \subseteq \boldsymbol{\mathcal{S}}^{soft}$.

Now consider the following two possible sets of changepoint vectors produced by combining the set of changepoint locations $\boldsymbol{\tau}$ with the two respective sets of affected variable subsets $\boldsymbol{\mathcal{S}}^{hard}$ and $\boldsymbol{\mathcal{S}}^{soft}$: $C_{\boldsymbol{\tau},\boldsymbol{\mathcal{S}}^{hard},n}$ and $C_{\boldsymbol{\tau},\boldsymbol{\mathcal{S}}^{soft},n}$. These sets denote all possible changepoint vectors using hard restriction and soft restriction, respectively. Since it has been shown that $\boldsymbol{\mathcal{S}}^{hard} \subseteq \boldsymbol{\mathcal{S}}^{soft}$, then because $C_{\boldsymbol{\tau},\boldsymbol{\mathcal{S}}^{hard},n}$ and $C_{\boldsymbol{\tau},\boldsymbol{\mathcal{S}}^{soft},n}$ are formed using the same set of changepoint locations $\boldsymbol{\tau}$, we must have $C_{\boldsymbol{\tau},\boldsymbol{\mathcal{S}}^{hard},n} \subseteq C_{\boldsymbol{\tau},\boldsymbol{\mathcal{S}}^{soft},n}$.

Therefore, if we perform A-SMOP using soft subset restriction, then there are two possible outcomes:

1. All possible changepoint vectors in the optimal configuration lie in $C_{\boldsymbol{\tau},\boldsymbol{\mathcal{S}}^{hard},n}$. Hence, application of hard restriction would also obtain the same optimal configuration, so that

$$\boldsymbol{c}^{soft} = \boldsymbol{c}^{hard} \quad \text{and} \quad F^{soft} = F^{hard}.$$

2. There exists at least one changepoint vector $c \in \boldsymbol{c}^{soft}$ such that

$$c \in \{C_{\boldsymbol{\tau},\boldsymbol{\mathcal{S}}^{soft},n} \setminus C_{\boldsymbol{\tau},\boldsymbol{\mathcal{S}}^{hard},n}\}.$$

   Hence, the overall cost of this configuration is lower than the cost of the optimal configuration obtained using hard restriction, i.e. $F^{soft} < F^{hard}$.

Therefore, we must have $F^{soft} \leq F^{hard}$, so that soft subset restriction produces a segmentation which has the same or a lower cost than the segmentation produced using hard subset restriction. □

Appendix C

changepointmv: An R Package for Multivariate Changepoint Analysis

A range of methodologies have been proposed for the detection of multivariate changepoints. However, despite the variety of these contributions, only a handful of R packages are available for implementing such methods. The most notable of these are the ecp and bcp packages of James and Matteson (2014) and Erdman and Emerson (2007) respectively. These packages take separate approaches to the multivariate changepoint problem: the ecp package utilises non-parametric energy statistics, whereas the bcp package adopts a Bayesian MCMC framework.

Recall from Section 4.2.1 that multivariate changepoints can be classified as *fully-multivariate* or *subset-multivariate*. For the former, all variables are changing at the changepoint location. For the latter, only a subset of the variables are affected by the change. The ecp and bcp packages are similar in that they both implement only fully-multivariate detection procedures. To our knowledge, no publicly available R package exists which permits the explicit detection of subset-multivariate changepoints. Motivated by this, we present the **changepointmv** R package. This is a software suite which implements both the SMOP and A-SMOP methodology described in Chapters 4 and 5. These methods detect both fully- and subset-multivariate changepoints through the use of a parametric framework, allowing for a variety of distributional models and types of change. This software is available for download at http://www.lancaster.ac.uk/~pickerin/software.html.

This appendix is structured as follows. We begin with an outline of the package structure in Section C.1. In Section C.2 we discuss the two main functions within the **changepointmv** package: smop and asmop. Case studies considering the application of asmop function are examined in Section C.3.

C.1 Package Structure and the cptmv class

There are two main functions within the **changepointmv** package implementing the multivariate changepoint detection methodology developed within this thesis. These are:

- smop: Performs the SMOP algorithm, described in Chapter 4.
- asmop: Performs the A-SMOP algorithm, described in Chapter 5.

The package also introduces a new S4 object class called cptmv. In a similar manner to the cpt class from the changepoint package (Killick et al., 2015), the cptmv class is used to store information relating to the results of the multivariate changepoint analysis performed by SMOP or A-SMOP. In particular, an object of type cptmv contains the following slots:

- data.set - an $n \times p$ matrix containing the sequence of multivariate observations. Each row represents a different p-variate observation, and each column represents the n-length series for that specific variable.
- cost.func - a character object providing the name of the function used to calculate the (unpenalised) cost, e.g. "norm.mean" for changes in the mean of multivariate Normally-distributed observations.
- cpt.type - a character object denoting the type of change(s) which are being detected, e.g. "mean" for mean, "mean and variance" for both mean and variance.

- `alpha` - the numeric value of the α penalty used within the SMOP and A-SMOP detection algorithms (see Chapter 4 for details).
- `beta` - the numeric value of the β penalty used within the SMOP and A-SMOP detection algorithms (see Chapter 4 for details).
- `num.cpt.vecs` - the total number of possible changepoint vectors considered by the detection procedure.
- `cpt.vecs` - a p-column matrix containing the final set of detected changepoint vectors. Each row contains a different changepoint vector.
- `like` - the numeric value of penalised likelihood of the estimated segmentation.
- `cpts` - a numeric vector containing the set of detected changepoint locations.
- `subsets` - a p-column matrix of logical values representing the sets of affected variables corresponding to each detected changepoint. The i^{th} row represents the subset for the i^{th} changepoint in `cpts`.
- `runtime` - the numeric value of the running time of the detection procedure, in seconds.

As `cptmv` is an S4 object, these slots can be accessed using the @ symbol (analgous to the $ symbol for S3 objects). To enable the end-user to easily visualise the results of their changepoint analysis, the **changepointmv** package contains a `plot` method for the `cptmv` class. The behaviour of this method is dependent of the type of change being detected. For example, for changes in variance the changepoint locations are shown by vertical lines. For changes in mean, the mean values are also shown using horizontal lines in each segment.

Recall from Chapter 4 that changepoint vectors are used to encapsulate information about *both* the locations of changepoints and the subsets of variables in which they occur. Specifically, a changepoint vector at a given time-point t, denoted c_t, contains the most recent changepoint locations in each variable up to and including time t. The changepoint vectors found in `cpt.vecs` represent the unique distinct changepoint vectors detected for the series. This is simply a different representation of the information contained in cpts and subsets.

Within the following sections we examine the key functions of the **changepointmv** package.

C.2 The smop and asmop Functions

The smop and asmop functions within the **changepointmv** package are used to implement the SMOP and A-SMOP algorithms presented in Chapters 4 and 5, respectively. Both functions share common architecture, but also have important differences in their arguments and output. Within this section we describe how to invoke these functions, delineate their structure and arguments, and consider some illustrative examples.

C.2.1 Usage

The smop function has the following structure:

```
smop(data, alpha = 2 * log(nrow(data)),
     beta = 2 * log(ncol(data)) * log(nrow(data)), min.dist = 2,
     cost.func = "norm.meanvar.seglen", class = TRUE, verbose = TRUE)
```

The details of these arguments are as follows:

- data – An $n \times p$ matrix representing a length n multivariate time series containing observations of p variables.

- alpha – The variable-specific penalty, used to penalise the addition of a given changepoint into a given variable. A non-negative numeric value.

- beta – The multivariate penalty, used to penalise the addition of a new changepoint into the model regardless of the variable(s) in which it occurs. A non-negative numeric value.

- min.dist – The minimum distance allowed between any two changepoints. Required to have an integer value of at least 2.

- cost.func – The name of the multivariate cost function used by the method, given as a string. Possible values include "norm.mean", "norm.var", "norm.meanvar", "norm.mean.seglen", "norm.var.seglen" and "norm.meanvar.seglen". Details of these of values are provided below.

- class – A logical value. If TRUE then an object of class cptmv is returned. If FALSE, a generic list is returned with identical slots to those in the cptmv object.

- verbose – A logical value. If TRUE then information regarding the check-list of possible changepoint vectors is printed during the algorithm.

The asmop function has a similar structure to smop:

```
asmop(data, alpha = 2 * log(nrow(data)),
  beta = 2 * log(ncol(data)) * log(nrow(data)), min.dist = 2,
  cost.func = "norm.meanvar.seglen", window.size,
  hard.restrict = TRUE, class = TRUE, verbose = FALSE)
```

The two additional arguments are:

- window.size – A non-negative integer representing the size of the window considered to the left and right of a given changepoint when performing subset restriction. Note that the choice of this value is entirely context dependent. See Section 5.2.2 for details on the role of this value and how it affects the behaviour of A-SMOP.

- hard.restrict – A logical value. If TRUE then hard subset restriction is used. If FALSE then soft subset restriction is used. See Section 5.2.2 for details regarding the differences between hard and soft subset restriction.

The argument cost.func, used by both smop and asmop, can take a range of possible functions, depending on the distribution of the data being considered and the type of change(s) being sought. To date, the following possible values have been implemented:

- `"norm.mean"` – Used for detecting changes in mean in multivariate Normally distributed data. Assumes fixed variance parameters (= 1) for each variable. The mean parameters are set to their maximum likelihood estimates. If the true variance is not 1, then the data can first be normalised for analysis by calculating the sample variance for each variable and dividing it into each observation in that variable.

- `"norm.var"` – Used for detecting changes in variance in multivariate Normally distributed data. Assumes fixed mean parameters (= 0) for each variable. The variance parameters are set to their maximum likelihood estimates. If the mean is not 0, then the data can first be normalised for analysis by calculating the sample mean for each variable and subtracting it from each observation in that variable.

- `"norm.meanvar"` – Used for detecting changes in both mean and variance in multivariate Normally distributed data. The mean and variance parameters are set to their maximum likelihood estimates.

- `"norm.mean.seglen"`, `"norm.var.seglen"` and `"norm.meanvar.seglen"` – Identical to `"norm.mean"`, `"norm.var"` and `"norm.meanvar"`, respectively, except these contain a $\log(\boldsymbol{seg}ment\ \boldsymbol{len}gth)$ penalty term in the likelihood for each variable. These functions are included to allow for the use of penalties akin to the modified BIC (Zhang and Siegmund, 2007).

C.2.2 Illustrative Examples

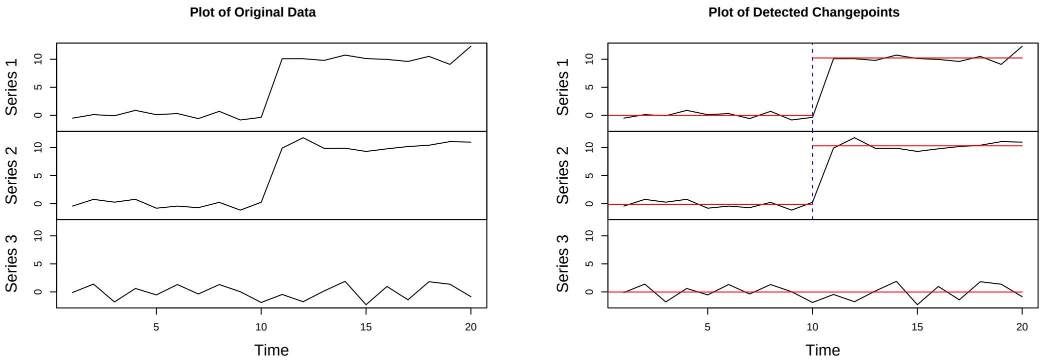
We now consider some examples that demonstrate the performance of the `smop` and `asmop` functions. The first example we consider is a single change in mean at the mid-point of 2 out of 3 Normally distributed variables, displayed in Figure C.2.1(a).

```
library(zoo) # for plotting

n = 20; p = 3

set.seed(100)
```

```
data.meanchange = matrix(NA, n, p)
data.meanchange[,1] = c( rnorm(n/2, 0, 1), rnorm(n/2, 10, 1) )
data.meanchange[,2] = c( rnorm(n/2, 0, 1), rnorm(n/2, 10, 1) )
data.meanchange[,3] = rnorm(n, 0, 1)
# plot multivariate time series:
plot.zoo(data.meanchange, ylim=range(data.meanchange))
```



C.2.1(a): Original data.

C.2.1(b): Detected changepoints.

Figure C.2.1: An example of a single change in mean at the mid-point of 2 out of 3 Normally distributed variables. Plot (a) shows the original data, and plot (b) shows the data with the changepoints detected by both `smop` and `asmop` (dashed blue lines), along with the means of the segments (sold red lines).

We apply both `smop` and `asmop` to this series. Since we are searching for changes in mean only, we set `cost.func` to `"norm.mean.seglen"` and use the default values for all other arguments. For the `asmop` function, a choice of value for `window.size` is also necessary. This value is entirely context dependent. Informally, it can be thought of as a tolerance for the slight misestimation of potential multivariate changepoint locations within the initial stages of the algorithm. A larger value means that estimated changepoints across different variables which are ‘close’ (in time) are more likely to be treated as the same changepoint across those variables. Since this example series is relatively short, we wish to have a small window size and hence set `window.size` to 2.

```
meanchange.results.smop = smop(data.meanchange,
    cost.func="norm.mean.seglen")
meanchange.results.asmop = asmop(data.meanchange,
    cost.func="norm.mean.seglen", window.size=2)
plot(meanchange.results.smop)
plot(meanchange.results.asmop)
# see how many changepoint vectors were considered:
meanchange.results.smop@num.cpt.vecs # 5833
meanchange.results.asmop@num.cpt.vecs # 3
# view the running times
meanchange.results.smop@runtime # 10.29 seconds
meanchange.results.asmop@runtime # 0.02 seconds
```

As demonstrated in Figure C.2.1(b), both the `smop` and `asmop` functions identify the true changepoint locations in the correct variables.

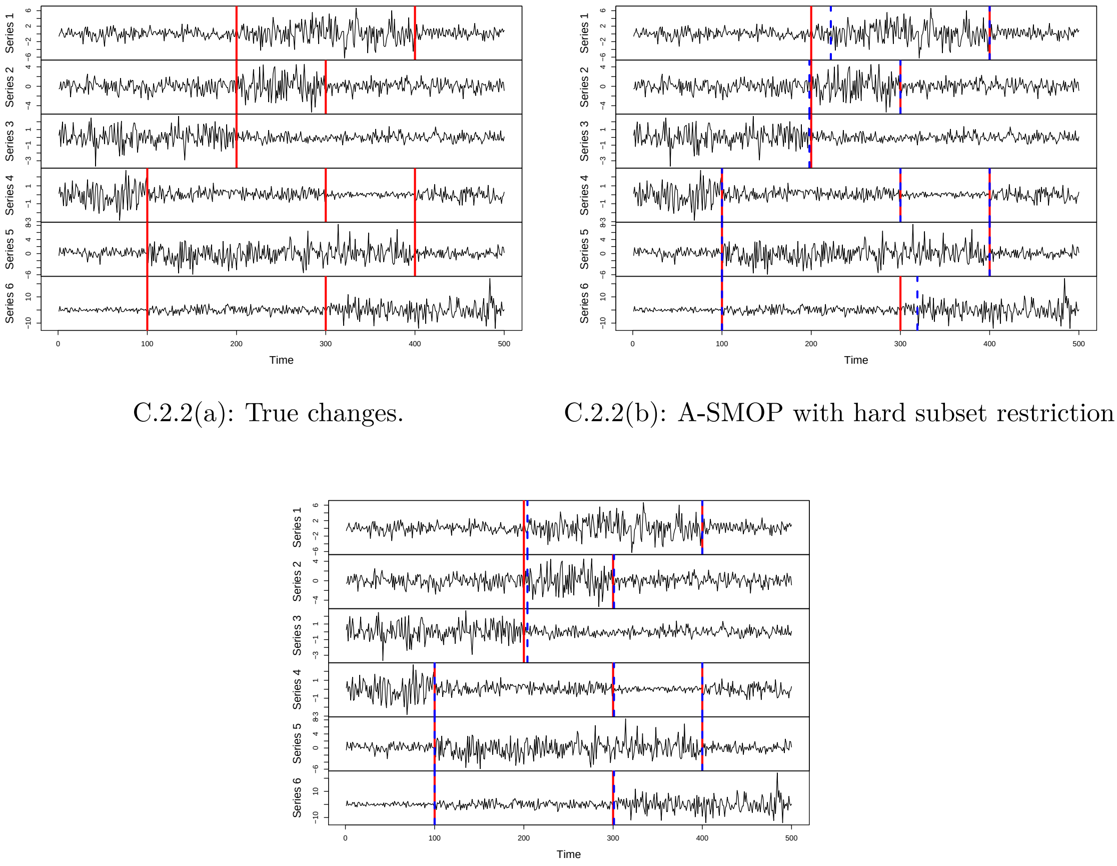
The key difference between the `smop` and `asmop` functions is the number of possible changepoint vectors considered within the procedure, and hence the running times of the functions. Indeed, the `num.cpt.vecs` slot of the results show that `smop` considers 5833 changepoint vectors, whereas `asmop` considers only 3. This is reflected in their running times, with `smop` requiring 10.29 seconds and `asmop` requiring only 0.02 seconds on an Intel i5 2.5GHz processor. We note that the changepoint vectors considered by `smop` will always include those considered by `asmop`. This consideration of additional changepoint vectors means that the `smop` function will always produce a solution which is at least as accurate as the solution produced by `asmop`.

In addition to this example, we consider the application of `asmop` to a larger series. Specifically, we examine a series containing 500 observations of six Normally distributed variables. This series has multiple changes in variance occurring in different subsets of variables, and is displayed in Figure C.2.2(a). Note that we do not apply `smop` to this data set due to the excessively long run-time of the method on a series of this size.

```
library(zoo) # for plotting
# load data from changepointmv package:
```

```
data("var.change.ex")

plot.zoo(var.change.ex)
```



C.2.2(a): True changes.

C.2.2(b): A-SMOP with hard subset restriction.

C.2.2(c): A-SMOP with soft subset restriction.

Figure C.2.2: An example of multiple changes in variance in six Normally distributed variables. Each change affects a different subset of the variables. The true changepoints are shown by the red solid lines, and the estimated changepoints in each case are shown by the blue dashed lines.

To demonstrate the difference between the hard-restricted and soft-restricted versions of A-SMOP, we perform two applications of `asmop` on this data: one with `hard.restrict=TRUE`, and another with `hard.restrict=FALSE` (i.e. soft restriction is used). We set `window.size=10` and use `cost.func="norm.var.seglen"`.

```
varchange.results.hard = asmop(data=var.change.ex,
    cost.func="norm.var.seglen", window.size=10, hard.restrict=TRUE)
```

```
varchange.results.soft = asmop(data=var.change.ex,
    cost.func="norm.var.seglen", window.size=10, hard.restrict=FALSE)
```

The plots of the results of these two applications are presented in Figures C.2.2(b) and C.2.2(c), respectively. These plots suggest that soft-restricted A-SMOP provides a more accurate segmentation compared to hard-restricted A-SMOP, with the key differences being the size of the affected variable subsets. Hard-restricted places a two-variable change at 198 and a one-variable change at 222, whereas soft-restricted places a single three-variable change at 204. Similar behaviour is exhibited later in the 300–350 range. Visual inspection suggests that these three-variable changepoints are more appropriate.

To assess this mathematically, the **changepointmv** package includes a function called `vmeasure`. The V-measure, introduced by Rosenberg and Hirschberg (2007), is a metric which quantifies the accuracy of a given segmentation (compared to the true segmentation) on the $[0, 1]$ scale, with a larger value (i.e. closer to 1) indicating a more accurate segmentation. The V-measure of a segmentation resulting from application of `smop` or `asmop` can be found using the `vmeasure` function as follows:

```
# create 'true' changepoint locations and subsets:
true.cpts = c(100, 200, 300, 400, 500) # includes end-point of data
true.subsets = matrix(NA, length(true.cpts), ncol(var.change.ex))
true.subsets[1,] = c(F,F,F,T,T,T)
true.subsets[2,] = c(T,T,T,F,F,F)
true.subsets[3,] = c(F,T,F,T,F,T)
true.subsets[4,] = c(T,F,F,T,T,F)
true.subsets[5,] = c(T,T,T,T,T,T) # end-point affects all variables
# calculate V-measure of hard- and soft-restricted segmentations:
vmeasure(varchange.results.hard, true.cpts, true.subsets) # 0.891
vmeasure(varchange.results.soft, true.cpts, true.subsets) # 0.980
# see how many changepoint vectors were considered:
varchange.results.hard@num.cpt.vecs # 280
varchange.results.soft@num.cpt.vecs # 15422
# view the running times
varchange.results.hard@runtimes # 0.46 seconds
varchange.results.soft@runtimes # 218.4 seconds = 3.64 mins
```

This gives V-measures of 0.891 and 0.980 for the segmentations produced using hard and soft restriction, respectively. This therefore confirms that soft restriction provides a more accurate segmentation than hard restriction. The higher accuracy of soft-restricted A-SMOP is to due its consideration of additional changepoint vectors: `num.cpt.vecs = 15422` for soft-restricted and `num.cpt.vecs = 280` for hard-restricted. Note that due to the definition of soft restriction, this will always be the case (see Section 5.2.2 for more details). This is subsequently reflected in the running times: 3.64 minutes and 0.46 seconds for soft- and hard-restricted, respectively.

C.3 Case Studies

We now consider application of the `asmop` function to two data sets:

- a multivariate series containing the flows of four rivers in Quebec; and
- a multivariate series containing the exchange rates of four currencies against the US Dollar.

Each series is examined in turn.

C.3.1 Quebec River Flows

This data set contains the annual January to June streamflow amounts for four rivers in Quebec (Baleine, Churchill Falls, Manicouagan and Romaine) from 1972 to 1994, measured in $L/(km^2s)$. This data is also analysed by Perreault et al. (2000) and is originally published by the Centre d'Expertise Hydrique Quebec. It is made available in the `bcp` package (Erdman and Emerson, 2007), from which we have obtained the data.

```
library(bcp) # for data

library(zoo) # for plotting

data("QuebecRivers")

plot.zoo(QuebecRivers)
```

A plot of this data is shown in Figure C.3.1.

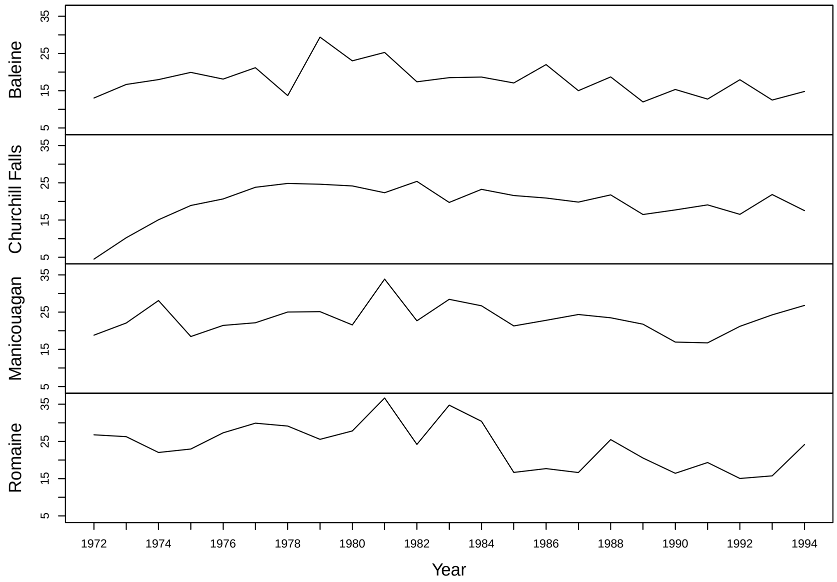


Figure C.3.1: The annual January to June streamflow amounts for four rivers in Quebec from 1972 to 1994, measured in $L/(km^2 s)$.

Interest lies in detecting changes in the streamflow of the rivers. Whilst Perreault et al. (2000) search only for shifts in the mean level, visual inspection of the data suggests that changes may be occurring in the mean and/or variance of the flow. Therefore, we consider changes in both properties. Inspection of the series for Churchill Falls may lead to the interpretation that it could be non-stationary near the beginning. If the end-user believes that this may be the case, then a non-stationary analysis of this univariate series could be performed, for example using the Locally Stationary Wavelet process (see Nason et al. (2000) for more details). The low-frequency components could then be filtered out to remove this behaviour and leave the information regarding the mean and variance relatively unaffected. However, in this instance we take the view that this apparent behaviour is simply due to the stochastic nature of the observations, and that the series will be segmented appropriately by the changepoint detection procedure.

Since it is feasible that some rivers may be affected by a change whilst others may not, it is prudent to search for subset-multivariate (rather than strictly fully-multivariate) changes. To this end, we apply the A-SMOP algorithm to the data using `asmop`. Soft subset restriction is used for greater accuracy. Since we are searching for changes in both mean and variance, we set `cost.func` to `"norm.meanvar.seglen"`. We use the default values for `alpha` ($= 2 \log n$), `beta` ($= 2 \log p \log n$), and `min.dist` ($= 2$). Since the series is relatively short, we wish to use a small window size and hence set `window.size=3`. We therefore run the following code:

```
# running A-SMOP with default beta value
quebec.results = asmop(data=QuebecRivers,
    cost="norm.meanvar.seglen", window.size=3, hard.restrict=FALSE)
plot(quebec.results)
# see the years at which changes occur
rownames(QuebecRivers)[quebec.results@cpts]
```

The resulting plot is shown in Figure C.3.2. We see that A-SMOP estimates two changepoints in the series, at the years 1975 and 1984. These two changes affect different rivers: the change at 1975 affects Churchill Falls, whereas the change at 1984 affects Baleine, Manicouagan and Romaine. We note that the detected locations correspond to the findings of Perreault et al. (2000), who search for a single changepoint and estimate one at 1984. The multiple changepoint approach of A-SMOP allows the detection of the additional changepoint. Furthermore, such intricate results detailing the specific affected variables provide additional information, and are not part of the output of the fully-multivariate approaches in the `bcp` and `ecp` packages.

Given the results in Figure C.3.2, we can either believe that there are two changes in the series which affect the respective subsets of rivers, or consider that there may be another segmentation which is more appropriate and try to obtain this by altering

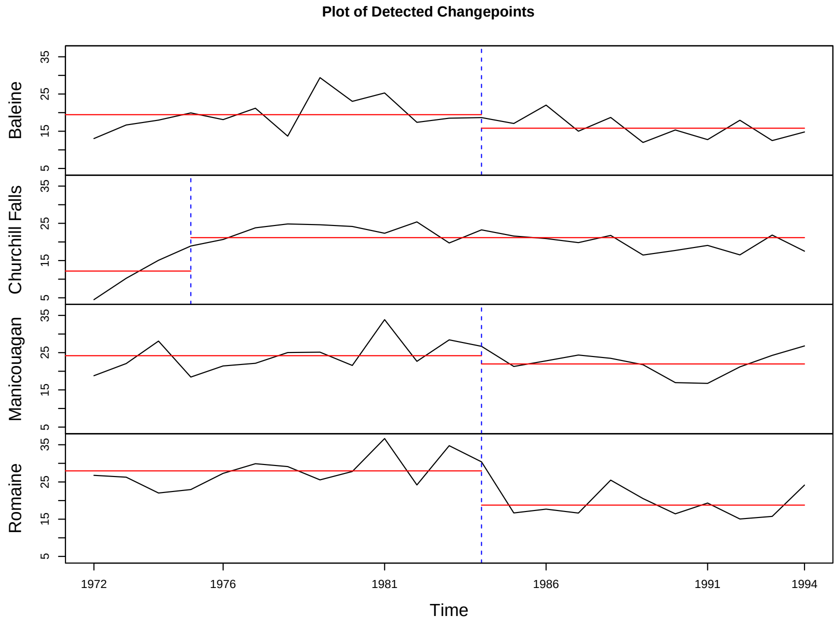


Figure C.3.2: The results of applying `asmop` to the Quebec river flows data set.

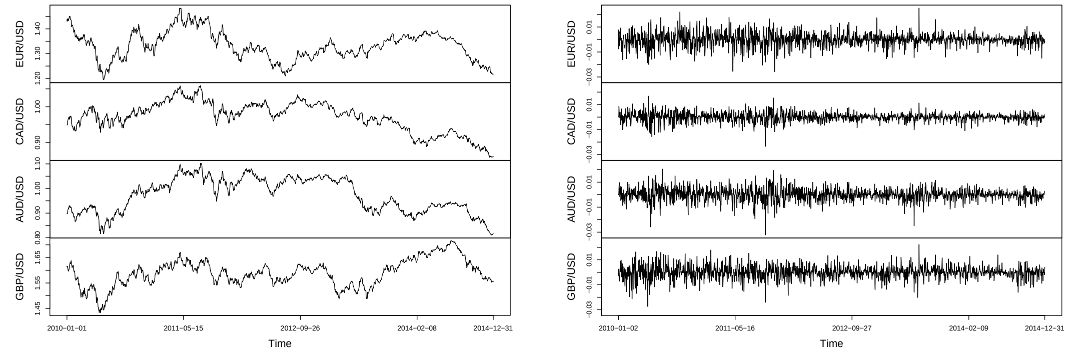
the α and/or β penalties. The choice of appropriate values for α and β is an open question and is dependent on many factors including the size of changes and the length of segments, both of which may be unknown prior to analysis. As demonstrated here, current practice for penalty choice assessment involves plotting the detected changepoints on the data to see if they seem reasonable.

C.3.2 Currency Exchange Rates

This data set is a multivariate series containing 1826 observations of the daily closing exchange rates of four currencies against the United States Dollar (USD). These are the Euro (EUR), Canadian Dollar (CAD), Australian Dollar (AUD) and British Pound (GBP). The rates are taken from 01/01/2010 to 31/12/2014. This data set can be obtained via the `quantmod` package (Ryan, 2015) using the following code:

```
library(quantmod) # for downloading exchange rate data
library(zoo) # for plotting
# store symbols of exchanges rates of interest:
currencies.usd = c("EUR/USD", "CAD/USD", "AUD/USD", "GBP/USD")
```

```
start.date = "2010-01-01"
end.date = "2014-12-31"
# create a sequence of the dates to use for row names of data matrix:
dates = as.character( seq(from=as.Date(start.date),
                        to=as.Date(end.date), by="1 day") )
# download the individual exchange rate series into the R environment:
getFX(currencies.usd, from=start.date, to=end.date)
# Adds four objects: EURUSD, CADUSD, AUDUSD, GBPUSD.
# Compile four univariate series into single multivariate series:
rates = matrix(NA, nrow=1826, ncol=4)
rates[,1] = EURUSD; rates[,2] = CADUSD
rates[,3] = AUDUSD; rates[,4] = GBPUSD
colnames(rates) = currencies.usd
row.names(rates) = dates
plot.zoo(rates) # plot multivariate series of exchange rates.
```



C.3.3(a): Original data.

C.3.3(b): First-differenced data.

Figure C.3.3: The daily closing exchange rates of four currencies against the United States Dollar (USD): EUR, CAD, AUD and GBP.

A plot of this data is shown in Figure C.3.3(a). Previous authors have modelled daily stock market returns as changes in volatility (see, for example, the analysis of Dow Jones Index returns by Killick et al. (2012)). From the plot of the first-differences of the series in Figure C.3.3(b), it appears reasonable to do the same for daily exchange rates. Visual inspection of these volatilities suggests that there may be changes in the variation of the four exchange rates, particularly towards the end of the series. Some changes appear to only affect certain currencies.

In an effort to detect any changes in the exchange rates, and identify which of the currencies are affected, we apply A-SMOP to the series. As in previous analyses of financial data we assume that the exchange rates are Normally distributed with constant mean and piecewise stationary variance (both of which are unknown). We therefore set the cost function as `norm.var.seglen` (so that very small segments are penalised) and use a `window.size` of 10 (so that if two currencies are affected by a change within 10 days of each other, we assume it is induced by the same event). Hard restriction is used to ensure a faster computation time, and use the default values for the other parameters (including the α and β penalties).

```
rates.diff = diff(rates)
currency.results = asmop(rates.diff, cost.func="norm.var.seglen",
    window.size=10, hard.restrict=TRUE)
plot(currency.results)
# see the dates at which changes occur
dates[currency.results@cpts]
```

The results of `asmop` are illustrated in Figure C.3.4. It can be seen that multiple changes are detected in the series, with different changes affecting different combinations of the exchange rates. A consideration of world events suggests that some of the detected changepoints correspond to certain developments. In particular, a sharp drop in UK unemployment was reported on 11/08/2010, which corresponds exactly to a detected change in the GBP. Similarly, a detected change in July 2011 in only the AUD corresponds to the introduction of the Minerals and Resource Rent Tax in Australia. Indeed, the occurrence of more global events correspond to the detected locations of changepoints which affect many of the exchange rates. Specifically, the formal end of the Iraq War in December 2011 correlates with a detected reduction in variance in the EUR, CAD and AUD. Likewise, the rise of ISIS and the ongoing

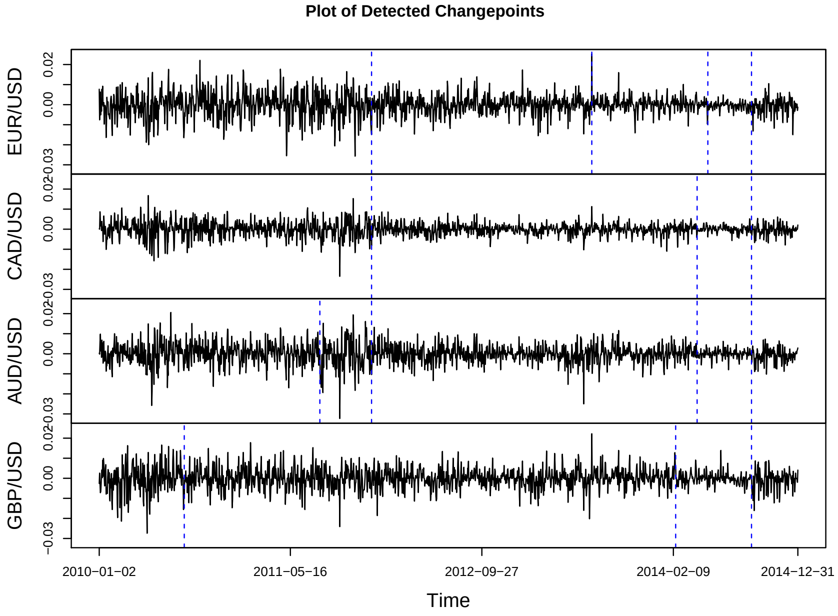


Figure C.3.4: The results of applying `asmop` to the exchange rate time series in Figure C.3.3(b).

Israel-Palestine conflict are likely an influence in the detected increase in variance of all four exchange rates in August 2014.

C.3.3 Summary

This appendix illustrates the application of the methodology available in the **changepointmv** package for performing changepoint analysis on multivariate time series. The functions available allow for the detection of a range of different types of change, including changes which occur in all variables and those which in occur in only subsets of variables. Further, the package provides the user with separate control over the penalisation of additional changepoints and additional affected variables for a given detected change. Consequently, the **changepointmv** package is useful the analysis of multivariate series where interest lies in both the locations of any changes and the identification of the affected variables. The **changepointmv** package can be obtained from `http://www.lancaster.ac.uk/~pickerin/software.html`.

Bibliography

Adams, R. P. and MacKay, D. J. Bayesian online changepoint detection. *arXiv preprint arXiv:0710.3742*, 2007.

Akaike, H. A new look at the statistical model identification. *IEEE Transactions on Automatic Control*, 19(6):716–723, 1974.

Andreou, E. and Ghysels, E. Structural breaks in financial time series. In *Handbook of Financial Time Series*, pages 839–870. Springer, 2009.

Aston, J. A. D. and Kirch, C. Evaluating stationarity via change-point alternatives with applications to fMRI data. *The Annals of Applied Statistics*, 6(4):1906–1948, 2012.

Aue, A., Hörmann, S., Horváth, L., Reimherr, M., et al. Break detection in the covariance structure of multivariate time series models. *The Annals of Statistics*, 37(6B):4046–4087, 2009.

Auger, I. E. and Lawrence, C. E. Algorithms for the optimal identification of segment neighborhoods. *Bulletin of Mathematical Biology*, 51(1):39–54, 1989.

Bardwell, L. and Fearnhead, P. Bayesian detection of abnormal segments in multiple time series. *arXiv preprint arXiv:1412.5565*, 2014.

Batsidis, A., Horváth, L., Martín, N., Pardo, L., and Zografos, K. Change-point detection in multinomial data using phi-divergence test statistics. *Journal of Multivariate Analysis*, 118:53–66, 2013.

Bollerslev, T. Generalized autoregressive conditional heteroskedasticity. *Journal of Econometrics*, 31(3):307–327, 1986.

Box, G. E., Jenkins, G. M., and Reinsel, G. C. *Time Series Analysis: Forecasting and Control*, volume 734. John Wiley & Sons, 2011.

Braun, J., Braun, R., and Müller, H. Multiple changepoint fitting via quasilikelihood, with application to DNA sequence segmentation. *Biometrika*, 87(2):301–314, 2000.

Brockwell, P. J. and Davis, R. A. *Time Series: Theory and Methods*. Springer, 2009.

Burnham, K. P. and Anderson, D. R. *Model Selection and Multimodel Inference: A practical information-theoretic approach*. Springer, 2002.

Byrd, R. H., Lu, P., Nocedal, J., and Zhu, C. A limited memory algorithm for bound constrained optimization. *SIAM Journal on Scientific Computing*, 16(5):1190–1208, 1995.

Chen, J. and Gupta, A. K. Testing and locating variance changepoints with application to stock prices. *Journal of the American Statistical Association*, 92(438): 739–747, 1997.

Chen, J. and Gupta, A. K. *Parametric statistical change point analysis*. Birkhauser, 2000.

Cho, H. and Fryzlewicz, P. Multiscale and multilevel technique for consistent segmentation of nonstationary time series. *Statistica Sinica*, 22:207–229, 2012.

Cho, H. and Fryzlewicz, P. 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.

Choudhuri, N., Ghosal, S., and Roy, A. Contiguity of the Whittle measure for a Gaussian time series. *Biometrika*, 91(1):211–218, 2004.

Cowpertwait, P. S. and Metcalfe, A. V. *Introductory time series with R*. Springer, 2009.

Cribben, I., Wager, T. D., and Lindquist, M. A. Detecting functional connectivity change points for single-subject fMRI data. *Frontiers in Computational Neuroscience*, 7, 2013.

Davis, R., Lee, T., and Rodriguez-Yam, G. Structural break estimation for nonstationary time series models. *Journal of the American Statistical Association*, 101 (473):223–239, 2006.

Dempster, A. P., Laird, N. M., and Rubin, D. B. Maximum likelihood from incomplete data via the em algorithm. *Journal of the royal statistical society. Series B (methodological)*, pages 1–38, 1977.

Eckley, I. A., Fearnhead, P., and Killick, R. Analysis of changepoint models. In Barber, D., Cemgil, A. T., and Chiappa, S., editors, *Bayesian Time Series Models*, pages 203–224. Cambridge University Press, 2011.

Erdman, C. and Emerson, J. W. bcp: An R Package for Performing a Bayesian Analysis of Change Point Problems. *Journal of Statistical Software*, 23(3):1–13, 2007.

Fearnhead, P. Exact and efficient bayesian inference for multiple changepoint problems. *Statistics and Computing*, 16(2):203–213, 2006.

Fearnhead, P. and Liu, Z. On-line inference for multiple changepoint problems. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 69(4):589–605, 2007.

Frick, K., Munk, A., and Sieling, H. Multiscale change point inference. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 76(3):495–580, 2014.

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

Fryzlewicz, P. and Subba Rao, S. Multiple-change-point detection for auto-regressive conditional heteroscedastic processes. *Journal of the Royal Statistical Society: series B (statistical methodology)*, 76(5):903–924, 2014.

Gallagher, K., Bodin, T., Sambridge, M., Weiss, D., Kylander, M., and Large, D. Inference of abrupt changes in noisy geochemical records using transdimensional changepoint models. *Earth and Planetary Science Letters*, 311(1):182–194, 2011.

Giraitis, L., Leipus, R., and Surgailis, D. The change-point problem for dependent observations. *Journal of Statistical Planning and Inference*, 53(3):297–310, 1996.

Gombay, E. Change detection in autoregressive time series. *Journal of Multivariate Analysis*, 99(3):451–464, 2008.

Gray, R. M. Toeplitz and circulant matrices: A review. *Communications and Information Theory*, 2(3):155–239, 2005.

Green, P. Reversible jump Markov chain Monte Carlo computation and Bayesian model determination. *Biometrika*, 82(4):711–732, 1995.

Haccou, P., Meelis, E., and Van De Geer, S. The likelihood ratio test for the change point problem for exponentially distributed random variables. *Stochastic processes and their applications*, 27:121–139, 1987.

Hannan, E. J. and Quinn, B. G. The determination of the order of an autoregression. *Journal of the Royal Statistical Society. Series B (Methodological)*, 41(2):190–195, 1979.

Haynes, K., Eckley, I. A., and Fearnhead, P. Efficient penalty search for multiple changepoint problems. *arXiv preprint arXiv:1412.3617*, 2014.

Hinkley, D. V. Inference about the change-point in a sequence of random variables. *Biometrika*, 57(1):1–17, 1970.

Hocking, T., Rigaill, G., Vert, J.-P., and Bach, F. Learning sparse penalties for change-point detection using max margin interval regression. In *Proceedings of The 30th International Conference on Machine Learning*, pages 172–180, 2013.

Horváth, L. and Hušková, M. Change-point detection in panel data. *Journal of Time Series Analysis*, 33(4):631–648, 2012.

Hosking, J. R. Fractional differencing. *Biometrika*, 68(1):165–176, 1981.

Hsu, C.-C. and Kuan, C.-M. Distinguishing between trend-break models: method and empirical evidence. *The Econometrics Journal*, 4(2):171–190, 2001.

Hurvich, C. Whittle's Approximation to the Likelihood Function. Lecture Notes, New York University Stern School of Business, 2002.

Hušková, M. and Kirch, C. Bootstrapping confidence intervals for the change-point of time series. *Journal of Time Series Analysis*, 29(6):947–972, 2008.

Inclan, C. and Tiao, G. C. Use of Cumulative Sums of Squares for Retrospective Detection of Changes of Variance. *Journal of the American Statistical Association*, 89(427):913, 1994.

Jackson, B., Sargle, J. D., Barnes, D., Arabhi, S., Alt, A., Gioumousis, P., Gwin, E., Sangtrakulcharoen, P., Tan, L., and Tsai, T. T. An algorithm for optimal partitioning of data on an interval. *IEEE, Signal Processing Letters*, 12(2):105–108, 2005.

James, N. A. and Matteson, D. S. ecp: An R package for nonparametric multiple change point analysis of multivariate data. *Journal of Statistical Software*, 62(7): 1–25, 2014.

James, N. A. and Matteson, D. S. Change points via probabilistically pruned objectives. *arXiv preprint arXiv:1505.04302*, 2015.

Jeng, X. J., Cai, T. T., and Li, H. Simultaneous discovery of rare and common segment variants. *Biometrika*, 100(1):157–172, 2013.

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

Killick, R., Fearnhead, P., and Eckley, I. A. Optimal detection of changepoints with a linear computational cost. *Journal of the American Statistical Association*, 107 (500):1590–1598, 2012.

Killick, R., Eckley, I. A., and Jonathan, P. A wavelet-based approach for detecting changes in second order structure within nonstationary time series. *Electronic Journal of Statistics*, 7:1167–1183, 2013.

Killick, R., Haynes, K., and Eckley, I. A. *changepoint: An R package for changepoint analysis*, 2015. R package version 2.2.

Kirch, C., Muhsal, B., and Ombao, H. Detection of Changes in Multivariate Time Series With Application to EEG Data. *Journal of the American Statistical Association*, 110(511):1197–1216, 2015.

Lai, T. L. Sequential changepoint detection in quality control and dynamical systems. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 613–658, 1995.

Lavielle, M. and Ludeña, C. The multiple change-points problem for the spectral distribution. *Bernoulli*, 6(5):845–869, 2000.

Lavielle, M. and Teyssiere, G. Detection of multiple change-points in multivariate time series. *Lithuanian Mathematical Journal*, 46(3):287–306, 2006.

Lebarbier, É. Detecting multiple change-points in the mean of Gaussian process by model selection. *Signal Processing*, 85(4):717–736, 2005.

Lévy-Leduc, C. and Roueff, F. Detection and localization of change-points in high-dimensional network traffic data. *The Annals of Applied Statistics*, 3(2):637–662, 2009.

Li, H., Munk, A., and Sieling, H. FDR-Control in Multiscale Change-point Segmentation. *arXiv preprint arXiv:1412.5844*, 2014.

Li, S. and Lund, R. Multiple Changepoint Detection via Genetic Algorithms. *Journal of Climate*, 25(2):674–686, 2012.

Liu, D. C. and Nocedal, J. On the limited memory BFGS method for large scale optimization. *Mathematical programming*, 45(1-3):503–528, 1989.

Lung-Yut-Fong, A., Lévy-Leduc, C., and Cappé, O. Distributed detection/localization of change-points in high-dimensional network traffic data. *Statistics and Computing*, 22(2):485–496, 2011a.

Lung-Yut-Fong, A., Lévy-Leduc, C., and Cappé, O. Homogeneity and change-point detection tests for multivariate data using rank statistics. *arXiv preprint arXiv:1107.1971*, 2011b.

Luong, T. M., Perduca, V., and Nuel, G. Hidden Markov Model Applications in Change-Point Analysis. *arXiv preprint arXiv:1212.1778*, 2012.

Maboudou, E. M. and Hawkins, D. M. Fitting Multiple Change-Point Models to a Multivariate Gaussian Model. In *Proceedings of International Workshop in Sequential Methodologies*, 2009.

Maboudou-Tchao, E. M. and Hawkins, D. M. Detection of multiple change-points in multivariate data. *Journal of Applied Statistics*, 40(9):1979–1995, 2013.

Maidstone, R., Hocking, T., Rigaill, G., and Fearnhead, P. On optimal multiple changepoint algorithms for large data. *arXiv preprint arXiv:1409.1842*, 2014.

Matteson, D. S. and James, N. A. A nonparametric approach for multiple change point analysis of multivariate data. *Journal of the American Statistical Association*, 109 (505):334–345, 2014.

Nam, C. F., Aston, J. A., and Johansen, A. M. Quantifying the uncertainty in change points. *Journal of Time Series Analysis*, 33(5):807–823, 2012.

Nam, C. F. H., Aston, J. A. D., Eckley, I. A., and Killick, R. The uncertainty of storm season changes: Quantifying the uncertainty of autocovariance changepoints. *Technometrics*, 57(2):194–206, 2015.

Nason, G. P. *Wavelet methods in statistics with R*. Springer Verlag, 2008.

Nason, G. P., Von Sachs, R., and Kroisandt, G. Wavelet processes and adaptive estimation of the evolutionary wavelet spectrum. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 62(2):271–292, 2000.

Olshen, A. B., Venkatraman, E. S., Lucito, R., and Wigler, M. Circular binary segmentation for the analysis of array-based DNA copy number data. *Biostatistics*, 5(4):557–72, 2004.

Ombao, H., von Sachs, R., and Guo, W. SLEX Analysis of Multivariate Nonstationary Time Series. *Journal of the American Statistical Association*, 100(470):519–531, 2005.

Ombao, H. C., Raz, J. A., Von Sachs, R., and Malow, B. A. Automatic Statistical Analysis of Bivariate Nonstationary Time Series. 96(454):543–560, 2001.

Page, E. S. Continuous inspection schemes. *Biometrika*, 41(1/2):100–115, 1954.

Perreault, L., Parent, E., Bernier, J., Bobée, B., and Slivitzky, M. Retrospective multivariate Bayesian change-point analysis: A simultaneous single change in the mean of serveral hydrological sequences. *Stochastic Environmental Research and Risk Assessment*, 14:243–261, 2000.

Picard, F., Robin, S., Lavielle, M., Vaisse, C., and Daudin, J.-J. A statistical approach for array CGH data analysis. *BMC Bioinformatics*, 6:27, 2005.

Polunchenko, A. S. and Tartakovsky, A. G. State-of-the-art in sequential change-point detection. *Methodology and computing in applied probability*, 14(3):649–684, 2012.

Polushina, T. and Sofronov, G. Change-point detection in biological sequences via genetic algorithm. In *2011 IEEE Congress on Evolutionary Computation (CEC)*, pages 1966–1971. IEEE, 2011.

Preuß, P., Puchstein, R., and Dette, H. Detection of multiple structural breaks in multivariate time series. *Journal of the American Statistical Association*, 110(510): 654–668, 2015.

R Development Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2011.

Reeves, J., Chen, J., Wang, X. L., Lund, R., and QiQi, L. A review and comparison of changepoint detection techniques for climate data. *Journal of Applied Meteorology and Climatology*, 46(6):900–915, 2007.

Rigaill, G. Pruned dynamic programming for optimal multiple change-point detection. *arXiv preprint arXiv:1004.0887*, 2010.

Rissanen, J. *Stochastic complexity in statistical inquiry*. World Scientific, 1989.

Rosenberg, A. and Hirschberg, J. V-measure: A conditional entropy-based external cluster evaluation measure. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 410–420, 2007.

Ruggieri, E., Herbert, T., Lawrence, K. T., and Lawrence, C. E. Change point method for detecting regime shifts in paleoclimatic time series: Application to δ18O time series of the Plio-Pleistocene. *Paleoceanography*, 24(1), 2009.

Ryan, J. A. *quantmod: Quantitative Financial Modelling Framework*, 2015. R package version 0.4-5.

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

Scott, A. and Knott, M. A cluster analysis method for grouping means in the analysis of variance. *Biometrics*, pages 507–512, 1974.

Shumway, R. H. and Stoffer, D. S. *Time series analysis and its applications: with R examples*, volume 3. Springer New York, 2000.

Siegmund, D., Yakir, B., Zhang, N. R., et al. Detecting simultaneous variant intervals in aligned sequences. *The Annals of Applied Statistics*, 5(2A):645–668, 2011.

Silkina, T. Application of distributed acoustic sensing to flow regime classification. Master's thesis, Department of Petroleum Engineering and Applied Geophysics, Norwegian University of Science and Technology, 2014.

Srivastava, M. S. and Worsley, K. J. Likelihood Ratio Tests for a Change in the Multivariate Normal Mean. *Journal of the American Statistical Association*, 81 (393):199–204, 1986.

Stark, D. R. and Spall, J. C. Computable rate of convergence in evolutionary computation. In *Proceedings of the Fifth International Conference on Information Fusion, 2002*, volume 1, pages 88–93. IEEE, 2002.

Tartakovsky, A. G., Polunchenko, A. S., and Sokolov, G. Efficient computer network anomaly detection by changepoint detection methods. *IEEE Journal of Selected Topics in Signal Processing*, 7(1):4–11, 2013.

Van der Horst, J., Den Boer, H., Wyker, B., Kusters, R., Mustafina, D., Groen, L., Bulushi, N., Mjeni, R., Awan, K., Rajhi, S., et al. Fiber optic sensing for improved wellbore production surveillance. In *IPTC 2014: International Petroleum Technology Conference*, Doha, Qatar, 2014.

Velis, D. R. Statistical segmentation of geophysical log data. *Mathematical Geology*, 39(4):409–417, 2007.

Venkatraman, E. S. *Consistency results in multiple change-point problems.* PhD thesis, Department of Statistics, Stanford University, 1993.

Vert, J.-P. and Bleakley, K. Fast detection of multiple change-points shared by many signals using group LARS. In Lafferty, J., Williams, C., Shawe-Taylor, J., Zemel, R., and Culotta, A., editors, *Advances in Neural Information Processing Systems 23*, pages 2343–2351. Curran Associates, Inc., 2010.

Viterbi, A. J. Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. *Information Theory, IEEE Transactions on*, 13(2):260–269, 1967.

Vostrikova, L. J. Detecting disorder in multidimensional random processes. *Soviet Mathematics Doklady*, 24:55–59, 1981.

Wald, A. Note on the consistency of the maximum likelihood estimate. *The Annals of Mathematical Statistics*, pages 595–601, 1949.

Whittle, P. *Hypothesis Testing in Time Series Analysis.* Almquist and Wicksell, 1951.

Wyse, J., Friel, N., et al. Approximate simulation-free Bayesian inference for multiple changepoint models with dependence within segments. *Bayesian Analysis*, 6(4): 501–528, 2011.

Xie, Y. and Siegmund, D. Sequential multi-sensor change-point detection. *The Annals of Statistics*, 41(2):670–692, 2013.

Xing, H., Mo, Y., Liao, W., and Zhang, M. Q. Genome-wide localization of protein-DNA binding and histone modification by a Bayesian change-point method with ChIP-seq data. *PLoS Computation Biology*, 8(7):e1002613–e1002613, 2012.

Yamaguchi, K. Estimating a change point in the long memory parameter. *Journal of Time Series Analysis*, 32(3):304–314, 2011.

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

Yau, C. Y. and Davis, R. A. Likelihood inference for discriminating between long-memory and change-point models. *Journal of Time Series Analysis*, 33(4):649–664, 2012.

Yuan, M. and Lin, Y. Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 68(1):49–67, 2006.

Yule, G. U. The applications of the method of correlation to social and economics statistics. *Journal of the Royal Statistical Society*, 72:721–730, 1909.

Yule, G. U. On a method of investigating periodicities in disturbed series, with special reference to wolfer's sunspot numbers. *Philosophical Transactions of the Royal Society of London. Series A, Containing Papers of a Mathematical or Physical Character*, 226(636-646):267–298, 1927.

Zhang, N., Siegmund, D., Ji, H., and Li, J. Detecting simultaneous change-points in multiple sequences. *Biometrika*, 97:631–645, 2010.

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