Changepoint Detection for Data Intensive Settings

Samuel Owen Tickle, M.A. (Cantab), M.Res

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

September 2019

STOR-i
excellence with impact

Abstract

Detecting a point in a data sequence where the behaviour alters abruptly, otherwise known as a changepoint, has been an active area of interest for decades. More recently, with the advent of the data intensive era, the need for automated and computationally efficient changepoint methods has grown. We here introduce several new techniques for doing this which address many of the issues inherent in detecting changes in a streaming setting. In short, these new methods, which may be viewed as non-trivial extensions of existing classical procedures, are intended to be as useful in as wide a set of situations as possible, while retaining important theoretical guarantees and ease of implementation.

The first novel contribution concerns two methods for parallelising existing dynamic programming based approaches to changepoint detection in the single variate setting. We demonstrate that these methods can result in near quadratic computational gains, while retaining important theoretical guarantees.

Our next area of focus is the multivariate setting. We introduce two new methods for data intensive scenarios with a fixed, but possibly large, number of dimensions. The first of these is an offline method which detects one change at a time using a new test statistic. We demonstrate that this test statistic has competitive power in a variety of possible settings for a given changepoint, while allowing the method to be versatile across a range of possible modelling assumptions.

The other method we introduce for multivariate data is also suitable in the streaming setting. In addition, it is able to relax many standard modelling assumptions. We discuss the empirical properties of the procedure, especially insofar as they relate to a desired false alarm error rate.

Acknowledgements

This work would not have been possible without the support of numerous people, chief among whom my supervisors. Firstly, Professor Idris Eckley, whose unique wisdom has ensured that this thesis has remained cogent, grounded and, above all, exciting. While the data problems of the future will inexorably shift from those we see today, Professor Eckley's enthusiasm and love for the problem, whatever it may be, is something that I devoutly hope will continue to be the mainstay. Secondly, Professor Paul Fearnhead, who for me has totally redefined the parameters of intelligence and quiet patience, the latter of which in particular has been of incalculable assistance during this process. Many superlatives have been penned regarding Professor Fearnhead's abilities across a staggeringly vast region of statistics, which of course includes those discussed herein. However, I fear no words can fully do him justice. Finally, Dr Kjeld Jensen of British Telecommunications plc (BT), who has been an enormously helpful source of ideas, constructive criticisms and interesting problem domains for our methods within BT. On which note, I'd like to pay my thanks to a multitude of people within the company for showing me such warmth and friendship, in addition to integrating me into the wonderfully rich data sandpit, during my visits. Special thanks in this regard to David Yearling and Trevor Burbridge. I would also like to thank BT, as well as the Engineering and Physical Sciences Research Council (EPSRC) for financial assistance throughout the project.

I'd like to take this opportunity to thank the reviewers, as well as the editor and associate editor, at the Journal of Computational and Graphical Statistics, for their time and effort in critiquing and greatly strengthening the parallelisation work discussed in Chapter 3. On this note, I should also like to thank the examiners of this

work - Professor George Michailidis of the University of Florida and Dr Alex Gibberd of Lancaster University - for having the unenviable task of reading through it all and providing many interesting discussion points. This thesis is much the richer for their comments and suggestions.

More widely, I wish to give my most heartfelt thanks to everyone within the STOR-i community, and Lancaster University as a whole, who have given me an experience on which I will look back, not just with fondness, but also with a profound regret that it all could not have lasted longer. To the StatScale group, in particular Dr Daniel Grose, who helped me fall in love with running all over again. To Professor Richard Samworth and Dr Rajen Shah, also of StatScale, for helpful discussions throughout the project. To the wider changepoint community to which Lancaster is affiliated, in particular Dr Guillem Rigaill, for further helpful discussions. To the STOR-i Time Series and Changepoints reading group, who helped keep things fresh and vibrant for me on those days when such a notion felt orthogonal to reality. In particular, I should like to thank Alex, Euan, Harjit, Sean and Tom for helpful discussions on one of the new methods I present here. To Professor Jonathan Tawn, one of the most personally brilliant and brilliantly personable people one could ever hope to meet, and the person on whom I feel a great deal of my nigh-on universally positive memories of STOR-i ultimately rest. To the many STOR-i students, past and present, far too numerous to name here, and yet each instrumental in being their own unique brand of wonderful exactly when I needed it. To my long-suffering housemate Euan, whose strength of character is matched only by his loyalty and kindness. Finally, to a group of nine extraordinary people to whom I owe so much, and with whom I am delighted to have shared this journey over the last four years. Harjit, Jake, Rob, Anna, Emily, David, Toby, Luke and Kathryn - thank you for teaching me to laugh again. It is an incredible privilege to count you all as friends.

Last but by no means least, I wish to thank my close friends outside of Lancaster - particularly Gar, without whom this thesis would contain many more errors than it does - and, of course, my family. Sadly (or perhaps not) this group is far too large to individually name here, so special thanks to my parents and my sister, Rachel, who

have followed my mathematical career ever since the first green bottle fell off the wall. Although we discuss change herein, they for me have been the irreplaceable constant.

Declaration

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

Chapter 3 has been accepted for publication as S. O. Tickle, I. A. Eckley, P. Fearnhead and K. Haynes. Parallelization of a Common Changepoint Detection Method. *Journal of Computational and Graphical Statistics*, 2019 (to appear).

Sam Tickle

Contents

List of Figures

List of Tables

List of Abbreviations

AIC	Akaike Information Criterion
AMOC	At Most One Change
BARD	Bayesian Abnormal Region Detector
BIC	Bayesian Information Criterion
Bin-Weight	Binary-Weight
CAPA	Collective and Point Anomalies
cdf	Cumulative distribution function
CROPS	Changepoints for a Range Of PenaltieS
CUSUM	CUmulative SUM
ED-PELT	Empirical Distribution Pruned Exact Linear Time
FPOP	Functional Pruning Optimal Partitioning
GFPOP	Generalized Functional Pruning Optimal Partitioning
GTD	Global Terrorism Database
HC	Hierarchical Clustering
H-SMUCE	Heterogenous Simultaneous MUltiscale Changepoint Estimator
i.i.d.	Independent and identically distributed
Inspect	Informative Sparse Projection for the Estimation of ChangepoinTs
KCP	Kernel Change Point
MBIC	Modified Bayesian Information Criterion
MOSUM	MOving SUM
Neg-Bin	Negative Binomial
NMCD	Nonparametric Multiple Changepoints Detection
OMEN	Online Multivariate Empirical Nonparametric changepoint detection method

OP	Optimal Partitioning
pDPA	pruned Dynamic Programming Algorithm
PELT	Pruned Exact Linear Time
PGIS	Pinkerton Global Intelligence Services
R-FPOP	Robust Functional Pruning Optimal Partitioning
RJMCMC	Reversible Jump Markov Chain Monte Carlo
RSS	Residual Sum of Squares
SAMC	Stochastic Approximate Monte Carlo
SIC	Schwarz Information Criterion
SMOP	Subset Multivariate Optimal Partitioning
SMUCE	Simulataneous MUltiscale Changepoint Estimator
S-R	Shiryaev-Roberts (procedure)
SUBSET	Sparse and Ubiquitous Binary Segmentation in Efficient Time
WBS	Wild Binary Segmentation

List of Symbols

$\mathcal{A}$	Restricted set of time points in a sequence over which we wish to search for changepoints.
a	$\max\{n, d\}$
α	Penalty for adding a variate to an affected set for a sparse multidimensional change.
$\mathcal{B}$	Restricted subset of time indices on which we may fit a changepoint.
b	Number of parameters estimated by the model.
β	Penalty added to the global cost when a changepoint is introduced.
β'	Penalty incurred in the univariate online setting when raising an alarm following a changepoint.
$\mathcal{C}(.)$	Segment cost function, e.g. squared error loss.
$c(.)$	Combination function for compressing a stream.
d	Number of variates/dimensions in a given data stream.
$D_{i,t}$	Difference in cost for the i^{th} variate if a change is placed at time t, especially when the cost is squared error loss in the Gaussian change in mean setting.
Δ_i	Absolute change in mean in variate i in the univariate setting. If this is the same for all variates which change, we use $\Delta\mu$, as in the univariate setting.
$\Delta\mu$	Absolute change in mean in the univariate setting.
Δp	Absolute change in success probability in the negative binomial setting.
$\mathcal{E}$	General event symbol.
ϵ	Strictly positive quantity (usually small).
$F(.)$	Optimal segment cost of a sequence up to the given point.
$\hat{F}(.)$	Empirical cumulative distribution function.
G	Data generating process.

$g(.|.)$ General pre-specified family of densities.

$\Gamma(.)$ Gamma function.

$\Gamma(.,.)$ Upper incomplete gamma function.

$\gamma(.,.)$ Lower incomplete gamma function.

$\mathcal{H}(.,.)$ A set of segmentations 'close' to the truth.

I_u Identity matrix of dimension $u \times u$.

K Penalty added to the model on the introduction of a dense change in the multivariate setting.

L Number of computer cores over which we may parallelise a procedure.

$L(.)$ Number of computer cores available in a theoretical setting for a given length of sequence.

$\mathcal{L}(.)$ Likelihood.

M Number of intervals drawn uniformly using Wild Binary Segmentation.

m True number of changepoints.

$\hat{m}$ Estimated number of changepoints.

$m_X(.)$ Moment generating function of a random variable X.

m_U Known upper bound on the number of changepoints (see also Q).

$\boldsymbol{\mu}$ Latent parameter (vector) for a segment.

μ Single segment parameter, typically the mean of a Gaussian.

$\hat{\mu}$ Estimated segment parameter.

$\mathbb{N}$ Set of natural numbers.

n Number of temporal points.

$\mathcal{O}(.)$ If $f_1(x) = \mathcal{O}(f_2(x))$, then $f_1(x)/f_2(x)$ is bounded as $x \to x_0$.

$\mathcal{O}_p(.)$ As for $\mathcal{O}(.)$, in probability under the same convergence.

$o(.)$ If $f_1(x) = o(f_2(x))$, then $f_1(x)/f_2(x) \to 0$ as $x \to x_0$.

$o_p(.)$ As for $o(.)$, in probability under the same convergence.

$\Omega(.)$ If $f_1(x) = \Omega(f_2(x))$, then $f_2(x)/f_1(x)$ is bounded as $x \to x_0$.

ω Information window: number of points in the past that an online algorithm recalls.

$\Phi^{-1}(.)$ Inverse cdf of a standard normal.

$\psi_X^*(.)$ Cramer transform of the random variable X.

$\rho_{\mathbf{v}}(.)$ Sort function, gives the argument according to the ascending order of the elements of $\mathbf{v}$.

Q Maximum number of estimated changes a method is permitted to place (see also m_U).

$Q_a(b,c)$ The largest integer such that $Q_a(b,c) \times b + (a \mod b) < c$.

$R(.,.)$ Log of the maximum likelihood ratio for a given change location and affected set in the multidimensional setting.

$\mathbb{R}$ Set of real numbers.

r In the negative binomial setting, this is the over-dispersion parameter.

$\mathcal{S}$ Set of variates affected by a changepoint in the multidimensional setting.

$\hat{\mathcal{S}}$ Estimated affected set.

S_t Test statistic (in the multivariate setting).

σ^2 Known variance in a Gaussian noise setting.

$\hat{\sigma}^2$ Maximum likelihood estimator for the variance in a Gaussian setting.

T Most recent time observed in the stream in an online setting.

T_k Typical placeholder for denoting the k^{th} scenario.

$T(.;.)$ Test statistic for a single changepoint at a point in a sequence.

t General (testing) point within a data sequence/stream.

τ True changepoint location.

$\hat{\tau}$ Estimated changepoint location.

θ Changepoint location as a proportion: $\lfloor \theta n \rfloor = \tau$.

$V(.)$ Overlap size in the Chunk procedure.

$Var(.)$ Variance.

$W_{i,t}$ CUSUM statistic for the i^{th} variate of the stream at time t.

$y_{1:n}$ General (univariate) data sequence.

Z A standard normal random variable.

ζ Real number > 0. Several of our theoretical results discuss the probability that we locate a tue changepoint to within $\lceil (\log n)^{1+\zeta} \rceil$ time points.

$\mathbb{1}\{A\}$ Indicator function; equal to 1 if event A occurs and 0 otherwise.

Chapter 1

Introduction

We live in an increasingly data-rich environment. With each year, the number of sensors monitoring a myriad of the minutiae of daily life multiplies. Indeed, the amount of data collected in 2017 and 2018 was nine times the total amount of data collected across recorded history up to and including 2016 (Petrov, 2019). While this fast-changing world affords countless opportunities for improvement and innovation, the practicalities of appropriately handling "Big Data" in a timely fashion are becoming ever more challenging.

One such challenge is to ensure that data can be appropriately inspected, features identified and necessary responses enacted - if required - in an unsupervised fashion, given that, for many systems, the scale of the data space entirely precludes human monitoring. The number of possible features of interest is vast; we herein focus on changepoints: points in a data series where some aspect of the system alters abruptly, if potentially subtly.

The benefits of quickly locating changepoints within data intensive settings are self-evident in numerous contexts from health to the environment to the stock market (see, for example, Chandola et al. (2013), Manogaran and Lopez (2018) and Gu et al. (2013) respectively).

One example application is the monitoring of changepoints in telecommunications data. As reported by, for example, Khomami (2016), in early February 2016 a major outage of the broadband network occurred across much of the UK. It later

transpired that this was due to a fault within one of the main core router units, which subsequently degraded the network to the extent experienced (Jackson, 2016). Figure 1.1 below displays the basic structure for a broadband network for a single gateway router. In broad terms, if a core or gateway router fails, then the effect on the network at large can be felt by the access layer, usually comprised of Edge Routers. A resultant ramification is then experienced by the computing servers of individual customers.

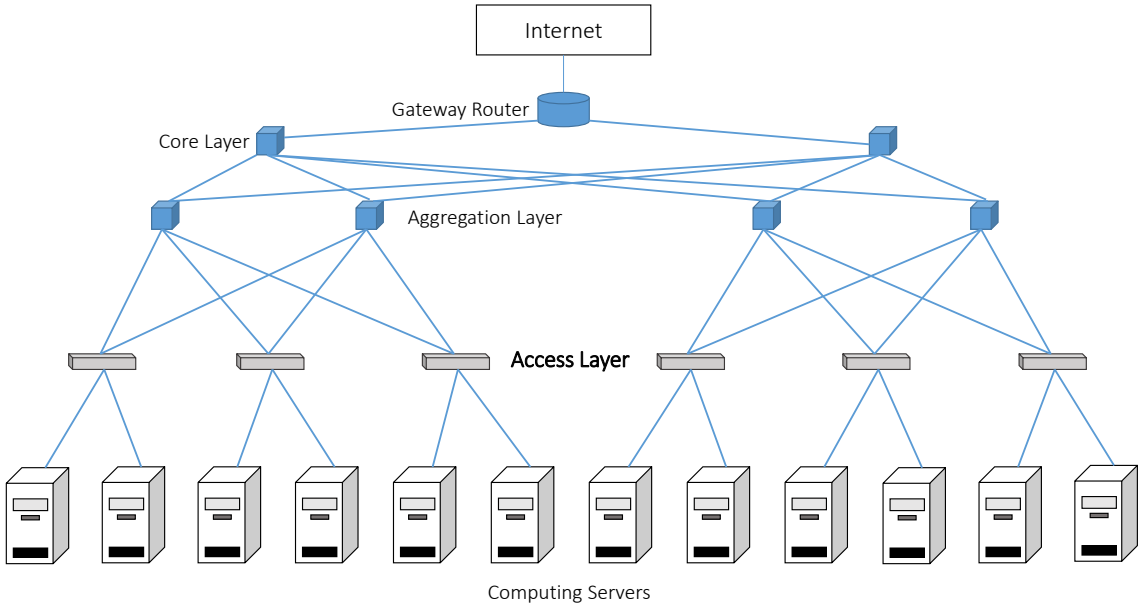

Figure 1.1: Basic topology of the broadband network for a single gateway router. Note that the routers within the access layer are usually referred to as Edge Routers. This image was inspired by a similar image from Fiandrino (2014).

British Telecommunications Ltd (BT) collects an extensive amount of information from each Edge Router at one-minute intervals. Every Edge Router is comprised of a number of shelves, while each shelf contains a number of ports. For each of these ports, a measurement is taken in a number of metrics. Even for a single Edge Router, this can lead to thousands of variates to analyse. Therefore, given the high sampling rate, subtle shifts in network performance can easily be missed, as indeed happened in early 2016, leading to a greater chance of a later, and more costly, failure.

On the other hand, there is also the potential for 'small-scale' changes occurring in a single shelf or port of an Edge Router, leading to a much more localised outage. In

such a situation, it is important for BT to be as 'targeted' as possible when reporting on a change in order to be suitably efficient with engineering resources. We discuss the dichotomy between localised and global changes further in Chapters 2 and 4.

Other authors have explored the challenge of finding changes within Edge Router data. For example, Rajaduray et al. (2004) examine the problem of handling highly non-smooth demand on an Edge Router by reacting with an 'Optimal Burst Switching' technique. More recently, Jutila (2016) investigated the idea of using changes in quality-of-service to trigger the implementation of 'adaptive edge computing solutions' in an Internet of Things context; while these and other technology-focused solutions are interesting and useful, it is the ability of the system to be reactive only when required in a data intensive setting that is the fundamental issue at hand. We therefore herein present novel algorithmic, computational and theoretical contributions to the changepoint detection problem in such settings which may be described as data intensive. This could either be because we are receiving data in an online fashion (i.e. a *data stream*), or else have a high-dimensional series, or simply need to analyse a (long) sequence of data as efficiently as possible.

In Chapter 2, we formally introduce the changepoint problem and give a summary of the current literature in multiple inference settings, with particular focus on online detection and change detection in multiple dimensions.

In Chapter 3, we consider the challenge of changepoint detection in the classical univariate, offline setting. In recent years, various means of efficiently detecting changepoints in such a setting have been proposed, with one popular approach involving minimising a penalised cost function using dynamic programming. In some situations, these algorithms can have an expected computational cost that is linear in the number of data points; however, the worst-case cost remains quadratic. We introduce two means of improving the computational performance of these methods by parallelising the dynamic programming approach. We establish that parallelisation can give substantial computational improvements: in some situations, the computational cost decreases roughly quadratically in the number of cores used. These parallel implementations are no longer guaranteed to find the true minimum of

the penalised cost. However, we show that they retain the same asymptotic guarantees in terms of their accuracy in estimating the number and location of the changes.

In Chapter 4, we extend the discussion to multiple dimensions in the offline setting. Detecting changepoints in datasets with many variates is a challenge of increasing importance. While several methods which are applicable in this domain have been introduced, the issue of timely and accurate location of changes remains, particularly if information on which variates are affected by the change is desired. In this chapter, we propose a method with these properties: SUBSET - a model-based approach which uses the penalised likelihood to detect changes for a wide class of parametric settings. We derive suitable values for the penalties using the Gaussian change in mean setting. Further, we demonstrate that, under these penalties, SUBSET provides theoretical power in detecting changepoints which can affect few or many of the variates. We also show that the method performs well empirically, even when the data are non-Gaussian. In addition, we demonstrate SUBSET's utility by considering count data on the number of terrorist incidents worldwide since the beginning of the 1970s.

In Chapter 5, we introduce a new method designed for the online, multivariate setting, which can be applied with very few parametric assumptions. Identifying changepoints across many variates while the data stream is still being observed is a challenging problem, but has a vast number of potential applications. Several methods for handling this problem have been proposed in recent years, however many of these make restrictive assumptions on the data generating processes of the stream. In addition, other methods generally require a great deal of tuning for specific problems, meaning limited versatility across multiple possible streams. We here introduce a new nonparametric method, OMEN, for which few assumptions on the data generating processes are required. Importantly, OMEN requires one value as an input, for which a sensible value can be found with minimal understanding of the stream. We show that OMEN has a good theoretical false alarm error rate, and exhibit this empirically. In addition, our synthetic examples show that OMEN has a competitive detection ability for even relatively 'difficult' types of change. The applicability of OMEN is

demonstrated on hourly records of wind speeds for various cities in Canada and Israel.

We conclude in Chapter 6, with a discussion of some potential avenues for future research. Additional materials may be found in the appendices. Appendix A provides proofs for some of the theoretical results given in Chapter 3, as well as further discussion of the parallelisation methods from a finite-sample perspective. The latter is conducted in the context of a second simulation study involving an increasing number of changepoints. Appendix B provides proofs for the theoretical results given in Chapter 4, as well as further empirical results on the use of SUBSET in simulated and real data settings. Appendix C gives a theoretical result on the false alarm error rate of OMEN. In addition, we provide further simulations to discuss a particular choice made in implementing the procedure, and conclude by showing the application of OMEN to another real data example.

Chapter 2

An Overview of Changepoint Detection

In this chapter, we discuss recent advances in the changepoint problem in order to place our new detection procedures into context, analysing the current state of the art while also discussing the precise issues which we return to in the chapters to follow.

Much historic work has focused on data which has been received in its entirety in advance of any inference, otherwise known as the *offline* setting. We therefore devote Section 2.1 to surveying well-established approaches to change detection for such data in the univariate case. In Section 2.2, we discuss how these have been extended to the offline setting under multiple variables, an area of increasing interest. Practically, the main issue when detecting changepoints in this setting has been striking an appropriate balance between computational feasibility and statistical power, and we explore this problem further. This issue is also a concern in the *online* setting. In this setting, given estimates for changepoints are required 'as fast as possible' - in particular, before we have collected all of the data. Therefore, keeping the number of false alarms as low as possible, while maintaining a useful true detection probability, is very important. We discuss existing approaches to the online changepoint detection problem in Section 2.3 for both univariate and multivariate data. We conclude with a general discussion in Section 2.4.

For the pertinent sections relating to each subsequent chapter, note that Chapter 3

concerns the offline detection of changepoints in the univariate setting by parallelising a common model-based approach. We discuss common model-based methods for changepoint detection in Section 2.1.2, where we also introduce the general framework in which our parallelisation methods operate.

Chapter 4 introduces a new offline method for detecting changepoints in the multivariate setting. To do this, the method computes a test statistic to find a single changepoint, before embedding this within one of a class of existing methods for detecting multiple changepoints given a test statistic for a single change. Methods which follow a similar strategy for the univariate setting are discussed in Section 2.1.1. Methods which are of this type in the multivariate setting are discussed in Section 2.2.1. One important contribution our new method of Chapter 4 makes is that it has competitive statistical power both for settings in which very few of the variates are affected by a changepoint, as well as in situations where most of the variates are affected. The problem of balancing power between these two settings is discussed in more detail at the beginning of Section 2.2. Note that the test statistic we use to detect a single change in Chapter 4 arises from considering a model-based approach in the multivariate setting under a single changepoint. We therefore briefly discuss the extension of model-based methods to the multivariate setting in Section 2.2.2.

Chapter 5 introduces a new online, nonparametric method for detecting changepoints in the multivariate setting. Our new method uses a 'memory window' in which we impose a two-stage test statistic for the presence of a changepoint. In Section 2.3.1, we compare this methodology with other established approaches which use a rolling test statistic to find changes in the univariate setting. We use Section 2.3.2 to give an indication of the current sparsity of the literature in the online location of changes in a multivariate setting.

2.1 Classical Univariate Changepoint Techniques

In this section, we present two classes of procedure for locating a changepoint in a univariate data sequence, and discuss the relative merits of each. The section concludes with a review of other recently proposed methods that do not readily fall into either of the classes described.

We first present the problem which the methods of this section seek to resolve. Let $y_{1:n} = (y_1, \ldots, y_n) \in \mathbb{R}^n$ be a data sequence. Suppose that there are $m < n$ changepoints in the system occurring at time points $0 = \tau_0 < \tau_1 < \tau_2 < \ldots < \tau_m < \tau_{m+1} = n$ with $(\tau_1, \ldots, \tau_m) \in \mathbb{N}^m$, such that

$$y_j \sim G_k \text{ for } \tau_{k-1} + 1 \leq j \leq \tau_k \text{ for } k \in \{1, \ldots, m+1\}. \tag{2.1.1}$$

Here $G_1, \ldots, G_{m+1}$ are a sequence of data-generating processes such that $G_k \nsim G_{k+1}$ for $k \in \{1, \ldots, m\}$. Of particular interest in the literature is the setting where these data-generating processes come from the same family of distributions and differ only in terms of some set of parameters. Examples of such parameters include the mean (Gupta and Chen, 1996; Lebarbier, 2005; Srivastava and Worsley, 1986; Wang et al., 2007), variance (Inclan, 1993; Tsay, 1988; Whitcher et al., 2000; Wichern et al., 1976), and event rate, among others. In these instances, (2.1.1) becomes

$$y_j \sim g\left(.|\boldsymbol{\mu}_k\right) \text{ for } \tau_{k-1} + 1 \leq j \leq \tau_k \text{ for } k \in \{1, \ldots, m+1\} \tag{2.1.2}$$

where $g(.|.)$ is some pre-specified family of densities, and $\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_{m+1}$ are a sequence of latent parameter vectors with $||\boldsymbol{\mu}_k - \boldsymbol{\mu}_{k+1}||_0 > 0$ for $k \in \{1, \ldots, m\}$. Note that for problem (2.1.2), we typically additionally assume that the data are conditionally independent given these latent parameter vectors. For most of this chapter, our focus shall be on problem (2.1.2) rather than the more general (2.1.1). However, we shall maintain a commentary on techniques suitable for the nonparametric setting, where appropriate.

Given either (2.1.1) or (2.1.2), one seemingly intuitive solution is to define a suitable test statistic, or score, for a given segmentation, enumerate all possible sets of changepoints in the sequence, and compare all score values. Practically,

however, this approach is computationally prohibitive, with 2^{n-1} possible models when m is unknown, and $\binom{n-1}{m}$ when m is known. Therefore, several existing methods circumvent this problem by detecting multiple changepoints one at a time. Broadly speaking, there are two popular ways of doing this. The first of these concerns the construction of a test statistic which can be used to search the data for the presence of a single changepoint. Such methods individually are referred to as At Most One Changepoint (AMOC) approaches. Multiple changepoints are then subsequently found by considering points before and after an estimated changepoint separately, recursively seeking further changepoints in each sub-region. The second class of methods uses a model-based approach. These typically include a 'pass' through the data using recursive updates, considering the likely history of the sequence to determine whether the current location may be labelled as a candidate changepoint. In the below, we review pertinent literature associated with both approaches.

2.1.1 AMOC Approaches and Extensions to the Detection of Multiple Changes

AMOC Detection

One of the most established approaches to detecting changes in the AMOC setting has been to define a suitable test statistic, $T(t; y_{1:n})$, for placing a changepoint at time $1 \leq t \leq n-1$. We can then test, against a null hypothesis of no change, for the presence of a change, by finding $\max_t T(t; y_{1:n})$ and comparing this to a suitable threshold value, say $\xi(n)$.

Perhaps the most common changepoint problem is detecting changes in mean. For this setting, a natural choice of $T(.;.)$ is the CUSUM statistic, defined as

$$T(t; y_{1:n}) = \sqrt{\frac{t(n-t)}{n}} \left| \frac{1}{n-t} \left(\sum_{j=t+1}^{n} y_j \right) - \frac{1}{t} \left(\sum_{j=1}^{t} y_j \right) \right|. \tag{2.1.3}$$

This has been used since at least Hinkley (1971) for the change in mean problem under Gaussian noise, building on a similar procedure of Page (1954). More recently, the CUSUM test statistic has been used by a variety of authors (Kass-Hout et al.,

2012; Kulkarni et al., 2015; Pranuthi et al., 2014; Tartakovsky et al., 2013) in contexts as diverse as detecting linguistic change to cybersecurity to flu epidemic modelling. This popularity is partially due to the well-established theory attached to the CUSUM statistic, with classical results from Lee et al. (2006), Lee et al. (2004) and Ploberger and Krämer (1992), among others. These results demonstrate that, in the null setting, the set of CUSUM statistics for $1 \leq t \leq n-1$ follows a Brownian bridge. This fact enables the setting of a penalty to give a worst-case probability of a Type I error. However, it should be noted that, in practice, such penalties are often much too conservative. Indeed, some authors, such as Gallagher et al. (2013), instead set penalties based on simulations from the appropriate null model.

Note that the CUSUM test statistic for a change at a specific location is equivalent to performing a likelihood ratio test for a change in mean at that location under a model with i.i.d. Gaussian noise with known variance. Hence, the CUSUM typically has high power in situations where the noise is i.i.d. Gaussian, or else well approximated by i.i.d. Gaussians. However, in other situations, such as highly correlated or heavy-tailed noise, the CUSUM statistic loses power compared to test statistics which make more appropriate modelling assumptions. Nevertheless, over the years the CUSUM statistic has been adapted to other settings. Inclán and Tiao (1994) introduce a normalised CUSUM statistic for the change in variance problem. Csörgő and Horváth (1988) discuss a scaled version of the CUSUM statistic in the nonparametric i.i.d. setting. Robbins et al. (2011) introduce an appropriately adjusted CUSUM for correlated data. We discuss some additional extensions to the multivariate setting in Section 2.2 and Chapter 4.

A closely-related alternative to the CUSUM is the Worsley likelihood ratio test (Worsley, 1979), also for the Gaussian change in mean problem, given by

$$T\left(t; y_{1:n}\right) = \frac{(n-2)^{\frac{1}{2}} V}{\left(1-V^2\right)^{\frac{1}{2}}}, \tag{2.1.4}$$

where

$$V = \max_{1 \leq t \leq n-1} \frac{\sqrt{\frac{t(n-t)}{n}} \left| \frac{1}{t} \left(\sum_{j=1}^{t} y_j \right) - \frac{1}{n-t} \left(\sum_{j=t+1}^{n} y_j \right) \right|}{\sqrt{\sum_{j=1}^{n} \left(y_j - \frac{1}{n} \sum_{i=1}^{n} y_i \right)^2}}. \tag{2.1.5}$$

This particular test statistic has been used by Pranuthi et al. (2014) and Shen (2016), among others. Note that the Worsley likelihood ratio test is equivalent to the CUSUM statistic scaled by an estimate of the variance. Therefore, unlike with the classical CUSUM given in (2.1.3), we do not require that the variance be known *a priori*. However, like the CUSUM, it will lose power outside of situations in which the noise is approximately i.i.d. Gaussian.

There has also been some success in recent years at deriving test statistics based on a "windowed-CUSUM" type approach. This has the advantage of just examining a test statistic within a small interval of the data. This mitigates the problem of there being potentially several changes within the sequence, which could corrupt the test statistic and cause a false negative result. A prominent example of a windowed approach is a procedure based on MOSUM statistics for which a bandwidth parameter is required (Hušková and Slabý, 2001; Eichinger and Kirch, 2018). While this method has been shown to be consistent and efficient, in practice the selection of this bandwidth parameter is extremely important. If it is too large, then multiple changes can be present in the window, and the main advantage over CUSUM is lost. If it is too small in relation to n, then the test statistic does not converge as desired in the null setting.

The final two choices of test statistic, $T(.;.)$, we mention here are the Mood and Mann-Whitney U test statistics. These two classical nonparametric tests for the presence of a changepoint are based on a computation of the ranks $(r(y_j))_{j=1}^{n}$, where $r(y_j) = \sum_{i \neq j} \mathbb{1}\{x_i \leq x_j\}$. In particular, $T(.,.)$ is given as

$$T(t; y_{1:n}) = \sum_{i=1}^{t} \left(r(y_i) - \frac{n+1}{2} \right)^2, \tag{2.1.6}$$

and

$$T\left(t; y_{1:n}\right) = \sum_{i=1}^{t} r\left(y_i\right) - \frac{t(t+1)}{2}, \tag{2.1.7}$$

for the Mood and Mann-Whitney U tests respectively; see, for example, Ross et al. (2011) for further discussion. Note that as ranking the data points is equivalent to sorting, the complexity of computing the Mood and Mann-Whitney test statistics is $\mathcal{O}\left(n \log n\right)$ in the worst case. This contrasts with the computation of the statistics in the CUSUM procedure, which is linear in n. However, these test statistics are invariant to monotone transformations of the data, meaning that they are robust to distributional assumptions.

Extending to Multiple Changes

In the previous section, we reviewed AMOC-based approaches. We now turn our focus to the setting of multiple changepoints. Following the computation of $T_n = \max_t T\left(t; y_{1:n}\right)$, we check if $T_n > \xi\left(n\right)$, where $\xi(.)$ is a threshold chosen based on asymptotic null behaviour of the test statistic, or through simulation from the null setting to achieve a desired Type I error. If $\xi(n)$ is exceeded, then a changepoint is placed at $\arg\max_t T\left(t; y_{1:n}\right)$. To locate potential further changepoints, a form of Binary Segmentation is then typically used. Binary Segmentation as a method for dividing time series into segments dates back to at least Scott and Knott (1974), who were in turn building on similar ideas from Edwards and Cavalli-Sforza (1965). For the changepoint detection problem in the classical univariate setting, Binary Segmentation proceeds as follows. To begin with, for a general test statistic $T(.;.)$, a changepoint is estimated at $\hat{\tau}_{(1)} = \arg\max_t T\left(t; y_{1:n}\right)$. Then, the sequence is segmented into two separate sequences, $y_{1:\hat{\tau}_{(1)}}$ and $y_{(\hat{\tau}_{(1)}+1):n}$. The process is then repeated, with $\max_t T\left(t; y_{1:\hat{\tau}_{(1)}}\right)$ and $\max_t T\left(t; y_{(\hat{\tau}_{(1)}+1):n}\right)$ compared to $\xi\left(\hat{\tau}_{(1)}\right)$ and $\xi\left(n - \hat{\tau}_{(1)}\right)$ respectively. This continues iteratively, such that if $\xi\left(.\right)$ is not exceeded for a particular subset of the sequence, it is concluded that no changepoints are present in this region. The region is then removed from further consideration.

Binary Segmentation, usually using the CUSUM test statistic, has been a popular multiple changepoint detection method for many years, largely due to its

computational efficiency and ease of implementation. Examples of its use include Hernandez-Lopez and Rivera (2014), Mahmoud et al. (2007), Yang (2004) and Zdansky (2006), where it is implemented in a wide range of practical contexts from video surveillance to NASA wind tunnel experiments to detecting periods of 'good form' within sports. Moreover, various theoretical results can be established for the Binary Segmentation procedure in the change in mean setting. For example, Venkatraman (1992) states that, under the assumptions that the number of changepoints remains fixed and that the changepoints are spaced apart by a minimum distance of $\mathcal{O}\left(n^{7/8}\right)$, the correct change locations will be found in probability as $n \to \infty$ with error at most $\mathcal{O}\left(n^{3/4}\right)$. Given that this error is $o(n)$, we can refer to Binary Segmentation in this setting as asymptotically consistent from an infill perspective. That is, if the changepoints are placed at fixed 'proportions' in the sequence, say $\theta_1, \ldots, \theta_m$ such that $\lfloor \theta_i n \rfloor = \tau_i$ for $i \in \{1, \ldots, m\}$, then as $n \to \infty$, these change proportions will be correctly estimated.

The theoretical properties of Binary Segmentation, and variant methods, are still a significant area of interest, with recent literature such as Chen et al. (2011), Cho and Fryzlewicz (2012) and Fryzlewicz (2014) improving upon the results of Venkatraman (1992). However, despite the good theoretical performance of the method, Binary Segmentation has some notable drawbacks. Most significantly, there is the issue of masking (Padmore, 1992). Masking is defined as those situations where the presence of multiple changepoints causes the test to fail to detect at least one change. This is especially problematic in a sequence with many changepoints, as not only is masking much more likely, but multiple tests for changes across relatively short segments increases the chance of overfitting. We illustrate the problems of overfitting and masking in a simple example in Figure 2.1. This was created with the aid of the `changepoint` package of Killick et al. (2016), using the default arguments (with `penalty=''BIC''`) for Binary Segmentation, with the maximum possible number of changepoint estimates set to 25. As can be seen from Figure 2.1, Binary Segmentation fails to detect three true changepoints whose effects cancel one another out, at $t = 200, 220$ and 240.

As a result of these problems, several authors have suggested variations to the Binary Segmentation approach. A well-known example of such an alternative is Wild Binary Segmentation (Fryzlewicz, 2014), which greatly reduces the problem of masking by uniformly drawing M intervals (where M is large) of the data sequence and performing the tests solely across the intervals in question. The central idea underpinning this approach is that, for sufficiently large M, the probability that there is an interval containing exactly one changepoint is high. One common criticism of the method is that the recommended default setting of M is often taken to be very large. This means that the method can be computationally cumbersome. In addition, it can lead to more false positives due to testing across many different intervals. In practice, the latter issue can be overcome by increasing the penalty appropriately. Meanwhile, the computational drawbacks were addressed in a recent article by the same author (Fryzlewicz, 2019), in which a similar method, referred to as WBS2, was introduced. It is broadly this segmentation procedure which we use to search for multiple changes in our novel method in Chapter 4. We illustrate the use of Wild Binary Segmentation on the same example as for Binary Segmentation in Figure 2.1. This plot was created with the aid of the `wbs` package of Baranowski and Fryzlewicz (2015), with the default parameters used. Note that the performance of Wild Binary Segmentation is improved over Binary Segmentation, however one change is still missed, at $t = 200$.

Another popular alternative to Binary Segmentation is Circular Binary Segmentation (Olshen et al., 2004; Venkatraman and Olshen, 2007), which simultaneously tests for persistent and epidemic changes. Note that the latter is defined as a change from a 'regular' regime to an 'irregular' regime and back again. Circular Binary Segmentation is most typically applied in the context of genomic data (Cheng et al., 2015; Verhaak et al., 2010; Zack et al., 2013). However, the issue of simultaneously testing the length and location of the abnormal interval reduces the efficiency of the method in general.

Work on finding further alternatives to Binary Segmentation continues, with very recent additions to the literature. See, for an additional example, the Narrowest-Over-Threshold approach of Baranowski et al. (2018). This indicates that

the popularity of Binary Segmentation methods, both directly in the changepoint literature and beyond, is likely to continue.

In the next section, we turn to consider another class of changepoint detection methods which do not require Binary Segmentation or alternatives to search for multiple changes. Informally, instead of searching for a changepoint that maximises some test statistic, these model-based methods search for changepoint candidates based on an observed history of the sequence.

2.1.2 Model-based Changepoint Detection with Recursive Updates

In the previous section, we examined changepoint detection methods which maximised some single test statistic and subsequently used Binary Segmentation to search for multiple changes. While Binary Segmentation can be very fast, the approach has the potential to incorrectly assign or miss changepoints, particularly if n is not sufficiently large or there are extremely short segments in the data. We therefore now turn to consider a second broad class of changepoint detection procedures for which interest lies in detecting each changepoint based on 'one pass' through the data sequence. Formally, the general setup involves the use of a model for the sequence, which we then recursively update with each successive point in the pass. Some post-processing is then typically required to find the location of the changepoints based on optimally resolving the model.

Cost Function Approaches

Many popular model-based methods involve the use of a penalised cost function. In such a formulation of the problem, which typically assumes that each of the data generating mechanisms from (2.1.1) or (2.1.2) are stationary, we require a *segment cost function*, $\mathcal{C}\left(y_{(i+1):j}\right)$. This measures how well we can fit data $y_{(i+1):j}$ without requiring a changepoint. A typical construction involves modelling the data within a segment and basing the cost function on the negative of the maximum log-likelihood

for the chosen model.

Additionally, a means of penalising the presence of changepoints is needed to avoid overfitting. The usual approach involves defining a single *penalty* value, β. This leads, for a choice of the number of changepoints, $\hat{m}$, and the corresponding change locations, $\hat{\tau}_1, \ldots, \hat{\tau}_{\hat{m}}$, to the general *global cost function*

$$C\left(\hat{m}, \hat{\tau}_1, \ldots, \hat{\tau}_{\hat{m}} | \mathcal{C}(.), f(.), \beta\right) = \sum_{i=1}^{\hat{m}+1} \mathcal{C}\left(y_{(\hat{\tau}_{i-1}+1):\hat{\tau}_i}\right) + \beta f(\hat{m}), \tag{2.1.8}$$

such that $\hat{\tau}_0 = 0$ and $\hat{\tau}_{\hat{m}+1} = n$, for some increasing function $f(.)$ and some appropriately chosen $\mathcal{C}(.)$ and β. We typically take $f(\hat{m}) = \hat{m}$, so that (2.1.8) becomes, up to one β term

$$C\left(\hat{m}, \hat{\tau}_1, \ldots, \hat{\tau}_{\hat{m}} | \mathcal{C}(.), \beta\right) = \sum_{i=1}^{\hat{m}+1} \left[\mathcal{C}\left(y_{(\hat{\tau}_{i-1}+1):\hat{\tau}_i}\right) + \beta\right]. \tag{2.1.9}$$

The challenge, given $\mathcal{C}\left(.\right)$ and β, is then to find $\hat{m}$ and $\hat{\tau}_1, \ldots, \hat{\tau}_{\hat{m}}$ such that (2.1.9) is minimised. Before introducing existing methods which do this, we discuss several typical choices for the penalty function $f(\hat{m})$, the segment cost function $\mathcal{C}(.)$, and the penalty β.

Cost Function Approaches: Choice of $f(\hat{m})$

As stated above, $f(\hat{m}) = \hat{m}$ is by far the most common penalty function selected within the changepoint literature. After this section, we assume that the total penalty increases by some fixed β on the detection of each new changepoint. However, this choice is by no means universal. Indeed, in the Gaussian change in mean setting (with variance σ^2), there are some alternatives which have received increasing interest in recent years. For example, several authors have suggested penalties of the form

$$\beta f(\hat{m}) = \frac{\hat{m}}{n}\sigma^2\left(c_1 \log\left(\frac{n}{\hat{m}}\right) + c_2\right), \tag{2.1.10}$$

with Lebarbier (2005) recommending simulation to set the constants c_1 and c_2, so that the model selection realises the minimax of an appropriate risk ratio. Indeed, many of the non-linear penalty functions arise from more general model selection problems.

For instance, Massart (2004) suggests the penalty form given by (2.1.10), in addition to several others, in the wider context of non-asymptotic model selection. However, the tuning required for many of these penalties, as for (2.1.10), remains a disadvantage in a changepoint context, as does the typical requirement that the problem is a change in mean under Gaussian noise.

Other recent work has focused on the application of penalties used within the regression analysis literature. (Note that we discuss further the contemporary place of changepoint detection with varying-coefficient models, including regression models, in Section 2.4.1.) Most notably, Harchaoui and Lévy-Leduc (2010) remark on the similarity between detection of changes in mean under Gaussian noise and the Least Absolute Shrinkage Selection Operator (LASSO) of Tibshirani (1996). They demonstrate that when a total penalty is taken which is proportional to the sum of the absolute differences between the estimated means between consecutive segments, then the problems become equivalent (assuming that we have chosen to minimise squared error loss - see the next section for more information). Several others, such as Tibshirani and Wang (2008), have examined similar total variation penalties for changepoint detection, with other penalties from regression analysis such as SCAD (Fan and Li, 2001), itself a function of the total variation, being applied. However, as some recent authors, such as Ng et al. (2018), have noted, such penalties have difficulty in detecting the true number of changepoints. They suggest an alternative penalty based on the bridge penalty of Frank and Friedman (1993) and Fu (1998). Using this penalty does give a greater guarantee of consistency than the other regression-based penalties we have discussed, although the authors note that this is subject to the number of changepoints not increasing 'too fast' as the length of the sequence grows.

Cost Function Approaches: Choice of Segment Cost Function

As previously stated, a very common choice for the segment cost function is based on the negative of the maximum log-likelihood for a particular model of the data. For example, if data within a segment are assumed to be i.i.d. from a family of models with density $f(y|\mu)$, then $\mathcal{C}(y_{(i+1):j}) = -2\max_{\mu}\sum_{k=i+1}^{j}\log f(y_k|\mu)$ is a natural choice;

see Eckley et al. (2011), Hawkins (2001) and others.

Alternatively, for a given type of change, we can use generic loss functions. For instance, in the change in mean setting we can take

$$\mathcal{C}(y_{(i+1):j}) = \min_{\mu} \sum_{k=i+1}^{j} l(y_k - \mu), \tag{2.1.11}$$

where a common choice for the loss function, $l(.)$, is squared error loss. One criticism with using squared error loss for the change in mean problem is that a typical procedure which minimises the global cost function may not then be robust to the presence of outliers. In addition, if a chosen likelihood model has heavy tails (where in practice this can just mean not sub-Gaussian), setting the penalty value to avoid overfitting while maintaining power becomes much more challenging. To mitigate the former issue somewhat, it is relatively common practice (Bai, 1995, 1998; Huber, 2011; Hušková, 2013) to use absolute error loss in place of residual sum of squares. Other robust choices include Huber loss and biweight loss (Huber, 2011; Fearnhead and Rigaill, 2019), which are defined as

$$\mathcal{C}\left(y_{(i+1):j}\right) = \sum_{k=i+1}^{j} \begin{cases} \left(y_k - \hat{\mu}(y_{(i+1):j})\right)^2 & \text{if } \left|y_k - \hat{\mu}(y_{(i+1):j})\right| < K \\ K\left|y_k - \hat{\mu}(y_{(i+1):j})\right| - K^2 & \text{otherwise,} \end{cases} \tag{2.1.12}$$

and

$$\mathcal{C}\left(y_{(i+1):j}\right) = \sum_{k=i+1}^{j} \begin{cases} \left(y_k - \hat{\mu}(y_{(i+1):j})\right)^2 & \text{if } \left|y_k - \hat{\mu}(y_{(i+1):j})\right| < K \\ K^2 & \text{otherwise,} \end{cases} \tag{2.1.13}$$

respectively. Here, K is a suitably chosen value and $\hat{\mu}(y_{(i+1):j})$ is an estimate for the segment parameter which minimises the segment cost function.

Many other popular current choices of cost function are detailed in Truong et al. (2019) and references therein. We make mention of two of these here. The first is based on the empirical cumulative distribution function, and is therefore naturally equipped to deal with change detection in the nonparametric setting

$$\mathcal{C}\left(y_{(i+1):j}\right) = -(j-i)\sum_{z=1}^{\omega} \frac{\hat{F}_{(i+1):j}(z)\log\hat{F}_{(i+1):j}(z) + (1-\hat{F}_{(i+1):j}(z))\log(1-\hat{F}_{(i+1):j}(z))}{(z-0.5)(\omega - z + 0.5)}. \tag{2.1.14}$$

Note that ω is a signal length/information window parameter and $\hat{F}_{(i+1):j}(.)$ is the empirical cumulative distribution defined here by

$$\hat{F}_{(i+1):j}(z) = \frac{1}{j-i}\left[\sum_{k=i+1}^{j} \mathbb{1}\{y_k < z\} + 0.5 \times \mathbb{1}\{y_k = z\}\right].$$

Another option in the non-parametric setting is to use a kernel-based cost function, as suggested by Harchaoui and Cappe (2007), Harchaoui et al. (2009), Garreau and Arlot (2018) and many others. As a cost function, we can set

$$\mathcal{C}\left(y_{(i+1):j}\right) = \sum_{k=i+1}^{j} \kappa(y_k, y_k) - \frac{1}{j-i}\sum_{k,l=i+1}^{j} \kappa(y_k, y_l), \tag{2.1.15}$$

where $\kappa(.,.)$ is a suitable kernel function. Note that if $\kappa(y_k, y_l) = y_k y_l$, then we obtain standard squared error loss, (2.1.16). Both of these nonparametric options can be inefficient, as we discuss shortly.

We conclude this section on segment cost functions by remarking that there can be close links between this approach to detecting changepoints and those methods discussed in Section 2.1.1. Indeed, some segment cost functions are equivalent to certain test statistics which we have discussed for the AMOC problem. The most important example is if we detect a change in mean using the cost function that is the sum of squared residuals. This gives a segment cost function of

$$\mathcal{C}\left(y_{(i+1):j}\right) = \sum_{k=i+1}^{j}\left(y_k - \frac{1}{j-i}\sum_{l=i+1}^{j} y_l\right)^2 = \sum_{k=i+1}^{j} y_k^2 - \frac{1}{j-i}\left(\sum_{k=i+1}^{j} y_k\right)^2. \tag{2.1.16}$$

This cost function has, in particular, been used in the Gaussian setting; see, for example, Xie et al. (2007) and Yao (1988). Under this choice of segment cost, the difference in cost of adding a single change at a specific location $\hat{\tau}_1$ is

$$\begin{aligned} C\left(0|\mathcal{C}(.),\beta\right) - C\left(1,\hat{\tau}_1|\mathcal{C}(.),\beta\right) &= \frac{1}{\hat{\tau}_1}\left(\sum_{k=1}^{\hat{\tau}_1} y_k\right)^2 + \frac{1}{n-\hat{\tau}_1}\left(\sum_{k=\hat{\tau}_1+1}^{n} y_k\right)^2 - \frac{1}{n}\left(\sum_{k=1}^{n} y_k\right)^2 - \beta \\ &= \frac{\hat{\tau}_1\left(n-\hat{\tau}_1\right)}{n}\left(\frac{1}{\hat{\tau}_1}\left(\sum_{k=1}^{\hat{\tau}_1} y_k\right) - \frac{1}{\left(n-\hat{\tau}_1\right)}\left(\sum_{k=\hat{\tau}_1+1}^{n} y_k\right)\right)^2 - \beta. \end{aligned}$$

which is just the square of the CUSUM statistic given in (2.1.3) for a change at $\hat{\tau}_1$ minus the penalty for adding a change, β. This means that if we used the penalised cost approach to detect a single changepoint by minimising (2.1.9) under

the constraint that $m \leq 1$; then our estimate of whether there is a change and where it occurs would be identical to that obtained by performing a CUSUM test for a single change with a threshold $\sqrt{\beta}$ for the test statistic. We further discuss this with a novel consistency result in Chapter 3. The link between the CUSUM statistic and minimising the penalised residual sum of squares is additionally further explored in Chapter 4.

Cost Function Approaches: Choice of Penalty

We now discuss the choice of penalty value β within (2.1.9). Classical information criteria are some of the most extensively employed options in this context. For example, see the default penalty options in the changepoint package of Killick et al. (2016), as well as Gupta and Chen (1996) and Eckley et al. (2011). One very popular information criterion used as a penalty in this setting is the Schwarz Information Criterion (SIC), also known as the Bayesian Information Criterion (BIC), first introduced by Schwarz (1978). For a general problem, if n is the number of samples and b is the number of parameters estimated by the model, then the BIC/SIC for the vector of estimated parameters $\hat{\boldsymbol{\mu}} \in \mathbb{R}^b$ is

$$C_{SIC}(\hat{\boldsymbol{\mu}}) = b \log n - 2 \log \mathcal{L}(\hat{\boldsymbol{\mu}}), \tag{2.1.17}$$

where $\mathcal{L}(.)$ is the likelihood function for the chosen model. Note that this is (2.1.9) with $\mathcal{C}(.)$ set as twice the negative log-likelihood within the segment and $b \log n = \hat{m}\beta$. For each new estimated changepoint, it is required that we estimate the parameters which may change either side of any newly-placed change. In addition, the estimated changepoint itself is an additional parameter. For example, in the scenario where we allow for a change in a single parameter only (say, the mean), each changepoint adds an additional two parameters to the model: the estimated change location, and an additional mean parameter. The latter is needed as there are now two "new" means either side of the change, where previously there was just one estimated mean within the segment. Therefore, under the BIC, $\beta = 2 \log n$. See Yao (1988) and Chapter 3 for a theoretical justification of this choice in the Gaussian change in mean setting.

As noted by Zhang and Siegmund (2007), the BIC penalty has difficulty in settings where the noise is not sub-Gaussian. In such a scenario, they propose using the Modified Bayesian Information Criterion (MBIC). When the segment cost function is twice the negative log-likelihood, the MBIC gives a total penalty of

$$\beta f\left(\hat{m}\right) = \left(2\hat{m} - 1\right) \log n + \sum_{i=1}^{\hat{m}+1} \log\left(\hat{\tau}_i/n - \hat{\tau}_{i-1}/n\right). \tag{2.1.18}$$

The final choice we mention here is the Akaike Information Criterion (AIC), first introduced by Akaike (1974), for which

$$C_{AIC}(\hat{\boldsymbol{\mu}}) = 2b - 2\log \mathcal{L}(\hat{\boldsymbol{\mu}}). \tag{2.1.19}$$

This gives $\beta = 4$ in the setting where a single parameter is subject to a change. Given the lack of scaling with n, it is unsurprising that the use of this penalty typically results in a very high false positive rate, as noted by Haynes et al. (2017a), Jones and Dey (1995), Reeves et al. (2007) and others. Therefore, AIC is rarely used in practice.

The choice of penalty, or alternatively the threshold $\xi(n)$ for AMOC detection methods, is often the most challenging modelling issue for a given problem. This is especially true in situations where sensible choices for the likelihood are not known *a priori*. Many of the theoretical results introduced in Chapters 3, 4 and 5 discuss the best setting of the penalty for a limited class of generating processes for a given method. For now, we focus on the use of typical segment cost functions and penalty choices in the literature to date.

Cost Function Approaches: Existing Methods

We now discuss some existing methods which minimise (2.1.9). One of the first changepoint detection procedures to use a cost function approach was introduced by Yao (1984). This method minimises a cost function of type (2.1.9) from a Bayesian perspective using forward and backward recursions. A similar means of minimising (2.1.9), known as Segment Neighborhood, is due to Auger and Lawrence (1989). In this method, a constraint is placed on the maximum number

of changepoints which may be estimated by the procedure, say $\hat{m} \leq Q < n$. Dynamic programming is then used to search through all possible segmentations with at most Q estimated changepoints. The segmentation with the smallest total cost is returned, giving the estimated change locations. Formally, this is done by defining $c^q_{a,b}$ as the cost of the best partitioning of $y_{a:b}$ into q segments. The objective is to find $c^Q_{1:n}$, and hence the best partition. The first step in this is to calculate

$$c^1_{i+1,j} = \mathcal{C}(y_{(i+1):j}), \forall i, j \text{ s.t. } i < j.$$

The method then proceeds recursively, using

$$c^q_{1,j} = \min_{v \in \{1,\ldots,j\}} \left(c^{q-1}_{1,v} + c^1_{v+1,j} \right).$$

One issue with Segment Neighborhood is that the computational cost is $\mathcal{O}(Qn^2)$. In settings where there is a great deal of uncertainty about the number of changes - in particular, in situations where very short segments are possible or even common - it is desirable from a modelling perspective to set $Q \sim n$. In this way, all possible segmentations are searched. However, by doing this, the computational cost incurred is $\mathcal{O}(n^3)$. This is prohibitive from the standpoint of quick decision-making, or even for attempting to extend the method to multiple dimensions.

The Optimal Partitioning method of Jackson et al. (2005) somewhat fixes the issue of computational cost. Like Segment Neighborhood, Optimal Partitioning uses dynamic programming to minimise (2.1.9). However, in addition, Optimal Partitioning conditions on the location of the most recent changepoint to determine whether a changepoint should also be placed at a particular location in the 'current segment'. Formally, the method defines

$$F(i) = \min_{m', 0 < \hat{\tau}_1 < \ldots < \hat{\tau}_{m'} < i} \sum_{k=1}^{m'+1} \left[\mathcal{C}\left(y_{(\hat{\tau}_{k-1}+1):\hat{\tau}_k} \right) + \beta \right],$$

the cost of the segmentation which minimises (2.1.9) for $y_{1:i}$. Note here we use $i = \hat{\tau}_{m'+1}$ for notational convenience, despite the fact that i itself may not be a changepoint. After setting $F(0) = -\beta$, a similar recursion step to the Segment Neighborhood procedure is then used, namely

$$F(j) = \min_{i<j} \left\{ F(i) + \mathcal{C}(y_{(i+1):j}) + \beta \right\}.$$

This enables the computation of $F(n)$, the optimal cost of the whole sequence. The estimated change locations then follow naturally.

Note that Optimal Partitioning does not require an upper limit on the number of estimated changes. In addition, the worst-case computation time of the method is $\mathcal{O}(n^2)$, as the computation of $F(j)$ is linear in j for each $j = 1, \ldots, n$. This is more desirable from a practical perspective, especially as Optimal Partitioning finds the exact solution to (2.1.9). However, we remark that, in general, the performance of Binary Segmentation is still preferable. (Binary Segmentation is often asymptotically linear in n unless, for example, the number of changepoints also grows linearly with n.)

A further computational saving is made by the Pruned Exact Linear Time (PELT) method of Killick et al. (2012), which adds an additional pruning step to Optimal Partitioning. This pruning step makes use of the observation that introducing a changepoint reduces the cost of the sequence (possibly up to the inclusion of the penalty). That is, if the optimal cost of the sequence up to time $n_2 > n_1$ satisfies

$$F(n_1) + \mathcal{C}\left(y_{(n_1+1):n_2}\right) + \beta \geq F(n_2),$$

then for any $n_3 > n_2$, n_1 cannot be the most recent changepoint. In other words, the optimal cost up to time n_2 is at most the best cost obtained conditioning on n_1 being the most recent change prior to n_2. In practice, this means that when conditioning on the location of the most recent changepoint, the method will typically only consider points since the most recent true changepoint. While this is intuitive, the computational gains from this over Optimal Partitioning are impressive. In particular, PELT has a linear computational cost in the setting where the maximum segment length remains bounded (for example, if the number of changepoints grows linearly with n). However, PELT is an $\mathcal{O}(n^2)$ method in the worst case, which can be seen in situations where the number of changepoints remains fixed as $n \to \infty$. We discuss this issue in more detail in Chapter 3, and develop two means of ensuring that the worst-case computational cost may become linear in n through parallelisation. The performance of PELT on the same example as for Binary Segmentation and Wild

Binary Segmentation is shown in Figure 2.1. Again, the `changepoint` package was used to generate the plot, with the same inputs as for Binary Segmentation.

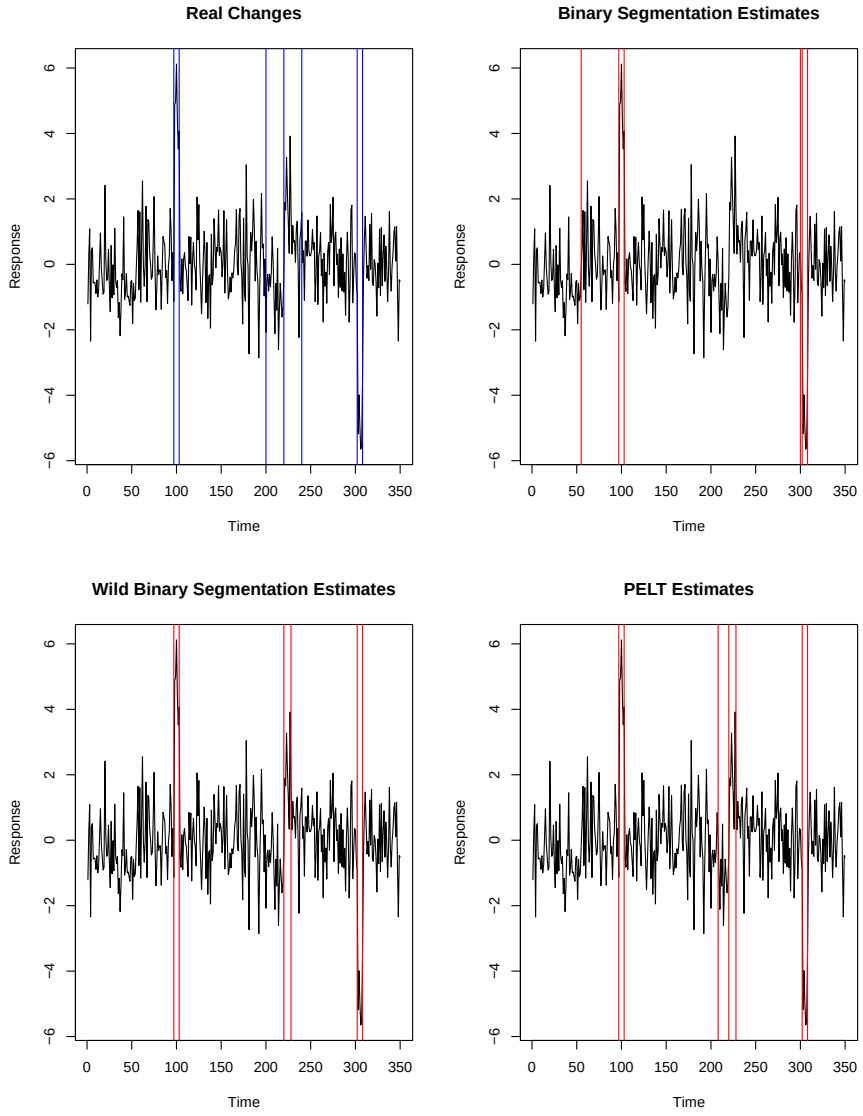

Figure 2.1: A sequence of length 350 exhibiting seven changes in mean - at the times shown by blue vertical lines - under Gaussian noise (top left). The change locations estimated by Binary Segmentation, Wild Binary Segmentation and PELT are shown as red vertical lines (top right, bottom left and bottom right respectively). Binary Segmentation fails to find the changes at $t = 200, 220$ and 240 due to masking, and incorrectly places two additional changespoints at $t = 55$ and $t = 300$. Wild Binary Segmentation does not place any spurious changes into the sequence, and detects all but one of the changepoints. PELT does not place any spurious changes into the sequence, and detects all changepoints present, albeit with a slight location error in two cases.

Despite the issue of a worst-case quadratic cost, PELT has become a very popular

changepoint detection method within the literature; see, for example, Figueroa et al. (2014), Hilborn et al. (2017), Murray et al. (2016) and Richardson et al. (2018). There are a multitiude of other examples in which PELT has been applied, in practical contexts as diverse as vegetation tracking, cancer risk and predator-prey population modelling. There are, therefore, many recent methods which successfully build on the PELT procedure for particular contexts. One example is the CAPA method of Fisch et al. (2019a), which can be used to detect anomalies. Another is the ED-PELT procedure of Haynes et al. (2017b). This uses a cost function approximating (2.1.14) within the PELT framework to give a computationally efficient means of locating changes in the nonparametric setting. The resulting procedure has an $\mathcal{O}(n \log n)$ complexity. Computationally, the approximation is necessary. For example, the NMCD algorithm of Zou et al. (2014) uses (2.1.14) within Segment Neighbourhood. As a result, the procedure is $\mathcal{O}(Qn^2 + n^3)$, where again Q is taken to be the maximum possible number of changes the method is allowed to insert into the sequence.

Another extension of the PELT procedure, which can additionally be applied to any method which exactly minimises (2.1.9), is the CROPS algorithm of Haynes et al. (2017a). Here, instead of specifying a penalty value β, a 'penalty interval', $[\beta_{min}, \beta_{max}]$, is an input. In other words, CROPS finds

$$\min_{\beta \in [\beta_{min}, \beta_{max}]} \left[\min_{\hat{m}, \hat{\tau}_1, \ldots, \hat{\tau}_{\hat{m}}} C\left(\hat{m}, \hat{\tau}_1, \ldots, \hat{\tau}_{\hat{m}} | \mathcal{C}(.), \beta\right) \right]$$

where $C(.|.)$ is as defined in (2.1.9). CROPS is therefore advantageous in situations where the 'optimal' penalty is unknown. Computationally, when used within the PELT procedure, CROPS has a worst-case computational cost of

$$\mathcal{O}\left\{n^2 \times \left(\hat{m}\left(\beta_{min}\right) - \hat{m}(\beta_{max}) + 2\right)\right\}.$$

Note here we adopt the notation that the number of changepoints estimated by the procedure using a penalty γ is $\hat{m}(\gamma)$. This worst-case cost is due to the fact that, as β is varied from β_{min} to β_{max}, $\hat{m}\left(\beta_{min}\right) - \hat{m}\left(\beta_{min}\right) + 2$ runs of PELT are required.

An extremely computationally efficient alternative to the PELT procedure, applicable in the single parameter change case, is the Functional Pruning Optimal

Partitioning (FPOP) method of Maidstone et al. (2017). For FPOP, a different form of pruning, known as functional pruning, is used in place of the inequality-based pruning favoured by PELT. The advantage of functional pruning is that the number of candidates considered as the most recent changepoint is drastically reduced, even compared to inequality-based pruning. This means that, when it can be applied, FPOP is much faster than PELT. This is especially true when there are particularly long segments in the data. The main disadvantage of FPOP is that it can be applied to fewer cost functions than PELT or Optimal Partitioning. Indeed, only a change in a single variable is permitted, with the method being most efficient when the cost function is piecewise quadratic. Despite this, there has been much recent interest in FPOP, with Hocking et al. (2018) introducing the Generalized FPOP (GFPOP) procedure, and Fearnhead and Rigaill (2019) formulating a robust version, R-FPOP. Importantly, these and other extensions to FPOP can be used to fit models with dependencies in the parameters across segments. This represents a significant advantage over PELT or Optimal Partitioning, where such costs cannot be minimised.

As an aside, we note that the Segment Neighbourhood search procedure can also be improved using functional pruning. This idea is the basis of the pruned Dynamic Programming Algorithm (pDPA) introduced by Rigaill (2010) and further discussed by Cleynen et al. (2014).

Other Model-based Approaches

We conclude this section on model-based approaches by briefly summarising some recent Bayesian changepoint detection techniques. Many Bayesian methods exist which include a conditioning on the most recent changepoint location. A number of these are based on a Hidden Markov Model approach, where the states are typically taken to be the regimes which the system is known to enter (for example, 'normal' or 'abnormal'). This has been a relatively popular idea in the literature for some time (Chib, 1998; Ko et al., 2015; Luong et al., 2013), particularly for applications in finance and genomics.

Another Bayesian method of note is the perfect simulation procedure of Fearnhead

(2006), which uses a product-partition model for the prior on the change locations and then recursively updates the posterior in a similar fashion to the updating of $F(.)$ within PELT. Like PELT, this method has an expected linear run time. We discuss online extensions to this approach, as well as other Bayesian methods, in Section 2.3.

2.1.3 Other Recent Approaches

There exist a number of other approaches to offline univariate changepoint detection. One example of a method which performs simultaneous estimation of the changepoint locations is the SMUCE procedure of Frick et al. (2014). This searches over the entire space of possible discrete step functions in data generated according to exponential-family processes. A similar method, H-SMUCE, for heterogeneous Gaussian regression models, was introduced by Pein et al. (2017). SMUCE and H-SMUCE are advantageous procedures in that, in addition to being computationally efficient, there is a natural means of obtaining a confidence set on the locations of the changepoints.

Another method of note is Hierarchical Clustering (HC), as discussed by Sullivan (2002). HC can essentially be thought of as 'backwards Binary Segmentation', as we begin by considering the entire sequence as n separate clusters consisting of the singleton points in the sequence. Neighbouring points are then merged if there is sufficient similarity between them. One issue with HC, potentially preventing the method from being more widely used, is the general lack of capability in 'un-merging' points. In addition, specifying a suitable stopping condition is challenging. Despite this, there has been some uptake of the method (Fryzlewicz, 2018; Harnish et al., 2009; Wang et al., 2005; Xing et al., 2007). In addition, some recent entries to the literature have used hierarchical clustering-type approaches to relax assumptions surrounding the generating processes. For example, the method of Khaleghi and Ryabko (2012) provides an interesting hybrid of Binary Segmentation and Hierarchical Clustering to relax many typical assumptions, such as within-segment independence.

2.2 Multivariate Changepoint Detection

Compared to the univariate problem, interest in detecting changepoints in the multivariate setting is much more recent. Letting the number of variates be d, the analogue to the univariate problem of (2.1.1) is now, for $i = 1, \ldots, d$

$$y_{i,j} \sim G_{i,k} \text{ for } \tau_{k-1} + 1 \leq j \leq \tau_k \text{ for } k \in \{1, \ldots, m+1\}. \tag{2.2.1}$$

Again we assume that $G_{i,1}, \ldots, G_{i,m+1}$ are a sequence of data generating processes. We additionally stipulate that $G_{i,k} \nsim G_{i,k+1}$ iff $i \in \mathcal{S}_k$, where $\mathcal{S}_k \subseteq \{1, \ldots, d\}$ is a non-empty *affected set* of variates. That is, $\mathcal{S}_k$ is the non-empty set of variates which undergo a change at τ_k.

Importantly, an implication of (2.2.1) is that we do not know which variates are changed at each changepoint. Therefore, the information about the change can be very different depending on whether only a few or many of the variates change. To see this, consider detecting a single changepoint which changes the mean of $|\mathcal{S}|$ of the variates. Two simple approaches to detecting a change in mean in multivariate data consist of either (i) assuming all variates change; or (ii) looking at each variate separately and considering whether there is a change in any of them.

To simplify the following exposition, assume we wish to test for a change at a single time point, t. Note that the intuition from the following argument applies for the more usual case of needing to test for changes at all locations - for example, see Enikeeva and Harchaoui (2019). If we use a likelihood ratio test, then, in the setting where we assume all variates change, we would have the test statistic

$$T(t; y_{1:d,1:n}) = \sum_{i=1}^{d} \left[\frac{1}{t} \left(\sum_{j=1}^{t} y_{i,j} \right)^2 + \frac{1}{n-t} \left(\sum_{j=t+1}^{n} y_{i,j} \right)^2 - \frac{1}{n} \left(\sum_{j=1}^{n} y_{i,j} \right)^2 \right],$$

in the case of Gaussian noise with known variance 1. Note that this test statistic has a χ^2_d distribution for a given t under the null. As we increase d, the quantiles of χ^2_d increase like

$$d + r\sqrt{d},$$

where r is dependent on the quantile in question. Under the alternative where, for instance, variate i changes by Δ_i at t, the test statistic is non-central chi-squared,

$\chi^2_d(\nu)$, where $\nu = \frac{t(n-t)}{n}\sum_{i\in\mathcal{S}}(\Delta_i)^2$ is the non-centrality parameter. Given that the mean of $\chi^2_d(\nu)$ is $\nu + d$, to have high power in detecting a change we therefore require $\nu + d$ to be much larger than the test threshold. Asymptotically, as d increases for fixed n and t, this occurs if

$$\sum_{i\in\mathcal{S}}(\Delta_i)^2 >> d^{1/2+\gamma},$$

for some pre-specified $\gamma > 0$. By contrast, we would not have high power for much smaller values of $\sum_{i\in\mathcal{S}}(\Delta_i)^2$.

To simplify further, imagine all $|\mathcal{S}|$ series affected by the change are altered by the same amount. Then if $|\mathcal{S}|$ is $\mathcal{O}(d^{\gamma_1})$ and the size of the change is $\mathcal{O}(d^{\gamma_2})$, then as $d \to \infty$ we can show we have power to detect the change that tends to 1 if $\gamma_1 + 2\gamma_2 > 0$. In particular, we can detect a change of fixed size providing the number of variates which change dominates $\sqrt{d}$ in order. Additionally, we can detect changes of decreasing size if $|\mathcal{S}|$ increases at a polynomial rate faster than $\sqrt{d}$. By comparison, if $\gamma_1 < 0$, we require $\gamma_2 > 0$.

In the setting where we examine at each variate separately, a natural choice for $T(t; y_{1:d,1:n})$ is the maximum of the likelihood ratio test statistics for each individual series

$$T(t; y_{1:d,1:n}) = \max_{i\in\{1,\ldots,d\}}\left[\frac{1}{t}\left(\sum_{j=1}^{t} y_{i,j}\right)^2 + \frac{1}{n-t}\left(\sum_{j=t+1}^{n} y_{i,j}\right)^2 - \frac{1}{n}\left(\sum_{j=1}^{n} y_{i,j}\right)^2\right].$$

Note that this is the maximum of d independent χ^2_1 distributions under the null. Therefore, an appropriately scaled version of the test statistic converges to a Gumbel distribution as $d \to \infty$. The threshold we use would need to increase like

$$2\log d - \log\log d,$$

as d increases. See, for example, Gasull et al. (2015) for further explanation. Therefore, if there is a change, and Δ is the largest change in mean for any series, the distribution of our test statistic is bounded below by $\chi^2_1(\nu)$, with $\nu = \frac{t(n-t)}{n}\Delta^2$. By Birgé (2001), the test statistic is therefore bounded below by $\nu + 1 - 2k\sqrt{1+2\nu}$ with high probability. Hence, ignoring the $t(n-t)/n$ factor, if

$$\Delta^2 >> 2\log d,$$

we would have high power to detect the change. Hence, if only a few variates within the dataset change, this is a much weaker condition than that obtained for (i). However, if many variates change, this is a much stronger condition. Therefore, in terms of the asymptotic behaviour as d increases, the question as to which of the tests is best depends on whether $|\mathcal{S}|$ increases faster or slower than $\sqrt{d}$.

We typically refer to those cases where (i) gives greater power, for example in situations where many variates change, as the *dense* setting. In contrast, those cases where (ii) gives greater power are referred to as the *sparse* setting. It is important for methods which are powerful in the sparse setting to give an exact idea of which series are changed. In this way, resources are not needlessly wasted or a problem mis-diagnosed (see, for example, the telecoms example discussed in Chapter 1). We refer the reader to Chapter 4 for a further discussion on sparse and dense changepoints in the multivariate setting.

Another issue in the multivariate setting is the significantly increased computational intensity of the problem. Some methods such as the E-Divisive procedure of Matteson and James (2014), or the Hierarchical Clustering approach of Székely and Rizzo (2005), scale in an undesirable way in either n or d. Therefore, most methods for multivariate changepoint detection are based on the AMOC approaches with Binary Segmentation discussed in Section 2.1.1.

2.2.1 AMOC Approaches and Extensions to Multiple Changes

There have been several attempts to extend the use of the CUSUM statistic to the multivariate setting, three of which are introduced in Groen et al. (2013) and Cho and Fryzlewicz (2015). We introduce these in more detail in Chapter 4, where we refer to the three statistics as Max, Mean and Bin-Weight. These multivariate statistics for a test of a single changepoint at t are, respectively, $T_{Max}(t; y_{1:d,1:n}) = \max_i W_{i,t}$, $T_{Mean}(t; y_{1:d,1:n}) = \sum_{i=1}^{d} W_{i,t}/d$ and $T_{Bin-Weight}(t; y_{1:d,1:n}) = \sum_{i=1}^{d} W_{i,t} \mathbb{1}\{W_{i,t} > \alpha\}$, for some α. Note that here $W_{i,t}$ refers to the standard univariate CUSUM statistic at

time t from (2.1.3), applied to the sequence $y_{i,1:n}$.

Another method which is equivalent to the weighted sum of CUSUMs is the Inspect method of Wang and Samworth (2018). Inspect seeks to compute the best projection direction of the multivariate series to maximise the signal-to-noise ratio in the univariate projected series. The optimal projection direction is the (normalised) difference of the multivariate means either side of the change. Using this would lead to no loss of information. The problem is then a question of how to estimate the projection direction, so that a suitable univariate detection method can then be used. The authors suggest solving a convex relaxation of the problem of finding the k-sparse leading left singular vector of the CUSUM transformation of the data stream. Note that in practice that this cannot be found directly as the problem is NP-hard.

For each of Max, Mean, Bin-Weight and Inspect, a form of Binary Segmentation is used to find multiple changes. For example, Bin-Weight was introduced with the Binary Segmentation alternative introduced by Cho and Fryzlewicz (2012), while Inspect uses Wild Binary Segmentation. Therefore, all of the methods are computationally efficient in both n and d. We discuss the empirical properties of the four methods in more detail in Chapter 4 and Appendix B. However, we remark here that the simultaneous attainment of competitive statistical power in both the sparse and dense settings remains a challenge.

Other changepoint tests based around a multivariate CUSUM are relatively common (Barigozzi et al., 2018; Cho, 2016; Dette and Gösmann, 2018; Enikeeva and Harchaoui, 2019; Tartakovsky et al., 2014; Wang and Reynolds, 2013; Zamba and Hawkins, 2006, 2009). However, again, the problems of sparse or dense power, sometimes coupled with computational complexity, remain an issue. In addition, we emphasise that the CUSUM is most effective in tracking changes in mean in the Gaussian setting. Hence, for general problems CUSUM-based techniques can be much less effective, as again we explore in Chapter 4.

There exist some nonparametric alternatives to the CUSUM suitable for the multivariate setting. In addition to the aforementioned E-Divisive approach of Matteson and James (2014), there is also the MultiRank procedure of Cabrieto et al.

(2017). This is primarily designed to detect changes in correlation structure, and uses the test statistic

$$T(t, y_{1:d,1:n}) = \frac{4}{n^2}\left[t\bar{\boldsymbol{r}}_{\mathbf{1}}^T\hat{\Sigma}^{-1}\bar{\boldsymbol{r}}_{\mathbf{1}} + (n-t)\bar{\boldsymbol{r}}_{\mathbf{2}}^T\hat{\Sigma}^{-1}\bar{\boldsymbol{r}}_{\mathbf{2}}\right].$$

Here, $\hat{\Sigma}$ is the empirical covariance matrix of the rank orders of the scores, and $\bar{\boldsymbol{r}}_{\mathbf{1}}$ and $\bar{\boldsymbol{r}}_{\mathbf{2}}$ are "phase specific vectors" consisting of deviations from the expected mean phase rank under the null. We note that, unusually for an AMOC approach, MultiRank does not use a Binary Segmentation method to search for multiple changes after the first changepoint. Instead, it then estimates the change locations simultaneously using a Segment Neighborhood type approach with a constraint on the maximum number of estimated changes. Thus, in addition to issues of performance under more challenging models, there remains the question of computational complexity.

We conclude this section by remarking that projection-based methods of the type considered by Wang and Samworth (2018) are becoming an increasingly popular multivariate changepoint approach. Auret and Aldrich (2010), Moskvina and Zhigljavsky (2003), Idé and Tsuda (2007) and Aston and Kirch (2012a) are among many to recently introduce projection-based procedures. Again, though, the recovery of information on the nature of which variates alter is typically more challenging. We remark that our new multivariate approaches introduced in Chapters 4 and 5 (and indeed many of the other methods discussed in this section) can in the strictest sense also be described as projection methods, given that both rely on aggregations from the univariate sequences within the dataset.

2.2.2 Model-based Approaches with Recursive Updates

In the multivariate setting, the central issue with cost function approaches of the type seen in Section 2.1.2 is computational cost. For example, as a close analogue to PELT, Pickering (2016) formulated a multivariate cost function for the multidimensional setting. This cost function allows for any number of the variates to alter at each changepoint. In addition, an exact means of resolving this cost function using pruned dynamic programming, Subset Multivariate Optimal Partitioning (SMOP),

was introduced. However, SMOP has a computational performance of $\mathcal{O}\left(d \times n^{2d}\right)$ in the worst case. This makes the method impractical except in very small data examples. Therefore, it is the usual approach to relax the need for exactness when resolving the cost function; see, for example, recent works such as Bardwell et al. (2019), Fisch et al. (2019b) and Lavielle and Teyssiere (2006), where the latter examines an adaptive penalisation procedure suitable for dependent processes. In Chapter 4, we introduce a similar means of finding an approximate resolution for a penalised cost function, albeit using an AMOC approach. The resulting method is exact under a single sparse change and approximate otherwise.

Other model-based methods applicable in the multivariate setting include the approach of Bulteel et al. (2014). This uses a moving sum type method to track changes in correlation using a specified information window. Another existing procedure is the Kernel Change Point (KCP) method of Arlot et al. (2012), which uses a kernel-based approach of the type discussed in Section 2.1.2. Garreau and Arlot (2018) show that KCP is consistent in identifying the correct number of changepoints. Moreover, KCP detects the changepoint locations at optimal rate, enabling the method to find many possible types of change. Note, however, that for fixed d the computation of the cost function matrix using KCP is $\mathcal{O}\left(n^4\right)$. In contrast, the novel kernel-based method of Celisse et al. (2018) is $\mathcal{O}\left(dn^2\right)$ in the worst case.

The final method we mention in this section is the nonparametric rank approach of Lung-Yut-Fong et al. (2012). This also uses a form of dynamic programming, although like SMOP the method loses computational efficiency if more than one change is to be located.

2.2.3 Other Recent Approaches

As for Section 2.1.3, we focus on approaches which can estimate the locations of all changepoints simultaneously. One such method is the Stochastic Approximate Monte Carlo (SAMC) algorithm of Liang (2007) and Liang (2009) which, as discussed by Cheon and Kim (2010) may be applied to multivariate multiple changepoint problems. Indeed, as they show for problems such as the estimation of the number

of changepoints, SAMC outperforms Reversible Jump Markov Chain Monte Carlo (RJMCMC) approaches. However, RJMCMC remains a popular means of detecting multiple changes in both the univariate and multivariate settings (Bolton and Heard, 2018; Ruggeri and Sivaganesan, 2005; Steward et al., 2016; Suparman et al., 2002; Xuan and Murphy, 2007).

Another recent Bayesian changepoint detection method in the multivariate setting is the procedure of Peluso et al. (2019). This method has the additional flexibility of relaxing parametric assumptions on the generating processes within segments by using a Dirichlet process mixture prior. Using such a prior is a general technique which has been growing in popularity (see Maheu and Yang (2016), Dufays (2016) and many others).

2.3 Online Changepoint Detection

In a data streaming setting, perhaps the most important problem is to identify salient features in as timely a fashion as possible, and certainly at a rate faster than the arrival of new data. In this way, pertinent decisions, which can subsequently influence the evolution of the stream (for example, to bring it back under control), can be made as the data are still being observed. Changepoints are, in this sense, an extremely important feature. Therefore, the online change detection problem is of great contemporary significance. This is especially true in scenarios where it is impractical for humans to monitor all elements within a data stream by eye.

The online challenge is also somewhat distinct from the offline setting in that the balance between the two central performance measures - i.e. a low false alarm rate and high true detection rate - is often bespoke to the situation. Hence, many entries in the literature attempt to address this trade-off directly. In this way, the problem of tuning multiple different parameters can largely be avoided.

We split the remainder of this section into separate discussions on univariate and multivariate online change detection.

2.3.1 Univariate Online Changepoint Detection

As traditional AMOC approaches with Binary Segmentation require us to observe the entire sequence in advance, they are typically impractical in an online setting. This is because we require the update on the arrival of a new piece of data to be $\mathcal{O}(1)$ in computational complexity. Therefore, the majority of online, or sequential, changepoint methods have been Bayesian. Early such examples include Smith and Cook (1980), Gamerman (1991) and Sarkar and Meeker (1998).

One popular approach is to use a latent state process for the position of the most recent changepoint, before updating recursively through the sequence using a generating function. For example, in Adams and MacKay (2007), this approach is referred to as updating the changepoint prior, while Fearnhead and Liu (2007) describe the iterative process as a filtering recursion. A typical choice for the generating function is to assume that the gap between successive changes can be modelled using a geometric distribution. This means that the system is effectively memoryless between changepoints. In this setting, greater emphasis is placed on the evolution of the process from one point to the next. Therefore, in practice, these methods are most effective either when (i) the presence of a changepoint gives a great deal of information on the location of the next changepoint; or (ii) when the noise in the system can be modelled reasonably accurately. Under such circumstances, the true detection rate can be very high, although the probability of a false alarm in the null setting as $n \to \infty$ still approaches 1.

We discuss a multivariate extension to this style of approach to the changepoint problem in Section 2.3.2. For a further discussion on online Bayesian (and non-Bayesian) methods, see Cook (2017), Caron et al. (2012) and Chowdhury et al. (2012) among others.

In contrast to the Bayesian literature, there is comparatively little on, for instance, cost function style approaches to the online changepoint detection problem. However, we note that some CUSUM-type approaches exist for a limited class of problems; see, for example, Cheifetz et al. (2012), Cheng et al. (2017), Höhle (2010) and Tsechpenakis et al. (2006). The traditional means of utilising the CUSUM in an

online environment is to use a finite data horizon of recently observed data. This is commonly referred to as a memory window for the recent past, beyond which the process 'forgets' the contents of the sequence. This enables the computational memory and time requirements at each stage to remain bounded.

We introduce a similar means of detecting changepoints (in the multivariate setting) in Chapter 5. There, we also make use of the penalised cost framework to raise an 'initial alarm' for a potential change within the stream.

2.3.2 Multivariate Online Changepoint Detection

Changepoint detection in the multivariate online setting is extremely challenging. This is particularly true if the dimension of the problem and intensity of data arrival can both be made very large. As such, the literature in this setting is sparse.

One notable recent method is the Bayesian Abnormal Region Detector (BARD) method of Bardwell and Fearnhead (2017). This is based on the same principles of Bayesian recursions as the procedures discussed in Section 2.3.1. With BARD, it is assumed that variates within the stream all begin in a "Normal" state. Subsequently, variates can only transition to an "Abnormal" state before transitioning back. While BARD is flexible in modelling which variates undergo a given change, the fact that each variate must return to a normal state does restrict the class of problems for which the method is useful. Additionally, some assumptions extraneous to within-segment independence are required. In Chapter 5, we attempt to relax these with our new multivariate, online method.

Another very recent method is that of Chen (2019b), which is based on a k-Nearest Neighbours approach to detecting changepoints first posited in an offline setting in Schilling (1986) and Henze (1988). Therefore, like the new method we introduce in Chapter 5, this approach is also nonparametric, and can additionally be applied in situations with a non-Euclidean structure, such as networks. We compare this method with our new approach in more detail in Chapter 5.

2.4 General Discussion

In this chapter, we have given an overview of many of the more popular changepoint detection methods. Additionally, we have discussed two important current spaces in the changepoint problem. In Section 2.2, we summarised the problem of detecting changepoints across many dimensions. In Section 2.3, we gave an overview of online approaches to detecting changes. While these are two important issues and, along with efficient inference, form the basis of the discussions in the chapters to follow, they are by no means the exclusive extant problems in the field.

A major issue of importance is that of the degree of confidence in a given changepoint estimate. While the Bayesian methods aforementioned in this chapter provide a natural means of doing this, specific focus on the problem in a non-Bayesian setting is a developing area of research. For example, Howard et al. (2019) examine the interesting problem of constructing non-asymptotic confidence sequences for "A/B Tests". This is a problem to which certain problems of the changepoint setting can be recast. For the remaining chapters, we consider accuracy only in terms of asymptotic or finite-sample consistency (which we discussed in more detail in Sections 2.1.2 and 2.1.1). This is an approach in accordance with the introduction of the vast majority of changepoint methods.

Another natural problem of importance, particularly in the streaming setting, is that of prediction. This is of interest, both in terms of behaviour following a changepoint (Galceran et al., 2017; Garnett et al., 2009; Steyvens and Brown, 2005) and the location of the next changepoint itself (Botezatu et al., 2016; Chen and Tsui, 2013; Garre et al., 2008). Once again, outside of a Bayesian framework, this is typically a very hard problem. In particular, very specific assumptions are usually required. Therefore, existing methods are typically very bespoke. While we do not confront the problem of changepoint prediction again directly, we discuss our new methods in this context for further development in Chapter 6.

2.4.1 Changepoint Detection in Context

We conclude this chapter with a brief comparison between the changepoint detection problems we have focused on and other varying-coefficient problems.

Changepoints in Regression Models

As alluded to in the discussion of Section 2.1.2, there is a natural link between the Gaussian change in mean problem and sparse regression. Several recent contributions have examined this connection further by explicitly considering the problem of detecting sparse changepoints in regression models. Notable among these is the work of Zhang et al. (2015), where the authors exploit the 'double sparsity' of the problem (i.e. sparsity of the number of changepoints relative to the number of samples, and sparsity of the changes in terms of the number of regression coefficients altered), to use a Sparse Group Lasso (Simon et al., 2013) approach based on a weighted sum of L_1 and L_2 penalisations. They note that the resulting algorithm is $\mathcal{O}(n^2 \log n)$ under standard assumptions, and establish consistency results on the changepoint estimators obtained.

Another recent paper (Leonardi and Bühlmann, 2016) relaxes the need for Gaussian residuals, and presents two approaches for changepoint detection in a high-dimensional regression context. These two approaches are very similar in character to the AMOC and model-based approaches we have discussed throughout this chapter, and theoretical guarantees on the locations of the changes, which they note are analogous to those obtained for the Lasso. In practice, however, a degree of tuning is needed, as the method requires an appropriate setting of two parameters. One of these regularises the high-dimensionality and sparsity, while the other regularises the number of segments. While a theoretical basis for setting the former is given, choosing the latter typically requires some knowledge of the minimum segment length.

Changepoints in Networks

An area which has experienced greatly increasing interest is that of the detection of changepoints within a network setting. Recent work includes Barnett and Onnela (2016), Masuda and Holme (2019), Yudovina et al. (2015) and Zhao et al. (2019), with applications as varied as social proximity in academic environments and polarisation of politics in the United States Congress. Modelling assumptions taken towards the detection of changepoints in a network context are varied, with stochastic block models (Ridder et al., 2016; Ludkin et al., 2018; Wills and Meyer, 2020) and Gaussian graphical models (Gibberd and Nelson, 2014; Gibberd and Roy, 2017; Kolar and Xing, 2012) perhaps two of the most commonly used frameworks in this space, with many examples in the latter setting using a group-fused lasso penalisation of the type discussed above.

Regardless of the modelling choices made, additional questions arise in the network setting extraneous to the considerations in the time series settings we have hitherto focused on this chapter. As a network naturally imbues a structure between the vertices, it is of interest to find changes in the structure of the network, particularly insofar as this relates to *communities* of nodes. (See, for example, Peel and Clauset (2015) for some of the possible types of changepoints involving community structure.) Several authors have utilised the additional structure of a network setting to search for changepoints on more challenging time series. For example, Xuan and Murphy (2007) use the Gaussian graphical model to search for changes in dependency across variates. Another recent paper to have explored the link between networks and changepoints under locally dependent data is that of Chen (2019a). Here, a test statistic based on a standardised edge-count of the similarity graph of the observations in the series is constructed. While this results in a test which can handle, for example, certain levels of autocorrelation, the construction of the test statistic can be computationally cumbersome.

Chapter 3

Parallelisation of a Common Changepoint Detection Method

3.1 Introduction

The challenge of changepoint detection has received considerable interest in recent years; see, for example, Rigaill et al. (2012), Chen and Nkurunziza (2017), Truong et al. (2018) and references therein. There are many algorithms for estimating the number and location of changepoints, for example Binary Segmentation, due to Scott and Knott (1974), and its variants such as Circular Binary Segmentation, Wild Binary Segmentation and Narrowest-Over-Threshold, due to Olshen et al. (2004), Fryzlewicz (2014) and Baranowski et al. (2018) respectively; and dynamic programming approaches that minimise a penalised cost, such as the Optimal Partitioning procedure of Jackson et al. (2005) or the Pruned Exact Linear Time (PELT) method of Killick et al. (2012).

In many applications, there are computational constraints that can affect the choice of method. We are interested in whether parallel computing techniques can be used to speed up algorithms such as Optimal Partitioning or PELT. The application of parallelisation is vast, with use in such areas as meta-heuristics, cloud computing and biomolecular simulation, as discussed in Alba (2005), Mezmaz et al. (2011), Schmid et al. (2012) and Wang and Dunson (2014) among many others. Some methods

are more easily parallelisable in that it is plain how to split a search space or other task between different nodes. These problems are often described as 'Embarrassingly Parallel'. For the changepoint detection problem, Binary Segmentation and Wild Binary Segmentation may be described as such. However, it is not so straightforward to parallelise dynamic programming methods.

This chapter makes a new contribution to this area by suggesting two new approaches for parallelising a penalised cost approach. In particular, we demonstrate in Section 3.3 that the computational cost of dynamic programming algorithms that minimise the penalised cost, such as PELT, can be reduced by a factor that can be quadratic in the number of computer cores. Further, we demonstrate empirically that super-linear gains in speed are achievable even in reasonably small sample settings in Section 3.4. One disadvantage with parallelising an algorithm such as PELT is that we are no longer guaranteed to find the segmentation which minimises the penalised cost. However, these approximations do not affect the asymptotic properties of the estimator of the number and locations of the changepoints: in Section 3.3 we show that, for the change in mean problem, our proposed approaches retain the same asymptotic properties as PELT.

The changepoint problem considers the analysis of a data sequence, $y_1, \ldots, y_n$, which is ordered by some index, such as time or position along a chromosome. We use the notation $y_{s:t} = (y_s, \ldots, y_t)$ for $t \geq s$. Our interest is in segmenting the data into consecutive regions. Such a segmentation can be defined by the changepoints, $0 = \tau_0 < \tau_1 < \ldots < \tau_m < \tau_{m+1} = n$, where the set of changepoints splits the data into $m + 1$ segments, with the j^{th} segment containing data-points $y_{\tau_{j-1}+1:\tau_j}$.

As mentioned, we focus on a class of methods which involve finding the set of changepoints that minimise a given cost. The cost associated with a specific segmentation consists of two important specifications. The first of these is $\mathcal{C}(.)$, the cost incurred from a segment of the data. Common choices for $\mathcal{C}(.)$ include quadratic error loss, Huber loss and the negative log-likelihood (for an appropriate within-segment model for the data); see Yao and Au (1989), Fearnhead and Rigaill (2017) and Chen and Gupta (2000) for further discussion. For example, using

quadratic error loss gives

$$\mathcal{C}(y_{s:t}) = \sum_{i=s}^{t} \left(y_i - \frac{1}{t-s+1} \sum_{j=s}^{t} y_j \right)^2 . \tag{3.1.1}$$

This cost is proportional to the negative log-likelihood for a piecewise constant signal observed with additive Gaussian noise. The second specification is β, the penalty incurred when introducing a changepoint into the model. Common choices for β include the Akaike Information Criterion, Schwarz Information Criterion and modified Bayesian Information Criterion; see Rigaill et al. (2013), Haynes et al. (2017a) and Truong et al. (2017) and references therein for further discussion. Finally, it is assumed that the cost function is additive over segments. The objective is then to find the segmentation which minimises the cost. In other words, we wish to find

$$\arg \min_{m;0=\tau_0<\ldots<\tau_m<\tau_{m+1}=n} \sum_{i=1}^{m+1} \left[\mathcal{C}(y_{\tau_{i-1}+1:\tau_i}) + \beta \right] . \tag{3.1.2}$$

Sometimes this minimisation is performed subject to a constraint on the minimum possible segment length. Optimal Partitioning, due to Jackson et al. (2005), uses dynamic programming to solve (3.1.2) exactly in a computation time of $\mathcal{O}(n^2)$. Killick et al. (2012) introduced the PELT algorithm, which also solves (3.1.2) exactly and can have a substantially reduced computational cost. In situations where the number of changepoints increases linearly with n, Killick et al. (2012) show that PELT's expected computational cost can be linear in n. However, the worst-case cost is still $\mathcal{O}(n^2)$.

The basis of these dynamic programming algorithms is a simple recursion for the minimum cost of segmenting the first t data points, $y_{1:t}$, which we denote as $F(t)$. It is straightforward to show that

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

The intuition is that we minimise over all possible values for the most recent changepoint prior to u, with the bracketed term being the minimum cost for segmenting $y_{1:u}$ with the most recent changepoint at t. By setting $F(0) = -\beta$ and solving this recursion for $u = 1, \ldots, n$, we obtain $F(n)$, the minimum value of (3.1.2).

At the same time, it is possible to obtain the set of changepoints which minimises the cost, see Jackson et al. (2005) for more details.

One of our approaches to parallelising algorithms such as PELT will use the fact that (3.1.2) can still be solved exactly when we restrict the changepoints to be from an ordered subset $\mathcal{B} = \{b_1, \ldots, b_k\} \subset \{1, \ldots, (n-1)\}$. Let $F_{\mathcal{B}}(b_s)$ denote the minimum cost of $y_{1:b_s}$ when we restrict potential changepoints to $\mathcal{B}$; this satisfies the recursion

$$F_{\mathcal{B}}(b_s) = \min_{t<s} \left\{ F_{\mathcal{B}}(b_t) + \mathcal{C}(y_{b_t+1:b_s}) + \beta \right\}.$$

Using the initial condition $F_{\mathcal{B}}(0) = -\beta$, this gives a means of recursively calculating $F_{\mathcal{B}}(b_k)$. For most cost functions, after a simple pre-processing step that is linear in n, the computational cost of solving these recursions will be, at worst, quadratic in the size of $\mathcal{B}$ rather than quadratic in n. This property is key to the near quadratic speed-ups we can obtain as we increase the number of cores. For both of the parallelisation methods we introduce, each core minimises the penalised cost whilst allowing changepoints at just a subset of locations. If we have L cores, then each core considers approximately n/L possible changepoint locations. Hence the worst-case cost of minimising the penalised cost on a given core is roughly a factor of L^2 less than that of running PELT on the full data. Furthermore, the parallelisation schemes we introduce involve no communication between cores other than a single post-processing step of the output from each core.

The general format of this chapter is as follows: Section 3.2 introduces two means of parallelising dynamic programming methods for solving (3.1.2), which we refer to as *Chunk* and *Deal*. In each case, we provide a description of the proposed algorithm with practical suggestions for implementation, followed by a short discussion of the theoretical justifications behind these choices. We devote Section 3.3 to examining this latter aspect in detail. In particular, we establish the asymptotic consistency of Chunk and Deal in a specific case with recourse to the asymptotic consistency of the penalised cost function method. Section 3.4 compares the use of parallelisation to other common approaches in a number of scenarios involving changes in mean. We conclude with a short discussion in Section 3.5. The proofs of all results may be found

in Section 3.6 and Appendix A.

3.2 Parallelisation of Dynamic Programming Methods

In this section, we introduce Chunk and Deal, two methods for parallelising dynamic programming procedures for changepoint detection. For convenience, we shall herein refer to this exclusively as the parallelisation of PELT.

We introduce the notation $PELT(y_{\mathcal{A}}, \mathcal{B})$ when referring to applying PELT to a dataset $y_{\mathcal{A}}$ but only allowing candidate changepoints to be fitted from within the set $\mathcal{B}$. Note that we trivially require $\mathcal{B} \subseteq \mathcal{A}$. Thus, for example, when performing PELT without any parallelisation, we may refer to this as $PELT\left(y_{\{1,\ldots,n\}}, \{1, \ldots, n-1\}\right)$.

In addition, we refer to the *parent core* as the core which is responsible for dividing the problem into sub-problems and then distributing these sub-problems to the other cores available. It then receives the output from each core (i.e. a set of estimated changepoints) and fits a changepoint model across the entire sequence using the results from these other cores.

Using this notation, the general setup for the parallelisation procedure then takes the following form.

- (Split Phase) We divide the space $\{1, \ldots, (n-1)\}$ into (not necessarily disjoint) subsets $\mathcal{B}_1, \ldots, \mathcal{B}_L$, where L is the number of computer cores available;
- Each of the cores $i = 1, \ldots, L$ then performs $PELT(y_{\mathcal{A}_i}, \mathcal{B}_i)$, returning a candidate set, $\hat{\boldsymbol{\tau}}_i$, of changes, which are returned to the parent core;
- (Merge Phase) The parent core then performs $PELT(y_{1:n}, \cup_{i=1}^{L} \hat{\boldsymbol{\tau}}_i)$, and the method returns $\hat{\boldsymbol{\tau}}$, the set of estimated changes found at this stage.

In short, $\mathcal{A}_i$ is the set of time points over which the i^{th} core is to fit a changepoint model, while $\mathcal{B}_i$ is the set of candidate changepoints passed to the i^{th} core. Note that

in the above we require $\cup_{i=1}^{L} \mathcal{A}_i = \{1, \ldots, n\}$. Our two methods for parallelisation differ in how they choose $\mathcal{A}_{1:L}$ and $\mathcal{B}_{1:L}$.

3.2.1 Chunk

The Chunk procedure consists of dividing the data into continuous segments and then handing each core a separate segment on which to search for changes. This splitting mechanism is shown in Figure 3.1. One problem with this division arises from changes which can be arbitrarily close to, or coincide with, the 'boundary points' of adjacent cores. This necessitates the use of an overlap - a set of points which are considered by both adjacent cores for potential changes, also shown in Figure 3.1. For a time series of length n, we choose an overlap of size $V(n)$ either side of the boundary for each core (with the exception of the first and final cores, which can each trivially only overlap in one direction). The full procedure for Chunk is detailed in Algorithm 1.

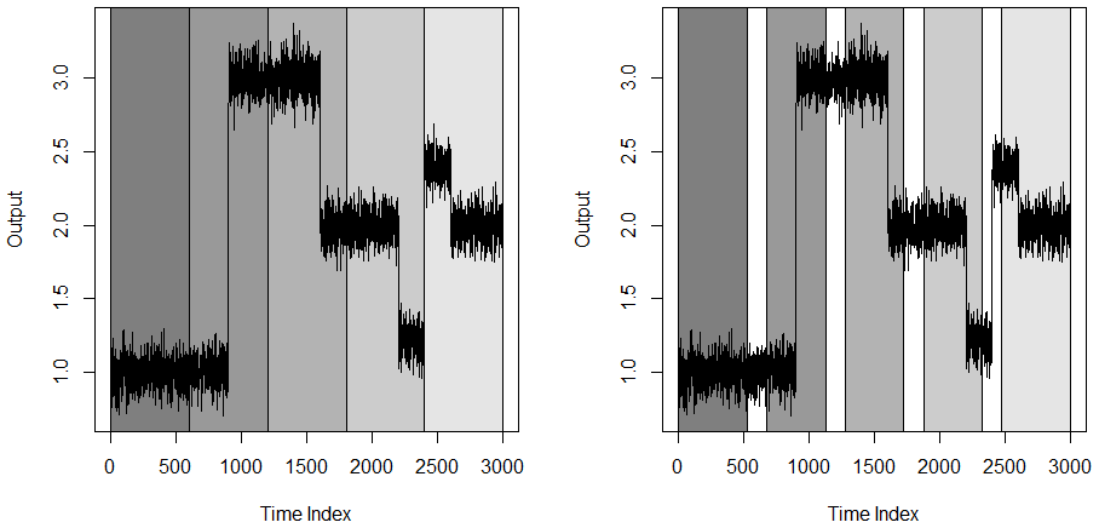

Figure 3.1: The time series is split into continuous segments by the Chunk procedure, in this case with 5 cores (l). An overlap is specified between the segments such that points within are considered by both adjacent cores (r).

Algorithm 1 Chunk for the PELT procedure

Data: A univariate dataset, $y_{1:n}$.

Result: A set of estimated **changepoint locations** $\hat{\tau}_1, \ldots, \hat{\tau}_{\hat{m}}$.

Step 1: Split the dataset into the subsets $\mathcal{B}_1, \ldots, \mathcal{B}_L$ such that

$\mathcal{B}_1 = \left\{1, \ldots, \left\lfloor \frac{n}{L} \right\rfloor + V(n)\right\}$,

$\mathcal{B}_i = \left\{(i-1)\left\lfloor \frac{n}{L} \right\rfloor - V(n), \ldots, i\left\lfloor \frac{n}{L} \right\rfloor + V(n)\right\}$ $\forall i \in \{2, \ldots, L-1\}$,

$\mathcal{B}_L = \left\{(L-1)\left\lfloor \frac{n}{L} \right\rfloor - V(n), \ldots, n\right\}$;

for $i = 1, \ldots, L$ **do**

| On core i, find $\hat{\boldsymbol{\tau}}_i = PELT\left(y_{\mathcal{B}_i}, \mathcal{B}_i\right)$;

end

Step 2: Sort $\cup_{i=1}^{L} \hat{\boldsymbol{\tau}}_i$ into ascending order;

Step 3: Calculate and return $(\hat{\tau}_1, \ldots, \hat{\tau}_{\hat{m}}) = PELT\left(y_{1:n}, \cup_{i=1}^{L} \hat{\boldsymbol{\tau}}_i\right)$.

Given that Algorithm 1 executes PELT multiple times, it is not immediate that Chunk represents a computational gain. We therefore briefly examine the speed of the procedure. Recall that PELT has a worst-case computational cost that is quadratic in the number of possible changepoint locations. Such a quadratic cost is observed empirically when the number of changepoints is fixed as n increases. Taking this worst-case computational cost, the cost of the split stage is $\mathcal{O}\left(\left(\frac{n}{L}\right)^2\right)$. The cost of the merge phase is dependent on the total number of estimated changes generated in the split phase. If we can estimate changepoint locations to sufficient accuracy, then as each change appears in at most two of the 'chunks', the number of returned changes ought to be at most $2m$. Thus the merge phase has a cost that is $\mathcal{O}(m^2)$. This intuition is confirmed later, in Corollary 3.3.3.

These calculations suggest that by increasing L we can decrease the computational cost by a factor of close to L^2. This is observed empirically for large n and few changepoints. In situations where there are many changepoints, the computational cost for PELT can be much faster than its worst-case cost, and the computational gains will be less.

To guarantee that the method does not overestimate the number of changes, some knowledge of the location error inherent in the PELT procedure is needed. This motivates the results of Section 3.3, which in turn imply various practical choices for the length of the overlap region, $V(n)$. In particular, using $V(n) = \lceil (\log n)^2 \rceil$ will give an effective guarantee of the accuracy of the method. Other sensible choices for $V(n)$ can be made based on the trade-off between accuracy and speed (see Section 3.3 for details).

3.2.2 Deal

The Deal procedure allows each core to segment the entire data sequence, but restricts them to considering a subset of possible changepoint locations. This is done analogously to dealing the possible changepoints locations to the cores, so that each core will receive every L^{th} possible location. A pictorial example of Deal is shown in Figure 3.2.

Formally, we define $Q_a(b, c)$ as the largest integer such that $Q_a(b, c) \times b + (a \bmod b) < c$. The split phase then partitions $\{1, \ldots, (n-1)\}$ as follows

$$\begin{aligned}
\mathcal{B}_1 &= \{1, L+1, 2L+1, \ldots, Q_1(L,n)L+1\}, \\
\mathcal{B}_2 &= \{2, L+2, 2L+2, \ldots, Q_2(L,n)L+2\}, \\
&\ldots \\
\mathcal{B}_L &= \{L, 2L, 3L, \ldots, Q_L(L,n)L\}.
\end{aligned}$$

This splitting mechanism is shown in Figure 3.2. On the k^{th} core, the objective function to be minimised then becomes

$$\min_{m,\tau_1,\ldots,\tau_m \in \mathcal{B}_k} \sum_{i=1}^{m+1} \left\{ \mathcal{C}(y_{(\tau_{i-1}+1):\tau_i}) + \beta \right\},$$

as discussed in Section 3.1. When the estimated changepoints from each core have been found and returned, the parent core then fits a changepoint model for the entire data sequence, using only points returned from the cores as changepoint candidates.

The full procedure for Deal is detailed in Algorithm 2.

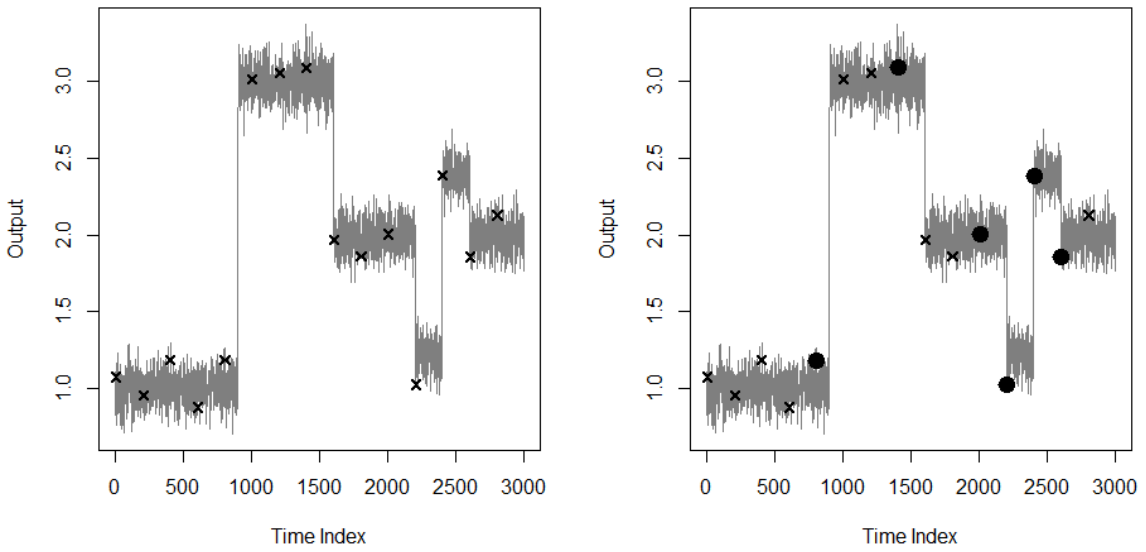

Figure 3.2: The time series is distributed across a number of cores by the Deal procedure. A particular core is given a certain collection of equally spaced points; for example, the points denoted by crosses (l). This core will then fit a changepoint model using only these points as candidate changes. The points estimated as changes are returned to the parent core. These points are circled (r).

Algorithm 2 Deal for the PELT procedure

Data: A univariate dataset, $y_{1:n}$.

Result: A set of estimated **changepoint locations** $\hat{\tau}_1, \ldots, \hat{\tau}_{\hat{m}}$.

Step 1: Split the dataset into subsets $\mathcal{B}_1, \ldots, \mathcal{B}_L$ such that $\mathcal{B}_i = \{i, L+i, \ldots, Q_i(L,n)L + i\}$;

for $i = 1, \ldots, L$ **do**

| On core i, find $\hat{\boldsymbol{\tau}}_i = PELT\left(y_{1:n}, \mathcal{B}_i\right)$;

end

Step 2: Sort $\cup_{i=1}^{L} \hat{\boldsymbol{\tau}}_i$ into ascending order;

Step 3: Calculate and return $(\hat{\tau}_1, \ldots, \hat{\tau}_{\hat{m}}) = PELT\left(y_{1:n}, \cup_{i=1}^{L} \hat{\boldsymbol{\tau}}_i\right)$.

As for the Chunk procedure, the implementation of Deal leads to computational gains. In a similar way to the previous section, the worst-case computational time

of the split phase of Deal will be $\mathcal{O}\left(\left(\frac{n}{L}\right)^2\right)$. The speed of the merge phase is again dependent on the number of changes detected at the split phase. We demonstrate in the proof of Corollary 3.3.3 that, with probability tending to 1, the number of changes detected by each core is at most $2m$, meaning that the worst-case performance of the merge phase is $\mathcal{O}_p\left(L^2\right)$.

We remark that while the Chunk and Deal procedures do not inherit the exactness of PELT in finding the optimal solution to (3.1.2), they nevertheless track the true optimum very closely. This can be seen in the empirical results of Section 3.4.

3.3 Consistency of Parallelised Approaches

Our two methods, Chunk and Deal, are no longer guaranteed to minimise (3.1.2). Thus, we turn to the question as to whether, regardless, the estimates of the number and location of the changepoints they give still retain desirable asymptotic properties. We investigate this question for the canonical change in mean model with infill asymptotics.

This corresponds to our time series, $y_1, ..., y_n$, having changepoints corresponding to proportions $\theta_1, ..., \theta_m$, for some fixed m, such that, for a given n, the changepoints $\tau_1, ..., \tau_m$ are defined as $\tau_i = \lfloor \theta_i n \rfloor \ \forall i$. For the asymptotic setting we consider, we take $\theta_{1:m}$ to be fixed.

With this framework in place, we note that the consistency results for Chunk and Deal we develop in Section 3.3.1 require one particular result not provided by Killick et al. (2012). Namely, consistency of PELT for the change in mean setting.

Proposition 1: *We consider the change in mean setting for the univariate time series*

$$Y_i = \delta_i + \mu_k, \text{ for } \tau_{k-1} + 1 \leq i \leq \tau_k \text{ and } k \in \{1, ..., m+1\}, \tag{3.3.1}$$

where $\mu_k \neq \mu_{k+1}$, for $k \in \{1, ..., m\}$ and $(\delta_1, ..., \delta_n)$ are a set of centered, independent and identically distributed Gaussian random variables with known variance σ^2. Take a series with m changes and true changepoint locations $\tau_1, ..., \tau_m$ (where $0 < \tau_1 < ... < \tau_m < n$). Apply the PELT procedure, minimising squared error loss, with a penalty of

$\beta = (2+\epsilon)\,\sigma^2 \log n$, for any $\epsilon > 0$, to produce an estimated set of $\hat{m}$ change locations $0 < \hat{\tau}_1 < ... < \hat{\tau}_{\hat{m}} < n$. Then, for any $\zeta > 0$, $\mathbb{P}(\mathcal{E}_n^\zeta) \to 1$ as $n \to \infty$, where

$$\mathcal{E}_n^\zeta = \left\{ \hat{m} = m; \max_{i=1,\ldots,m} |\hat{\tau}_i - \tau_i| \leq \left\lceil (\log n)^{1+\zeta} \right\rceil \right\}.$$

Proof: See Section A.2.

This result indicates that the probability of PELT misspecifying the number of changes, or the location of the true changes by more than a log-power factor, tends to 0 asymptotically. Note that this is with the Schwarz Information Criterion penalty in this setting, namely $2\,(1+\epsilon)\,\sigma^2 \log n$. Whilst this proposition, and the related results given in the next section, assume that the data have Gaussian distributions with common variance, it is straightforward to extend the results to sub-Gaussian random variables, or to allow the variance to vary across the time-series provided the variance is upper-bounded. In the latter case we would need to replace σ^2 in the condition for the penalty with the maximum value the variance could take.

Proposition 1 also extends naturally to the same problem in the multivariate setting with d dimensions, with a penalty of $(d+1)\,(1+\epsilon)\,\sigma^2 \log n$ (see Section A.2 for details). For the univariate case, the proof of Proposition 1 follows a similar pattern to that of Yao (1988), though we relax Yao's condition that an upper bound on the estimated number of changes is specified *a priori*.

3.3.1 Consistency and Computational Cost of Chunk and Deal

We now extend the consistency result in the unparallelised setting to obtain equivalent results for Chunk and Deal. If we fix the number of cores, L, as we increase n, many of the asymptotic results would follow trivially from existing results. For example, if we consider the Chunk approach and fix L as n increases then consistency would follow directly by the consistency of the analysis of data from each of the cores. Thus, in the following, we allow the number of cores to potentially increase as n increases, and use $L\,(n)$ to denote the number of cores used for a given sample size n.

Theorem 3.3.1. *For the change in mean setting specified in (3.3.1), assume that for a data series of length n we have $L(n)$ cores across which to parallelise a changepoint detection procedure, and an overlap of $V(n)$ between adjacent cores. For any $\zeta > 0$, define $\mathcal{E}_n^\zeta$ as for the previous results. In addition to the assumptions of Proposition 1, assume that (i) $L(n) = o(n)$ with $L(n) \to \infty$, (ii) that there exists a $\gamma > 1$ such that $V(n)/(\log n)^\gamma \to \infty$ and (iii) $V(n) = o(n)$. Then estimates from the Chunk procedure applied to a minimisation of the least squared error under a penalty of $\beta = (2+\epsilon)\sigma^2 \log n$, satisfy $\mathbb{P}(\mathcal{E}_n^\zeta) \to 1$ as $n \to \infty$.*

Proof: See Section 3.6.

In our simulation study we set $V(n) = \lceil (\log n)^2 \rceil$, which satisfies the condition of the theorem.

Theorem 3.3.2. *If $L(n) \geq \left\lceil (\log n)^{1+\zeta} \right\rceil$, then the same result as for Theorem 3.3.1 holds with the Deal parallelisation procedure.*

Proof: See Section 3.6.

Note that the conditions on $L(n)$ are stronger for Deal than for Chunk, with a lower bound corresponding with the maximum location error inherent in the event $\mathcal{E}_n^\zeta$. We believe the constraint on $L(n)$ is an artefact of the proof technique. Intuitively we would expect the statistical accuracy of Deal to be larger for smaller $L(n)$ as, for example, $L(n) = 1$ corresponds to optimally minimising the cost. Practically, setting $L(n) = \lceil (\log n) \rceil$ is unlikely to be problematic for typical values of n, a notion which we confirm empirically in Section 3.4.

Finally, given these results, we are now in a position to give a formal statement on the worst-case computational cost for both Chunk and Deal, when the computational cost of setting up a parallel environment is assumed to be negligible.

Corollary 3.3.3. *Under the change in mean setting outlined in Proposition 1, with probability tending to 1 as $n \to \infty$, the worst-case computational cost for Chunk when parallelising the PELT procedure using $L(n)$ computer cores is $\mathcal{O}_p\left(\max\left(\left(\frac{n}{L(n)}\right)^2, m^2\right)\right)$, while for Deal the worst-case cost is*

$\mathcal{O}_p\left(\max\left(\left(\frac{n}{L(n)}\right)^2, (L(n))^2\right)\right)$, *compared to a worst-case cost of* $\mathcal{O}(n^2)$ *for unparallelised PELT.*

Proof: See Section 3.6.

In the best case, we achieve a computational gain which is quadratic in $L(n)$. These results also show there is a limit to the gains of parallelisation as we continue to increase the number of cores. This is particularly true for Deal, where larger values of $L(n)$ can lead to more candidate changepoints considered in merge phase. For large $L(n)$, the cost of the merge phase will then dominate the overall cost of the Deal procedure. Setting $L(n) \sim n^{\frac{1}{2}}$ in Corollary 3.3.3 guarantees a worst-case computational cost of $\mathcal{O}_p(n)$ for both Chunk and Deal, no matter the performance of PELT. We emphasise again that this result ignores the cost of setting up a parallel environment, which can lead to PELT performing better computationally for small n. Therefore, we now conduct a simulation study in order to understand the likely practical circumstances in which parallelisation is a more efficient option.

3.4 Simulations

We now turn to consider the performance of these parallelised approximate methods on simulated data.

While these suggested parallelisation techniques do speed up the implementation of the dynamic programming procedure underlying, say, PELT, the exactness of PELT in resolving (3.1.2) is no longer guaranteed. We therefore compare parallelised PELT with Wild Binary Segmentation (WBS), proposed by Fryzlewicz (2014), a non-exact changepoint method which has impressive computational speed. To implement WBS, we used the `wbs` R package of Baranowski and Fryzlewicz (2015).

Simulated time series with piecewise normal segments were generated. Five scenarios, with changes at particular proportions of the time series, were examined in detail in the study. For time series of length 100000, these scenarios are shown in Figure 3.3.

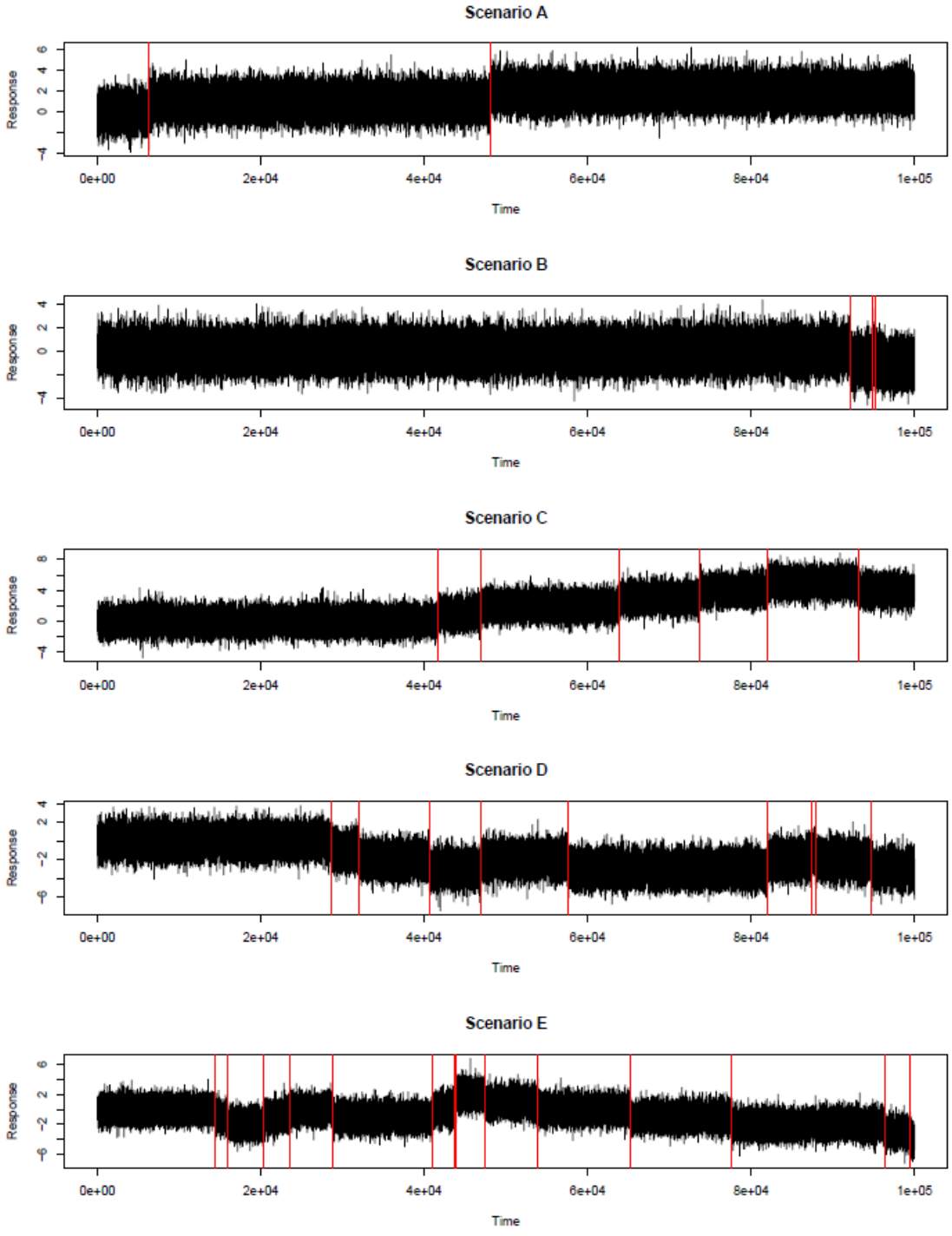

Figure 3.3: Five scenarios under examination in the simulation study. From top to bottom are scenarios A, B, C, D and E with 2, 3, 6, 9 and 14 true changes respectively.

Different lengths of series for each of the five scenarios - keeping the changepoints fixed at particular proportions in the time series as per the asymptotic regime outlined at the beginning of Section 3.3 - were used to examine the statistical power of PELT, Chunk, Deal and WBS under 200 replications for the error terms. In addition, four change magnitudes ($\Delta\mu = 0.25$, 0.5, 1 and 2) were used to examine the behaviour of the algorithms in each of the scenarios as $\Delta\mu$ was increased. When using PELT, Chunk and Deal, we assumed a minimum spacing between consecutive changes of at least two points.

The number of false positives (which were counted as the number of estimated changes more than $\lceil \log n \rceil$ points from the closest true change) and missed changes (the number of true changes with no estimated change within $\lceil \log n \rceil$ points), as well as the maximum observed location error and average location error across all repetitions were measured. Finally, the average cost of the segmentations (using mean squared error) generated by the methods, relative to the optimal given by PELT, were recorded.

As can be seen from Tables 3.1-3.3, Chunk and Deal closely mirror WBS and PELT in statistical performance in finding approximately the same number of changes in broadly similar locations. This was particularly evident in situations where the length of the series was 10^5. Here, the performance of Chunk and Deal becomes indistinguishable from PELT and WBS in most cases. However, as the number of changes and series length was increased, WBS was generally outperformed by both Chunk and Deal in terms of location accuracy. One additional aspect of note is that WBS was generally slightly more effective than the cost function based approaches at detecting the full set of changepoints in the scenarios with very short segments lengths (B, D and E) - see Table 3.2 for the full picture.

From Table 3.4, we note that, in practice, Deal often outperforms Chunk in terms of computational speed for a given number of cores. This is due to the fact that the Deal procedure will rarely perform at the worst-case computational speed during the split phase (which typically dominates the computation time), as one of the candidates around a true change is very likely to be chosen as a candidate changepoint (see

the proof of Theorem 3.3.2). This means that more candidates for the most recent changepoint are pruned than for Chunk. PELT was observed to be the fastest method for the smallest value of n across all scenarios. It was at the larger values of n where the super-linear gains in speed of Chunk and Deal became apparent, as can also be seen in Figure 3.4, which indicates that both Chunk and Deal exhibit a super-linear gain in speed in most situations. The exception to this is the use of the Chunk algorithm in Scenario E, which has a comparatively large number of true changepoints. As a result of this, the maximum segment length in the series in Scenario E remains similar in both the PELT and Chunk settings, even as the number of cores is increased. Hence, the computation gains here are less impressive.

An additional point of interest from Tables 3.4 and 3.5 is that PELT generally outperforms Chunk and Deal computationally when the time series is of length 10^3 or 10^4. This is due to the fact that the setting up of the parallel cluster takes around one second to complete, while the PELT algorithm takes significantly less time than this for shorter data sequences.

Finally, from Table 3.6, both Chunk and Deal are seen to track PELT very closely in terms of the final cost of the model. This appears to be particularly true for the datasets of greater length, where the average cost seen under both Chunk and Deal was seen to be the same as PELT (up to our stated precision) for almost all situations we investigated. In light of the behaviour seen from Tables 3.1-3.3, however, this should not be surprising.

Caution should be exercised when discussing these results in the context of the general statistical performance of Chunk and Deal, as only the value of $L = 4$ was tested.

All simulations were run in R using a Linux OS on a 2.3GHz Intel Xeon CPU. Simulations were run in batches of 20, grouped by length of series and detection method. When testing the PELT procedure, each job within a batch was assigned a separate core without any parallelisation or external packages involved. For Chunk and Deal, although jobs were again run in batches of 20, each was assigned the number of cores across which the algorithm was to be parallelised. (This was 4 in

all cases, except to run the simulations to generate Figure 3.4.) Parallelisation was implemented using the `doParallel` and `foreach` packages of Calaway et al. (2018) and Calaway and Weston (2017) respectively. Note that the `doParallel` package uses multiprocessing as opposed to multithreading.

In addition to the simulation study we have conducted here, please see the further study in Section A.3, in which we examine the performance of Chunk and Deal relative to PELT for a situation with an increasing number of changepoints.

3.5 Discussion

We have proposed two new methods for changepoint detection, Chunk and Deal, each based on parallelising an existing method, PELT. These methods represent a substantial computational gain in many cases, particularly for large n. In addition, by establishing the asymptotic consistency of PELT, we have been able in turn to show the asymptotic consistency of the Chunk and Deal methods, such that the error inherent to all three is $\mathcal{O}(\log n)$ in terms of the maximum location error of an estimated change relative to the corresponding true change. We have demonstrated empirically that an implication of this is that Chunk and Deal, while not inheriting the exactness of PELT, do perform well in finding changes in practice.

There are other approaches to reduce the computational cost of changepoint methods, whilst retaining the same asymptotic statistical properties. A suggestion, made by a reviewer, is that we could implement the Deal algorithm but with fewer candidates per core. Providing there is at least one core with a candidate close to the true change, say within $\log n$ of it, then under infill asymptotics of the kind discussed in Section 3.3 we will still detect the change with probability tending to 1 as n increases. Our empirical experience with such a method is that it can lose power at detecting changes in practical, non-asymptotic settings. Such a strategy has similarities to the ideas presented in Lu et al. (2018), and could be sensible in situations that they consider where n is exceedingly large, and it is computationally infeasible to analyse all the data.

Average False Alarms		Length = 10^3				Length = 10^4				Length = 10^5			
		$\Delta\mu$				$\Delta\mu$				$\Delta\mu$			
Scenario	Method	0.25	0.5	1	2	0.25	0.5	1	2	0.25	0.5	1	2
A	PELT	0.65	0.72	0.24	0.01	1.36	0.72	**0.15**	**0.00**	1.28	**0.59**	**0.10**	**0.00**
(2 changes)	Chunk4	0.67	0.87	**0.21**	**0.01**	1.49	0.72	0.16	**0.00**	1.29	**0.59**	**0.10**	**0.00**
	Deal4	0.64	0.69	0.22	0.01	1.35	0.72	**0.15**	**0.00**	1.28	**0.59**	**0.10**	**0.00**
	WBS	**0.54**	**0.66**	0.29	0.08	**1.20**	**0.66**	0.16	**0.00**	**1.26**	**0.59**	**0.10**	**0.00**
B	PELT	0.17	0.26	**0.15**	**0.01**	0.75	0.46	0.14	**0.00**	0.98	**0.83**	**0.09**	**0.00**
(3 changes)	Chunk4	**0.15**	**0.24**	0.16	**0.01**	0.70	0.46	0.14	**0.00**	0.98	**0.83**	**0.09**	**0.00**
	Deal4	0.16	0.27	**0.15**	**0.01**	0.75	0.46	0.14	**0.00**	0.98	**0.83**	**0.09**	**0.00**
	WBS	**0.15**	0.25	0.19	0.07	**0.55**	**0.45**	**0.12**	0.02	**0.97**	0.93	0.24	0.10
C	PELT	0.87	1.01	**0.68**	**0.12**	2.79	**2.08**	0.37	**0.00**	**3.94**	1.89	0.20	**0.00**
(6 changes)	Chunk4	0.89	**1.00**	0.73	0.15	**2.66**	2.11	0.37	**0.00**	3.96	**1.88**	**0.19**	**0.00**
	Deal4	**0.84**	1.02	0.69	**0.12**	2.81	**2.08**	**0.36**	**0.00**	**3.94**	1.89	0.20	**0.00**
	WBS	0.86	1.23	1.07	0.23	2.73	2.40	0.66	0.08	4.11	2.17	0.53	0.11
D	PELT	1.03	1.17	0.61	**0.09**	3.42	2.83	**0.60**	0.11	5.16	**2.73**	**0.43**	**0.00**
(9 changes)	Chunk4	1.02	1.16	0.63	0.12	**3.10**	**2.81**	**0.60**	**0.10**	**5.14**	**2.73**	**0.43**	**0.00**
	Deal4	1.01	**1.11**	**0.60**	**0.09**	3.41	2.83	0.61	0.11	5.16	**2.73**	**0.43**	**0.00**
	WBS	**0.97**	1.27	1.01	0.17	3.20	3.10	0.90	0.20	5.42	3.26	0.79	0.17
E	PELT	0.94	1.16	**0.64**	**0.07**	3.93	3.64	**0.86**	**0.07**	8.12	4.07	**0.59**	**0.05**
(14 changes)	Chunk4	0.99	1.27	0.91	0.30	**3.85**	3.64	0.90	0.10	8.16	**4.06**	**0.59**	**0.05**
	Deal4	**0.92**	**1.15**	0.65	0.09	3.91	**3.63**	**0.86**	**0.07**	8.11	4.07	**0.59**	**0.05**
	WBS	1.01	1.67	1.24	0.24	3.86	4.23	1.24	0.18	8.14	4.50	1.08	0.18

Table 3.1: The average number of false alarms recorded across all 200 repetitions for each of the 5 scenarios A, B, C, D and E. A false alarm is defined as an estimated changepoint which is at least $\lceil(\log n)\rceil$ points from the closest true changepoint. Bold entries show the best performing algorithm.

Average Num. Missed		Length = 10^3				Length = 10^4				Length = 10^5			
		$\Delta\mu$				$\Delta\mu$				$\Delta\mu$			
Scenario	Method	0.25	0.5	1	2	0.25	0.5	1	2	0.25	0.5	1	2
A	PELT	**1.78**	**1.14**	0.22	**0.01**	**1.38**	0.71	**0.14**	**0.00**	1.28	**0.59**	**0.10**	**0.00**
(2 changes)	Chunk4	1.95	1.39	**0.21**	**0.01**	1.56	0.72	0.15	**0.00**	1.29	**0.59**	**0.10**	**0.00**
	Deal4	**1.78**	1.15	0.22	**0.01**	**1.38**	0.71	**0.14**	**0.00**	1.28	**0.59**	**0.10**	**0.00**
	WBS	1.84	1.29	0.22	**0.01**	1.45	**0.66**	0.16	**0.00**	**1.26**	**0.59**	**0.10**	**0.00**
B	PELT	**2.63**	**2.06**	**1.19**	1.02	**2.47**	**1.94**	1.22	**0.00**	2.45	**0.86**	**0.09**	**0.00**
(3 changes)	Chunk4	2.65	2.15	1.22	1.03	2.48	1.95	1.22	**0.00**	2.45	**0.86**	**0.09**	**0.00**
	Deal4	**2.63**	2.08	**1.19**	1.03	**2.47**	1.95	1.25	**0.00**	2.44	**0.86**	**0.09**	**0.00**
	WBS	2.65	2.13	1.29	**0.91**	2.51	1.95	**1.06**	0.01	**2.43**	1.02	0.16	0.01
C	PELT	5.55	**4.87**	2.29	0.95	**4.85**	2.08	0.37	**0.00**	**3.94**	1.89	0.20	**0.00**
(6 changes)	Chunk4	5.69	4.99	2.56	1.00	5.01	2.11	0.37	**0.00**	3.96	**1.88**	**0.19**	**0.00**
	Deal4	**5.54**	**4.87**	2.38	0.98	4.88	**2.08**	**0.36**	**0.00**	**3.94**	1.89	0.20	**0.00**
	WBS	5.57	4.71	**1.22**	**0.08**	4.90	2.36	0.56	0.03	4.05	2.08	0.48	0.04
D	PELT	8.26	7.10	4.67	2.80	**7.51**	**4.39**	1.78	0.74	**6.43**	2.76	**0.44**	**0.00**
(9 changes)	Chunk4	8.40	7.19	4.78	2.98	7.67	4.43	1.79	0.73	**6.43**	**2.75**	**0.44**	**0.00**
	Deal4	8.26	7.07	4.68	2.87	7.53	4.40	1.81	0.74	6.45	2.76	**0.44**	**0.00**
	WBS	**8.22**	**6.66**	**2.65**	**0.66**	7.79	4.57	**1.07**	**0.07**	6.48	3.21	0.67	0.02
E	PELT	**13.0**	11.8	9.43	7.62	**12.3**	7.75	3.54	2.29	9.90	**4.75**	**0.82**	0.20
(14 changes)	Chunk4	13.2	12.1	9.91	8.04	12.4	7.89	3.63	2.40	9.95	**4.75**	**0.82**	0.20
	Deal4	**13.0**	11.9	9.53	7.71	**12.3**	7.78	3.54	2.29	9.89	4.76	**0.82**	0.20
	WBS	13.1	**11.2**	**6.09**	**1.53**	**12.3**	**7.46**	**2.51**	**0.16**	10.2	5.00	0.97	**0.04**

Table 3.2: The average number of missed changes across all 200 repetitions for each of the 5 scenarios A, B, C, D and E. A missed change is defined as a true changepoint for which no estimated change lies within $\lceil(\log n)\rceil$ points. Bold entries show the best performing algorithm.

Average Location Error		Length = 10^3 $\Delta\mu$				Length = 10^4 $\Delta\mu$				Length = 10^5 $\Delta\mu$			
Scenario	Method	0.25	0.5	1	2	0.25	0.5	1	2	0.25	0.5	1	2
A	PELT	58.0	18.6	5.04	**1.23**	70.1	**11.5**	**3.25**	**1.19**	**46.0**	**11.7**	3.21	**1.26**
(2 changes)	Chunk4	**51.2**	18.8	**3.16**	1.24	90.3	12.1	3.35	1.18	47.4	**11.7**	3.21	**1.26**
	Deal4	61.0	**15.5**	3.21	**1.23**	57.3	**11.5**	**3.25**	**1.19**	**46.0**	**11.7**	3.21	**1.26**
	WBS	86.2	34.7	12.7	10.7	**52.4**	12.3	3.40	1.20	**46.0**	12.1	**3.18**	**1.26**
B	PELT	70.3	**31.5**	11.7	3.74	76.1	42.8	3.66	1.25	47.5	12.1	3.00	1.27
(3 changes)	Chunk4	77.5	37.2	12.5	**1.16**	72.3	41.6	3.59	**1.24**	**47.0**	12.1	3.00	1.27
	Deal4	70.3	32.9	**11.8**	3.77	74.6	41.6	3.65	**1.24**	47.1	**12.0**	**3.00**	**1.27**
	WBS	**59.9**	38.7	17.4	13.8	**32.2**	**11.0**	**3.25**	1.52	47.4	14.4	5.82	3.07
C	PELT	25.9	15.0	**4.38**	**1.53**	64.2	**11.9**	3.29	**1.23**	**50.3**	12.5	3.04	**1.23**
(6 changes)	Chunk4	26.3	**14.1**	4.53	1.77	**60.9**	12.7	3.29	**1.23**	50.7	**12.4**	**3.01**	1.24
	Deal4	25.5	14.8	**4.38**	1.54	64.3	12.0	**3.28**	**1.23**	**50.3**	12.5	3.04	**1.23**
	WBS	**21.8**	**14.1**	5.87	2.51	65.1	17.7	5.79	1.88	80.7	24.0	5.62	1.93
D	PELT	18.9	10.4	**3.57**	**1.43**	58.3	**13.2**	**3.52**	**1.47**	**86.0**	11.7	**3.32**	**1.25**
(9 changes)	Chunk4	19.6	10.9	3.71	1.54	63.6	13.8	3.68	**1.47**	86.8	**11.6**	**3.32**	**1.25**
	Deal4	18.8	**9.90**	**3.57**	1.44	**56.6**	**13.2**	3.53	**1.47**	86.2	11.7	**3.32**	**1.25**
	WBS	**17.6**	10.4	4.41	4.12	58.3	20.0	5.29	1.76	199	20.4	6.47	2.39
E	PELT	13.0	**8.68**	**3.78**	**1.44**	51.7	**13.3**	**3.60**	**1.39**	50.9	12.7	**3.48**	**1.44**
(14 changes)	Chunk4	15.0	9.88	4.91	2.09	65.8	15.0	4.14	1.73	52.0	**12.6**	**3.48**	**1.44**
	Deal4	**12.9**	9.01	**3.78**	**1.44**	**51.4**	**13.3**	3.64	**1.39**	50.8	12.8	**3.48**	**1.44**
	WBS	13.7	9.67	4.20	2.58	56.9	17.1	9.05	1.64	70.6	36.3	5.18	1.90

Table 3.3: The average location error between those true changes which were detected by the algorithms and the corresponding estimated change across all 200 repetitions for each of the 5 scenarios. Bold entries show the best performing algorithm.

Mean Time Taken (seconds)		Length = 10^3 $\Delta\mu$				Length = 10^4 $\Delta\mu$				Length = 10^5 $\Delta\mu$			
Scenario	Method	0.25	0.5	1	2	0.25	0.5	1	2	0.25	0.5	1	2
A	PELT	**0.06**	**0.06**	**0.05**	**0.05**	**1.61**	**1.44**	1.47	1.49	108	107	113	109
(2 changes)	Chunk4	1.48	1.49	1.37	1.13	1.90	1.89	1.83	**1.54**	23.9	24.0	21.1	24.1
	Deal4	1.59	1.23	1.59	1.49	1.72	1.70	**1.45**	1.69	**12.1**	**10.7**	**11.9**	**11.1**
B	PELT	**0.06**	**0.06**	**0.06**	**0.06**	2.23	2.27	2.35	2.57	147	144	154	165
(3 changes)	Chunk4	1.38	1.37	1.13	1.38	**1.78**	1.82	**1.55**	1.78	23.9	24.1	24.2	31.6
	Deal4	1.49	1.49	1.24	1.16	1.82	**1.45**	1.59	**1.59**	**16.2**	**16.5**	**16.5**	**16.4**
C	PELT	**0.06**	**0.05**	**0.04**	**0.03**	**1.23**	**0.94**	**0.93**	**0.88**	72.2	71.8	70.7	72.1
(6 changes)	Chunk4	1.48	1.13	1.38	1.48	1.84	1.50	1.73	1.85	22.3	20.0	23.2	29.7
	Deal4	1.58	1.58	1.49	1.15	1.46	1.42	1.28	1.37	**8.33**	**7.58**	**7.60**	**7.30**
D	PELT	**0.05**	**0.05**	**0.03**	**0.04**	**1.12**	**0.82**	**0.73**	**0.75**	60.6	55.5	56.9	55.4
(9 changes)	Chunk4	1.37	1.37	1.48	1.37	1.79	1.73	1.85	1.77	22.5	22.5	19.8	29.8
	Deal4	1.49	1.23	1.58	1.58	1.65	1.36	1.40	1.26	**6.66**	**6.58**	**6.26**	**6.91**
E	PELT	**0.05**	**0.05**	**0.04**	**0.04**	**1.03**	**0.69**	**0.63**	**0.58**	60.9	40.0	37.2	37.7
(14 changes)	Chunk4	1.42	1.38	1.48	1.37	2.15	1.65	1.74	1.65	28.8	14.3	16.0	16.0
	Deal4	1.50	1.58	1.48	1.23	1.55	1.38	1.56	1.33	**8.92**	**5.23**	**4.95**	**5.44**

Table 3.4: The time taken across 200 repetitions for each of the scenarios in question for PELT, Chunk and Deal (using 4 cores). Bold entries show the best performing algorithm.

Average Relative Gain In Computation Speed		Length = 10^3 $\Delta\mu$				Length = 10^4 $\Delta\mu$				Length = 10^5 $\Delta\mu$			
Scenario	Method	0.25	0.5	1	2	0.25	0.5	1	2	0.25	0.5	1	2
A	Chunk4	**0.04**	0.04	**0.03**	**0.04**	0.85	0.76	0.80	**0.97**	4.53	4.44	5.34	4.53
(2 changes)	Deal4	**0.04**	**0.05**	**0.03**	0.03	**0.94**	**0.85**	**1.01**	0.88	**8.94**	**9.97**	**9.46**	**9.83**
B	Chunk4	**0.04**	**0.05**	**0.06**	**0.05**	**1.25**	1.25	**1.52**	1.44	6.14	5.96	6.37	5.21
(3 changes)	Deal4	**0.04**	0.04	0.05	**0.05**	1.23	**1.57**	1.48	**1.62**	**9.05**	**8.71**	**9.34**	**10.0**
C	Chunk4	**0.04**	**0.04**	**0.03**	0.02	0.67	0.63	0.54	0.47	3.24	3.59	3.05	2.43
(6 changes)	Deal4	**0.04**	0.03	**0.03**	**0.03**	**0.84**	**0.66**	**0.72**	**0.64**	**8.67**	**9.47**	**9.31**	**9.88**
D	Chunk4	**0.04**	0.03	**0.02**	**0.03**	0.63	0.48	0.39	0.42	2.69	2.47	2.87	1.86
(9 changes)	Deal4	0.03	**0.04**	**0.02**	0.02	**0.68**	**0.61**	**0.52**	**0.59**	**9.10**	**8.43**	**9.08**	**8.01**
E	Chunk4	**0.04**	**0.03**	**0.03**	**0.03**	0.48	0.42	0.36	0.35	2.11	2.79	2.32	2.36
(14 changes)	Deal4	0.03	**0.03**	**0.03**	**0.03**	**0.66**	**0.50**	**0.40**	**0.44**	**6.82**	**7.64**	**7.51**	**6.94**

Table 3.5: The average relative computation gain of the Chunk and Deal methods relative to the PELT method across 200 repetitions for each of the scenarios in question. These values are calculated by dividing corresponding values from Table 3.4. Bold entries show the best performing algorithm.

Average Cost - Optimal		Length = 10^3 $\Delta\mu$				Length = 10^4 $\Delta\mu$				Length = 10^5 $\Delta\mu$			
Scenario	Method	0.25	0.5	1	2	0.25	0.5	1	2	0.25	0.5	1	2
A	Chunk4	1.70	1.57	0.03	0.01	3.17	0.05	0.01	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**
(2 changes)	Deal4	**0.01**	**0.03**	**0.01**	**0.00**	**0.02**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**
B	Chunk4	0.12	0.51	0.13	0.09	0.19	0.01	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**
(3 changes)	Deal4	**0.01**	**0.05**	**0.02**	**0.04**	**0.01**	**0.01**	0.04	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**
C	Chunk4	1.65	1.85	2.44	6.52	3.44	0.39	0.02	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**
(6 changes)	Deal4	**0.03**	**0.04**	**0.07**	**0.05**	**0.01**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**
D	Chunk4	2.30	2.23	2.90	7.44	4.10	1.13	1.42	0.01	0.10	**0.00**	**0.00**	**0.00**
(9 changes)	Deal4	**0.05**	**0.06**	**0.13**	**0.17**	**0.00**	**0.01**	**0.01**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**
E	Chunk4	2.41	4.02	8.43	24.2	7.45	4.21	6.75	19.7	0.10	**0.00**	**0.00**	**0.00**
(14 changes)	Deal4	**0.05**	**0.11**	**0.19**	**0.29**	**0.02**	**0.02**	**0.03**	**0.10**	**0.00**	**0.00**	**0.00**	**0.00**

Table 3.6: The average error, across 200 repetitions, between the penalised residual sum of squares using Chunk and Deal with 4 cores and PELT (which is optimal). Bold entries show the best performing algorithm.

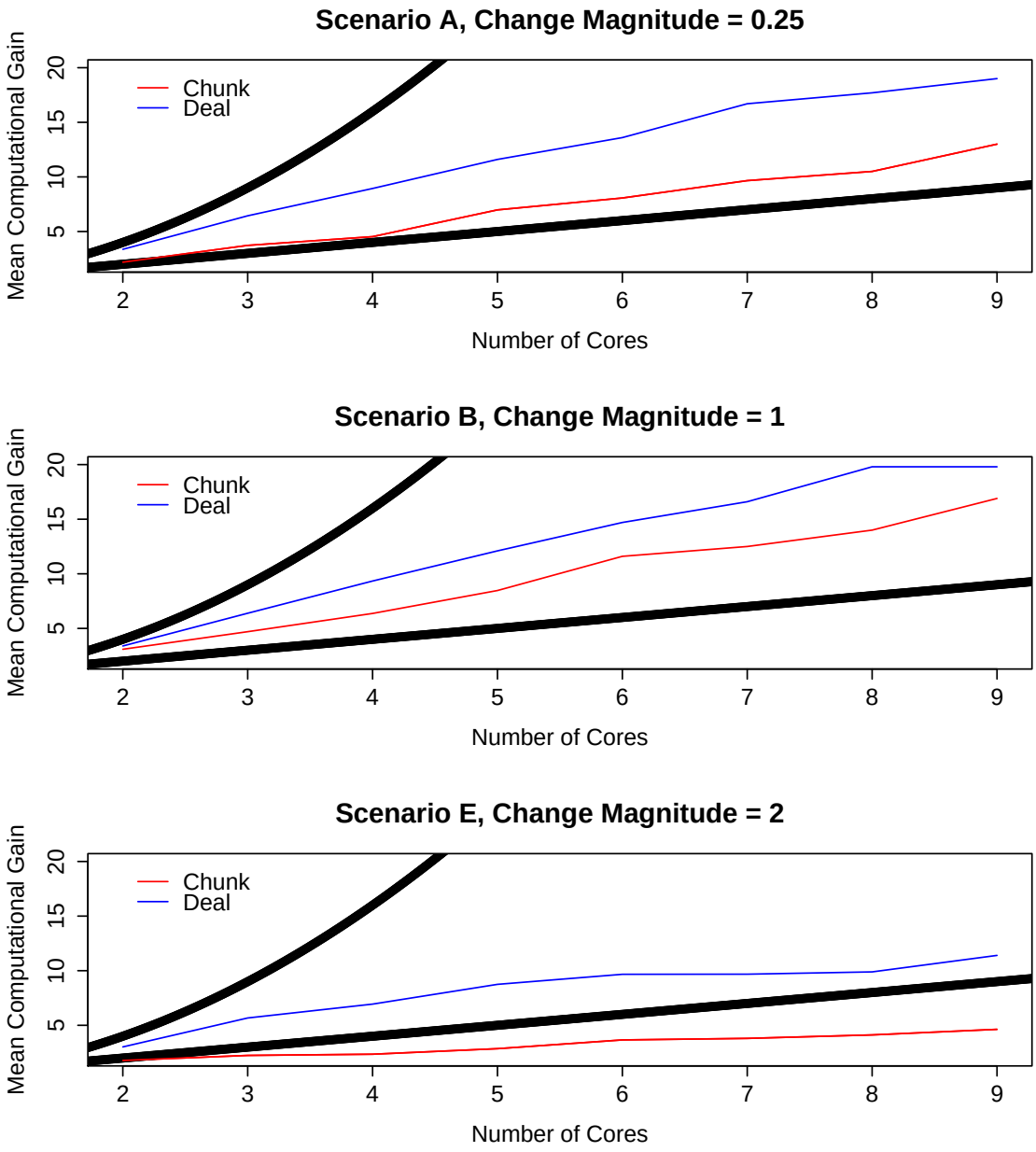

Figure 3.4: Mean computational gain (y) across 200 repetitions for Chunk and Deal compared to PELT across a differing number of cores (x) under three specific scenarios. The lines $y = x$ and $y = x^2$ are shown for comparison.

3.6 Proofs

The following results will be stated with respect to a general $\zeta > 0$. Theoretically, this means that any $\zeta > 0$ can be used in Algorithm 1 or Algorithm 2, however in the simulation study detailed in Section 3.4, $\lceil (\log n)^2 \rceil$ was used as the overlap length (for Chunk), while the cutoff value for closeness detailed in the merge phase (Step 3) of both procedures was taken as $\lceil (\log n) \rceil$.

Proof of Theorem 3.3.1: The Chunk procedure involves obtaining a set of candidate changepoints from analysing the data sent to each core, and then finding the best segmentation using these changepoints in the merge phase. We claim that to show Chunk is consistent, it is sufficient to show that, with probability tending to 1, there will be a segmentation using m of the candidate changepoints that gives an RSS that is within $o_p(\log n)$ of the RSS we obtain for the true segmentation.

This claim follows from a simple adaptation of the proof of Proposition 1. In that proof we show that, with probability tending to 1, for any penalty $(2+\epsilon)\sigma^2 \log n$ with $\epsilon > 0$, a segmentation with $\hat{m} > m$ changepoints will have a worse penalised cost than the true segmentation. Furthermore, any segmentation with $\hat{m} \leq m$ which is not in $\mathcal{E}_n^\zeta$ will miss one or more changepoints by more than $(\log n)^{1+\zeta}$ and will have a worse penalised cost than a segmentation with $\hat{m} > m$ changepoints (i.e. a segmentation obtained by adding three changepoints for each changepoint that is not estimated well enough). Thus, to show our claim, we need only show that, with probability tending to 1, we do not overestimate the number of changepoints.

Assume we use a penalty of $(2+\epsilon)\sigma^2 \log n$ for Chunk. From the argument in the proof of Proposition 1 applied to the penalised cost with a penalty $(2+2\epsilon)\sigma^2 \log n$, we have that with probability tending to 1, for all $\hat{\tau}_{1:\hat{m}}$ with $\hat{m} > m$,

$$\mathrm{RSS}(y_{1:n}; \hat{\tau}_{1:\hat{m}}) - \mathrm{RSS}(y_{1:n}; \tau_{1:m}) + (\hat{m} - m)(2+2\epsilon)\sigma^2 \log n > 0, \text{ so}$$

$$\mathrm{RSS}(y_{1:n}, \hat{\tau}_{1:\hat{m}}) - \{\mathrm{RSS}(y_{1:n}; \tau_{1:m}) + o_p(\log n)\} + (\hat{m} - m)(2+\epsilon)\sigma^2 \log n > \epsilon\sigma^2 \log n + o_p(\log n),$$

as required.

We now show that we will have a suitable set of candidate changepoints for the merge phase in two steps. The first of these steps establishes that each changepoint will be estimated within $\log \log n$.

By the set-up of Chunk, each changepoint will appear in the non-overlap region of data assigned to precisely one core. Furthermore, as $L(n) \to \infty$ and $V(n) = o(n)$, then for large enough n the core that a changepoint is assigned to will have data which contains only that changepoint.

Consider the data associated with each such core. Such a core will have data with just a single changepoint and a minimum segment length that is at least $V(n)$. As for sufficiently large n, $V(n) > \lceil \log n \rceil^{1+\gamma}$, for some $\gamma > 0$, then, by a simple adaptation of the argument in Section A.1, it is straightforward to show that, with probability tending to 1, we will detect precisely one changepoint for this data. Standard results (for example, see Lemma 3 of Yao and Au (1989)) for detecting a single changepoint from Gaussian data shows that the error in the location is $\mathcal{O}_p(1)$, and hence with probability tending to 1 we will detect the changepoint within an error of $\log \log n$.

As there are a finite number of changepoints, with probability tending to 1 we will detect precisely one changepoint with an error less than $\log \log n$ for all cores with a changepoint in the non-overlap region.

We now define, for a true segmentation of $\tau_{1:m}$ and sequence of length n, a *good set of segmentations*, $\mathcal{H}(\tau_{1:m}, n)$, such that

$$\mathcal{H}(\tau_{1:m}, n) = \{\hat{\tau}_{1:\hat{m}} | \hat{m} = m, |\hat{\tau}_i - \tau_i| \leq \log \log n \text{ for } i \in \{1, \ldots, m\}\}.$$

The second phase is to show that for any set of changepoints $\hat{\tau}_{1:\hat{m}} \in \mathcal{H}(\tau_{1:m}, n)$, the maximum difference between the RSS for fitting changepoints at $\hat{\tau}_{1:m}$ and the RSS for fitting changepoints at the true locations is $\mathcal{O}_p(\log \log n)$.

Define $\Delta\mu_k := |\mu_k - \mu_{k+1}|$. For any appropriate $\hat{\tau}_{1:m}$ we have

$$\begin{aligned}\text{RSS}(y_{1:n}; \hat{\tau}_{1:m}) - \text{RSS}(y_{1:n}; \tau_{1:m}) \leq & \sum_{i=1}^{m+1} \left\{ \frac{1}{\tau_i - \tau_{i-1}} \left(\sum_{j=\tau_{i-1}+1}^{\tau_i} Z_j \right)^2 - \frac{1}{\hat{\tau}_i - \hat{\tau}_{i-1}} \left(\sum_{j=\hat{\tau}_{i-1}+1}^{\hat{\tau}_i} Z_j \right)^2 \right\} \\ & + \sum_{k=1}^{m} (\Delta\mu_k)^2 \log \log n + G,\end{aligned}$$

where the fourth term, G, depends on $\hat{\tau}_{1:m}$ with $G \sim N\left(0, 4\sigma^2 \sum_{k=1}^{m} (\Delta\mu_k)^2 \log \log n\right)$. Note that the first term in this inequality does not depend on $\hat{\tau}_{1:m}$ and has a χ^2_{m+1} distribution, and so is $\mathcal{O}_p(\log \log n)$; the second term is negative and the third term is a constant multiple of $\log \log n$. So it only remains to check that $G = \mathcal{O}_p(\log \log n)$

uniformly across all members of $\mathcal{H}$. This follows trivially from standard bounds on a Gaussian distribution together with a Bonferroni correction over the $(2\log\log n)^m$ possibilities for $\hat{\tau}_{1:m}$. □

Proof of Theorem 3.3.2: Recall that $L(n) \geq \left\lceil (\log n)^{1+\zeta} \right\rceil$ and $L(n) = o(n)$. The idea will be to show that the core which is 'dealt' a particular true change, τ_i, will always return this true change as a candidate changepoint for the merge phase. By Yao (1988), letting $\hat{\tau}_{1:m}$ be a set of estimated changes which miss the true change τ_i by at least $\left\lceil (\log n)^{1+\zeta} \right\rceil$, then again by the proof of Corollary A.1.3 the cost of this segmentation is strictly worse than the cost of also fitting changes at the points $\tau_i - L(n)$ and $\tau_i + L(n)$. By then considering the difference

$$\text{Diff} := \text{RSS}(y_{1:n}; \hat{\tau}_{1:m}, \tau_i - L(n), \tau_i + L(n)) - \text{RSS}(y_{1:n}; \hat{\tau}_{1:m}, \tau_i - L(n), \tau_i, \tau_i + L(n)),$$

in a similar fashion to the proof of Corollary A.1.3, it can be shown that in probability

$$\frac{\text{Diff}}{L(n)} \to (\Delta\mu_{i-1})^2,$$

where again $\Delta\mu_{i-1}$ is the absolute change in mean at the changepoint τ_i. □

Proof of Corollary 3.3.3: It is sufficient to prove the following Claim regarding the number of candidate changes each core returns.

Claim: With probability tending to 1, and for any candidate set given to the cores in accordance with the conditions of Theorem 3.3.1 and Theorem 3.3.2

(I): under the Chunk procedure, the maximum number of points returned for the merge phase is bounded above by $2m$,

(II): under Deal, the maximum number of points recorded as estimated changes is bounded above by $2m$ for each core.

Proof of Claim:

Proof of (I): We note that when $L(n)$ is constant, the result is immediate from the proof of Proposition 1.

When $L(n) \to \infty$, it suffices to show that across all cores which are given no true changes, the probability of any of these cores returning a true change converges to 0.

Given that the number of cores which are given a change is fixed (and bounded above at $2m$ - as each change could fall inside an overlap), the result is then immediate from the proof of Theorem 3.3.1.

Considering a single core with no true changes, we adapt the argument from the proof Proposition 1. For a quantity U_{k+1} which is distributed according to a χ^2_{k+1} distribution, then by Laurent and Massart (2000)

$$\mathbb{P}(U_{k+1} \geq d \log n) \leq n^{-\frac{d}{2}+\delta}, \text{ for any } \delta > 0.$$

Fitting $k > 0$ changes across a core will give that the residual sum of squares, relative to a fit of no changes across the same core, follows a χ^2_{k+1} distribution. Therefore, following the application of a Bonferroni correction across all possible placings of k changes gives that the difference between the null fit and the best possible fit of k changes is then bounded in probability as

$$\mathbb{P}(\text{Diff}_k \geq d \log n) \leq n^{-\frac{d}{2}+\delta} \times \left(\frac{n}{L(n)}\right)^k.$$

In particular, setting $d = 2k\,(1+\epsilon)$ and $\delta = \epsilon/2$ as before, gives that

$$\begin{aligned}\sum_{k=1}^{n/L(n)} \mathbb{P}(\text{Diff}_k \geq 2k\,(1+\epsilon) \log n) &\leq \sum_{k=1}^{n/L(n)} \frac{n^{-\frac{(2k-1)\epsilon}{2}}}{(L(n))^k} \\ &= \frac{n^{-\frac{\epsilon}{2}}}{L(n)} \left(\frac{1 - n^{-\epsilon \frac{n}{L(n)}} L(n)^{-\frac{n}{L(n)}}}{1 - n^{-\epsilon} L(n)^{-1}}\right) \to 0, \ \forall \epsilon > 0,\end{aligned}$$

and so scaling this by $L(n)$

$$\mathbb{P}(\text{A core with no true changes overfits}) \to 0 \ \forall \epsilon > 0.$$

Therefore, the computation time of the merge phase of Chunk is $\mathcal{O}_p(m^2)$ in the worst case, which along with the worst-case cost from the split phase of $\mathcal{O}\left(\left(\frac{n}{L(n)}\right)^2\right)$ gives the worst-case computation time for the whole procedure.

Proof of (II): We introduce the set of points, a subset of the points given to a particular core under the Deal procedure, with exactly $2m$ elements. Each element in this set is the closest time point given to the core immediately before and after each true change. That is, we define, for a given core under the Deal procedure

$$\mathcal{U}_1 = \left\{s_1^{(1)}, s_1^{(2)}, s_2^{(1)}, s_2^{(2)}, \ldots, s_m^{(1)}, s_m^{(2)}\right\},$$

where $s_i^{(1)}$ is the final point given to the core which is strictly before τ_i, and $s_i^{(2)}$ is the first point given to the core which is after τ_i. In the same way as for the proof of Proposition 1, we examine the best possible segmentations which include $\mathcal{U}_1$ as a subset of the estimated changepoints for a core, and show that all are rejected in favour of $\mathcal{U}_1$ in probability. We then show that this is true across all cores in probability.

For a given core, suppose $\mathcal{U}_2$ is a set of points estimated as changes under the Deal procedure such that $\mathcal{U}_1 \subset \mathcal{U}_2$. By construction of $\mathcal{U}_1$, all points in $\mathcal{U}_2 \cap \mathcal{U}_1^c$ must lie in a region between two points of $\mathcal{U}_1$ which also does not contain any true changes. We can therefore apply the same argument as for Proposition 1 to the difference

$$\text{Diff} := \text{RSS}(y_{\mathcal{A}}; \mathcal{U}_1) - \text{RSS}(y_{\mathcal{A}}; \mathcal{U}_2),$$

where $\mathcal{A}$ refers to any such region between two consecutive points of $\mathcal{U}_1$ which contains a point found only in $\mathcal{U}_2$. Uniformly across such regions, and supposing $k > 0$ such estimated changes are found within $\mathcal{A}$, it can be seen that the positive term in the expression of the difference above is distributed as χ^2_{k+1}. Thus letting $\tilde{n} = \frac{n}{L(n)}$ and again with recourse to the Bonferroni correction argument as in Proposition 1, for a given $\epsilon > 0$

$$\begin{aligned}\sum_{k=1}^{\tilde{n}} \mathbb{P}(\text{Diff}_k \geq 2k\,(1+\epsilon)\log n) &\leq \sum_{k=1}^{\tilde{n}} \frac{n^{-\frac{(2k-1)\epsilon}{2}}}{(L(n))^k} \\ &= \frac{n^{-\frac{\epsilon}{2}}}{L(n)} \left(\frac{1 - n^{-\tilde{n}\epsilon} L(n)^{-\tilde{n}}}{1 - n^{-\epsilon} L(n)^{-1}} \right) \to 0, \ \forall \epsilon > 0.\end{aligned}$$

Note that this argument does not consider segmentations which do not contain $\mathcal{U}_1$ as a proper subset. In order to extend this argument, we define the following three sets of segmentations (with respect to a given core)

$$\begin{aligned}\mathcal{GU}_2 &= \{\hat{\boldsymbol{\tau}} : |\hat{\boldsymbol{\tau}}| = 2m; \hat{\tau}_{2t-1} \leq \tau_t, \hat{\tau}_{2t} > \tau_t, \forall t \in \{1, ..., m\}\}, \\ \mathcal{GU}_1 &= \{\hat{\boldsymbol{\tau}} : |\hat{\boldsymbol{\tau}}| \leq 2m; |\hat{\boldsymbol{\tau}} \cap \{\tau_t + 1, ..., \tau_{t+1}\}| \geq 1, \forall t \in \{0, ..., m\}; \\ &\qquad |\hat{\boldsymbol{\tau}} \cap \{\tau_t + 1, ..., \tau_{t+1}\}| = 1, \text{ some } t \notin \{0, m\}\} \\ \mathcal{GU}_0 &= \{\hat{\boldsymbol{\tau}} : |\hat{\boldsymbol{\tau}}| \leq 2m; |\hat{\boldsymbol{\tau}} \cap \{\tau_t + 1, ..., \tau_{t+1}\}| = 0, \text{ some } t\}.\end{aligned}$$

In short, $\mathcal{GU}_2$ is the set of segmentations containing exactly $2m$ points where between two consecutive true changes there are exactly two estimated changepoints.

Additionally, there is exactly one estimated change prior to the first true change and exactly one estimated changepoint following the final true change. Meanwhile, $\mathcal{GU}_1$ is the set of segmentations in which at least one estimated change is placed between two consecutive true changes in every case, and, for at least one case, exactly one estimate is placed between two consecutive true changes. Finally, $\mathcal{GU}_0$ is the set of segmentations with at most $2m$ estimated changes, where in at least one case no estimated changes are placed between two consecutive true changes.

Note that $\mathcal{U}_1 \in \mathcal{GU}_2$. In addition, the argument showing that any segmentation $\mathcal{U}_2$ containing $\mathcal{U}_1$ is rejected uniformly in favour of $\mathcal{U}_1$ may be extended to any element of $\mathcal{GU}_2$. This in turn shows that any segmentation with more than $2m$ estimated changes in total, and which has at least two estimated changes between each true change, is uniformly dominated by a corresponding element of $\mathcal{GU}_2$.

In the same way, let us now consider extensions from a general element, $\mathcal{T}_1 \in \mathcal{GU}_1$, where here an extension is defined as a superset of $\mathcal{T}_1$ which also contains additional estimated changes from regions between two estimated changes within $\mathcal{T}_1$ not containing a true change. Let, for example

$$\mathcal{T}_1 = \left\{ s_1^{(1)}, s_1^{(2)}, ..., s_{i-1}^{(2)}, s_i^{(k)}, s_{i+1}^{(1)}, ..., s_m^{(2)} \right\} \subset \mathcal{U}_1,$$

for some $k \in \{1, 2\}$ and $i \in \{1, ..., m\}$. Then, any extension of $\mathcal{T}_1$ consists of placing further estimated changes in any of the regions between the changes above with the exception of either (if $k = 1$) the region $\left(s_i^{(1)}, s_{i+1}^{(1)}\right)$ or (if $k = 2$) the region $\left(s_{i-1}^{(2)}, s_i^{(2)}\right)$. Let $\mathcal{T}_1'$ be an arbitrary such extension, and again let $\mathcal{A}$ be any region between two consecutive points of $\mathcal{T}_1$ which contains a point found only in $\mathcal{T}_1'$. As before, uniformly across such regions, and supposing again that $k > 0$ such estimated changes are found within $\mathcal{A}$, letting

$$\text{Diff} := \text{RSS}(y_{\mathcal{A}}; \mathcal{T}_1) - \text{RSS}(y_{\mathcal{A}}; \mathcal{T}_1'),$$

then again Diff is distributed as χ^2_{k+1}. With recourse to the same argument as before (noting again that any such region $\mathcal{A}$ will have at most $\tilde{n} = \frac{n}{L(n)}$ candidate points for the extension - no matter which base element of $\mathcal{GU}_1$ we pick), and extending to other elements of $\mathcal{GU}_1$, we conclude that any segmentation with more than $2m$ estimated

changes which places just one estimated change between two true changes in at least one case will be rejected uniformly (and for all cores) in favour of an element of $\mathcal{GU}_1$.

Finally, we consider all segmentations with more than $2m$ changes which place no estimated changes between two true changes in at least one case. We again compare with $\mathcal{T}_0 \in \mathcal{GU}_0$. Let, for example

$$\mathcal{T}_0 = \left\{ s_1^{(1)}, s_2^{(2)}, \ldots, s_{i-1}^{(2)}, s_{i+1}^{(1)}, \ldots, s_m^{(2)} \right\},$$

for some $i \in \{1, \ldots, m\}$. Then any extension of $\mathcal{T}_0$ consists of placing any further estimated changes in any of the regions between the changes above, with the exception of the region $\left(s_{i-1}^{(2)}, s_{i+1}^{(1)}\right)$. Let $\mathcal{T}_0'$ be an arbitrary such extension, and again let $\mathcal{A}$ be any region between two consecutive points of $\mathcal{T}_0$ which contains a point found only in $\mathcal{T}_0'$. Then again letting

$$\text{Diff} := \text{RSS}(y_{\mathcal{A}}; \mathcal{T}_0) - \text{RSS}(y_{\mathcal{A}}; \mathcal{T}_0'),$$

then for $k > 0$ changes in the region $\mathcal{A}$, Diff is distributed as χ^2_{k+1}. We can again extend this argument to extensions of other elements of $\mathcal{GU}_0$ to conclude that segmentations with more than $2m$ changes which have no estimated changepoints between two consecutive true changes in at least one case will be uniformly rejected in favour of an element of $\mathcal{GU}_0$.

Therefore, as any segmentation with more than $2m$ changes for any core is an extension of an element of $\mathcal{GU}_0$, $\mathcal{GU}_1$ or $\mathcal{GU}_2$ (as such a segmentation must contain a region between two consecutive true changes with at least three estimated changes), then across all cores, a segmentation must be picked from within one of the classes $\mathcal{GU}_0$, $\mathcal{GU}_1$ or $\mathcal{GU}_2$ in probability. Thus, the maximum number of estimated changepoints that a core can return in the Deal procedure is $2m$.

The number of candidates returned for the merge phase of the Deal procedure is therefore bounded in probability by $2mL(n)$, so that the maximum computation time of the merge phase is $\mathcal{O}_p\left((L(n))^2\right)$ in the worst case. This gives the stated total worst-case computation time for the whole procedure. □

Chapter 4

Computationally Efficient Multivariate Changepoint Detection

4.1 Introduction

Changepoint detection concerns inferring those points in a data sequence where some aspect of the data generating mechanism alters abruptly. Classical examples of aspects which may undergo a change include the mean (Hinkley, 1971; James et al., 1987; Kokoszka and Leipus, 1998, among others), variance (Hsu, 1977; Inclán and Tiao, 1994; Chen and Gupta, 1997, among others), slope (Miao, 1989; Julious, 2001; Aue et al., 2006, among others), event rate (Raftery and Akman, 1986; Yao, 1986; Henderson, 1990, among others) or distribution (Lombard, 1987; Carlstein, 1988; Barry and Hartigan, 1992, among others).

Changepoint detection continues to be an area of intense activity and practical concern, particularly due to the large amount of data that is routinely collected and interest in segmenting such data into regions with homogeneous behaviour. Application areas are wide-ranging, from climate change (Manogaran and Lopez, 2018) to brain imaging (Jewell et al., 2019) and Bitcoin volatility (Thies and Molnár, 2018). Recent contributions include Anastasiou and Fryzlewicz (2019), Eichinger and

Kirch (2018), Plasse and Adams (2019) and Roy et al. (2017).

Whilst detecting changes in univariate data sequences has a long history, there has been much less work on methods for detecting potentially multiple changepoints in multivariate datasets. Univariate approaches can be readily adapted to the multivariate setting if we are willing to assume all variates change at each changepoint; see, for example, Wessman (1998), Wolfe and Chen (1990) and Zhou et al. (2010). However, this may not be appropriate in applications where some, but not all, variates are affected by each changepoint, or where it is not known *a priori* whether a change will only affect a very small number or many of the variates.

Within the multivariate changepoint setting, the change in mean problem has to date received the most substantial focus; see, for example, Sen and Srivastava (1973), Bardet and Dion (2019) and many others. In this setting, evidence for a change in a single series can, for example, be quantified using CUSUM statistics – a weighted difference in the empirical mean before and after the potential changepoint. The simplest ways of combining evidence across time series are to (i) perform some form of averaging of the CUSUM statistics; or (ii) take the maximum value of the CUSUM statistics. Enikeeva and Harchaoui (2019) study the properties of these two approaches under an asymptotic regime where both the number of variates and the number of observations per variate increase. The detection boundaries, that is, how large a change in mean is needed in order that the presence or absence of a change can be determined with probability tending to 1, for approaches based on (i) and (ii) are very different. In particular, which of (i) and (ii) is better depends on the proportion of variates that undergo a sizeable change. If we let d be the number of variates, a change is said to be sparse if it affects $o(d^{1/2})$ of the variates, and dense otherwise. Then methods based on averaging CUSUM statistics are able to detect smaller changes in the dense setting. By contrast, using the maximum can detect smaller changes in the sparse setting. Enikeeva and Harchaoui (2019) propose combining these two approaches in order to have a high detection chance across all types of change.

Alternative ways of aggregating evidence for a change across variates try to

strike a balance between approaches (i) and (ii). For example, Cho and Fryzlewicz (2015) and Cho (2016) sum only CUSUM statistics that exceed a certain threshold. Conversely, Wang and Samworth (2018) consider sparse projections of the data. This is equivalent to using a weighted average of CUSUM statistics. These approaches can demonstrate strong empirical performance, but neither has been shown theoretically to simultaneously work as well as (i) in the dense setting *and* (ii) in the sparse setting. For example, the method of Wang and Samworth (2018) was designed for detecting sparse changes, and its theory establishes strong performance in only that setting.

In this chapter, we propose an alternative approach for detecting changes in multivariate datasets, based on likelihood ratio test statistics. One challenge with performing a likelihood ratio test is that we do not know how many, and which, of the variates change at any potential changepoint. Consequently, as Chapter 4 of Pickering (2016) identified, we introduce a penalised version of the likelihood ratio test statistic. Here, the penalty depends on how many variates are assumed to change. We then maximise these penalised statistics over all possible subsets of variates and changepoint locations. The method we propose has good computational properties, with an approximately linear computation time in the number of temporal points when the number of dimensions is fixed, and vice versa. Note that our approach is distinct from Chapter 4 of Pickering (2016), as their procedure exactly minimises a multivariate cost function. As they discuss, this leads to a heavy computational burden.

Since our approach is penalised, we show how to choose the penalties so that, for the change in mean problem, it has good asymptotic properties simultaneously for both sparse and dense changes. The method can be applied to detect a range of different types of change, providing we use an appropriate likelihood model for the data within a segment on which we base our likelihood ratio test statistic. The theoretical properties for the change in mean case solely use the chi-squared distribution of the likelihood ratio statistic, and thus the penalties we have developed in that setting will be appropriate more generally, providing that the likelihood ratio test statistic is approximately chi-squared distributed. This is demonstrated

empirically for count data under a negative binomial model in Section 4.4.

Whilst our new approach gives a test for detecting a single changepoint and estimating where it occurs, we can embed this within a recent wild binary segmentation procedure (Fryzlewicz, 2019) in order to efficiently detect multiple changes. We also introduce a fast post-processing step that estimates which series change at each changepoint. Importantly, this is done using information about all the estimated changepoints. In doing so, we reduce the problem that estimates of which variates change at a given point can be corrupted by other variates changing at nearby time points. We call the resulting multivariate changepoint algorithm Sparse and Ubiquitous Binary Segmentation in Efficient Time (or SUBSET), given that the method has computational efficiency which is competitive with other existing methods, while also being equipped to estimate the subset of variates within the dataset which are affected by each change.

The chapter is organised as follows: Section 4.2 formally introduces the multivariate changepoint detection problem, whilst Section 4.3 provides a complete description of the SUBSET procedure, including theoretical justifications in the change in mean setting. Section 4.4 compares the SUBSET method against a number of competitor methods in a simulation study covering both at most one change and multiple change scenarios. Section 4.5 applies SUBSET to the Global Terrorism Database (GTD). The GTD is a global historical record of terrorist incidents maintained by the National Consortium for the Study of Terrorism and Responses to Terrorism at the University of Maryland (Jensen, 2018). Specifically, we search for changepoints in overall terror activity in different regions of the world since 1970. We conclude with a discussion in Section 4.6.

4.2 Problem Formulation

We are interested in the problem of changepoint detection for multivariate data. One typical complication in such a setting, as identified by, for example, Chapter 4 of Pickering (2016), is the nature of the change in question, i.e. whether a change affects

all variates simultaneously, or just some subset (see Figure 4.1).

Figure 4.1: Four univariate sequences comprise this example dataset. There are three changepoints, which each affect a different number of variates: the first change affects the first variate only, the second change affects all variates and the third change alters the third and fourth variates.

Suppose that the data sequence for each variate, $(y_{i,j})_{j=1}^{n}$ for $i = 1, \ldots, d$, within the dataset, $\mathbf{y}_{1:n}$, can be segmented by changepoints, which are often shared across variates within the data. Following Chapter 4 of Pickering (2016), we define the set of changepoints to be points where at least one variate undergoes a change. Therefore, for each changepoint, there is an associated *affected set* of variates which undergo a change. Formally, let $0 = \tau_0 < \tau_1 < \ldots < \tau_m < \tau_{m+1} = n$ be the changepoints with corresponding affected sets $\mathcal{S}_1, \ldots, \mathcal{S}_m$. We will assume a parametric model for the data within a segment for each variate, and assume that the segment parameter for this model only changes at changepoints which affect that variate. To simplify the exposition, assume that the data are conditionally independent given the segment parameters. In other words, we have

$$y_{i,j} \sim g(.|\mu_{i,k}), \tag{4.2.1}$$

for some family of densities $g(.|.)$, where $k = |\{v : \tau_v < j\}| + 1$.

We remark that, in many practical applications, it can be expected that data will have dependence across the different variates within the system. While some recent works - see, for example, Aston and Kirch (2012b) and Bücher et al. (2014) - have considered this problem for contexts where assuming independence is much

less reasonable, such as financial time series, in general the current state of the art is to assume independence in everything except the changepoint locations. One likely effect of using the test statistic we propose on highly dependent data would be to depress the number of sparse changepoints detected, while inflating the number of dense changepoints detected. This is one important reason why we propose the post-processing step to our method presented in Section B.3. This post-processing step enables each variate to be considered separately, in a computationally efficient fashion, to remove any possible overfitting effects. For instance, as can be seen from our example application in Section 4.5, the only dense changepoint found by our method following post-processing can be explained by a change in the data collection method.

4.3 SUBSET

In this section, we introduce our new method for detecting multiple changepoints in the multivariate setting. We begin by discussing the detection of a single change.

4.3.1 Detecting a Single Changepoint

We begin with a derivation of the test statistic used by SUBSET in the single change setting. The log-likelihood ratio statistic for detecting a changepoint at time τ, affecting variates in set $\mathcal{S}$, is

$$R(\tau, \mathcal{S}) = 2\left[\sum_{i\in\mathcal{S}}\left\{\max_{\mu}\sum_{t=1}^{\tau}\log g(y_{i,t}|\mu) + \max_{\mu}\sum_{t=\tau+1}^{n}\log g(y_{i,t}|\mu) - \max_{\mu}\sum_{t=1}^{n}\log g(y_{i,t}|\mu)\right\}\right].$$

To simplify the notation, let $\mathcal{C}(y_{i,s:t}) = -2\max_{\mu}\sum_{t=s}^{t}\log g(y_{i,t}|\mu)$. Then we can define

$$D_{i,t} = \mathcal{C}(y_{i,1:n}) - \mathcal{C}(y_{i,1:t}) - \mathcal{C}(y_{i,t+1:n})$$

to be the contribution from the i^{th} series to the log-likelihood ratio statistic, if this variate is assumed to change at time t. Then $R(\tau, \mathcal{S}) = \sum_{i\in\mathcal{S}} D_{i,\tau}$.

Directly using the log-likelihood test statistic is complicated due to the fact we do not know τ or $\mathcal{S}$. In addition, different choices of $\mathcal{S}$ will allow for different numbers of

series to change. We therefore consider a penalised version of the test statistic, where the penalty depends on the number of variates that change, $|\mathcal{S}|$. We then maximise over possible choices of τ and $\mathcal{S}$. That is, we use $\max_t S_t$ as our test statistic where, for $t = 1, \ldots, n-1$,

$$S_t = \max_{\mathcal{S}} \sum_{i \in \mathcal{S}} D_{i,t} - \text{Pen}(|\mathcal{S}|),$$

for some suitable penalty function $\text{Pen}(\cdot)$.

As we shall describe in detail in Section 4.3.2, we suggest a piecewise linear penalty of the form $\text{Pen}(p) = \min\{\beta + \alpha p, K\}$ for some suitable constants α, β and K. We then detect a change if $\max_t S_t > 0$, with the location at $\hat{\tau} = \arg\max_t S_t$ and the set of affected variates estimated by $\arg\max_{\mathcal{S}} \sum_{i \in \mathcal{S}} D_{i,\hat{\tau}} - \text{Pen}(|\mathcal{S}|)$. Here we choose a piecewise linear penalty as this makes the maximisation over $\mathcal{S}$ computationally efficient. In particular, we define $D'_{i,t} = \max\{D_{i,t} - \alpha, 0\}$, and then

$$S_t = \max\left\{\sum_{i=1}^{d} D'_{i,t} - \beta, \sum_{i=1}^{d} D_{i,t} - K\right\}.$$

The two terms in the maximisation above correspond to the two different linear regimes in the penalty function. As we shall see later, the $\beta + \alpha p$ part of the penalty function determines the test statistic's behaviour for detecting sparse changes. Meanwhile, the constant term, K, is needed to improve power for detecting dense changes. Note here that if $S_t = \sum_{i=1}^{d} D'_{i,t} - \beta > 0$, then we say that we have detected a *sparse* change, with evidence for a change only in those variates i such that $D'_{i,t} > \alpha$. If, however, $S_t = \sum_{i=1}^{d} D_{i,t} - K > 0$, then all changes are labelled as affected by the estimated changepoint. In this situation, the change is described as *dense*.

4.3.2 Theory for a Change in Mean

To understand the behaviour of the test statistic for a single change, we study its theoretical properties for the canonical change in mean problem with Gaussian noise and a common, known variance, σ^2. As we are considering just a single change, we will simplify notation so that $\mu_{i,1}$ is the initial mean of series i. If there is a change, $\mu_{i,2}$ will be the mean after the change, and $\mu_{i,1} = \mu_{i,2}$ if $i \notin \mathcal{S}_1$. Thus, the data-generating

model is

$$Y_{i,j} = \epsilon_{i,j} + \begin{cases} \mu_{i,1} \text{ for } 1 \leq j \leq \tau, \\ \mu_{i,2} \text{ for } \tau + 1 \leq j \leq n, \end{cases} \quad \text{for } i \in \{1, \ldots, d\} \tag{4.3.1}$$

where the $\epsilon_{i,j}$, for $i = 1, \ldots, d$, and $j = 1, \ldots, n$ are a set of centred, independent and identically distributed Gaussian random variables.

For this particular problem, we have that

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

and consequently it follows that

$$D_{i,t} = \frac{1}{\sigma^2} \left[\frac{1}{t} \left(\sum_{k=1}^{t} y_{i,k} \right)^2 + \frac{1}{n-t} \left(\sum_{k=t+1}^{n} y_{i,k} \right)^2 - \frac{1}{n} \left(\sum_{k=1}^{n} y_{i,k} \right)^2 \right].$$

Hence $D_{i,t}$ is chi-squared distributed with 1 degree of freedom when no changepoint is present. We use this fact to establish false positive and detection probability results in the single change setting under Gaussian noise when $\max_t S_t$ is taken as the test statistic.

Our first theoretical contribution concerns the false positive rate of the chosen test statistic. As we shall explain shortly, this result motivates our specific choices for β, α and K in this setting.

Theorem 4.3.1. *Suppose we are in setting (4.3.1), and without loss of generality that in addition $\mu_{i,1} = \mu_{i,2}$ $\forall i$ and $Var\left(\epsilon_{i,j}\right) = 1$ $\forall i, j$. Take $\alpha = 2 \log d$, $\beta = (J + \epsilon) \log n$ and $K = \beta + d + \sqrt{2\beta d}$ for some $\epsilon > 0$; then*

$$\mathbb{P}\left(\max_t S_t > 0 \right) \leq C n^{1 - \frac{J}{2} - \epsilon/2},$$

where C is an absolute constant bounded above for all $d > 1$.

Proof: See Section B.2.

Note that taking $J = 2$ in the above corresponds to the standard BIC penalty for a change in a single parameter. However, we take $J = 4$ herein, as we later use a form of Binary Segmentation where we want to control the probability of $\max S_t > 0$ for

$\mathcal{O}(n)$ different regions of data in order to detect multiple changes (see Section 4.3.4). Note that this is under the assumption that the number of changes grows at most linearly with n.

We additionally remark that, as this result just follows from using the marginal chi-squared distribution of $D_{i,t}$ when there is no change, the penalties derived in Theorem 4.3.1 would be natural choices in other settings if the test statistic is based on the log-likelihood ratio statistic for a regular model. In practice, for such cases we recommend choosing α as above, but then tuning both β and K using simulated data. This helps to ensure that we have an appropriate overall false positive rate (e.g. 1%) and that we have similar rates of false positives where $\sum_{i=1}^{d} D'_{i,t} > \beta$ as where $\sum_{i=1}^{d} D_{i,t} > K$.

Given these choices for the penalty values, we next establish a result on the power of this procedure.

Theorem 4.3.2. *Assume that we are again in setting (4.3.1) with $\sigma^2 = 1$, and now we have that $\mu_{i,1} \neq \mu_{i,2}$ whenever $i \in \mathcal{S}_1 \subseteq \{1, \ldots, d\}$. Let $\Delta_i := |\mu_{i,2} - \mu_{i,1}|$. Then for $\delta > 0$ and $a = \max\{n, d\}$, we have that $\mathbb{P}(\max_t S_t > 0) \geq 1 - (a)^{-\delta}$, providing that, for $K_{\mathcal{S}_1} := \beta + |\mathcal{S}_1|\alpha$*

$$n\theta(1-\theta)\sum_{i\in\mathcal{S}_1}(\Delta_i)^2 \geq \begin{cases} V_S & \text{for a sparse change} \\ V_D & \text{for a dense change.} \end{cases}$$

Here $V_S := 4\delta\log a + K_{\mathcal{S}_1} - |\mathcal{S}_1| + 2\sqrt{\delta\log a\,(4\delta\log a + 2K_{\mathcal{S}_1} - |\mathcal{S}_1|)}$, $V_D := 4\delta\log a + K - d + 2\sqrt{\delta\log a\,(4\delta\log a + 2K - d)}$ and $\theta = \frac{\tau}{n}$ is fixed strictly between 0 and 1. Additionally, $2 > \delta > 0$ is required in the dense setting.

Proof: See Section B.2.

Note that $K_{\mathcal{S}_1} = \beta + |\mathcal{S}_1|\alpha$ corresponds to the total penalty incurred in the sparse setting. We introduce this notation to emphasise the link between the sufficiency conditions in the sparse and dense settings. We additionally remark that in the setting where $n = \max\{n, d\} \to \infty$, we require only that $\sum_{i\in\mathcal{S}_1}(\Delta_i)^2 > 0$ under both types of change asymptotically. In contrast, when $d = \max\{n, d\} \to \infty$, the leading

order term of the condition for detecting a sparse change is $\mathcal{O}\left(\sqrt{|\mathcal{S}_1|}\log d\right)$, while in the dense condition this is $\mathcal{O}\left(\sqrt{d\log d}\right)$. We also remark on the following. Assume that all series which change are altered by the same amount, Δ. Allow d to increase as $n \to \infty$. Then, for our choice of β, α and K, if $|\mathcal{S}_1| = o(\sqrt{d})$ - corresponding to a sparse change - we have power tending to 1 for detecting changes when $\Delta = \Omega\left(\sqrt{\frac{\log a}{n}}\right)$. By contrast, in the dense setting, we have power tending to 1 when $\Delta = \Omega\left(\frac{(\log a)^{1/4}}{n^{1/2}}\right)$.

We remark that the boundary of $|\mathcal{S}_1| = o(\sqrt{d})$ between sparse and dense changepoints corresponds to the sparsity boundary discussed in Enikeeva and Harchaoui (2019), who also propose a test statistic with 'two regimes'. For both their procedure and ours, this can be seen from considering the power of performing a likelihood ratio test across all variates when there is a change of Δ in $|\mathcal{S}_1|$ of the variates. If $|\mathcal{S}_1|$ dominates $\sqrt{d}$ in order, then this test has very high power in detecting the change. Conversely, if $|\mathcal{S}_1| = o(\sqrt{d})$, considering the maximum of the likelihood ratios across the variates gives a weaker requirement on the size of the changepoint. For a more detailed account of this transition, see the beginning of Section 2.2.

Note that this transition boundary is a distinct idea from the more traditional phase transition often discussed in changepoint detection. The latter typically refers to a boundary on the signal to noise ratio relative to the length of the sequence, below which consistency for any changepoint detection procedure becomes impossible. This boundary has been the subject of much recent interest. For example, Wang et al. (2019a) give results for the classical univariate change in mean setting under sub-Gaussian noise, and Wang et al. (2018) discuss the boundary for the change in covariance problem in a high dimensional setting.

4.3.3 Relationship to other Multivariate Changepoint Tests

For the change in mean setting, it is possible to draw strong comparisons between our approach and other multivariate changepoint tests, with the main difference being the means of aggregating evidence for a change across different variates. These alternative approaches use the CUSUM statistic for each variate within the dataset. The CUSUM

statistic is defined, in the known σ^2 case, as

$$W_{i,t} := \frac{1}{\sigma}\sqrt{\frac{t\,(n-t)}{n}}\left|\frac{1}{n-t}\left(\sum_{j=t+1}^{n} y_{i,j}\right) - \frac{1}{t}\left(\sum_{j=1}^{t} y_{i,j}\right)\right|,$$

for $i = 1, \ldots, d$ and $t = 1, \ldots, n-1$. Note in particular that $D_{i,t} = W_{i,t}^2$. Therefore, for the Gaussian change in mean setting, our test statistic can be expressed in terms of the CUSUM statistic as

$$S_t = \max\left\{\sum_{i=1}^{d} \max\{W_{i,t}^2 - \alpha, 0\} - \beta, \sum_{i=1}^{d} W_{i,t}^2 - K\right\}.$$

For comparison, three previously proposed test statistics, which we refer to herein as **Mean** (Groen et al., 2013), **Max** (Groen et al., 2013) and **Bin-Weight** (Cho and Fryzlewicz, 2015), are defined as follows

$$\begin{aligned} S_t^{(\text{mean})} &= \frac{1}{d}\sum_{i=1}^{d} W_{i,t} - \beta, \\ S_t^{(\text{max})} &= \max_i W_{i,t} - \beta, \\ S_t^{(\text{bin-weight})} &= \sum_{i=1}^{d} W_{i,t}\mathbb{1}\left\{W_{i,t} > \alpha\right\} - \beta. \end{aligned}$$

From the results in Enikeeva and Harchaoui (2019), we know that $S_t^{(\text{mean})}$ will have high power for dense changes, but lose power for sparse changes. By comparison, $S_t^{(\text{max})}$ will have higher power in the sparse case and lower power in the dense case. Enikeeva and Harchaoui (2019) propose combining both test statistics as a way of having higher power across both settings. Meanwhile, empirically, the behaviour of $S_t^{(\text{bin-weight})}$ depends on the choice of α. Specifically, if $\alpha = \mathcal{O}\left(\sqrt{d}\right)$ then it has high power for sparse changes, whereas if α is fixed as we increase d, it will have high power for dense changes.

4.3.4 Sparse and Ubiquitous Binary Segmentation in Efficient Time

We now formally introduce SUBSET (**S**parse and **U**biquitous **B**inary **S**egmentation in **E**fficient **T**ime), the full procedure for the use of the test statistic $\max_t S_t$ given

in Section 4.3.1. Given this threshold penalty approach, SUBSET is designed to detect both sparse and dense changes, the latter of which are labelled by SUBSET as affecting all variates within the data.

In order to detect multiple changes within the data, SUBSET uses a hybrid of Wild Binary Segmentation (Fryzlewicz, 2014), namely a very close variant of Wild Binary Segmentation 2 (Fryzlewicz, 2019). This enables a fast, approximate search for changes in the multivariate setting. We randomly generate M intervals of the dataset, on which we then subsequently compute the test statistic and search for the most significant point in the dataset across all intervals. We label the resulting point as a change if the penalty in the sparse or dense setting is exceeded. The procedure then repeats either side of the change.

For a sensible choice of M, the above results in a computationally efficient procedure with an execution time which is linear in the number of entries in the dataset. For example, in the simulation study that we report in Section 4.4, we use $M = 5$ and for Section 4.5's example we set $M = 10$. Note that these values of M are similar to those recommended in the use of Wild Binary Segmentation 2 by Fryzlewicz (2019), where here it was suggested that to obtain an equivalent guarantee on the detection of changes as Wild Binary Segmentation (Fryzlewicz, 2014), M should be set to $\mathcal{O}(\log n)$. (Throughout our simulations, $n = 1000$.) However, an advantage of a Wild Binary Segmentation 2 approach is that, in settings with a potentially large number of changes, such as in our real data example of Section 4.5, the value of M can be set slightly higher - but still much lower than the equivalent number of intervals needed in Wild Binary Segmentation - to allow a significant chance of all changes being detected. Fryzlewicz (2019) recommends $M = 100$ for such situations where the data are recorded across at most a few thousand time points, with M set at half the square of the length of sub-series at each stage of the procedure in the worst-case setting. Given the computational difficulty of these more conservative recommendations in the high dimensional setting, we suggest setting $M \approx \lfloor \log n \rfloor$ unless it is suspected that there may be a particularly high concentration of changes. In such cases, $M \approx k\lfloor \log n \rfloor$ for $k = 1, 2, \ldots$ can be tried for successive values of k

until similar results are yielded for two consecutive integer values.

One practical challenge with SUBSET is that while the estimates of $\hat{\boldsymbol{\tau}}$ tend to be fairly reasonable, the estimates of $\hat{\boldsymbol{\mathcal{S}}} = \left(\hat{\mathcal{S}}_1, \ldots, \hat{\mathcal{S}}_{\hat{m}}\right)$ are more prone to misspecification due to masking from other changepoints. This is especially true for variates which may also have a particularly strong change at a nearby time point. To mitigate this, we propose using a post-processing step where we individually analyse data from each variate conditional on the set of estimated changes, $\hat{\boldsymbol{\tau}} = (\hat{\tau}_1, \ldots, \hat{\tau}_{\hat{m}})$. When analysing a single variate, we only allow changes to occur within the set $\hat{\boldsymbol{\tau}}$. We detect the changes by minimising the univariate version of our penalised cost. That is, for variate i, we find

$$\arg \min_{0 \leq m' \leq \hat{m}; \{\xi_1, \ldots, \xi_{m'}\} \subseteq \hat{\boldsymbol{\tau}}} \sum_{k=1}^{m'+1} \left[\mathcal{C}\left(y_{i,(\xi_{k-1}+1):\xi_k}\right) + \beta \right].$$

This can be done efficiently using dynamic programming; see, for example, Section 2 of Tickle et al. (2018).

For the specific post-processing step we use to complete the SUBSET procedure, please see Section B.3. We include the post-processing step in the implementation of SUBSET in Sections 4.4 and 4.5, however for brevity we detail the SUBSET procedure without the post-processing step in Algorithm 3.

We remark that, under this procedure, it may be the case that we check for the possibility of a single changepoint within an arbitrary region of the dataset which contains no true change. In this case, a slight modification to Theorem 4.3.1 is required. The result which follows outlines that the probability of SUBSET locating an erroneous change in the multiple change setting remains low.

Corollary 4.3.3. *Consider the setting of Theorem 4.3.1. Using the SUBSET procedure with the penalties β, α and K as derived in Theorem 4.3.1, with $J = 4$, gives that the probability of erroneously placing an estimated changepoint within the dataset is bounded above by $Cn^{-\epsilon/2}$, where C is an absolute constant bounded above for all $d > 1$.*

Proof: See Section B.2.

Algorithm 3 SUBSET (without post-processing).

Data: A multivariate dataset, $(y_{i,j})_{i=1,\ldots,d,j=1,\ldots,n}$; variate penalty function, $\alpha(.,.)$; changepoint penalty function, $\beta(.,.)$; threshold penalty function, $K(.,.)$; segment cost function, $\mathcal{C}(.)$; an interval number, M; a sort function with respect to vector $\mathbf{v}$, $\rho_{\mathbf{v}}(.)$.

Result: An estimated set of changepoints $\hat{\tau}_1, \ldots, \hat{\tau}_{\hat{m}}$ and corresponding estimated affected sets $\hat{\mathcal{S}}_1, \ldots, \hat{\mathcal{S}}_{\hat{m}}$.

Step 0: Set $l = 1$, $u = n$, $\hat{\boldsymbol{\tau}} = NULL$, $\hat{\boldsymbol{\mathcal{S}}} = NULL$

Step 1: $l_{M+1} = l, u_{M+1} = u$

for $j \in \{1, \ldots, M+1\}$ **do**

$r \sim U\{l, \ldots, u\}$, $s \sim U\{l, \ldots, u\}$, $(l_j, u_j) = (\min(r,s), \max(r,s))$

if $u_j - l_j > 1$ **then**

for $t \in \{l_j, \ldots, u_j\}$ **do**

$S_{1,t} = \sum_{i=1}^{d} \max\left\{\mathcal{C}\left(y_{i,l_j:u_j}\right) - \mathcal{C}\left(y_{i,l_j:t}\right) - \mathcal{C}\left(y_{i,(t+1):u_j}\right) - \alpha(d, u_j - l_j + 1), 0\right\}$

$S_{2,t} = \sum_{i=1}^{d} \left\{\mathcal{C}\left(y_{i,l_j:u_j}\right) - \mathcal{C}\left(y_{i,l_j:t}\right) - \mathcal{C}\left(y_{i,(t+1):u_j}\right)\right\}$

$S_t = \max\{S_{1,t} - \beta(d, u_j - l_j + 1), S_{2,t} - K(d, u_j - l_j + 1)\}$

end

if $\max_t S_t > 0$ **then**

$q^j = \arg\max S_t$, $T_{q^j} = \max S_t$

if $S_{q^j} = S_{1,q^j} - \beta(d, u_j - l_j + 1)$ **then**

$\mathcal{T}_{q^j} = \left\{i : \mathcal{C}\left(y_{i,l_j:u_j}\right) - \mathcal{C}\left(y_{i,l_j:q^j}\right) - \mathcal{C}\left(y_{i,(q^j+1):u_j}\right) - \alpha(d, u_j - l_j + 1) > 0\right\}$

else

$\mathcal{T}_{q^j} = \{1, \ldots, d\}$

end

else

$\left(q^j, T_{q^j}, \mathcal{T}_{q^j}\right) = (NULL, 0, \emptyset)$

end

else

$\left(q^j, T_q^j, \mathcal{T}_{q^j}\right) = (NULL, 0, \emptyset)$

end

end

Step 2: Set $\mathbf{q} = (q^1, q^2, \ldots, q^{M+1})$

if $||\boldsymbol{q}||_0 \geq 1$ **then**

$\gamma = \arg\max_{j \in \{1,\ldots,M+1\}} T_{q^j}$, $\eta = q^\gamma$, $\mathcal{U} = \mathcal{T}_\eta$, $\hat{\boldsymbol{\tau}} = (\hat{\boldsymbol{\tau}}, \eta)$, $\hat{\boldsymbol{\mathcal{S}}} = \left(\hat{\boldsymbol{\mathcal{S}}}, \mathcal{U}\right)$

SUBSET $\left(\mathbf{y}_{l:\eta}, \alpha, \beta, K, \mathcal{C}(.)\right)$, SUBSET $\left(\mathbf{y}_{\eta+1:u}, \alpha, \beta, K, \mathcal{C}(.)\right)$

$\hat{\boldsymbol{\tau}} = \rho_{\hat{\boldsymbol{\tau}}}(\hat{\boldsymbol{\tau}})$, $\hat{\boldsymbol{\mathcal{S}}} = \rho_{\hat{\boldsymbol{\tau}}}\left(\hat{\boldsymbol{\mathcal{S}}}\right)$

else

$\eta = NULL$, $\mathcal{U} = \emptyset$

end

4.4 Simulation Study

In this section, we examine the properties of the SUBSET method against the CUSUM aggregation procedures discussed in Section 4.3.3. In addition, we compare these methods against **Inspect** (Wang and Samworth, 2018). To implement Inspect, we use code from the `InspectChangepoint` package (Wang and Samworth, 2016).

All simulations were run in R using a Linux OS on a 2.3GHz Intel Xeon CPU. We examine multivariate series with pairwise independent Gaussian noise with variance 1, and count data generated according to a negative binomial likelihood model under various different dispersion parameters. For all scenarios considered, 200 repetitions were simulated.

Throughout this section, $(\alpha, \beta, K) = \left(2\log d, 4\log n, d + 4\log n + \sqrt{8d\log n}\right)$ for the SUBSET method, as per the result of Theorem 4.3.1. The threshold penalty for Inspect and the β values for the CUSUM-based methods were computed using simulations from the null model, such that the false alarm rate was fixed at 5%. Note that for the Bin-Weight procedure, the α value was taken to be $\sqrt{2\log n}$. The justification for this arises from a consideration of the theoretical false alarm error rate under an aggregation of CUSUMs; see, for example, Lemma 4 in the Supplementary Materials of Wang and Samworth (2018).

4.4.1 Gaussian Setting, At Most One Change in Mean

Our first examination concerns the false alarm error rate of each of the methods. As stated, we fix this at 5% for Bin-Weight, Inspect, Max and Mean. The penalty choices for SUBSET lead to no false positives across all simulated data scenarios ($n = \{1000, 10000, 100000\}$ and $d = \{5, 10, 50, 100, 500, 1000\}$).

To check the power of the methods in the single change setting, we examine five scenarios for the location of the changepoint. These correspond to the change being found at proportions $0.050, 0.081, 0.184, 0.266$ and 0.383 (to 3 d.p.) respectively along the series. We increase $\Delta\mu$, the absolute change in mean - for each variate in which a change occurs - from 0.01 to 1.00 in increments of 0.01, and record the proportion

of tests which yield a missed change in each case. We do this for $n = 1000$, $d = \{5, 10, 50, 100, 500, 1000\}$ and for densities of change corresponding to 0.5%, 1%, 5%, 10%, 50% and 100% (where feasible) of the variates affected by the change.

Figure 4.2 shows the result of this for $n = d = 1000$ when the location of the change is 5% of the way along the time series, for each of the densities of change, and for each of the methods under investigation. These results indicate that SUBSET is at least competitive with other methods, and often yields a smaller Type II Error. In particular, we observe that SUBSET and Bin-Weight appear to give the most 'balanced' performances of all the methods present, exhibiting competitive power for all the regimes. This is in contrast to some of the other procedures. For example, for the dense regimes, we see that the Mean method performs best, while the Max method is the worst performing method. The situation is exactly reversed in the sparse examples. Similar patterns are seen for the other cases - please refer to Section B.4.

We next compare the average location errors of the methods. We again consider the same $n = d = 1000$ cases as in Figure 4.2 across the same set of values for $\Delta\mu$. The results are shown in Figure 4.3. It is interesting to note that the SUBSET method gives a relatively small location error in most settings, even in comparison to the closest competitor methods. Indeed, in comparing Figure 4.3 to Figure 4.2 we see that once the probability of a Type II error falls below 1 for the SUBSET procedure, the location accuracy among those instances where SUBSET correctly identifies the presence of a changepoint is high.

The final point of interest we mention here is the potential misspecification of the true affected set at the change by SUBSET. Note that the competitor methods do not give information on the affected subset of variates at a changepoint, so no comparison is possible here. We again examine the same scenarios as for the power and location error. The results are shown in Figure 4.4. Figure 4.4 indicates that, when the Type II error is below 1 (again, see Figure 4.2 for comparison), SUBSET is effective at estimating the true affected variate set in the various sparse settings. SUBSET also exhibits a low 'Variate Error' in the case where all variates change. Note that for the instance where, for example, 10% of the variates are affected by

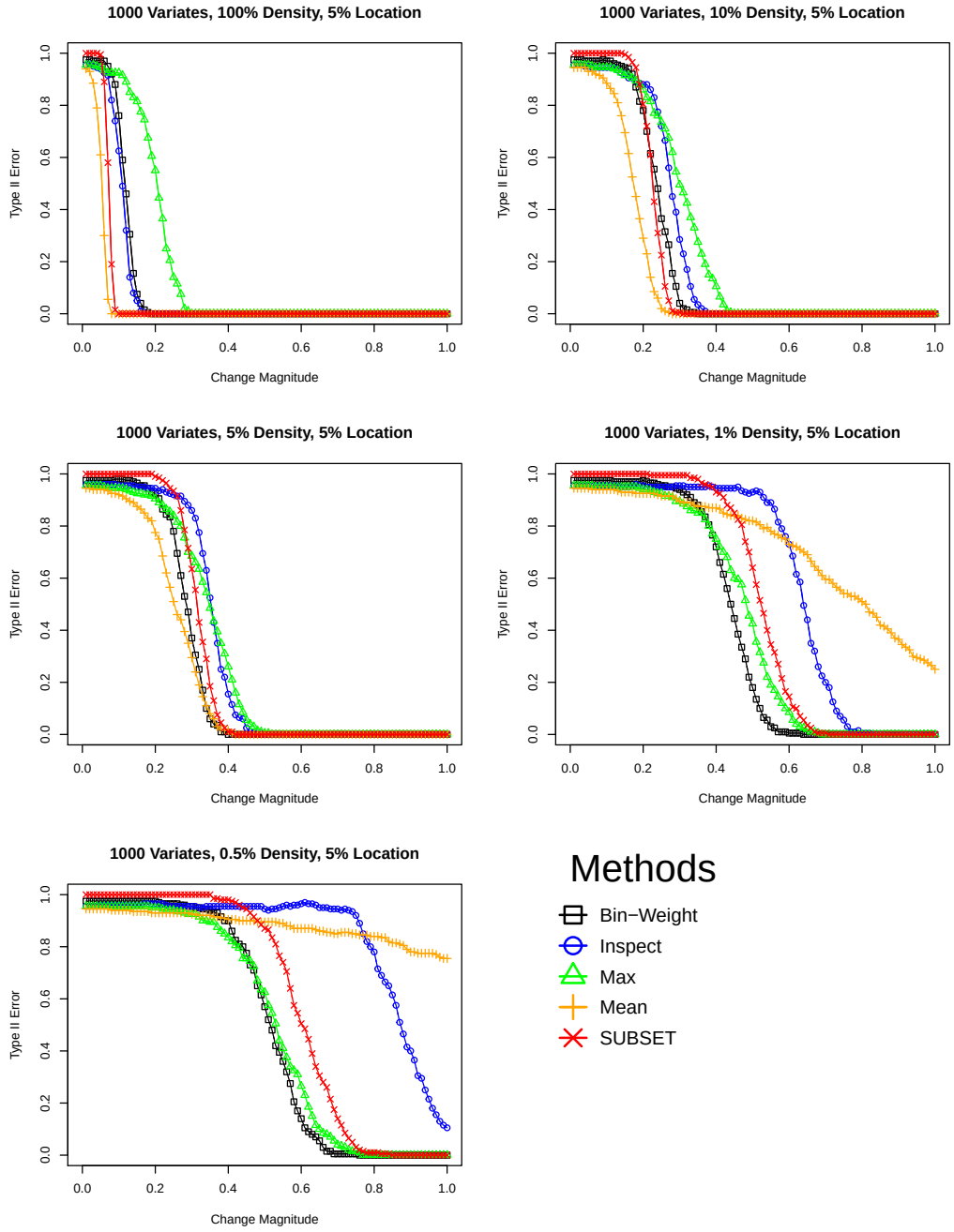

Figure 4.2: Type II Errors (in the AMOC setting) across a range of values for $\Delta\mu$ between 0.01 and 1 for each of the five methods under investigation for different subset densities of the changepoint, keeping the temporal location of the changepoint fixed at 5% of the way along the series and $n = d = 1000$. 200 repetitions were simulated in each case.

the change, we see the effect of the threshold penalty K. For smaller values of the change magnitude, SUBSET detects only very sparse effects for smaller $\Delta\mu$. A dense effect is then correctly identified at a threshold value of $\Delta\mu$ of just above 0.6. This

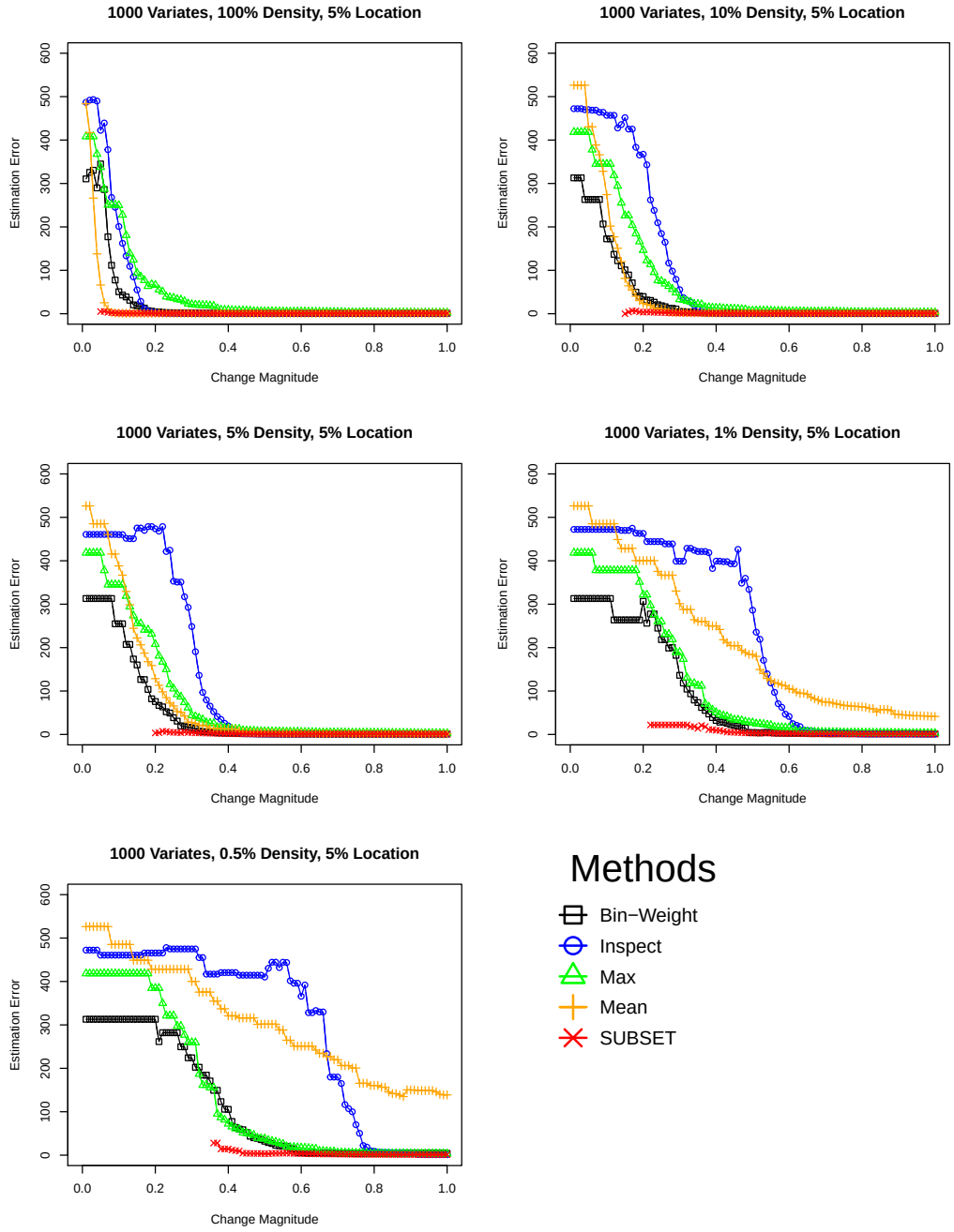

Figure 4.3: Location Errors across a range of values for $\Delta\mu$ between 0.01 and 1 for each of the five methods under investigation for particular densities of change (i.e. percentage of variates affected). Note that $n = d = 1000$, and the changepoint is fixed at 5% of the way along the series. In addition, there are no values for SUBSET below certain change magnitudes as no changepoints are estimated by the procedure in these cases (compare with Figure 4.2). 200 repetitions were simulated in each case.

phenomenon is empirical proof that the conditions from Theorem 4.3.2 on detecting a dense change are stricter than for a sparse charge.

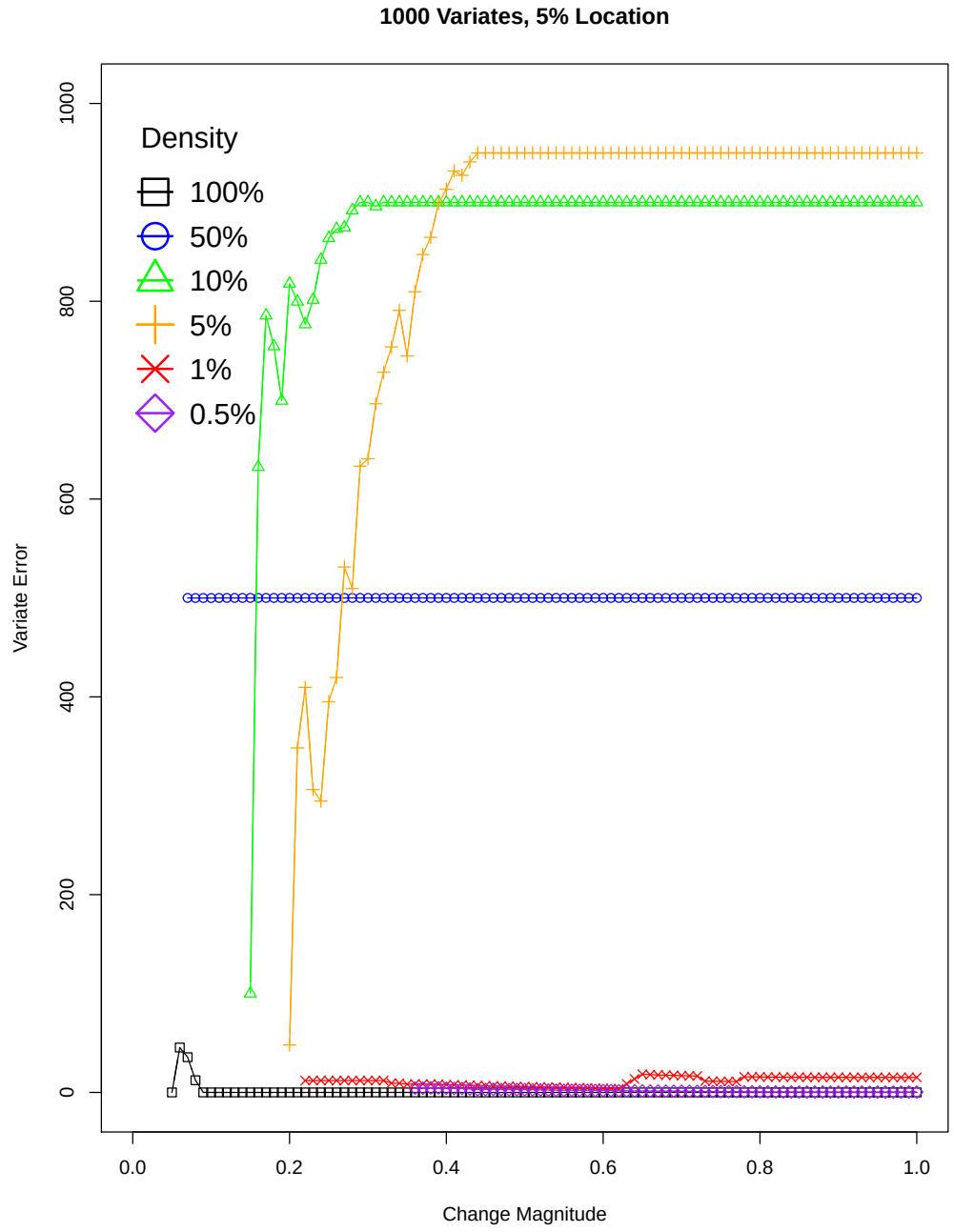

Figure 4.4: Variate Errors across a range of values for $\Delta\mu$ between 0.01 and 1 for the SUBSET method under different densities of change. Note that again $n = d = 1000$, and the changepoint is fixed at 5% of the way along the series. In addition, there are no values below certain change magnitudes as no changepoints are estimated by the procedure in these cases (compare with Figure 4.2). 200 repetitions were simulated in each case.

4.4.2 Gaussian Setting, Multiple Changes in Mean

We now turn to the more complex task of extending to the multiple changepoint setting. We examine five scenarios, which we label as F, G, H, I and J here, each with three changepoints present. In each case, the changepoints may be found at proportions 0.124, 0.394 and 0.989 of the way along the series. The only difference between scenarios is the size of each affected set of variates at each change. Thus, the scenarios imply different affected sets depending on the value of d. The scenarios are summarised below for $d = 1000$. Note that we once again fix $\sigma^2 = 1$ in all cases.

F : All three changes affect all variates.

G : The first and third changes affect all variates; the second change affects 0.5% of variates.

H : The first and third changes affect 0.5% of variates; the second change affects all variates.

I : All changes affect 1% of variates.

J : The first, second and third changes affect 0.5%, 1% and 5% of variates respectively.

Here, we restrict ourselves to examining the power of the methods. Note that we herein define a 'missed change' as being a true changepoint for which the methods do not place an estimated change within $\lceil \log n \rceil$ points. Table 4.1 shows the average number of changes missed by each of the methods in each of the five scenarios for $n = d = 1000$, when $\Delta\mu = 1$ for all variates which undergo a change at any changepoint. As can be seen from Table 4.1, the best performing methods across most of the scenarios are SUBSET, Bin-Weight and Inspect. The average number of missed changes for these three methods is generally very similar across all tested instances.

4.4.3 Negative Binomial Setting

We now turn to consider the multivariate changepoint detection problem for data distributed according to a negative binomial. In particular, we parameterise the negative binomial with a success probability p and an over-dispersion number, r. The

Average Number Missed (Average False Alarms)	Method				
Scenario	SUBSET	Mean	Max	BW	Inspect
F	0.06 (**0.00**)	0.05 (0.23)	0.24 (0.35)	**0.00** (69.7)	**0.00** (0.89)
G	0.12 (**0.00**)	1.06 (0.40)	0.34 (0.45)	0.01 (49.3)	**0.00** (0.89)
H	0.44 (**0.01**)	1.97 (0.59)	0.46 (0.37)	0.03 (30.2)	**0.03** (0.86)
I	0.22 (**0.01**)	1.97 (1.41)	0.47 (0.34)	0.05 (13.9)	**0.02** (0.94)
J	0.19 (**0.01**)	1.60 (1.26)	0.38 (0.36)	0.03 (16.4)	**0.03** (0.96)

Table 4.1: The average number of changes missed by each of the methods with $n = d = 1000$ fixed in all cases and $\Delta\mu = 1$ for any variate undergoing a change. Each of the scenarios F, G, H, I and J has 3 changepoints, and the percentage of variates affected by each change in each scenario is discussed at the beginning of Section 4.4.2. Bold entries show the best performing algorithm. 200 repetitions were simulated in each case.

latter represents the number of failures before an experiment is stopped. Formally, we have

$$y_{i,j} \sim \begin{cases} \text{Neg-Bin}\left(r_{i,1}, p_{i,1}\right) \text{ for } 1 \leq j \leq \tau_1, \\ \text{Neg-Bin}\left(r_{i,2}, p_{i,2}\right) \text{ for } \tau_2 + 1 \leq j \leq \tau_2, \\ \ldots \\ \text{Neg-Bin}\left(r_{i,m+1}, p_{i,m+1}\right) \text{ for } \tau_m + 1 \leq j \leq n \end{cases} \quad \text{for } i \in \{1, \ldots, d\}, \quad (4.4.1)$$

for some sequence, $(r_{i,k})_{k=1}^{m+1}$, of unknown over-dispersion parameters and some sequence, $(p_{i,k})_{k=1}^{m+1}$, of unknown success probabilities. Given the difficulty of computing the maximum likelihood estimators for the former at a given stage of the procedure, we assume that the over-dispersion parameter changes only if the unknown success probability also changes. Subsequently, we compute a methods of moments estimator (Savani and Zhigljavsky, 2006) for these over-dispersion parameters at each stage of the procedure.

Note that while SUBSET extends naturally to the negative binomial setting through adapting the $D_{i,t} = \mathcal{C}(y_{i,1:n}) - \mathcal{C}\left(y_{i,1:t}\right) - \mathcal{C}\left(y_{i,t+1:n}\right)$ quantities to an appropriate $\mathcal{C}(.)$, the other methods examined in Section 4.4.1 and 4.4.2 are not designed for this case. While they can still be applied, they work poorly - see Section B.4.

We firstly examine the null setting for $p_{i,1} = 0.5\ \forall i$, $r = \{1, 100\}$ ($\forall i$ in either case), $n = 1000$ and $d = \{5, 50, 100, 1000\}$. In all cases apart from $(r, d) = (1, 1000)$ - which gives a 1% false alarm rate - we record no false alarms.

We then check the multiple change setting, again using scenarios F, G, H, I and J from Section 4.4.2 with the three changes at the same points in the series. Table 4.2 summarises the results for $r = 20$ and $n = d = 1000$. At each changepoint, for those variates which are affected, the change manifests as a shift in the success probability parameter by 0.1, where each series starts with $p_{i,1} = 0.5$. The results show that SUBSET misses few of the changes in any of the scenarios, while consistently giving a very low false alarm error rate. Note that a false alarm here is defined as in Section 4.4.2.

Average Number Missed (Average False Alarms)	Method
Scenario	SUBSET
F	0.07 (0.02)
G	0.10 (0.01)
H	0.29 (0.02)
I	0.16 (0.01)
J	0.19 (0.02)

Table 4.2: The average number of changes missed by SUBSET across 200 repetitions in the negative binomial setting, with an over-dispersion parameter of 20, $d = n = 1000$ fixed in all cases, and $\Delta p = 0.1$ for any variate undergoing a change. Each of the scenarios F, G, H, I and J has 3 changepoints, and the percentage of variates affected by each change in each scenario is discussed at the beginning of Section 4.4.2. 200 repetitions were simulated in each case.

4.5 Detecting Changes in Global Terrorism

Please note that the Global Terrorism Database is copyrighted to the University of Maryland (2018).

The Global Terrorism Database (GTD), first introduced to the literature by LaFree and Dugan (2007), was built from the Pinkerton Global Intelligence Services (PGIS) database. This collated all terrorist incidents from 1 January 1970 onwards. PGIS defined terrorism as 'events involving "threatened" or actual use of illegal force and violence to attain a political, economic, religious or social goal through fear, coercion or intimidation.' Please see LaFree and Dugan (2007) for a more in-depth discussion on this, including further refinements to the definition in order to bring the number of events down to a record-able level.

Previous examinations of the GTD in LaFree and Dugan (2007), LaFree (2010) and LaFree et al. (2014) have highlighted several important points. These include the high preponderance of terrorism in Europe in the 1970s; a period of unusually high terrorist activity in Latin America between 1980 and 1997; and a more general note regarding the concentration of most incidents within geographic space. This last observation appears to be a result of the fact that most terrorist incidents in the period of interest have been domestic. For another analysis of this dataset, see Santifort et al. (2012), where changes in the 'arrival rate' of terrorist incidents in the univariate setting were found using a Reversible Jump Markov Chain Monte Carlo (RJMCMC) approach. Other analyses of similar data include Clauset and Young (2005), which examines the period between 1968 and 2004. However, the emphasis here was on the severity, rather than number, of incidents for a given area at a given moment in time.

We approach the problem of analysing the GTD from a multivariate changepoint perspective. The GTD naturally stratifies the globe into twelve regions: Australasia & Oceania, Central America & Caribbean, Central Asia, East Asia, Eastern Europe, Middle East & North Africa, North America, South America, South Asia, Southeast Asia, Sub-Saharan Africa and Western Europe. Given that these political terms may

be somewhat fluid geographically, we show this division pictorially in Figure B.1 in Section B.5. For each of the twelve regions, we aggregated all incidents for each month to produce one univariate time series of counts for each region. Each of these is of length 564, one for each month between January 1970 and December 2017 inclusive. Note that 1993 is not included, as that year's data are missing from the publicly available copy of the database. The resulting incident count by region is shown in Figure B.2 in Section B.5.

As the resulting series consist of count data, we model the data for each series as realisations from a negative binomial with changing 'success' probability parameter. We then apply SUBSET as per the study in Section 4.4.3. The results of this are displayed in Table B.7 and Figure B.3 in Section B.5; these document the months in which the estimated changepoints of the period occurred, and the corresponding estimated geographical regions affected.

We here summarise the results given for the Middle East and North Africa, North America and Western Europe regions. The plots showing the dates of the changes which affect these regions are given in Figure 4.5.

Some notable features are apparent: for example, one of the very few dense changes located by SUBSET (in January 1998) corresponds to an alteration in the data collection method for all regions. Else, most of the estimated changes are in fact sparse. This corresponds to the commentary found in, for example, LaFree et al. (2014), which asserts that most causes of terrorism remain localised. For instance, the change in the Middle East and North Africa in early 2013 appears to correspond to the beginning of the so-called 'Arab Winter'. Meanwhile the period of more intense activity in Western Europe in the later 1970s seems to broadly align with some of the worst years of the Troubles.

4.6 Discussion

We have proposed a means of computationally efficient multivariate changepoint detection. This method incorporates a penalised likelihood approach with that of

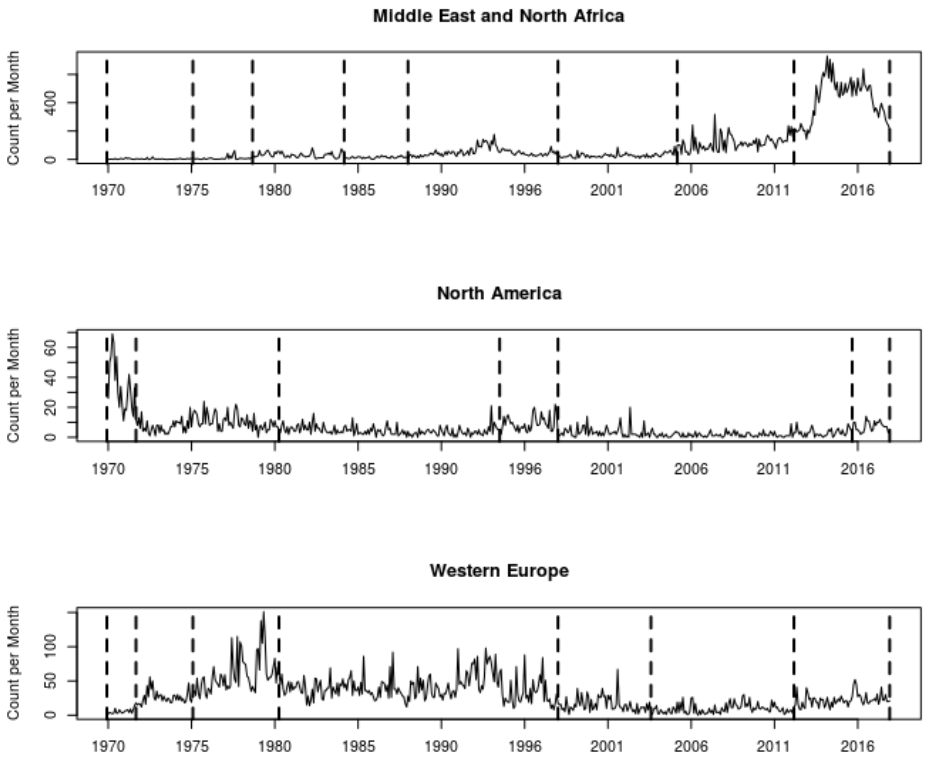

Figure 4.5: Terrorism incident count per month for the Middle East and North Africa (top), North America (middle) and Western Europe (bottom) from January 1970 to December 2017. Changes found by the SUBSET method using a negative binomial cost function are overlaid as dashed vertical lines.

a recently introduced, computationally efficient variant of Wild Binary Segmentation. We have demonstrated that the method has good theoretical and computational properties in a variety of cases. These cases range from the at most one change in mean problem, to more complex multiple change problems which potentially exhibit more difficult behaviour at each changepoint. In addition, we believe that the suggestions for implementation made here, such as the appropriate settings of the penalty values, will be of use to practitioners.

Some remaining challenges include an explicit algorithmic treatment of correlated or lagged changes to provide a clearer quantitative picture of a common cause of a change. Presently, this is an issue of penalty adjustment. Another issue to overcome is the fact that this method is best employed under specific parametric assumptions. It would be desirable to find a setting for this method under which these may be relaxed.

Perhaps the most important issue from a data streaming perspective, however, is that this method, while efficient, is highly offline.

It is this latter challenge in particular, namely achieving a reliable sequential changepoint detection algorithm in a general high-dimensional setting, that we believe forms the basis of the most interesting problem arising from this method.

Chapter 5

An Online, Nonparametric Method for the Detection of Multivariate Changepoints

5.1 Introduction

Correctly identifying time points in a data series where a phenomenon changes, usually referred to as changepoint detection, is a problem which is currently receiving considerable attention. Many recent authors have explored the problem of changepoint detection for contexts as varied as autonomous vehicle navigation, hyperspectral imaging and European flood risk (Alcantarilla et al., 2018; Merz et al., 2012; J. López-Fandiño et al., 2019).

With the growing preponderance of data generated in a streaming context, interest in the changepoint community is increasingly focusing on the challenging problem of detecting changepoints in a multivariate setting while collection is still in progress. We refer to this as the multivariate online detection problem herein. Several contributions have lately been seen in this area. For example, Tran (2019) uses an approach building on K-means clustering (see, for instance, Hartigan and Wong (1979) among many others) within two 'rolling windows'; Ahmad et al. (2017) introduce a new means of detecting changepoints and anomalies using Hierarchical Temporal Memory, a

deep learning method; and Sethi and Kantardzic (2017) present a method based on the analysis of margin density, where the margin is defined as the portion of space most vulnerable to misclassification. In short, existing approaches to the problem are extremely varied and often interestingly distinct from procedures developed for classical settings (i.e. univariate data or an offline context or both).

Many of the traditional approaches in the univariate, offline setting rely on dynamic programming to minimise a well-chosen cost function. Typically, this dynamic programming setup, for example with methods such as Optimal Partitioning (Jackson et al., 2005) and Pruned Exact Linear Time (Killick et al., 2012), performs a scan through the data sequence, conditioning on the time of the most recent changepoint. Given that the most recent changepoint is unknown, it is subsequently inferred following the computation of the minimum global cost. Hence, the optimal locations of the changepoints for minimising this cost are found.

This idea of considering the most recent changepoint has been popular in the univariate online setting of the changepoint problem, too. Many Bayesian approaches, such as those in Fearnhead and Liu (2007), Niekum et al. (2015), Ruggieri and Antonellis (2016) and others adopt this style of approach. For such methods, the use of a hidden state Markov model is a natural, and common, choice. In addition, other non-Bayesian sequential methods such as the Shiryaev-Roberts (S-R) procedure - see, for example, Polunchenko and Tartakovsky (2010) - 'reset' on raising an alarm (i.e. estimating a change). In short, considering a history of the stream only up to the most recent change is a natural mechanism for avoiding computational overhead.

One major issue with existing online methods is that very specific assumptions are often required on how a system behaves. For example, the S-R procedure assumes that the densities prior to and even following a change are known. This is in addition to the more typical assumption that the data are i.i.d. either side of an unknown changepoint location, τ. Meanwhile, many of the aforementioned Bayesian methods require prior beliefs on (i) the parametric family of the generating process of the data prior to or following the change; or (ii) the evolution of the system from one point to the next, assuming no change; or (iii) the time between successive changepoints;

or some combination of (i), (ii) and (iii). Indeed, the more recent multivariate online methods discussed earlier almost universally make assumptions of type (i), (ii) or (iii). In a general streaming setting, such assumptions are highly undesirable, and the need for a nonparametric approach is clear.

We therefore present a novel nonparametric approach to the changepoint detection problem in the streaming setting. This is an **O**nline, **M**ultivariate, **E**mpirical, **N**onparametric method (OMEN) for change detection, suitable in a variety of contexts where only within-segment i.i.d. observations can be assumed. In Section 5.2, we formally introduce the nonparametric changepoint detection problem and discuss existing approaches to changepoint detection on which OMEN builds. In Section 5.3, we describe the OMEN procedure and its computational properties, and establish a false alarm result. We also provide a comparison between our method and another recent online, multivariate, nonparametric method (Chen, 2019b). In Section 5.4, we compare the performance of OMEN, the method of Chen (2019b) and a current popular multivariate technique (Wang and Samworth, 2018) in an 'as if online' simulation study. In Section 5.5, we apply OMEN to hourly observations of wind speeds between 2012 and 2017 in various cities in Canada and Israel. We conclude with a discussion in Section 5.6.

5.2 Background

We are interested in detecting changepoints in an online fashion in the multivariate setting. We suppose that we observe a d-dimensional data stream. Let $\mathbf{y}_t := (y_{1,t}, \ldots, y_{d,t})$ be the observation at time t, and let $\mathbf{y}_{1:T} := (\mathbf{y}_1, \ldots, \mathbf{y}_T)$ be the set of observations up to and including time T, the most recent time point we have observed.

We make the assumption that each variate of the stream follows some stationary process, which is then potentially affected by a changepoint. Suppose that, up to time T, these changepoints occur at times $0 = \tau_0 < \tau_1 < \ldots < \tau_m < T$, for some (typically unknown) m. Additionally, we say that some non-empty subset, or *affected set*, of the

variates, $\mathcal{S}_j \subseteq \{1, \ldots, d\}$, within the system are affected by the changepoint τ_j.

Formally, for an individual variate $i \in \{1, \ldots, d\}$, for which we have seen $y_{i,1:T}$ up to time T, we can then define its set of changepoints. Let $m_i = |\{k : \tau_k < T, i \in \mathcal{S}_k\}|$ be the number of changepoints, and let $\tau^{(i)}_{1:m_i}$ be the set of changepoints which affect the i^{th} variate of the stream. Then

$$y_{i,t} \sim \begin{cases} G_{i,1} & \text{for } t \in \left\{1, \ldots, \tau^{(i)}_1\right\} \\ G_{i,2} & \text{for } t \in \left\{\tau^{(i)}_1 + 1, \ldots, \tau^{(i)}_2\right\} \\ & \ldots \\ G_{i,m_i+1} & \text{for } t \in \left\{\tau^{(i)}_{m_i}, \ldots, T\right\}. \end{cases} \tag{5.2.1}$$

Here we have used $G_{i,1}, G_{i,2}, \ldots, G_{i,m_i+1}$ to refer to time-independent data generating processes, such that observations drawn from within the same segment are also independent and exchangeable.

Note that we have placed no stipulations on the nature of a change between two consecutive generating process, $G_{i,j}$ and $G_{i,j+1}$. It could be that the processes differ only in one parameter, or else several parameters, or else are drawn from entirely different classes of distribution. In short, (5.2.1) is the nonparametric changepoint problem under assumptions of independence and exchangeability within a segment. Nonparametric change detection is a well-studied field. Efforts from Carlstein (1988), Csörgő and Horváth (1988), Dümbgen (1991) and Wolfe and Schechtman (1984) were among the first to describe the problem and propose detection methods for the single change case in the univariate setting. More recently, nonparametric techniques such as those of Zou et al. (2014), Haynes et al. (2017b) and Wang et al. (2019b) have arisen to detect multiple changes in a univariate offline setting. Still others, such as Ross et al. (2011) and Matteson and James (2014), have had success in either the online or multivariate domain. However, resolving (5.2.1) in an online fashion remains a largely open challenge (with only a small number of very recent exceptions, such as Chen (2019b) - see Section 5.3.3 for more information). We attempt to address this with our new method, OMEN.

5.3 Methodology

In this section, we introduce the OMEN method, an online nonparametric method for detecting changepoints in multivariate data. We describe the properties of OMEN, giving a false alarm error rate result and discussing its other theoretical properties, as well as covering the computational aspects of the procedure.

5.3.1 An Online, Multivariate, Empirical, Nonparametric changepoint detection method (OMEN)

As for Section 5.2, we assume we have observed, by time T, the data stream $\mathbf{y}_{1:T}$.

For $T < \omega$, we simply collect more data, meaning we cannot raise the alarm for any change - true or not - during this time. (Although a change can still be flagged within this period after the fact, as we shall see.) In short, ω can be thought of as a 'learning window', a popular concept in the online changepoint detection literature (Cao et al., 2018; Guo et al., 2016; Harel et al., 2014; Keogh et al., 2001; Malladi et al., 2013). We discuss the best choice for ω in Section 5.3.2.

At $T = \omega$, we compute, for $i = 1, \ldots, d$, the empirical cumulative distribution function - used in nonparametric changepoint detection since at least Pettitt (1980) - for each variate as

$$\hat{F}_i^{\omega}(x) = \frac{1}{\omega} \sum_{t=1}^{\omega} \mathbb{1}\left\{y_{i,t} \leq x\right\}, \quad x \in (-\infty, \infty). \tag{5.3.1}$$

We then use the empirical cdfs obtained in (5.3.1) to transform the incoming stream. This begins with the observations already recorded, so that, for $i = 1, \ldots, d$, $t = 1, \ldots, \omega$

$$z_{i,t} := \hat{F}_i^{\omega}(y_{i,t}). \tag{5.3.2}$$

Note that, for each i, $(z_{i,t})_{t=1}^{\omega}$ will form the sequence $\frac{1}{\omega}, \ldots, \frac{\omega}{\omega}$ in some order. If there has been no changepoint in the first ω points, and our assumptions of independence and exchangeability from Section 5.2 hold, then the sequence obtained will be equivalent to sampling ω times from $U\left(\{1/\omega, \ldots, \omega/\omega\}\right)$ without replacement. Therefore, as $\omega \to \infty$, values in the sequence resemble draws from $U\left[0, 1\right]$.

For applicability to a wide class of possible changepoint tests, it is useful to then further transform the stream so that the data are standard normal assuming no change. One option here is to use the inverse cdf of a standard normal, $\Phi^{-1}(.)$, to transform the data. However, in practice, a large value of ω is required to give the use of the inverse cdf more power than our method. Instead, we use the Box-Muller transform (Box and Muller, 1958). This results in two streams for $i = 1, \ldots, d$, $t = 1, \ldots, \omega$

$$a_{i,t} = \sqrt{-2 \log u_{i,t}} \cos(2\pi z_{i,t})$$
$$b_{i,t} = \sqrt{-2 \log u_{i,t}} \sin(2\pi z_{i,t})$$

where $u_{i,t}$ (for the same range of i and t) are a set of simulated, independent realisations from $U[0,1]$. By the properties of the Box-Muller transformation, corresponding entries in the resulting two streams are independent, as well as distributed according to the standard Gaussian. It is for these transformed streams for which we then run a test for a changepoint.

We remark that regardless of the original distributions of the variates in the stream, $|\cos(2\pi z_{i,t})| < 1$ (and similarly $|\sin(2\pi z_{i,t})| < 1$). Hence $-|\sqrt{-2 \log u_{i,t}}| < a_{i,t} < |\sqrt{-2 \log u_{i,t}}|$ (and similarly for $b_{i,t}$). However, as $u_{i,t}$ is a standard uniform, then $\mathbb{P}\left(|\sqrt{-2 \log u_{i,t}}| \geq x\right) = \exp\left(-\frac{1}{2}x^2\right)$ for $x > 0$. Therefore, both $a_{i,t}$ and $b_{i,t}$ are stochastically dominated by a sub-Gaussian random variable, and so are also sub-Gaussian. We use this fact later, to prove a false alarm result for our method (see Lemma 5.3.1).

For $T > \omega$, we compute $z_{i,T} = \hat{F}_i^{\omega}(y_{i,T})$, $a_{i,T}$ and $b_{i,T}$ for $i = 1, \ldots, T$. We then test for a change in each variate separately to examine the case for a change in the stream. Note that the Box-Muller transform ensures that the stream remains sub-Gaussian, even if there has been a change. We therefore use a test for a change from a normal with mean 0 and variance 1 to a normal with unknown mean μ and variance σ^2. The

likelihood ratio test statistic for a change in variate i at time k in, say, stream $\mathbf{a}$ is

$$\begin{aligned} S(k; a_{i,(T-\omega+1):T}) &= -2\log\left(\frac{\prod_{t=T-\omega+1}^{T}\frac{1}{\sqrt{2\pi}}\exp(-a_{i,t}^2/2)}{\prod_{t=T-\omega+1}^{k}\frac{1}{\sqrt{2\pi}}\exp(-a_{i,t}^2/2)\prod_{t=k+1}^{T}\frac{1}{\sqrt{2\pi\hat{\sigma}^2}}\exp\left(-(a_{i,t}-\hat{\mu})^2/2\hat{\sigma}^2\right)}\right) \\ &= \sum_{t=k+1}^{T} a_{i,t}^2 - (T-k)\log(\hat{\sigma}^2) - \sum_{t=k+1}^{T}(a_{i,t}-\hat{\mu})^2/\hat{\sigma}^2 \\ &= \sum_{t=k+1}^{T} a_{i,t}^2 - (T-k)\left[\log(\hat{\sigma}^2)+1\right], \end{aligned}$$

where $\hat{\mu} = \frac{1}{T-k}\sum_{t=k+1}^{T} a_{i,t}$ and $\hat{\sigma}^2 = \frac{1}{T-k}\sum_{t=k+1}^{T}(a_{i,t}-\hat{\mu})^2$.

If, for a particular variate i, $\max_{T-\omega+1\leq k\leq T-1} S\left(k; a_{i,T-\omega+1:T}\right) > \beta'$, for some β' - the choice of which we discuss in Section 5.3.2 - then we check that

$$\max_{T-\omega+1\leq k\leq T-1} S\left(k; b_{i,T-\omega+1:T}\right) > \beta'.$$

If β' is exceeded for both sequences, then we compute a test statistic for a multivariate change in the last ω time points. We here use a further test statistic, S', to capture changes which have a 'marked' effect on the stream, likely causing many variates to change. (A change of this kind is often referred to as a 'dense' changepoint.) We set

$$S' = \max\left\{\max_{T-\omega+1\leq t\leq T-1}\sum_{i=1}^{d} W_t\left(a_{i,T-\omega+1:T}\right)^2, \max_{T-\omega+1\leq t\leq T-1}\sum_{i=1}^{d} W_t\left(b_{i,T-\omega+1:T}\right)^2\right\}, \tag{5.3.3}$$

where

$$W_t\left(x_{1:h}\right) = \sqrt{\frac{t(h-t)}{h}}\left|\frac{1}{h-t}\sum_{k=t+1}^{h} x_k - \frac{1}{t}\sum_{k=1}^{t} x_k\right|$$

is the standard CUSUM transform in variate i for a change at time j. We note that this test statistic is equivalent to the likelihood ratio test for a multivariate change in mean under standard Gaussian noise. We therefore claim a change has occurred iff $S' > \beta$ for $\beta = (d+1)\log\omega$, with this choice of β arising from the Supplementary Materials of Tickle et al. (2018). We remark that other choices of S' based on combinations of $W_j\left(a_{i,T-\omega+1:T}\right)$ across $i = 1, \ldots, d$ are possible; see, for example, Groen et al. (2013), Cho and Fryzlewicz (2015), Enikeeva and Harchaoui (2019) and Wang and Samworth (2018).

If $S' > \beta$, then a changepoint is reported at

$$\xi^a = \arg\max_{T-\omega+1\leq t\leq T-1}\sum_{i=1}^{d} W_t\left(a_{i,T-\omega+1:T}\right)^2$$

if $\max_{T-\omega+1\leq t\leq T-1}\sum_{i=1}^{d} W_t \left(a_{i,T-\omega+1:T}\right)^2 > \beta$, and at

$$\xi^b = \arg \max_{T-\omega+1\leq t\leq T-1} \sum_{i=1}^{d} W_t \left(b_{i,T-\omega+1:T}\right)^2$$

otherwise. We then reset the entire procedure, beginning with the storing of the next ω time points for each variate in the stream. Once this is done, the empirical cumulative distribution functions are re-calculated.

If, however, $S^{'} \leq \beta$, or if the alarm for a change in a single variate using $S(.;.)$ is not raised for any i, then we do not report a changepoint. Instead, we collect the next time point $y_{1:d,(T+1)}$, and transform this using the pre-calculated empirical cdfs and Box-Muller. In the meantime, the system 'forgets' $y_{1:d,T-\omega}$, $a_{1:d,T-\omega}$ and $b_{1:d,T-\omega}$. Therefore, in the computation of the test statistics, only the most recent ω points are considered. This continues as long as the stream itself persists.

We summarise OMEN in Algorithm 4. Algorithm 5, on which the OMEN procedure detailed in Algorithm 4 has dependence, gives the pseudocode for the update step in which we compute the test statistic for each variate. Note that in Algorithm 5 we use $a_i := a_{i,1:g}$ for brevity.

Algorithm 4 OMEN.

Data: A multivariate dataset, $\mathbf{y}_{1:n}$, of dimension d; a penalty for introducing a multivariate change to the model, β; a penalty for a single variate to raise an alarm, $\beta^{'}$; an information collection/memory window, ω; the CUSUM transformation of a vector for a change at time k, $W_k(.)$.

Result: A sequence of decisions on a declaration of a change or otherwise for each time point at least ω after the most recent declaration of a changepoint.

Step 0: Set $r = 0$.

Step 1: Receive $\mathbf{y}_{(r+1):(r+\omega)}$. Then construct the following:

- For each variate, $i = 1, \ldots, d$, the empirical cumulative distribution function, $\hat{F}_i^{\omega}(.)$.
- The transformed data stream $\mathbf{z}_{(r+1):(r+\omega)}$, such that $z_{i,j} = \hat{F}_i^{\omega}(y_{i,j})$.
- $d\omega$ independent draws from a $U[0,1]$, $(u_{i,j})_{i=1,\ldots,d;j=(r+1),\ldots,(r+\omega)}$.
- The final streams for testing, $(a_{i,j})_{i=1,\ldots,d;j=(r+1),\ldots,(r+\omega)}$ and $(b_{i,j})_{i=1,\ldots,d;j=(r+1),\ldots,(r+\omega)}$, such that $a_{i,j} = \sqrt{-2\log u_{i,j}}\cos(2\pi z_{i,j})$ and $b_{i,j} = \sqrt{-2\log u_{i,j}}\sin(2\pi z_{i,j})$.

Step 2: Receive a new point, $\mathbf{y}_{r+\omega+1}$. Simulate $\mathbf{u}_{r+\omega+1}$. Compute $\mathbf{z}_{r+\omega+1}$, $\mathbf{a}_{r+\omega+1}$ and $\mathbf{b}_{r+\omega+1}$.
(I_{change}, τ) = Algorithm 5$\Big(\mathbf{a}_{(r+1):(r+\omega+1)}, \mathbf{b}_{(r+1):(r+\omega+1)}, \beta^{'}, r\Big)$.

if $I_{change} = 1$ **then**

  $\mathcal{W}^a = \max_{r+1\leq k\leq r+\omega} \sum_{i=1}^{d} W_k\left(a_{i,(r+1):(r+\omega+1)}\right)^2$

  $\xi^a = \arg\max_{r+1\leq k\leq r+\omega} \sum_{i=1}^{d} W_k\left(a_{i,(r+1):(r+\omega+1)}\right)^2$

  $\mathcal{W}^b = \max_{r+1\leq k\leq r+\omega} \sum_{i=1}^{d} W_k\left(b_{i,(r+1):(r+\omega+1)}\right)^2$

  $\xi^b = \arg\max_{r+1\leq k\leq r+\omega} \sum_{i=1}^{d} W_k\left(b_{i,(r+1):(r+\omega+1)}\right)^2$

  if $\mathcal{W}^a > \beta$ **then**

    Print ξ^a. Set $r = r + \omega$. Return to Step 1.

  if $\mathcal{W}^b > \beta$ **then**

    Print ξ^b. Set $r = r + \omega$. Return to Step 1.

  else

    $r = r + 1$. Return to Step 2.

  end

else

  $r = r + 1$. Return to Step 2.

end

Algorithm 5 Update step within OMEN.

Data: Two transformed streams, $(a_{i,j})_{i=1,\ldots,d;j=1,\ldots,g}$, $(b_{i,j})_{i=1,\ldots,d;j=1,\ldots g}$; a penalty incurred, $\beta^{'}$, for raising an alarm; the time point in the stream, r, which falls immediately before the beginning of the current memory window.

Result: An indicator, I, determining if there is sufficient evidence of a changepoint having occurred within the memory window; and a changepoint location, τ, which is returned as $NULL$ if $I = 0$.

Step 1: For $i = 1, \ldots, d$, take the sequences $(a_{i,j})_{j=1}^{g}$, and compute:

- cumulative sums, $(s_k^{a_i})_{k=1}^{g} := \left(\sum_{j=1}^{k} a_{i,j}\right)_{k=1}^{g}$;
- cumulative sums of squares, $(ss_k^{a_i})_{k=1}^{g} := \left(\sum_{j=1}^{k} a_{i,j}^2\right)_{k=1}^{g}$;
- test statistics for a change at k, for $k = 1, \ldots, g-1$,
  $S(k; a_i) = ss_g^{a_i} - ss_k^{a_i} + (g-k)\left\{\log(g-k) - 1 - \log\left(ss_g^{a_i} - ss_k^{a_i} - \frac{1}{g-k}\left(s_g^{a_i} - s_k^{a_i}\right)^2\right)\right\}$;
- overall test statistic for a change $S^a = \max_{1\leq i\leq d} \max_{1\leq k\leq g-1} S(k; a_i)$;

Step 2: **if** $S^a < \beta^{'}$ **then**
  $I = 0$
else
  Complete Step 1 for each of the sequences $(b_{i,j})_{j=1}^{g}$, for $i = 1, \ldots, d$.
  if $S^b < \beta^{'}$ **then**
    $I = 0$
  else
    $I = 1$
  end
end

5.3.2 Computational Considerations and Choices for ω and $\beta^{'}$

We now consider the computational burden of the OMEN method.

Trivially, the size of the storage required for the method prior to the calculation of the empirical cdfs is simply the size of the stream itself, meaning that ωd 'raw' data points are stored. At this point, we then require $\hat{F}_i^{\omega}(.)$ for each i, the computation and storage of which are $\mathcal{O}(d \log \omega)$ and $\mathcal{O}(d\omega)$ respectively. Computing and then storing the two transformed streams using Box-Muller is $\mathcal{O}(\omega d)$.

For each subsequent time point, the storage required does not increase, as each new transformed stream element replaces a 'forgotten' point from ω points earlier in the stream. This calculation of the new elements in the stream is $\mathcal{O}(d \log \omega)$, given that the computation of $\hat{F}_i^{\omega}(x)$ is $\mathcal{O}(\log \omega)$. Meanwhile, the computation of all the 'per stream' test statistics is $\mathcal{O}(\omega d)$ in the worst case. Therefore, the worst-case per-iteration cost of OMEN is $\mathcal{O}(d(\omega + \log \omega))$. Hence, from a computational standpoint, it is important to keep ω as small as possible. We therefore suggest setting ω to be the smallest possible value under which all 'usual' behaviours are observed. For example, in a time series with a weekly cycle (such as per hour water usage in a given building) we recommend setting ω to be the number of observations seen in a given week. Note that as the system completely refreshes following the detection of a change, ω can also be thought of as a 'minimum segment length'. This strengthens the case for our particular recommendation for ω. If such a time period of 'usual behaviour' is not clear from the context of the data, we recommend setting $\omega = 30$ as a baseline, with the justification for the choice arising from Haynes et al. (2017b).

We remark that, the computational considerations given above notwithstanding, it would also be possible to have an extending learning window size, rather than simply fixing this at the same length as the memory window, ω, or indeed another prescribed value. The primary advantage of allowing a variable learning window would be in allowing the OMEN procedure to be data-adaptive. For example, say at time T OMEN is beginning a new learning phase, having detected a set of changepoints $1 < \hat{\tau}_1 < \hat{\tau}_2 < ... < \hat{\tau}_{\hat{m}} < T$. If the changepoints are particularly 'close together' - for instance, if $f_{\hat{m}-1} := \min_{1 \leq h \leq \hat{m}-1} \frac{\hat{\tau}_{\hat{m}} - \hat{\tau}_{\hat{m}-h}}{h\omega} < 1$ - this could be an indication that the current learning window size has been set too low, and that OMEN is finding changepoints within the normal behaviours of the system. The learning size can then be increased appropriately as a function of $f_1, \ldots, f_{\hat{m}-1}$. Conversely, if the gap between successive changes is sufficiently large - for instance, if $g_{\hat{m}-1} := \max_{1 \leq h \leq \hat{m}-1} \frac{(\hat{\tau}_{\hat{m}} - \hat{\tau}_{\hat{m}-h})h}{(h-1)T} > 1$ - this could be an indication that the learning window can be decreased.

In practice, altering the length of the learning window works best for systems where it is suspected *a priori* that the frequency of changes may alter over time. (For example, hourly financial returns in a bull market versus a bear market.) In general, work on considering an adapting learning window in the changepoint literature remains sparse, with existing approaches typically being very bespoke. Although, see, for example, Poddar et al. (2016). For OMEN, under the assumption that a fixed learning window is sufficient to capture all normal behaviours of the system, we recommend taking a single value for the length of the learning window throughout. However, exploring an adaptive learning window, particularly for systems where there may be missing data, is a very interesting open problem.

We additionally remark that the worst-case per-iteration cost is also linear in d. For most examples this may not be an issue, particularly if parallel computation of the test statistics is an option. However, for an arbitrary data stream with large d, an alternative approach could be to combine the transformed variates in the stream. For a very dense changepoint, this could be done by sampling from the variates uniformly at random. In this way, the per-iteration computational cost does not increase linearly with d.

There remains the choice of β'. We recommend setting this based on a false alarm error rate. Let N be the period of time over which we wish to control the false alarm error rate. This can be set to n, the total number of time points which will be observed, if this is known (and not too large). We are then in a position to state the following result.

Lemma 5.3.1. *Define Λ such that the single variate penalty in OMEN is $\beta' = 2\log\Lambda$, and let ω be the information window/minimum segment length of the procedure. Then, under a stream of dimension d in which no change occurs for the first N time points, the probability of the single variate test statistic being violated for, without loss of generality, series $\boldsymbol{a}$, is at most*

$$\frac{d\,(N-\omega)\,(\omega-2)}{V^{\left(\frac{1}{2}-\sqrt{\frac{2}{\beta}}\right)}}e.$$

Proof: See Section C.1.

From the final probability value given in Lemma 5.3.1, we see that, for instance, setting $\Lambda = \omega^2 N^3 d^3$ (corresponding to $\beta' = 4\log\omega + 6\log N + 6\log d$) gives that the probability that series **a** or **b** will incorrectly flag a change is less than $C/\sqrt{Nd}$, for some absolute constant C. Therefore, the probability that the multivariate test statistic is erroneously called is $\mathcal{O}(1/Nd)$.

Note that the proof of Lemma 5.3.1 exploits the sub-Gaussianity of the data following the Box-Muller transform by using a bound from Fisch et al. (2019a). This bound is in turn derived from a Chernoff-type approach. (See, for example, Chapter 2 of Boucheron et al. (2013).) Therefore, Lemma 5.3.1 is applicable in a finite-sample setting.

We remark that the choice of ω also greatly affects the probability of detecting a changepoint, if present. Clearly, this is also heavily dependent on the nature of the change itself. We illustrate this point through the example of the change in mean problem under Gaussian noise with constant variance 1. Suppose the change is of size Δ. If Δ is sufficiently small as to have a low or negligible impact on the ranks of the values in the stream relative to the values of the stream in the learning window, then the power of OMEN will be low. For example, suppose $\Delta < \Phi^{-1}\left(\frac{1}{2} + \frac{1}{\omega}\right)$. In this setting, the ranks obtained will be altered by at most one from what they would have been under no change. Given the naturally 'smoothing' effect of our proposed application of the Box-Muller transform, this makes it very difficult for any slight change in the ranks to be apparent in the transformed stream. Indeed, simply to provide a guarantee that the vast majority of ranks are altered by at least one, we require that $\Delta > \Phi^{-1}\left(\frac{2}{\omega}\right) - \Phi^{-1}\left(\frac{1}{\omega}\right)$. In this setting, the transformed stream would be very slightly less likely to exhibit certain small ranges of values than before the changepoint. In practical terms, we would require these ranges to be sufficiently large to have a measurable effect on the resulting stream to ensure timely detection. For example, it is comparatively fairly likely that OMEN will detect a change in mean if $\Delta > \Phi^{-1}\left(\frac{6}{\omega}\right) - \Phi^{-1}\left(\frac{1}{\omega}\right)$, which roughly corresponds to $\Delta > 0.99$ in the case when $\omega = 30$.

Outside of the Gaussian change in mean setting, advisable values of ω for sufficient

detectability become even more difficult to determine. However, we remark that, if the changepoint does not change first or second moment behaviour, then the power of OMEN will remain low almost independently of the choice of ω, as we demonstrate empirically in Section 5.4.

5.3.3 A Comparison with the approach of Chen (2019b)

A very recent method, presented by Chen (2019b), shares some similarities with our proposed approach. Like OMEN, this method is applicable in streaming data contexts under multiple variates, while being nonparametric. Additionally, the method can be applied to non-Euclidean forms of data (e.g. networks). We refer to the method as gstream herein, as per its naming in the corresponding package of Chen and Chu (2019).

In a similar fashion to OMEN, gstream proceeds by taking in a period of 'historical observations' (of length N_0) analogous to the learning window of OMEN. Importantly, however, unlike the learning window, it is explicitly assumed that no changepoint takes place within the historical observations. Following the observation of the historical period, gstream then computes a test statistic for a changepoint having occurred at some point in the recent past. This is constructed by computing the (non-symmetric) matrix A_J^k of indicators on the most recent J observations, such that $A_{J,ij}^k = 1$ if observation $\mathbf{y}_{T-J+j}$ is one of the k nearest neighbours (with respect to some norm, $||.||$) of $\mathbf{y}_{T-J+i}$ among observations $\mathbf{y}_{T-J+1}, \ldots, \mathbf{y}_T$. This matrix is then added to its transpose. Meanwhile, another matrix of indicators is constructed such that an entry is 1 if, following a random permutation of the indices, say $\mathbf{P}(.)$, either $\mathbf{P}(T-J+j) \leq t < \mathbf{P}(T-J+i)$ or $\mathbf{P}(T-J+i) \leq t < \mathbf{P}(T-J+j)$ such that t is the point at which we wish to test for a change. We remark that the censoring of memory in considering only the J most recent points invokes the memory window used within the OMEN procedure.

The result is then normalised by its mean and standard deviation, to give a test statistic which is standard normal under the assumption of no change, exactly as for our method. If there is a changepoint, the test statistic will become large for values

of t close to the true change, τ. This further processing of the data to produce a test statistic in this way recalls the use of the Box-Muller transform in the OMEN procedure. Note that our transform also serves the purpose of controlling the rate of false positives, as demonstrated in Lemma 5.3.1.

As a stopping rule, Chen (2019b) recommends computing the test statistic for $T - n_1 \leq t \leq T - n_0$, and then declaring a change at the point in this window which maximises the test statistic, the first instance at which any time point in the window gives a test statistic value above some threshold.

Unlike OMEN, in which we would typically advise tuning only the ω parameter, gstream involves a number of selections, namely choices for N_0, J, n_0, n_1 and, most importantly, k and $||.||$. Chen (2019b) notes that the selection of k in particular greatly affects the performance of gstream, with an appropriate choice for this value being sensitive to the choice of L, as well as the number of dimensions. We take the suggested values in all simulations in the next section.

We additionally remark that, while gstream is clearly applicable to a wider range of applications than OMEN, there is no natural framework for determining the nature of an affected set at each change, if we have a classical multivariate stream. We note that this is relatively simple with OMEN: for example, we can replace the overall test statistic given by (5.3.3) with the SUBSET test statistic of Chapter 4.

5.4 Simulations

We here examine seven scenarios for the generating processes. Each of these shall extend for $n = 1000$ time points for a differing number of variates and proportion of variates which undergo a change (where appropriate).

For the first and second scenarios, which we label as T_1 and T_2 respectively, we have no changes within the system. The generating processes for variate i at time j in these examples are $T_1^{i,j} \sim N(i, 1)$, and $T_2^{i,j} \sim \text{Neg-Bin}\left(r = 2, \theta = \left[(i \mod 5) + 1\right]/6\right) + N(0, 10^{-6})$ respectively. (Note that the second scenario is not a pure negative binomial, as a constant stream of 0s can cause OMEN to fault.)

For the third scenario, T_3, we impose a single change at time $t = 600$. If variate i undergoes a change, the generating processes are $T_{3,1}^{i,j} \sim N(0,1)$ independently for $i = 1, \ldots, d$, $j = 1, \ldots, 600$ and $T_{3,2}^{i,j} \sim N(3,1)$ for $i = 1, \ldots, d$, $j = 601, \ldots, 1000$. If variate i does not experience a change, then $T_{3,1}^{i,j}$ is followed for $j = 1, \ldots, 1000$.

For the fourth through to the seventh scenarios, T_4, T_5, T_6 and T_7, we impose up to three changepoints per variate in increasingly more 'challenging' situations. The fourth and fifth scenarios again feature the normal and negative binomial distributions. If a variate i undergoes all three changepoints, at times $t = 300, 600$ and 900 respectively, then

$$
\begin{aligned}
T_{4,1}^{i,j} &\sim N(3,1) & T_{5,1}^{i,j} &\sim \text{Neg-Bin}\,(2, 0.05) + N(0, 10^{-6}) \\
T_{4,2}^{i,j} &\sim N(0,1) & T_{5,2}^{i,j} &\sim \text{Neg-Bin}\,(2, 0.4) + N(0, 10^{-6}) \\
T_{4,3}^{i,j} &\sim N(2,2) & T_{5,3}^{i,j} &\sim \text{Neg-Bin}\,(7, 0.4) + N(0, 10^{-6}) \\
T_{4,4}^{i,j} &\sim N(5,4) & T_{5,4}^{i,j} &\sim \text{Neg-Bin}\,(4, 0.9) + N(0, 10^{-6}).
\end{aligned}
$$

The sixth and seventh scenarios both feature more challenging sets of changes. Here we have

$$
\begin{aligned}
T_{6,1}^{i,j} &\sim \text{Pareto}\,(x_m = 1, \alpha = 0.5) & T_{7,1}^{i,j} &\sim \text{Exp}\left(\sqrt{12}\right) \\
T_{6,2}^{i,j} &\sim \text{Pareto}\,(3, 2) & T_{7,2}^{i,j} &\sim N\left(\frac{1}{\sqrt{12}}, \frac{1}{12}\right) \\
T_{6,3}^{i,j} &\sim \text{Pareto}\,(5, 0.75) & T_{7,3}^{i,j} &\sim U\left(\frac{-3+\sqrt{3}}{6}, \frac{3+\sqrt{3}}{6}\right) \\
T_{6,4}^{i,j} &\sim \text{Pareto}\,(7, 3) & T_{7,4}^{i,j} &\sim \frac{-3+\sqrt{3}}{6} + \text{Beta}\left(\frac{1}{2}, \frac{1}{2}\right).
\end{aligned}
$$

Note that in the sixth scenario, the mean is not bounded for $T_{6,1}$ or $T_{6,3}$, with the variance being unbounded for $T_{6,1}, T_{6,2}$ and $T_{6,3}$. Meanwhile, in the seventh scenario, neither the mean nor variance changes at any of the changepoints, except between $T_{7,3}$ and $T_{7,4}$ where, although the mean does not change, the variance is altered from 1/12 to 1/8 (this is in order to preserve the support between the two regimes).

We remark that for the fourth through to the seventh scenarios, if a variate does

not undergo one of the changepoints in question, then the 'current' generating function for variate i carries until a change is experienced. Therefore, if a variate i experiences, say, just the change at $t = 900$, then for this variate for $j = 1, \ldots, 900$, the data are generated according to the first regime, and then for $j = 901, \ldots, 1000$ according to the fourth regime.

Several of these scenarios feature more challenging changes than can typically be handled by, for example, CUSUM-based methods or those reliant on a parametric cost function. The typical approach to heavy-tailed data in a penalty-based setting is to increase the penalty incurred for flagging a changepoint; see, for example, Jeon et al. (2016), Knoblauch et al. (2018), Zoubir and Brcich (2002) and others.

Throughout this study, we keep $\left(\beta, \beta'\right) = ((d+1)\log n, 4\log\omega + 6\log n + 6\log d)$. We additionally set $\omega = 30$ for OMEN. We compare the performance with gstream (see Section 5.3.3) and Inspect (Wang and Samworth, 2018), the latter of which is an offline multivariate approach for which code can be found in the `InspectChangepoint` package (Wang and Samworth, 2016). Note that we used the default parameter values for the Inspect procedure as given in the package. Note also that Inspect is designed for the Gaussian setting, and in general is not robust to non-Gaussian noise. We therefore only include it for comparison in scenarios 1, 3 and 4. For gstream, we used the code from the package `gStream` (Chen and Chu, 2019). We set many of the parameters according to recommendations within Chen (2019b). In particular, we took $L = 10$, $N_0 = 10$, $k = 3$, $n_0 = 2$ and $n_1 = 8$. For the other inputs, we set the Average Run Length (ARL) to be the length of the series (1000), and the probability of an 'early stop', `alpha`, to be 0.05. In addition, we used the weighted test statistic out of the choice of four available in the package. Finally, we computed the Euclidean distance between the $d-$dimensional points as the distance norm.

We examine two sizes of the affected sets of the changes within each scenario, and look at three cases for the total number of variates. The metrics of interest were taken as the false alarm error rate, the number of missed changes and the average location error of the change. Note that a false alarm is hereby defined as an estimated changepoint falling at least ω temporal points away from the closest change

(or within said tolerance of the true changepoint if another estimated changepoint is closer to the true changepoint). In addition, a missed change is hereby defined as a true changepoint for which no estimated change was fitted within ω temporal points. Finally, the average location error of the change is the number time points of separation between each true change and associated estimated changepoint.

All simulations were run in R using a Linux OS on a 2.3GHz Intel Xeon CPU. Under each scenario, number of variates and size of affected set, we perform 200 repetitions and report the average of the aforementioned metrics.

Note that in Tables 5.1-5.3, we denote a change which affects 100% of the variates as D, and a change which affects a 'middling' number of variates as M. Therefore, for example, (D, D, D) denotes that the changes at $t = 300, 600$ and 900 each affect 100% of the variates. Note that here a 'middling' change here affects 3 variates in the 5 variate setting, 5 variates in the 10 variate setting and 50 in the 100 variate setting.

Table 5.1 examines the average number of false alarms triggered by each of the three methods in each of the seven scenarios. We note that in almost all situations, OMEN reports the fewest false alarms (with a very low false alarm error rate in all scenarios except 4 and 5).

Table 5.2 gives the average number of missed changes in each of the scenarios. Again, the performance of OMEN was encouraging, with most changes detected in most of the scenarios. However, it did perform poorly in scenarios 6 and 7. Note that the difficulty for the OMEN method in detecting changes in scenarios 6 and 7 may be due to the fact that these situations do not exhibit conventional mean or variance changes. As our test statistics for detecting changepoints within the window are, fundamentally, checking for a change from a Gaussian to another sub-Gaussian random variable, then it is clear that OMEN will be most powerful in detecting mean and variance changes. We remark that gstream generally outperforms OMEN in terms of the number of missed changes, however this should be viewed in the context of the false alarm results, which indicate that gstream is much more likely than OMEN to overfit. Additionally, gstream also has some difficulty detecting the changepoints in the more difficult scenarios, in particular scenario 7. This may be due to the chosen

distance metric only allowing for significant detection of changepoints under a change in first or second moment behaviour.

Table 5.3 gives the average location error for the estimated changepoints which were not previously labelled as false alarms. That is, the distance from the corresponding true change of any estimated change which was the closest of all estimated changes to this true change while being within at most ω time points. We again note that, while the performance of Inspect is impressive, given the number of false alarms a small location error is very much to be expected. It is clear that OMEN's accuracy improves with the number of variates present, which is unsurprising. However, at first glance, the lower location error for the (M, M, M) cases relative to the corresponding (D, D, D) regimes is more curious. By comparing Table 5.3 to Table 5.2, we see that this can be explained by the fact that OMEN estimates fewer changes in the scenarios where the change occurs in fewer of the variates. In short, OMEN can be described as a parsimonious method. However, the method nevertheless raises an alarm quickly in those situations where the change is more drastic.

In Section C.2, we give a comparison of OMEN's performance under each of these metrics in these scenarios for different values of ω.

As a direct illustrative comparison between OMEN and gstream, we show the results of applying both methods to the five variate setting in scenarios 3, 4, 5 and 7. In each case, we set all of the variates to change at each of the changepoints and display only the first variate for greater clarity. We then ran both of the methods once, using the same inputs as in the simulation study for this section, to get an idea of a typical segmentation given by the two procedures. The results are shown in Figure 5.1.

5.5 Real Data Example - Wind Speeds

We examine hourly wind speed data, measured to the nearest m/s, across 3 Canadian and 2 Israeli cities from 1am on 1 October 2012 to 12am on 28 October 2017, for a total

Average False Alarms	Method					
	5 Variates		10 Variates		100 Variates	
Scenario, Method	(D, D, D)	(M, M, M)	(D, D, D)	(M, M, M)	(D, D, D)	(M, M, M)
1, OMEN	**0.01**	**0.01**	**0.00**	**0.00**	**0.00**	**0.00**
1, Inspect	**0.01**	**0.01**	0.02	0.02	0.02	0.02
1, gstream	5.41	5.41	4.70	4.70	4.33	4.33
2, OMEN	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**
2, gstream	8.33	8.33	6.80	6.80	5.46	5.46
3, OMEN	**0.01**	0.10	**0.00**	0.10	**0.00**	**0.00**
3, Inspect	0.02	**0.03**	0.02	**0.03**	0.02	0.02
3, gstream	9.07	7.91	8.77	8.38	8.69	8.34
4, OMEN	**0.44**	**0.32**	**0.19**	**0.17**	**0.00**	**0.00**
4, Inspect	2.83	0.96	8.89	1.41	77.8	55.3
4, gstream	11.0	8.60	13.6	10.1	17.0	16.5
5, OMEN	**0.37**	**0.36**	**0.18**	**0.14**	**0.00**	**0.00**
5, gstream	12.4	7.67	14.0	5.43	18.8	8.02
6, OMEN	**0.02**	**0.02**	**0.01**	**0.01**	**0.00**	**0.00**
6, gstream	13.2	8.89	13.3	5.32	14.9	5.16
7, OMEN	**0.01**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**
7, gstream	4.33	4.88	3.71	3.92	3.63	4.88

Table 5.1: The average number of false alarms incurred by OMEN, Inspect and gstream under each of the scenarios. Bold entries show the best performing algorithm. 200 repetitions were simulated in each case.

Average Num Missed	Method					
	5 Variates		10 Variates		100 Variates	
Scenario, Method	(D, D, D)	(M, M, M)	(D, D, D)	(M, M, M)	(D, D, D)	(M, M, M)
3, OMEN	0.01	0.12	**0.00**	0.13	**0.00**	**0.00**
3, Inspect	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**
3, gstream	0.01	0.04	**0.00**	0.01	**0.00**	0.01
4, OMEN	1.15	1.27	1.05	1.26	1.00	1.14
4, Inspect	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**
4, gstream	0.43	0.88	0.08	0.50	**0.00**	**0.00**
5, OMEN	1.11	1.78	0.92	**1.73**	0.34	1.50
5, gstream	**0.15**	**1.64**	**0.03**	1.88	**0.00**	**1.25**
6, OMEN	2.95	2.96	2.93	2.96	2.77	2.93
6, gstream	**0.69**	**1.34**	**0.69**	**1.99**	**0.71**	**2.29**
7, OMEN	2.98	2.99	3.00	3.00	3.00	3.00
7, gstream	**2.43**	**2.31**	**2.42**	**2.40**	**2.51**	**2.24**

Table 5.2: The average number of changes missed by OMEN, Inspect and gstream under each of the scenarios. Bold entries show the best performing algorithm. 200 repetitions were simulated in each case.

Average Location Error	Method					
	5 Variates		10 Variates		100 Variates	
Scenario/Method	(D, D, D)	(M, M, M)	(D, D, D)	(M, M, M)	(D, D, D)	(M, M, M)
3, OMEN	8.22	6.37	7.61	3.59	3.90	1.28
3, Inspect	**0.00**	**0.01**	**0.00**	**0.00**	**0.00**	**0.00**
3, gstream	3.06	3.66	2.92	3.08	2.83	2.90
4, OMEN	8.68	7.61	7.65	4.86	2.34	0.98
4, Inspect	**0.05**	**0.15**	**0.01**	**0.05**	**0.00**	**0.00**
4, gstream	4.05	4.81	3.16	3.89	2.89	2.84
5, OMEN	7.71	**5.75**	5.92	**3.25**	**2.79**	**0.71**
5, gstream	**3.90**	10.1	**3.42**	9.86	2.80	8.24
6, OMEN	5.60	**4.50**	**4.07**	**3.44**	**2.93**	**1.00**
6, gstream	**4.29**	8.26	4.15	12.1	4.12	13.4
7, OMEN	**11.7**	20.0	**0.00**	21.0	**0.00**	**0.00**
7, gstream	15.6	**13.8**	15.1	**14.0**	14.1	13.6

Table 5.3: The average location error of the OMEN, Inspect and gstream under each of the scenarios. Bold entries show the best performing algorithm. 200 repetitions were simulated in each case.

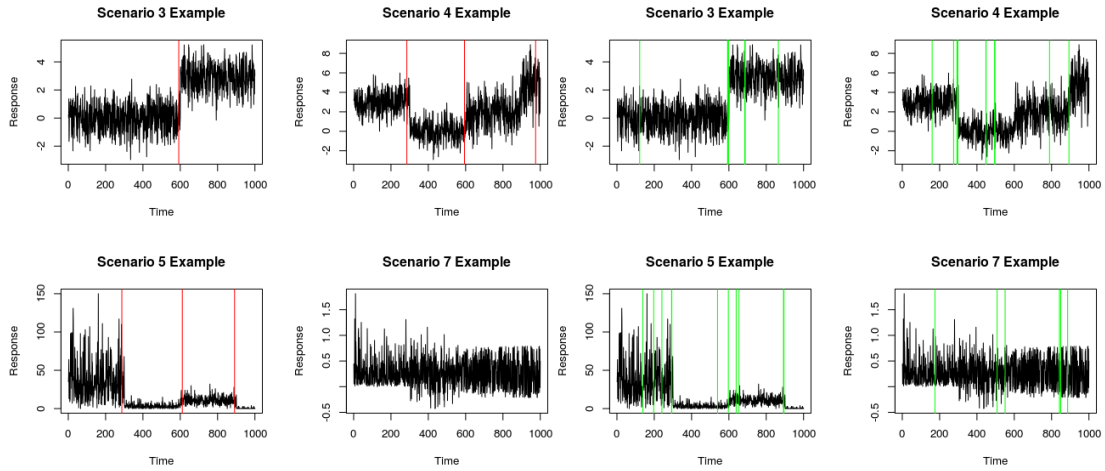

Figure 5.1: Results from a single run of OMEN (four leftmost plots) and gstream (four rightmost plots), on each of scenarios 3, 4, 5 and 7. Changes found by OMEN are overlaid as red vertical lines. Changes found by gstream are overlaid as green vertical lines. In each case, the total number of variates was 5 and the change affected all variates.

of 44460 observations for each city. These data can be found on Kaggle (Beniaguev, 2017). Specifically, our interest lies in the wind speed records from each of these cities. We discuss the application of the OMEN method to another dataset in Section C.3.

5.5.1 Canada

We examine wind speed recordings from Montreal, Toronto and Vancouver using the OMEN procedure with the same standard settings for β and β' as in Section 5.4. The value of the learning window was set at $\omega = 148$, roughly corresponding to the number of observations made over six days. The plotted series are shown for these three cities in Figure 5.2, with the changes found by OMEN overlaid.

As can be seen from Figure 5.2, OMEN places relatively few changes in the series, with the exception of 2014-15. In this particular period, the wind activity notably fell out of step in all three cities with other years. For example, in Toronto, a hint of a seasonal effect can be observed, with greater wind speeds more likely from late autumn to early spring. This is not seen in the winter of 2014-15. Most other changes found by OMEN seem to detect some aspect of this seasonal effect, with an average of two changepoints per year around the turn of each year. We remark that OMEN also seems detects the culmination of the period in which there was a particularly

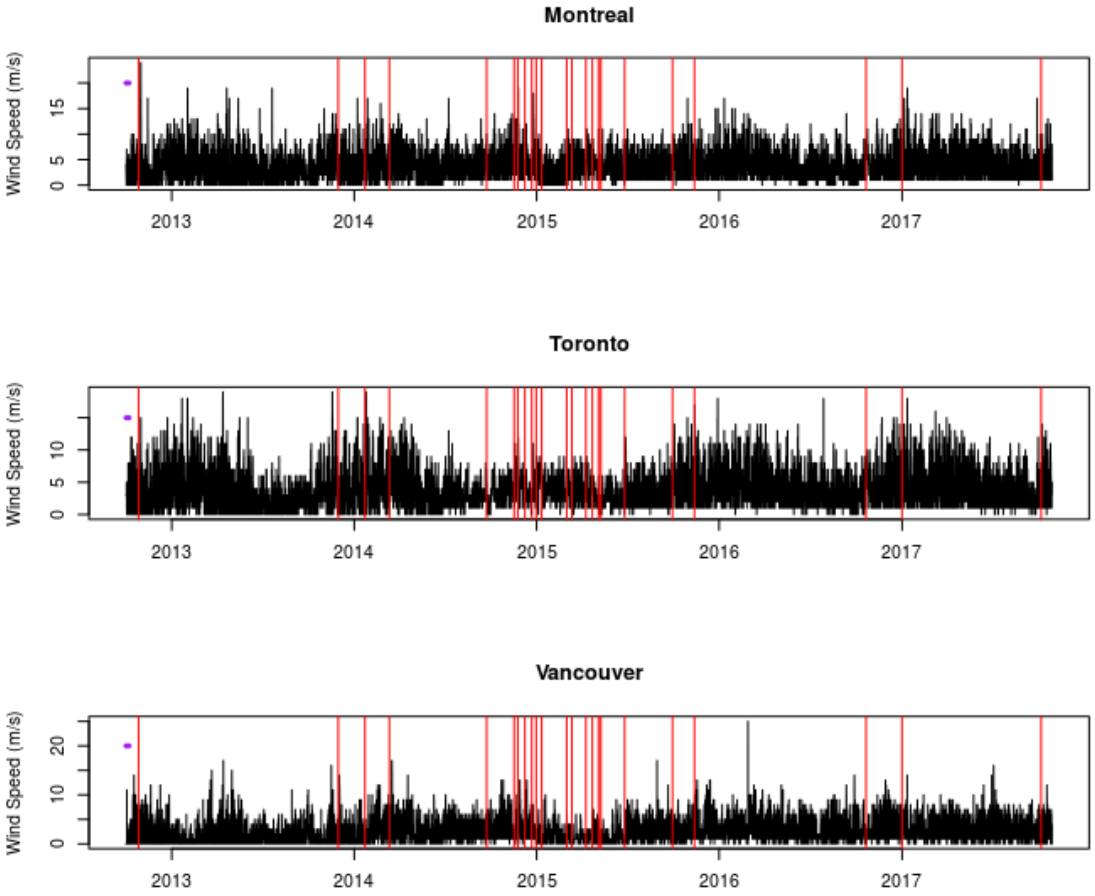

Figure 5.2: Hourly Wind Speeds - to the nearest (m/s) - in three Canadian cities from October 2012 to October 2017. Changes found by the OMEN method, with a minimum segment length corresponding to the number of observations made in one week, are also shown as red vertical lines. The span of the original learning window is indicated by the horizontal purple line on each plot.

notable number of hours in which a speed of zero was recorded. (This approximately corresponds to the beginning of 2014.)

5.5.2 Israel

We examine wind speed recordings from Eilat and Tel Aviv District. Note that, while the original dataset had series for six cities in Israel, four of these contained imputed data, so they are ignored here. The plotted series are shown for these two cities in Figure 5.3, with the changes found by OMEN overlaid. As for the Canada series, we used the usual settings for the penalty values and took $\omega = 148$.

Very few changes are detected by OMEN throughout the period of interest, which

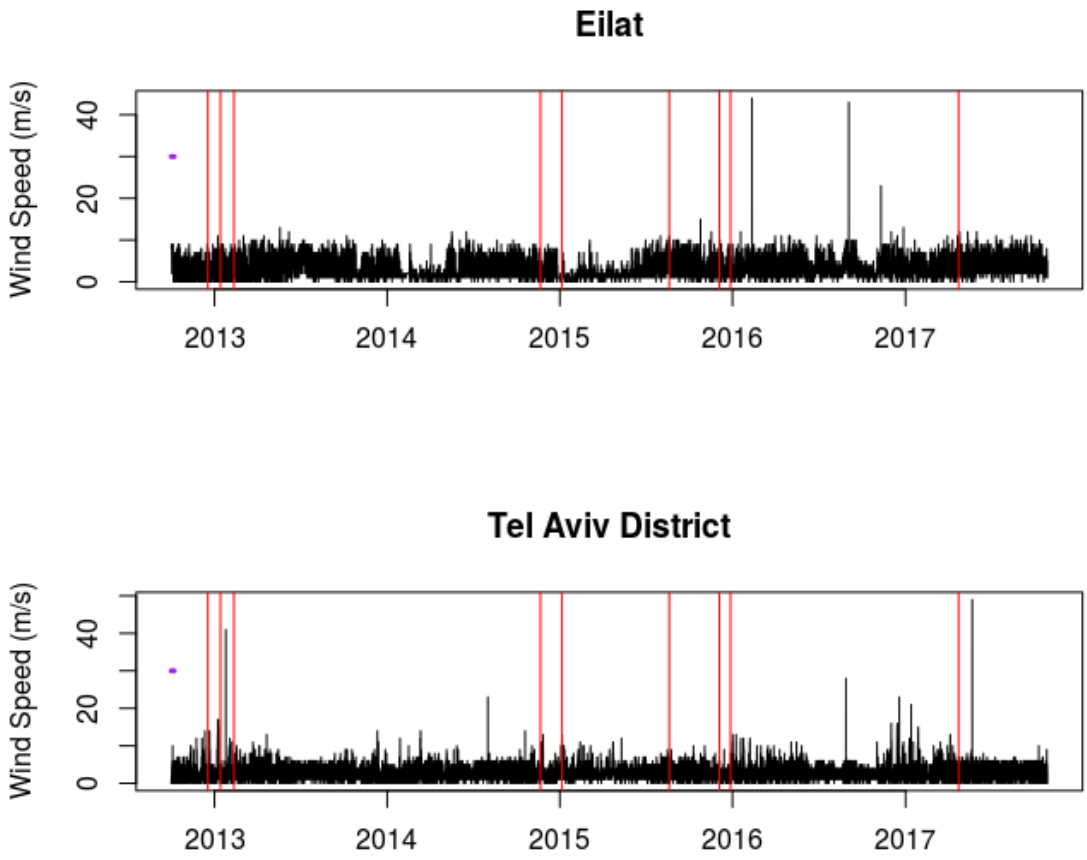

Figure 5.3: Hourly Wind Speeds - to the nearest (m/s) - in two Israeli cities from October 2012 to October 2017. Changes found by the OMEN method, with a minimum segment length corresponding to the number of observations made in one week, are also shown as red vertical lines. The span of the original learning window is indicated by the horizontal purple line on each plot.

appears to be unsurprising. Some of the changes which are found seem to correspond to points after which the chance of an 'unusually high' observation becomes more likely. For example, the change roughly corresponding to the beginning of 2016 is followed by a period of around one year in which wind speeds of 10 or more become more likely (with the change in 2017 seemingly marking the end of this period). We note that the method appears to be robust to the presence of single anomalous results. In particular, only one changepoint (at the beginning of 2013) is seemingly flagged due to the appearance of one exceptional observation.

5.5.3 Sensitivity of OMEN to ω

We now briefly comment on the behaviour of OMEN under alternative choices of ω for these real datasets. We firstly remark that for larger values of ω (but still keeping $\omega << 44460$), a very similar change profile emerges. Indeed, the only notable difference in the performance of OMEN is a tendency to place slightly fewer estimated changes, with this becoming more pronounced for larger values of ω. The opposite is true when ω is decreased, and for significantly smaller values (say, $\omega = 30$, as for the central comparison in the simulation study in Section 5.4), OMEN degenerates completely, giving a changepoint almost at every $k\omega$, for $k = 1, 2, \ldots$. This suggests that the dataset exhibits a degree of non-stationarity which can be overcome somewhat by picking a suitably high value for ω, with the trade-off that more true changes are likely to be missed.

5.6 Discussion

We have introduced OMEN, a new online, nonparametric means of detecting changepoints in the multivariate setting. This approach is inspired by empirical quantile estimation, as well as by existing cost function methods for resolving the offline changepoint problem. We have shown OMEN performs well from the perspective of the false alarm rate across a variety of circumstances. In addition, our simulation study in Section 5.4 also suggests that, for those situations where there is a change in mean or variance (even if this is the result of another parameter or distribution change), OMEN will still likely locate the change in question. However, the performance for changepoints which do not exhibit a mean or variance change is much less impressive. It would, therefore, be interesting to try to extend our method to ideas in uniform quantile estimation. This could be done by building on the results of, for example, Polonik (1997), Lei et al. (2013), Lei et al. (2018) and many others. The central question under such an alternative formulation would concern the number of empirical quantiles to track and the accuracy in estimating said quantiles. These are important considerations, informing the convergence rate and hence a suitable

setting for the length of the learning window.

On the subject of the learning window, it would clearly be of interest to link the power of the OMEN procedure as it stands to the length of this window. In this way, an idea of the likely effects which could be missed can be gleaned. For instance, in the wind speed setting (see Section 5.4), it is not immediately clear that all pertinent behaviours are seen, such as the period of suspiciously low wind speeds in Eilat in early 2014. Indeed, this is potentially symptomatic of the wider inability of OMEN to consider potential subsets of variates which could alter at the changepoint. At present, the method simply labels all or no variates as having altered. This is another important consideration for making OMEN more suitable for a data streaming setting.

Chapter 6

Conclusions

In the previous chapters, we have introduced novel techniques for handling time series with changepoints in different data intensive settings. While there remain several shortcomings of these new procedures, as well related wider questions of interest, we first summarise the important new material of this work.

6.1 Key Findings

In Chapter 3, we demonstrated the empirical and asymptotic properties of Chunk and Deal, two means of parallelising exact dynamic programming methods for changepoint detection in the univariate setting. Our consistency results for Chunk and Deal rely on assumptions for the number of cores used, $L(n)$, for a given length of sequence, n. Note that these two results rely on a new consistency result for unparallelised changepoint detection in the cost function setting. While all of this theory is strictly applicable only in the asymptotic setting, other recent work such as Wang et al. (2019a) has successfully derived finite-sample results in the unparallelised setting.

The most important new theoretical contribution of Chapter 3 concerns the worst-case computational cost of Chunk and Deal. This shows that, if the setting up of the parallel environment is not the computational bottleneck of the procedure, then, under particular choices of the number of cores, the worst-case computational cost is linear in the length of the sequence. This is an improvement over previous

techniques which minimise a segment cost function, where typically the worst-case computational cost is quadratic in the length of the sequence. While all of these results have here only been established for the Gaussian change in mean setting, in instances where a sequence may consist of many observations - for example, in the BT setting, with an observation recorded every minute - Chunk and Deal provide important new means of giving accuracy and speed for any changepoint problem where the segment cost function is given.

In Chapter 4, we introduced SUBSET, a multivariate changepoint detection technique using both Binary Segmentation and cost function approaches to locate changepoints in sparse and dense settings. As well as showing the good empirical performance of SUBSET on synthetic data, we established its strong finite-sample theoretical performance in the single Gaussian change in mean setting. These results highlight the negligible false alarm error rate of SUBSET, while showing that the conditions required for high-probability detectability of a change are weak in both the sparse and dense cases.

One critical advantage of SUBSET over alternative approaches is that, subject to choosing an appropriate cost function, it can be applied to many different model settings. These of course include the terrorism database example, as well as nonparametric change detection, where for the latter we may use cost functions of the type suggested in the ED-PELT method of Haynes et al. (2017b).

SUBSET uses a very recent Wild Binary Segmentation idea (Fryzlewicz, 2019), which allows it to be data-driven when detecting multiple changepoints. In particular, while the masking problems of classical Binary Segmentation are removed, the heavier computational overhead of Wild Binary Segmentation (Fryzlewicz, 2014) is avoided, except in situations where the system has many changepoints. Additionally, the computationally light post-processing step allows SUBSET to consider each variate separately, in order to, for example, overcome any issues of dependency between the different variates in the dataset.

In Chapter 5, we introduced OMEN, an online, multivariate, nonparametric means of detecting changepoints. OMEN uses several classical techniques to transform each

variate of the stream into two sub-Gaussian sequences. Importantly, these sequences are independent and approximately standard normal under the assumption of no changepoint. We use this fact to derive a two-stage test statistic for a changepoint. A novel result given in Chapter 5 shows that the false alarm rate using this transform and test statistic is low, regardless of the original distribution of the data. We demonstrated this on several synthetic examples, in which we also observed that the detection ability of the methods was at least comparable to other state of the art procedures in the multivariate setting.

Several open challenges remain from the work conducted. In addition, there are several highly prescient potential directions in which each of the new methods can be taken. We discuss some of these ideas in more detail in Section 6.3. We first revisit the BT example introduced in Chapter 1, and canvass the applicability of the new methods to that particular setting.

6.2 A Return to the BT Example

Recall from Chapter 1 the BT setting of interest, in which the performances of individual ports within Edge Routers - the components of the access layer of broadband networks - are measured per minute according to a number of metrics. For a typical Edge Router, this gives a data stream comprised of around one thousand variates, however this number may vary depending on the number of customers the Edge Router is assigned to serve. Given the number of customers served by the broadband network in the UK, the number of variates across the entire access layer of Edge Routers is therefore many millions.

None of the methods we have presented here have anything remotely close to the required capability to simultaneously analyse the entire access layer in an online fashion. Indeed, it is arguable that any such method currently exists, despite the recent surge of interest in Big Data. However, each of the methods we have introduced here may still be used to give valuable insights for reduced variants of the problem.

For Chunk and Deal, while we have not proposed any means of parallelising

techniques for multivariate changepoint detection, the established worst-case computational cost and location accuracy indicate that either method is a good candidate for verifying the results given by other methods. In particular, for those instances where a Major Service Outage (MSO) is seen to affect a single port by any overarching detection method, it would be useful to employ a reliable but efficient single variate method such as Chunk or Deal to verify the presence of a change. This is particularly true in those cases where a fault may not have been reported by a customer, given the cost of incorrectly sending an engineer to fix a very specific problem.

The benefits of SUBSET within an analysis of Edge Router data are even more apparent. While the method is not online, it could potentially be applied on a rolling basis (e.g. once a day), to analyse the output of a single Edge Router, where the number of variates is much more manageable. Certain additional problems would need to be rectified, however. Most notably, data returned from each router would exhibit strong daily, weekly and seasonal patterns. SUBSET would struggle in such situations given the requirement of a segment cost function, which assumes stationary data. Therefore, a pre-processing of the data to remove these normal behaviours would be needed. As the UK moves towards the 5G era, the assumptions made by a de-seasonalising procedure may not hold indefinitely, potentially requiring repeated manual interference as infrastructure is updated.

The problem of seasonality is potentially less of an issue with OMEN, providing a sensible learning window is chosen in advance. (Although such a window would be unlikely to take into account 'one-off' events such as public holidays or major sporting events in which behaviours may change markedly.) Under OMEN, providing the data remain within the pattern of normal behaviours, even if these behaviours are technically non-stationary, no change will be detected. Conversely, if something drastically affects components of the router in such a way that a fast response is required (for example, an MSO that affects an entire router or a distributed denial-of-service (DDOS) cyber-attack), OMEN would be likely to raise a timely alarm, under the assumption of behaviour well outside the normal. The issue with

using OMEN under non-stationary data are that changes which, for instance, affect the shape of the data, but have little impact on the first or second moment, would have a lower chance of being flagged.

6.3 Open Challenges and Future Directions

We now turn to some of the broader challenges arising from our findings.

We first discuss some of the questions surrounding our proposed parallelisation methods in the univariate, offline setting. As noted in Section 6.1, our theoretical results for Chunk and Deal apply only asymptotically. In the wake of recent finite sample results on the consistency of unparallelised changepoint detection, it would be interesting to derive a new result in the finite n setting for both Chunk and Deal. Moreover, new theory on the detection probability and accuracy of the methods may help to relax the assumptions we imposed in Chapter 3 on the number of cores, $L(n)$, for increasing n. For example, it would be interesting to explore the requirements for the detection of all changes in a sequence for an arbitrary fixed, or at least bounded, L. This would be of particular importance in the Deal setting, not only because our assumptions on the number of cores were stricter, but also in determining the likely output of a single core in the small L setting. If this output is sufficiently close to a true segmentation in cases without extremely short segments, then using a variant of Chunk where, following the split phase, only one core fits a changepoint model, could give an extremely impressive computational cost with little loss of accuracy.

We additionally remark that the splitting methods for Chunk and Deal are somewhat naive. In particular, there is little possibility for either procedure to be data-driven following the discovery of any changepoints in the sequence. Moreover, the stipulation that each point is considered as a changepoint by at least one core places a constraint on the computational gains which can be made. Possible splitting techniques which could ameliorate these issues include sampling on a steadily finer dyadic grid of the sequence. In addition, from a parallelisation perspective, it would be relatively straightforward to employ a 'work-stealing' mechanism between cores,

as seen in other parallel implementations (Acar et al., 2013; Fernando et al., 2019; Pedro and Abreu, 2010). Such a work-stealing mechanism would prevent the analysis of any one level of resolution of the sequence being a computational bottleneck, while simultaneously ensuring that relatively few cores are required for parallelisation. Areas in which changes are found or suspected in the sequence could then undergo further analysis in the merge phase.

Finally, it would be of interest to explore and implement parallelisation for multivariate settings, where the computational gain over methods such as SMOP (Pickering, 2016) or MultiRank (Cabrieto et al., 2017) has the potential to be extremely impressive. We note, however, that such an approach would preclude online implementation (assuming that we again choose a technique where each point is observed by at least one core).

We now turn to consider pertinent questions arising from the SUBSET procedure. Although we derived novel theory on the probabilities of a false alarm and missing a change in the single change setting, it would be prudent to also give a result on the location accuracy. This would then give an indication of how the method theoretically extends to the multiple change setting under the Wild Binary Segmentation scheme we adopted, as well as show more broadly how the method behaves in settings with many changes under either large n or large d. Other important missing theory includes a consideration of the penalties for data settings outside the toy example of the Gaussian change in mean. As demonstrated in the negative binomial example, while using the recommended penalties works well in cases where the over-dispersion parameter is sufficiently large, for instances where this is close to 0 it is much more advisable to use simulations from the null to set the appropriate penalty values. It would not, however, be too difficult to extend our proof techniques to, for example, sub-exponential settings, to give an idea of how the penalties may scale appropriately. Indeed, recent work such as Zheng et al. (2019) has examined penalty setting for a range of changepoint problems, such as detection under exponential decay. However, one complication in our considered setting would be to ensure an appropriate balance between sparse and dense false positives.

Another setting of interest which was neglected in Chapter 4 was that of binary streams. These are now a very common consideration in the changepoint literature, given the potential to, for instance, apply such thinking to networks (see the discussion in Section 2.4.1 for some recent examples). Indeed, the terrorism example we discuss in Chapter 4 could very naturally be recast as a sparse time series of binary responses. It would be very interesting to corroborate the results of our analysis of the GTD with an appropriate setting of SUBSET for binary data.

Finally, it would be interesting to implement the SUBSET test statistic within the OMEN procedure of Chapter 5, in place of the multivariate AMOC detector for a single Gaussian change in mean. This could potentially also provide a mechanism for determining which variates alter at the changepoint. However, we remark that, in such a scenario, additional care would need to be taken with the setting of the penalty, given the problem of testing for many different types of change within the memory window each time an alarm is flagged.

Other questions arising from Chapter 5 predominantly concern the power of the method, particularly for those situations in which the mean and variance do not change. In particular, it would be useful to link the length of the learning window, ω, to the probability that OMEN detects a particular type of change, for example a change in mean in the Gaussian setting of magnitude Δ across all variates. While this would by no means be a complete result from the perspective of a general change in distribution of the type given in problem (2.1.1), it would be an important first step in understanding the power of the method outside an empirical context. More generally, it would be interesting to investigate other implementations of the ideas within the OMEN algorithm. One natural approach is to use the inverse-cdf-transformation, rather than the Box-Muller transformation, to convert the data to a stream that is standard Gaussian under the null distribution. One challenge here is the question of how to deal with the discreteness of the transformation when the empirical cdf is inverted. This means that, for small ω, the transformed data will be resemble a discrete approximation to a Gaussian.

Another interesting potential extension to OMEN arises from considering the

situation where variates may 'drop in and out' of the stream. This style of modelling is useful even in contexts where the number of variates may be fixed, given that entries into the stream can go missing due to poor data handling, the loss of a sensor etc. Other authors to have considered the issue of missing data in a changepoint context include Xie et al. (2013), who use an interesting submanifold approach to handle high dimensional data in an online context, and Muniz-Terrera et al. (2011), who assume a data missing-at-random approach within a logistic regression model. Many authors, such as Oca et al. (2010), typically resort to using interpolation for small numbers of missing observations, which can create severe issues when the number of missing entries is significant (for example, with the missing year of data from the Global Terrorism Database). Indeed, all of these existing methods would be significantly challenged in the scenario where variates may enter and leave the system of the stream 'at will'.

A related issue to the missing data problem is that of differing sampling rates across the variates. A natural approach in this setting may be to consider data at time points corresponding to the observation points of the variate being sampled at the lowest frequency. However, this can involve the loss of a significant amount of information. Of the few authors to have previously addressed variants of the problem, Brauckhoff et al. (2006) examine anomaly detection under differing sampling rates in the telecoms setting, but strictly in an offline context. In general, however, this problem remains very much an open, and prescient, area of research, particularly given the volume of data generated at irregular intervals, such as from social media activity. Indeed, as Petrov (2019) records, in the first six weeks of 2019, over 1.5 billion tweets were sent. Developing a toolkit to track features in such data and automatically report changes of interest is an exciting possible future avenue of research.

One other barrier to such developments is the relative lack of literature on the analysis of text data from the changepoint perspective. Such efforts include Chandola et al. (2013), which uses a CUSUM procedure alongside a text analysis in a health care claims setting, and Kulkarni et al. (2015), who construct a 'distributional time series' for specific words to track linguistic shifts (again using CUSUM statistics) over

long periods of time. There is clearly scope to build on these approaches, as well as ideas from traditional National Language Processing in the machine learning domain, to develop changepoint methodology for text data in a streaming setting.

While these problems are still some way from a satisfactory solution, the challenge of intensive data settings and the detection of changepoints therein is now justifiably receiving attention commensurate to its importance. The new algorithms we have introduced here are intended, in this sense, as a bridge between an interesting, if classical, statistical problem and the practicalities of a data world which never ceases to change.

Appendix A

Chunk and Deal

A.1 Yao's Results and Extension

The following two lemmas are due to Yao (1988).

Lemma A.1.1. *Suppose* $Z_1, ..., Z_n \sim^{i.i.d.} N(0, \sigma^2)$. *Then, for any* $\epsilon > 0$, *as* $n \to \infty$

$$\mathbb{P}\left(\max_{0 \leq i < j \leq n} \frac{(Z_{i+1} + ... + Z_j)^2}{(j-i)} > 2(1+\epsilon)\sigma^2 \log n \right) \to 0. \tag{A.1.1}$$

Lemma A.1.2. *Let* m_U *be an upper bound on the number of changes, and let* $(\hat{\tau}_1, ..., \hat{\tau}_{\hat{m}})$ *be the set of estimated changes generated (by Yao's procedure). For every* $\hat{m}$ *s.t.* $m < \hat{m} \leq m_U$ *and* $1 \leq r \leq m$,

$$\mathbb{P}((\hat{\tau}_1, ..., \hat{\tau}_{\hat{m}}) \in B_i^2(n)) \to 0$$

as $n \to \infty$, *where*

$$B_i^{\delta}(n) = \{(\xi_1, ..., \xi_t) : 0 < \xi_1 < ... < \xi_t < n \text{ and } |\xi_s - \tau_r| \geq \left\lceil (\log n)^{\delta} \right\rceil \text{ for } 1 \leq s \leq \hat{m}\}.$$

Corollary A.1.3. *Lemma A.1.2 can be extended to* $B_i^{1+\zeta}(n)$, *for any* $\zeta > 0$.

Proof of Corollary A.1.3: The argument for the location accuracy being $(\log n)^2$ in Yao (1988) comes from showing that the residual sum of squares for a segmentation that misses a change by more than this amount can be reduced by an amount that is greater than $3(2+\epsilon)\log n$ (with probability tending to 1 as n increases), by adding

three changes at the changepoint plus or minus $(\log n)^2$. Thus, such a segmentation cannot be optimal, as the penalised cost for the latter segmentation will be less than the original one. We therefore need only show that this argument holds if we replace an accuracy of $(\log n)^2$ with $(\log n)^{1+\zeta}$ for any $\zeta > 0$.

To do this, it suffices to show that a segmentation $\hat{\tau}_1, \ldots, \hat{\tau}_{\hat{m}}$ which misses a particular change τ_i by at least $\left\lceil (\log n)^{1+\zeta} \right\rceil$ has a residual sum of squares between the points $\tau_i - \left\lceil (\log n)^{1+\zeta} \right\rceil$ and $\tau_i + \left\lceil (\log n)^{1+\zeta} \right\rceil$, which when normalised by the true fit has term of leading order $\left\lceil (\log n)^{1+\zeta} \right\rceil$.

For a segmentation $\hat{\tau}_{1:\hat{m}}$, define $\mathrm{RSS}(y_{s:t}; \hat{\tau}_{1:\hat{m}})$ to be the residual sum of squares obtained if we fit the changepoints to the subset of data $y_{s:t}$. Note that this will only depend on the changepoints, if any, that lie between time points s and t. Then, for any $\hat{\tau}_{1:\hat{m}} \in B_i^{1+\zeta}(n)$

$$\mathrm{RSS}(y_{1:n}; \hat{\tau}_{1:\hat{m}}) \geq \mathrm{RSS}\left(y_{1:n}; \hat{\tau}_{1:\hat{m}}, \tau_1, \ldots, \tau_{i-1}, \tau_i - \left\lceil (\log n)^{1+\zeta} \right\rceil, \tau_i + \left\lceil (\log n)^{1+\zeta} \right\rceil, \tau_{i+1}, \ldots, \tau_m\right). \tag{A.1.2}$$

As Yao (1988) remarks, RHS of (A.1.2) can be decomposed as

$$\begin{aligned}&\mathrm{RSS}(y_{1:\tau_1}; \mathcal{T}_1) + \ldots + \mathrm{RSS}\left(y_{\tau_{i-1}+1:\tau_i - \lceil (\log n)^{1+\zeta} \rceil}; \mathcal{T}_i\right) + \mathrm{RSS}\left(y_{\tau_i - \lceil (\log n)^{1+\zeta} \rceil + 1:\tau_i + \lceil (\log n)^{1+\zeta} \rceil}; \emptyset\right) \\ &+ \mathrm{RSS}\left(y_{\tau_i + \lceil (\log n)^{1+\zeta} \rceil + 1:\tau_{i+1}}; \mathcal{T}_{i+1}\right) + \ldots + \mathrm{RSS}\left(y_{\tau_m+1:n}; \mathcal{T}_{m+1}\right),\end{aligned}$$

where $\mathcal{T}_h$ is the subset of $\hat{\tau}_{1:\hat{m}}$ which falls inside the corresponding segment of the univariate time series. By Lemma A.1.1, each term in this decomposition involving $\mathcal{T}_h$ is such that

$$\mathrm{RSS}(y_{a+1:b}; \mathcal{T}_h) = \sum_{j=a+1}^{b} Z_j^2 + O_p(\log n),$$

while, if without loss of generality we assume that the mean at the changepoint τ_i changes from 0 to μ, then letting $c_n^{(\zeta)} = \left\lceil (\log n)^{1+\zeta} \right\rceil$

$$\begin{aligned}\mathrm{RSS}\left(y_{\tau_i - c_n^{(\zeta)} + 1:\tau_i + c_n^{(\zeta)}}; \emptyset\right) &= \sum_{j=\tau_i - c_n^{(\zeta)} + 1}^{\tau_i + c_n^{(\zeta)}} \left(Y_i - \bar{Y}\left(\tau_i - c_n^{(\zeta)} + 1, \tau_i + c_n^{(\zeta)}\right)\right)^2 \\ &= \sum_{j=\tau_i - c_n^{(\zeta)} + 1}^{\tau_i + c_n^{(\zeta)}} Z_j^2 + \frac{\mu^2}{2} c_n^{(\zeta)} - \frac{1}{2c_n^{(\zeta)}} \left(\sum_{j=\tau_i - c_n^{(\zeta)} + 1}^{\tau_i + c_n^{(\zeta)}} Z_j \right)^2 + D,\end{aligned}$$

where $D \sim N\left(0, 2\sigma^2 c_n^{(\zeta)} \mu^2\right)$. Therefore

$$\left\{ \sum_{j=\tau_i - c_n^{(\zeta)}+1}^{\tau_i + c_n^{(\zeta)}} Z_j^2 - \text{RSS}\left(y_{\tau_i - c_n^{(\zeta)}+1:\tau_i + c_n^{(\zeta)}}; \emptyset\right) \right\} / c_n^{(\zeta)} = \frac{\mu^2}{2} - \frac{1}{2\left(c_n^{(\zeta)}\right)^2} \left(\sum_{j=\tau_i - c_n^{(\zeta)}+1}^{\tau_i + c_n^{(\zeta)}} Z_j \right)^2 + D/c_n^{(\zeta)}$$
$$\to \frac{\mu^2}{2} \text{ by Lemma A.1.1.}$$

In particular, $\forall \hat{\tau}_{1:\hat{m}} \in B_i^{1+\zeta}(n)$

$$\left\{ \text{RSS}\left(x_{1:n}; \hat{\tau}_{1:\hat{m}}, \tau_{1:n}^{-i}, \tau_i - c_n^{(\zeta)}, \tau_i + c_n^{(\zeta)}\right) - \sum_{j=1}^{n} Z_j^2 \right\} / c_n^{(\zeta)} \to \frac{\mu^2}{2}.$$

Thus, as any segmentation from $B_i^{1+\zeta}(n)$ is strictly worse than a corresponding segmentation, which in turn is worse (in probability) than fitting the truth under a penalty of $\beta = 2(1+\epsilon)\log n$, then uniformly in $B_i^{1+\zeta}(n)$, $\mathbb{P}(\hat{\tau}_{1:\hat{m}} \in B_i^{1+\zeta}(n)) \to 0$. □

A.2 Unparallelised Consistency Results

Proof of Proposition 1: Let $\hat{m}$ be the number of changes estimated by the procedure. The aim is firstly to show that

(a): $\mathbb{P}(\hat{m} > m) \to 0$,

(b): $\mathbb{P}(\hat{m} < m) \to 0$.

Proof of (a): Under Corollary A.1.3, for $\hat{m} > m$, with probability 1 as $n \to \infty$, it must be the case that m of the estimated changes are within $(\log n)^{1+\zeta}$, some $\zeta > 0$, of the true changes. We will now show that, with probability tending to 1, these segmentations cannot be optimal.

To do this, we will compare the penalised cost of any such segmentation with the penalised cost of the true segmentation. The latter cost can be bounded above by $\sum_{t=1}^{n} Z_t^2 + m(2+\epsilon)\log n$. Our approach is to split the comparison of the residual sum of squares of a segmentation $\hat{\tau}_{1:\hat{m}}$ with $\sum_{t=1}^{n} Z_t^2$ into comparisons for a fixed number of regions of data. To do this, define $c_n^{(\zeta)} = \left\lceil (\log n)^{1+\zeta} \right\rceil$, $u_0 = 0$, $l_{m+1} = n$,

and for $i = 1, \ldots, m$, $l_i = \tau_i - c_n^{(\zeta)}$ and $u_i = \tau_i + c_n^{(\zeta)}$. We can partition the time points $1, \ldots, n$ into regions $\mathcal{M}_i = \{u_{i-1} + 1, \ldots, l_i\}$, for $i = 1, \ldots, m+1$ and regions $\mathcal{L}_i = \{l_i + 1, \ldots, \tau_i\}$ and $\mathcal{R}_i = \{\tau_i + 1, \ldots, u_i\}$ for $i = 1, \ldots, m$. These can be viewed as regions more than $c_n^{(\zeta)}$ from a changepoint, and regions of length $c_n^{(\zeta)}$ that are respectively left and right of a changepoint.

It is straightforward to show that for any segmentation

$$\mathrm{RSS}(y_{1:n}; \hat{\tau}_{1:\hat{m}}) \geq \sum_{i=1}^{m+1} \mathrm{RSS}(y_{\mathcal{M}_i}; \hat{\tau}_{1:\hat{m}}) + \sum_{i=1}^{m+1} \mathrm{RSS}(y_{\mathcal{L}_i}; \hat{\tau}_{1:\hat{m}}) + \sum_{i=1}^{m+1} \mathrm{RSS}(y_{\mathcal{R}_i}; \hat{\tau}_{1:\hat{m}}).$$

The proof proceeds by showing that, on each region $\mathcal{M}_i$, if we have $k = k(\hat{\tau}_{1:\hat{m}})$ changepoints that lie within this region, then with probability tending to 1

$$\max_{\hat{\tau}_{1:\hat{m}}} \left\{ \mathrm{RSS}(y_{\mathcal{M}_i}; \hat{\tau}_{1:\hat{m}}) + 2\,(1 + \epsilon/2)\, k \log n - \sum_{t=u_i+1}^{l_i} Z_t^2 \right\} > -4 \log \log n.$$

We then show that, on each region $\mathcal{L}_i$ (and similarly each region $\mathcal{R}_i$), that if there are $k = k(\hat{\tau}_{1:\hat{m}})$ changepoints, then with probability tending to 1

$$\max_{\hat{\tau}_{1:\hat{m}}} \left\{ \mathrm{RSS}(y_{\mathcal{L}_i}; \hat{\tau}_{1:\hat{m}}) - \sum_{t=u_i+1}^{l_i} Z_t^2 \right\} > -4\,(k+1) \log \log n.$$

Taken together we have, with probability tending to 1, a uniform bound on the difference in cost between any segmentation with more than m changepoints, and that also has one change within $c_n^{(\zeta)}$ of each true change, and the true segmentation. As such a segmentation can only have, at most, $\hat{m} - m$ changes in regions $\mathcal{M}_i$, this difference is bounded by

$$(\hat{m} - m)\epsilon \log n - 4(2m+3) \log \log n > \epsilon \log n - 4(2m+3) \log \log n,$$

which is positive for large enough n.

Note that on each region $\mathcal{M}_i, \mathcal{L}_i, \mathcal{R}_i$, there are no true changes. Therefore, any estimated changes we do fit inside these regions will involve fitting changes to the noise. Take a generic region of length $\tilde{n}$ which contains no true changes. We examine the reduction in the residual sum of squares when we add 0 and $k > 0$ estimated changes. Note that, in the former case, it is true that

$$-\mathrm{RSS}(y_{\mathcal{A}_i}; \hat{\tau}_{1:\hat{m}}) + \sum_{t=a_i+1}^{a_i+\tilde{n}} Z_t^2 = \frac{1}{\tilde{n}} \left(\sum_{t=a_i+1}^{a_i+\tilde{n}} Z_t \right)^2,$$

where $\mathcal{A}$ is used as a placeholder to refer to any of the three types of region such that $\mathcal{A}_i = \{a_i+1, \ldots, a_i+\tilde{n}\}$. Thus, the negative of the expression of interest is distributed according to χ^2_1. Therefore, for sufficiently large n, the probability that this quantity is greater than $4 \log \log n$ tends to 0.

So we need focus only on the case where $k > 0$. Label, without loss of generality, the estimated changes which lie in the region $\mathcal{A}_i$ as $\hat{\tau}_1, \ldots, \hat{\tau}_k$, and let

$$\text{Diff} = \text{RSS}(y_{\mathcal{A}_i}; \hat{\tau}_{1:\hat{m}}) - \sum_{t=a_i+1}^{a_i+\tilde{n}} Z_t^2.$$

Then

$$\text{Diff} = \frac{1}{\hat{\tau}_1 - a_i} \left(\sum_{t=a_i+1}^{\hat{\tau}_1} Z_t \right)^2 + \ldots + \frac{1}{a_i + \tilde{n} - \hat{\tau}_k} \left(\sum_{t=\hat{\tau}_k+1}^{a_i+\tilde{n}} Z_t \right)^2.$$

We demonstrate that this difference is less than $2k(1+\epsilon)\log \tilde{n}$, for any $\epsilon > 0$. Note that, collectively, the positive terms in the expression follow a χ^2_{k+1} distribution. By Laurent and Massart (2000), for any quantity U which follows a chi-squared distribution with D degrees of freedom, for any $x > 0$

$$\mathbb{P}\left(U - D \geq 2\sqrt{Dx} + 2x \right) \leq \exp(-x), \tag{A.2.1}$$

letting $D = k+1$ and $x = \frac{d \log \tilde{n} - \sqrt{(2d \log \tilde{n} - (k+1))(k+1)}}{2}$, for some $d > 0$ such that $\tilde{n} \geq e^{\frac{k+1}{2d}}$. In practice $d > k$ (see below) so almost all positive integer values of $\tilde{n}$ will be sufficient. With this choice of x, the LHS of (A.2.1) corresponds to $\mathbb{P}(U > d \log \tilde{n})$, and for large enough $\tilde{n}$ (A.2.1) becomes

$$\mathbb{P}(U \geq d \log n) \leq \tilde{n}^{-\frac{d}{2}+\delta}, \text{ for any } \delta > 0. \tag{A.2.2}$$

There are then $\binom{\tilde{n}}{k}$ possible segmentations of these (incorrectly) fitted changes in this region. Given that $\binom{\tilde{n}}{k} < \frac{\tilde{n}^k}{k!}$, then by employing a Bonferroni correction, for the best segmentation involving k changes in the region

$$\begin{aligned}\mathbb{P}(\text{Diff} \geq d \log \tilde{n}) &\leq \tilde{n}^{-\frac{d}{2}+\delta} \tilde{n}^k \\ &= \tilde{n}^{k+\delta-\frac{d}{2}} \to 0 \text{ for } d = 2k(1+\epsilon), \text{ if we set, for example, } \delta = \epsilon/2.\end{aligned}$$

(For $d = 2k(1+\epsilon)$, if $\delta = \epsilon/2$ - as (A.2.2) permits any strictly positive value of δ - then $k + \delta - \frac{d}{2} = -(2k-1)\epsilon/2 < 0$.)

Note that this establishes the appropriate bound only in the case where k is fixed and positive. To obtain the uniform bound over all k, we must sum over all $k = 1, \ldots, \tilde{n}$. So, for a given $\tilde{n}$ and ϵ

$$\sum_{k=1}^{\tilde{n}} \mathbb{P}(\text{Diff} \geq 2k\,(1+\epsilon)\log\tilde{n}) \leq \sum_{k=1}^{\tilde{n}} \tilde{n}^{-(2k-1)\epsilon/2} = \frac{\tilde{n}^{-\epsilon/2}\left(1-\tilde{n}^{-\tilde{n}\epsilon}\right)}{1-\tilde{n}^{-\epsilon}} \to 0,\ \forall \epsilon > 0.$$

This establishes the required results for both regions of type $\mathcal{M}_i$ and $\mathcal{L}_i$ ($\mathcal{R}_i$) by substituting $\tilde{n} = \lambda n$, $\lambda \leq 1$ and $\tilde{n} = \left\lceil (\log n)^{1+\zeta} \right\rceil$ (for $\zeta < 1$ to obtain the constant 4 in the two initial statements) respectively.

Hence $\mathbb{P}(\hat{m} > m) \to 0$.

Proof of (b): Now suppose we have that $\hat{m} < m$. For n sufficiently large, it is guaranteed that there is at least one true change (which shall be labelled τ) such that the closest estimated change is at least $\left\lceil (\log n)^{1+\zeta} \right\rceil$ time points away. Thus, by the proof of Corollary A.1.3, given that a change has been missed by this error, adding in estimated changes to the model at the points $\tau - \left\lceil (\log n)^{1+\zeta} \right\rceil$, τ, $\tau + \left\lceil (\log n)^{1+\zeta} \right\rceil$ gives that the reduction in the RSS is greater than the incurred penalty for adding 3 changes. Thus, the original segmentation was not optimal.

Hence $\mathbb{P}(\hat{m} < m) \to 0$.

Lastly, we need to establish that, when $\hat{m} = m$, the event that each of the estimated changes is within $\left\lceil (\log n)^{1+\zeta} \right\rceil$ of a true change tends to 1. Suppose we have a segmentation with $\hat{m} = m$ which contains a true change, τ_i, with no estimated changes within $\left\lceil (\log n)^{1+\zeta} \right\rceil$. Then by comparing this segmentation to an equivalent segmentation which also fits estimated changes at $\tau_i - \left\lceil (\log n)^{1+\zeta} \right\rceil, \tau_i, \tau_i + \left\lceil (\log n)^{1+\zeta} \right\rceil$, we again obtain a saving of greater than the cost of adding 3 changes by Yao (1988) and Corollary A.1.3. □

Note that this result extends naturally to a multivariate analogue.

Lemma A.2.1. *Take a procedure which exactly minimises the squared error loss for the multivariate problem*

$$\boldsymbol{Y}_i = \boldsymbol{\epsilon}_i + \boldsymbol{\mu}_k,\ \textit{for } \tau_{k-1} + 1 \leq i \leq \tau_k,\ \textit{and } k \in \{1, ..., m+1\}, \tag{A.2.3}$$

where $\boldsymbol{Y}_i = \left(Y_i^{(1)}, ..., Y_i^{(d)}\right)^T$, $\forall i \in \{1, ..., n\}$; $\boldsymbol{\mu}_k \neq \boldsymbol{\mu}_{k+1}$, $\forall k \in \{1, ..., m\}$; $\boldsymbol{\epsilon}_i \sim^{i.i.d.} N_d(\boldsymbol{0}, \sigma^2 I_d)$, *some* d. *In addition, take the penalty for fitting a change to be* $(d+1)(1+\epsilon)\log n$, *for any* $\epsilon > 0$. *Then, defining* $\mathcal{E}_n^\zeta$ *as for Proposition 1 for any* $\zeta > 0$, *again gives that* $\mathbb{P}(\mathcal{E}_n^\zeta) \to 1$ *as* $n \to \infty$.

Proof of Lemma A.2.1: We define the natural extension of the residual sum of squares in the multivariate case as

$$\begin{aligned}\mathrm{RSS}(\mathbf{y}_{1:n}; \hat{\tau}_{1:\hat{m}}) &= \sum_{i=1}^{\hat{\tau}_1} (\mathbf{y}_i - \hat{\boldsymbol{\mu}}_1)^T (\mathbf{y}_i - \hat{\boldsymbol{\mu}}_1) + ... + \sum_{i=\hat{\tau}_{\hat{m}}+1}^{n} (\mathbf{y}_i - \hat{\boldsymbol{\mu}}_{\hat{m}+1})^T (\mathbf{y}_i - \hat{\boldsymbol{\mu}}_{\hat{m}+1}) \\ &= \sum_{j=1}^{\hat{m}+1} \sum_{i=\hat{\tau}_{j-1}+1}^{\hat{\tau}_j} \sum_{k=1}^{d} (y_{i,k} - \hat{\mu}_{j,k})^2,\end{aligned}$$

where $\hat{\mu}_{j,k} = \frac{1}{\hat{\tau}_j - \hat{\tau}_{j-1}} \sum_{i=\hat{\tau}_{j-1}+1}^{\hat{\tau}_j} y_{i,k} = \bar{y}_{j,k}$. Using this, we proceed along the same trajectory as for the previous proof. Suppose that $\hat{m}$ changes are estimated by the procedure. Then we first show that

(a): $\mathbb{P}(\hat{m} > m) \to 0$,

(b): $\mathbb{P}(\hat{m} < m) \to 0$.

Proof of (a): Again let $c_n^{(\zeta)} = \left\lceil (\log n)^{1+\zeta} \right\rceil$. Note first that an equivalent result to Corollary A.1.3 holds in the multivariate case, as the residual sum of squares between the points $\tau_i - c_n^{(\zeta)}$ and $\tau_i + c_n^{(\zeta)}$ (where τ_i is some true change missed by the procedure as before) satisfies

$$\begin{aligned}\frac{\mathrm{RSS}\left(\mathbf{y}_{\tau_i - c_n^{(\zeta)}+1:\tau_i+c_n^{(\zeta)}}; \emptyset\right) - \sum_{k=1}^{d} \sum_{j=\tau_i - c_n^{(\zeta)}+1}^{\tau_i + c_n^{(\zeta)}} Z_{j,k}^2}{c_n^{(\zeta)}} &= \sum_{k=1}^{d} \frac{\left(\mu_k^{(i)} - \mu_k^{(i+1)}\right)^2}{2} \\ &\quad - \frac{1}{2c_n^{(\zeta)}} \sum_{k=1}^{d} \left(\sum_j Z_{j,k}\right)^2 + \frac{D_k}{c_n^{(\zeta)}} \\ &\to \sum_{k=1}^{d} \frac{\left(\mu_k^{(i)} - \mu_k^{(i+1)}\right)^2}{2} \text{ as } n \to \infty,\end{aligned}$$

where D_k is normally distributed with a variance equivalent to the deterministic term scaled by $4\sigma^2$.

Hence, as per the previous proof, we can compare the residual sum of squares of the fit of a set of estimated changes with $\hat{m} > m$ across (equivalent) regions $\mathcal{M}_i, \mathcal{L}_i, \mathcal{R}_i$ to the null fit. Across a region bounded by the points (a, b) containing estimated changes $\hat{\tau}_1, ..., \hat{\tau}_p$, the relevant difference term is

$$\text{Diff} = \sum_{k=1}^{d} \left[\frac{1}{\hat{\tau}_1 - a} \left(\sum_{j=a+1}^{\hat{\tau}_1} Z_{j,k} \right)^2 + ... + \frac{1}{b - \hat{\tau}_p} \left(\sum_{j=\hat{\tau}_p+1}^{b} Z_{j,k} \right)^2 \right],$$

giving that $\text{Diff} \sim \chi^2_{d(p+1)}$. A similar argument to before then gives that

$$\mathbb{P}(\text{Diff} \geq p\,(d+1)\,(1+\epsilon)\log n) \to 0,$$

and in particular

$$\sum_{p=1}^{n} \mathbb{P}(\text{Diff} \geq p\,(d+1)\,(1+\epsilon)\log n) \to 0.$$

Hence $\mathbb{P}(\hat{m} > m) \to 0$.

Proof of (b): This follows immediately from considering the multivariate equivalent to Corollary A.1.3 shown above, inferring the presence of a missed change, τ_i, and fitting three estimated changes at $\tau_i - \left\lceil (\log n)^{1+\zeta} \right\rceil, \tau_i, \tau_i + \left\lceil (\log n)^{1+\zeta} \right\rceil$. This segmentation will produce a lower residual sum of squares than the original with probability approaching 1.

Hence $\mathbb{P}(\hat{m} < m) \to 0$.

All that remains is to show that this correct number of changes falls within $\left\lceil (\log n)^{1+\zeta} \right\rceil$. However, this again follows the same line of reason as for the univariate case by the result established above. □

A.3 Additional Simulations: Parallelisation Under an Increasing Number of Changepoints

We here examine the behaviour of Chunk and Deal compared to PELT in situations with an increasingly large number of changes. We again focus on the Gaussian change in mean setting, beginning with a mean of 0. At every changepoint, the mean changes

to 2 if the mean was previously at 0, and changes to 0 if the mean was previously at 2. Gaussian noise of variance 1 is added to each time point.

For the following scenarios (which we label as $p = 1, \ldots, 7$), the length of the series was taken as $n = 1024$, while the number of changes was taken as $m = 2^p - 1$, with the set of changepoints in scenario p being $\tau_i^{(p)} = \frac{2^{10}}{2^p} \times i$, for $i = 1, \ldots, 2^p - 1$.

As for the main study in Chapter 3, all simulations were run in R using a Linux OS on a 2.3GHz Intel Xeon CPU. The average for all metrics was calculated across 200 repetitions in each case.

In Table A.1, we show the average number of false alarms incurred in each setting by Chunk and Deal under a differing number of cores, L, for each of the scenarios $p = 1, \ldots, 6$. In Table A.2, we show the average number of changes missed for each of the scenarios $p = 1, \ldots, 7$. In Table A.3, we show the average location area of the methods for each of the scenarios $p = 1, \ldots, 7$. The performance of unparallelised PELT is shown for comparison in each case.

We see from all three tables that parallelisation has little effect on the accuracy, with all of the methods, for example, missing many changepoints in the more difficult settings as p is made larger. Indeed, with regards to the false alarm performance, there was no change at all in the performance of the methods with parallelisation up to 10 cores, even with just 16 points per segment for $p = 6$.

We remark on the interesting case of $p = 4$, however, which corresponds to a setting with 15 changepoints. Here, a slight detrimental effect is observed, with Chunk struggling to reach the performance of PELT as L is increased and Deal experiencing a higher number of missed changes and larger average location error even for $L = 2$. Comparing the results here to the result given in Table 3.2, we see that Chunk and Deal (with 4 cores) are in fact performing better than for the case with 14 changes (which we labelled as scenario E). This is somewhat unsurprising given that this latter case had irregularly spaced changepoints, leading to some very short segments, making some changepoints more difficult to detect. (Indeed, PELT also does much worse.) In addition, each changepoint in scenario E shifts the mean signal by 1, whereas here there is a shift of 2 which, particularly for a small number of cores, increases detection

Average False Alarms		Scenario (p), $\Delta\mu = 2$, $n = 1024$					
Method	L	1	2	3	4	5	6
PELT	1	**0.00**	1.00	**1.00**	**2.00**	**0.00**	**0.00**
Chunk	2	**0.00**	1.00	**1.00**	**2.00**	**0.00**	**0.00**
Chunk	3	**0.00**	1.00	**1.00**	**2.00**	**0.00**	**0.00**
Chunk	4	**0.00**	1.00	**1.00**	**2.00**	**0.00**	**0.00**
Chunk	5	**0.00**	1.00	**1.00**	**2.00**	**0.00**	**0.00**
Chunk	6	**0.00**	**0.00**	**1.00**	**2.00**	**0.00**	**0.00**
Chunk	7	**0.00**	**0.00**	**1.00**	**2.00**	**0.00**	**0.00**
Chunk	8	**0.00**	**0.00**	**1.00**	**2.00**	**0.00**	**0.00**
Chunk	9	**0.00**	**0.00**	**1.00**	**2.00**	**0.00**	**0.00**
Chunk	10	**0.00**	**0.00**	**1.00**	**2.00**	**0.00**	**0.00**
Deal	2	**0.00**	1.00	**1.00**	**2.00**	**0.00**	**0.00**
Deal	3	**0.00**	1.00	**1.00**	**2.00**	**0.00**	**0.00**
Deal	4	**0.00**	1.00	**1.00**	**2.00**	**0.00**	**0.00**
Deal	5	**0.00**	1.00	**1.00**	**2.00**	**0.00**	**0.00**
Deal	6	**0.00**	1.00	**1.00**	**2.00**	**0.00**	**0.00**
Deal	7	**0.00**	1.00	**1.00**	**2.00**	**0.00**	**0.00**
Deal	8	**0.00**	1.00	**1.00**	**2.00**	**0.00**	**0.00**
Deal	9	**0.00**	**0.00**	**1.00**	**2.00**	**0.00**	**0.00**
Deal	10	**0.00**	1.00	**1.00**	**2.00**	**0.00**	**0.00**

Table A.1: The average number of false alarms recorded across all 200 repetitions for each of the scenarios $p = 1, \ldots, 6$. A false alarm is defined as an estimated changepoint which is at least $\lceil(\log n)\rceil$ points from the closest true changepoint. Note that this is why we do not report scenario 7 here, as any spuriously placed changepoint will be sufficiently close to a true change as to not be flagged as a false alarm. Bold entries show the best performing algorithm for each scenario.

probability, again as we saw in Table 3.2.

In Chapter 3, we discussed the asymptotic performance of Chunk and Deal using a novel asymptotic consistency result for PELT. These results assumed an infill asymptotic setting. That is, the number of changes remaining at fixed positions (i.e. proportions) of the data sequence for increasing n. The simulation study we have conducted here demonstrates that Chunk and Deal are not noticeably worse in terms of statistical performance than unparallelised PELT, even for settings very far removed from the assumptions made.

Average Number Missed		Scenario (p), $\Delta\mu = 2$, $n = 1024$						
Method	L	1	2	3	4	5	6	7
PELT	1	**0.00**	**0.00**	**1.00**	**2.00**	**30.0**	**62.0**	**125**
Chunk	2	**0.00**	**0.00**	**1.00**	**2.00**	**30.0**	**62.0**	**125**
Chunk	3	**0.00**	**0.00**	**1.00**	**2.00**	**30.0**	**62.0**	**125**
Chunk	4	**0.00**	**0.00**	**1.00**	6.00	**30.0**	**62.0**	**125**
Chunk	5	**0.00**	**0.00**	**1.00**	6.00	**30.0**	**62.0**	**125**
Chunk	6	**0.00**	**0.00**	**1.00**	**2.00**	**30.0**	**62.0**	**125**
Chunk	7	**0.00**	**0.00**	**1.00**	6.00	**30.0**	**62.0**	**125**
Chunk	8	**0.00**	**0.00**	**1.00**	6.00	**30.0**	**62.0**	**125**
Chunk	9	**0.00**	**0.00**	**1.00**	6.00	**30.0**	**62.0**	**125**
Chunk	10	**0.00**	**0.00**	**1.00**	6.00	**30.0**	**62.0**	**125**
Deal	2	**0.00**	**0.00**	**1.00**	6.00	**30.0**	**62.0**	**125**
Deal	3	**0.00**	**0.00**	**1.00**	6.00	**30.0**	**62.0**	**125**
Deal	4	**0.00**	**0.00**	**1.00**	6.00	**30.0**	**62.0**	**125**
Deal	5	**0.00**	**0.00**	**1.00**	6.00	**30.0**	**62.0**	**125**
Deal	6	**0.00**	**0.00**	**1.00**	6.00	**30.0**	**62.0**	**125**
Deal	7	**0.00**	**0.00**	**1.00**	6.00	**30.0**	**62.0**	**125**
Deal	8	**0.00**	**0.00**	**1.00**	6.00	**30.0**	**62.0**	**125**
Deal	9	**0.00**	**0.00**	**1.00**	6.00	**30.0**	**62.0**	**125**
Deal	10	**0.00**	**0.00**	**1.00**	6.00	**30.0**	**62.0**	**125**

Table A.2: The average number of missed changes across all 200 repetitions for each of the scenarios $p = 1, \ldots, 7$. A missed change is defined as a true changepoint for which no estimated change lies within $\lceil (\log n) \rceil$ points. Bold entries show the best performing algorithm.

Average Location Error		Scenario (p), $\Delta\mu = 2$, $n = 1024$						
Method	L	1	2	3	4	5	6	7
PELT	1	**5.00**	52.2	**3.14**	3.27	**6.00**	**6.00**	**1.00**
Chunk	2	**5.00**	52.2	**3.14**	3.27	**6.00**	**6.00**	**1.00**
Chunk	3	**5.00**	52.2	**3.14**	3.27	**6.00**	**6.00**	**1.00**
Chunk	4	**5.00**	52.2	**3.14**	3.73	**6.00**	**6.00**	**1.00**
Chunk	5	**5.00**	52.2	**3.14**	3.64	**6.00**	**6.00**	**1.00**
Chunk	6	**5.00**	**2.33**	**3.14**	**3.07**	**6.00**	**6.00**	**1.00**
Chunk	7	**5.00**	**2.33**	**3.14**	3.73	**6.00**	**6.00**	**1.00**
Chunk	8	**5.00**	**2.33**	**3.14**	3.73	**6.00**	**6.00**	**1.00**
Chunk	9	**5.00**	**2.33**	3.86	3.73	**6.00**	**6.00**	**1.00**
Chunk	10	**5.00**	**2.33**	**3.14**	3.64	**6.00**	**6.00**	**1.00**
Deal	2	**5.00**	52.2	**3.14**	3.73	**6.00**	**6.00**	**1.00**
Deal	3	**5.00**	52.2	**3.14**	3.73	**6.00**	**6.00**	**1.00**
Deal	4	**5.00**	52.2	**3.14**	3.73	**6.00**	**6.00**	**1.00**
Deal	5	**5.00**	52.2	**3.14**	3.73	**6.00**	**6.00**	**1.00**
Deal	6	**5.00**	52.2	**3.14**	3.73	**6.00**	**6.00**	**1.00**
Deal	7	**5.00**	52.2	**3.14**	3.73	**6.00**	**6.00**	**1.00**
Deal	8	**5.00**	52.2	**3.14**	3.73	**6.00**	**6.00**	**1.00**
Deal	9	**5.00**	**2.33**	**3.14**	3.73	**6.00**	**6.00**	**1.00**
Deal	10	**5.00**	52.2	**3.14**	3.73	**6.00**	**6.00**	**1.00**

Table A.3: The average location error between those true changes which were detected by the algorithms and the corresponding estimated change, across all 200 repetitions for each of the 7 scenarios. Bold entries show the best performing algorithm.

As we noted earlier in this section, the performance of Chunk, Deal and PELT is noticeably better in the $p = 4$ case than for scenario E of Section 3.4, despite the latter having fewer changepoints. Indeed, it is clear that all the methods based on dynamic programming suffer when there is a very short segment. For example, we see from Table 3.2 in scenario B that, even for $\Delta\mu = 2$ when $n = 1000$, the methods all miss at least one change on average, with WBS noticeably the best performing procedure. However, as n is increased under this same scenario, there is a very marked reduction in the false alarm rate even for lower values of $\Delta\mu$. This is consistent with a recent work in which the finite-sample properties of methods such as PELT are explored for the change in mean setting (Wang et al., 2019a). Here, it is shown that providing

$$\frac{\Delta\mu\sqrt{\min_{1\leq i\leq m+1}(\tau_i - \tau_{i-1})}}{\sigma} \geq C\sqrt{(\log n)^{1+\iota}}, \tag{A.3.1}$$

for some sufficiently large constant C and some $\iota > 0$, then the probability of detecting all changepoints to within $D \log n$, for some D, is at least $1 - e \times n^{3-c}$ for some $c > 3$. Assumption (A.3.1) is useful in giving a broad description of the trade-off between $\min_{1\leq i\leq m+1}(\tau_i - \tau_{i-1}), \Delta\mu, \sigma, n$ and L under Chunk and Deal. For example, under the Chunk procedure, we would require that the minimum segment length within the chunk given to a particular core is at least

$$\frac{C\sigma}{\Delta\mu}\sqrt{(\log n - \log L)^{1+\iota}}.$$

Note that, from a computational perspective, we recommended setting $L \sim n^{\frac{1}{2}}$ in Section 3.3.1, which simply gives an adjustment of the constant, such that the minimum segment length should still be $\Omega(\sqrt{\log n})$, meaning little degeneracy in performance with parallelisation. However, the two obvious caveats are that (i) this argument ignores any new minimum segment length induced by the placing of a boundary; and (ii) we are required to perform a correction on the lower bound of the detection probability stated above. To mitigate (i) in the infill setting, we took an overlap of length $\lceil(\log n)^{1+\xi}\rceil$, for some $\xi > 0$, and made the assumption that at most one change fell within each chunk for sufficiently large n. In a finite-sample case with potentially many changes, this argument is not valid. Indeed, we note it is no

longer possible to guarantee a minimum segment length within a chunk in general without, for example, assuming that adjacent changes must be at least $2\lceil(\log n)^{1+\xi}\rceil$ points apart. With regards to (ii), note that in the case of $L \sim n^{\frac{1}{2}}$ we would require a sequence of length n^2 before Chunk had comparable detection in probability to unparallelised PELT.

We remark that many of the issues discussed above for Chunk also apply to Deal, as the central assertion required when proving the consistency of Deal was that the core which is 'dealt' the true change will necessarily return that point as a change. If L dominates the minimum segment length, however, this is not guaranteed.

Appendix B

SUBSET

B.1 Preliminary Lemmas

In this section, we establish several lemmas required to prove the central results of Chapter 4 (please see Section B.2 for the main proofs). Our general approach with this section is to establish the stated results in either the sparse or the dense setting, and then combine these results appropriately in Section B.2. Throughout this section, we repeatedly use the following two Lemmas.

Lemma B.1.1. *Suppose* $G \sim \chi^2_k$. *Then for any* $x > 0$

$$\mathbb{P}\left(G \geq k + 2\sqrt{xk} + 2x\right) \leq \exp\left(-x\right).$$

Proof: See, for example, Laurent and Massart (2000).

Lemma B.1.2. *Suppose* $H \sim \chi^2_k(\nu)$. *Then for any* $y > 0$

$$\mathbb{P}\left(H \geq k + \nu - 2\sqrt{(k+2\nu)y}\right) \geq 1 - \exp(-y),$$

and

$$\mathbb{P}\left(H \geq k + \nu + 2\sqrt{(k+2\nu)y} + 2y\right) \leq \exp(-y).$$

Proof: See Birgé (2001).

We now give two results on the Type I error of the SUBSET procedure. Lemma B.1.3 gives a bound on the Type I error in the sparse setting, under particular

choices for the penalties α and β, and Lemma B.1.4 gives an equivalent result in the dense setting, under an additional choice for the dense penalty K.

Lemma B.1.3. *Suppose we are in the same setting as for Theorem 4.3.1 of Chapter 4. Let $D'_{i,t}$, α and β be as defined in Section 4.3 of Chapter 4. Define $S_{1,t} = \sum_{i=1}^{d} D'_{i,t} - \beta$. Let $\sqrt{\beta} = \sqrt{2d\frac{\Gamma\left(\frac{1}{2},\frac{\alpha}{2}\right)}{\Gamma\left(\frac{1}{2}\right)}} + C\sqrt{\log n}$ for some constant C, then*

$$\mathbb{P}\left(\max_t S_{1,t} > 0\right) \leq n^{1-\frac{C^2}{2}} \exp\left(\frac{(1+\vartheta)^2}{4\vartheta}\frac{d}{\exp\left(\frac{\alpha}{2}\right)}(1+\alpha)^{-\frac{1}{2}}\right), \text{ some } \vartheta,$$

providing that

$$\frac{\gamma\left(\frac{1}{2},\frac{\alpha}{2}\right)}{\Gamma\left(\frac{1}{2}\right)} \geq \vartheta > 0, \tag{B.1.1}$$

and

$$\beta > 2d\frac{\Gamma\left(\frac{1}{2},\frac{\alpha}{2}\right)}{\Gamma\left(\frac{1}{2}\right)}. \tag{B.1.2}$$

Lemma B.1.4. *Suppose again that we are in the same setting as for Theorem 4.3.1 of Chapter 4. Let $D_{i,t}$ and K be defined as in Section 4.3 of Chapter 4. Define $S_{2,t} = \sum_{i=1}^{d} D_{i,t} - K$. Setting $\beta = (J+\epsilon)\log n$, $\alpha = 2\log d$ and $K = d + \sqrt{2\beta d} + \beta$ gives that*

$$\mathbb{P}\left(\max_t S_{2,t} > 0\right) \leq n^{1-\frac{(J+\epsilon)}{2}}.$$

Proof of Lemma B.1.3: Fix τ. Note if $f_{D'_{i,\tau}}(x)$ is the density function for $D'_{i,\tau}$, then it is straightforward to show that for $x > 0$, $d\log f_{D'_{i,\tau}}(x)/dx < -\frac{1}{2}$; therefore, $D'_{i,\tau}$ is stochastically dominated by $N_{i,\tau}$, where

$$N_{i,\tau} = \begin{cases} 0 & \text{w.p. } p_\alpha \\ \text{Exp}\left(\frac{1}{2}\right) & \text{w.p. } 1-p_\alpha, \end{cases}$$

such that $p_\alpha = \mathbb{P}\left(D'_{i,\tau} = 0\right) = \frac{\gamma\left(\frac{1}{2},\frac{\alpha}{2}\right)}{\Gamma\left(\frac{1}{2}\right)}$. We define for convenience $q_\alpha = 1 - p_\alpha = \frac{\Gamma\left(\frac{1}{2},\frac{\alpha}{2}\right)}{\Gamma\left(\frac{1}{2}\right)}$.

Let $A_\tau = \sum_{i=1}^{d} N_{i,\tau}$; then the moment generating function of A_τ is

$$m_{A_\tau}(\lambda) = \left(p_\alpha + \frac{q_\alpha}{1-2\lambda}\right)^d.$$

We seek the Cramer transform, $\psi^*_{A_\tau}(r)$, of A_τ, such that

$$\psi^*_{A_\tau}(r) = \sup_{\lambda \geq 0}\left\{\lambda r - d\log\left(p_\alpha + \frac{q_\alpha}{1-2\lambda}\right)\right\};$$

it is easy to see that for $\lambda < \frac{1}{2}$ the supremum is achieved close to $\lambda = \frac{1}{2p_\alpha} - \frac{1}{p_\alpha}\sqrt{\frac{dq_\alpha}{2r}}$, such that we obtain

$$\begin{aligned}
\mathbb{P}\left(A_\tau \geq \beta\right) &\leq \exp\left(-\psi^*_{A_\tau}(\beta)\right) \\
&\leq \exp\left(-\left(\frac{1}{2p_\alpha} - \frac{1}{p_\alpha}\sqrt{\frac{dq_\alpha}{2\beta}}\right)\beta\right)\left(\frac{p_\alpha}{1-\sqrt{\beta q_\alpha}2d}\right)^d \\
&\leq \exp\left(-\frac{1}{2p_\alpha}\left(\sqrt{\beta} - (1+p_\alpha)\sqrt{\frac{dq_\alpha}{2}}\right)^2\right)\exp\left(\frac{(1+p_\alpha)^2 dq_\alpha}{4p_\alpha} + d\log p_\alpha\right) \\
&\leq \exp\left(\frac{(1+\vartheta)^2}{4\vartheta}Q\right)\exp\left(-\frac{1}{2}\left(\sqrt{\beta} - \sqrt{2Q}\right)^2\right),
\end{aligned}$$

for $Q = dq_\alpha$, where the penultimate line follows from considering F such that $\left(1/p_\alpha - \frac{1}{p_\alpha}\sqrt{\frac{\beta q_\alpha}{2d}}\right)^{-d} \leq \exp(\sqrt{\beta}F)$ and performing a Taylor Series expansion, and the final line follows from conditions (B.1.1) and (B.1.2).

Let $\sqrt{\beta} = \sqrt{2Q} + C\sqrt{\log n}$, some C. We now use the fact that $\frac{\Gamma(v,w)}{\Gamma(v)} \leq e^{-w}\left(1+\frac{w}{v}\right)^{v-1}$ (which can be shown using Jensen's Inequality), to assert that $q_\alpha \leq e^{-\alpha/2}(1+\alpha)^{-1/2}$, and that therefore

$$\mathbb{P}\left(A_\tau \geq \beta\right) \leq n^{-\frac{C^2}{2}}\exp\left(\frac{(1+\vartheta)^2}{4\vartheta}\frac{d}{\exp\left(\frac{\alpha}{2}\right)}\left(1+\alpha\right)^{-\frac{1}{2}}\right);$$

performing a Bonferroni correction for the position of τ in the data then gives the stated result. □

Proof of Lemma B.1.4: In the scenario where there is no true change, the difference in cost between selecting the point τ as a change (with affected subset $\mathcal{S} = \{1, \ldots, d\}$) and simply finding the (correct) null model is

$$\begin{aligned}
\text{Diff} &= \text{RSS}\left(\mathbf{y}_{1:n}; \emptyset\right) - \text{RSS}\left(\mathbf{y}_{1:n}; \tau; \mathcal{S}\right) - K \\
&:= W - K.
\end{aligned}$$

Note that here we use the notation $RSS(\mathbf{z}; \xi; \mathcal{T})$ to denote the residual sum of squares of the vector $\mathbf{z}$, while also enforcing a changepoint at time ξ with affected set $\mathcal{T}$. Note that $W \sim \chi^2_d$. By Lemma B.1.1, to establish the result we require K and x such that

$$\begin{aligned}
K &= d + 2\sqrt{xd} + 2x \\
\exp\left(-x\right) &= n^{-\frac{J}{2}-\epsilon/2},
\end{aligned}$$

giving $x = (J+\epsilon)/2\log n = \beta/2$, and consequently $K = d + \sqrt{2\beta d} + \beta$ as required. □

We later use these lemmas to establish Theorem 4.3.1 and Corollary 4.3.3. Before this, we give further results which are needed in establishing the other central result of Chapter 4.

Lemma B.1.5. *Assume that we are in the same setting as for Lemma B.1.3, except now we have that $\mu_{i,1} \neq \mu_{i,2}$ whenever $i \in \mathcal{S} \subseteq \{1,\ldots,d\}$. For $i \in \mathcal{S}$, let $\Delta_i := |\mu_{i,2} - \mu_{i,1}|$. Then for $\delta > 0$ and $a = \max\{n,d\}$, a sparse changepoint will be detected by SUBSET with probability greater than $1-(a)^{-\delta}$ providing that*

$$\sum_{i\in\mathcal{S}} (\Delta_i)^2 \geq \frac{4\delta\log a + \beta + |\mathcal{S}|(\alpha-1) + 2\sqrt{\delta\log a\,((2\alpha-1)|\mathcal{S}| + 2\beta + 4\delta\log a)}}{n\theta(1-\theta)},$$

where here we have that $\theta = \frac{\tau}{n}$ is fixed strictly between 0 and 1.

Lemma B.1.6. *Assume that we are in the same setting as for Lemma B.1.5, except with the threshold penalty regime. Then, again with probability greater than $1-a^{-\delta}$, for $2 > \delta > 0$ and $a = \max\{n,d\}$, providing that*

$$\sum_{i=1}^{d} (\Delta_i)^2 \geq \frac{4\delta\log a + K - d + 2\sqrt{\delta\log a\,(4\delta\log a + 2K - d)}}{n\theta(1-\theta)}$$

a changepoint will be detected in the dense setting.

Proof of Lemma B.1.5: Suppose there is a true change at location τ which affects a non-empty, sparse subset $\mathcal{S} \subset \{1,\ldots,d\}$ of variates, such that the magnitude of change in variate i is Δ_i. We compare the cost of fitting no change in such a scenario against the cost of fitting the truth; i.e. let

$$\text{Diff} := \sum_{i\in\mathcal{S}} D_{i,\tau} - \beta - |\mathcal{S}|\alpha,$$

where $D_{i,\tau}$ is as defined in Chapter 4. Note that $D_{i,\tau} \sim \chi^2_1\left(n\theta(1-\theta)(\Delta_i)^2\right)$, so

$$\text{Diff} + \beta + |\mathcal{S}|\alpha \sim \chi^2_{|\mathcal{S}|}\left(n\theta(1-\theta)\sum_{i\in\mathcal{S}}(\Delta_i)^2\right).$$

Therefore, by Lemma B.1.2, letting $\gamma = n\theta(1-\theta)\sum_{i\in\mathcal{S}}(\Delta_i)^2$

$$\mathbb{P}\left(\text{Diff} + \beta + |\mathcal{S}|\alpha \geq |\mathcal{S}| + \gamma - 2\sqrt{(|\mathcal{S}| + 2\gamma)\, y}\right) > 1 - \exp(-y).$$

Note that if Diff > 0, then a changepoint will be detected in probability. Therefore, we require that

$$\gamma \geq 4y + \beta + |\mathcal{S}|(\alpha - 1) + \sqrt{4y\left((2\alpha - 1)|\mathcal{S}| + 2\beta + 4y\right)}.$$

We may set $y = \delta \log a$, for $a = \max\{n, d\}$, to give that $\mathbb{P}(\text{Diff} > 0) > 1 - a^{-\delta}$, providing that

$$\sum_{i\in\mathcal{S}}(\Delta_i)^2 \geq \frac{4\delta\log a + \beta + |\mathcal{S}|(\alpha-1) + 2\sqrt{\delta\log a\left((2\alpha-1)|\mathcal{S}| + 2\beta + 4\delta\log a\right)}}{n\theta\left(1-\theta\right)},$$

as required. □

Proof of Lemma B.1.6: When comparing a fit at the true location $\tau = \theta n$ under a total penalty of K to the null fit, the difference in cost (in favour of the non-null fit) is distributed as a non-central chi-squared distribution with d degrees of freedom and non-centrality parameter $n\theta\left(1-\theta\right)\sum_{i=1}^{d}(\Delta_i)^2$. By Lemma B.1.2 and the definition of K, we therefore see that setting $\nu - 2\sqrt{y}\sqrt{d+2\nu} \geq 2\sqrt{dx} + 2x$ for $\nu = n\theta\left(1-\theta\right)\sum_{i=1}^{n}(\Delta_i)^2$ gives that $\mathbb{P}\left(\chi^2_d(\nu) > K\right) \geq 1 - \exp\left(-y\right)$.

Resolving the inequality $\nu - 2\sqrt{y}\sqrt{d+2\nu} \geq 2\sqrt{dx} + 2x$ gives that

$$\nu \geq 4y + 2x + 2\sqrt{xd} + 2\sqrt{y\left(4y + \left(\sqrt{4x} + \sqrt{d}\right)^2\right)}; \tag{B.1.3}$$

as $x = \beta/2$, and setting $y = \delta\log a$, (B.1.3) becomes

$$\sum_{i=1}^{d}(\Delta_i)^2 \geq \frac{4\delta\log a + K - d + 2\sqrt{\delta\log a\left(4\delta\log a + 2K - d\right)}}{n\theta\left(1-\theta\right)},$$

as required. □

With these lemmas, we are now in a position to prove the results of Chapter 4.

B.2 Proofs of Main Results

In this section, we combine the preliminary results of Section B.1 to give proofs of the results stated in Chapter 4.

Proof of Theorem 4.3.1: From Lemma B.1.3, letting $g(n,d) = d$ and $C = \sqrt{J+\varrho}$, some $\varrho > 0$, gives that $\alpha = 2\log d$ and $\sqrt{\beta} = \sqrt{2d\frac{\Gamma\left(\frac{1}{2},\log d\right)}{\Gamma\left(\frac{1}{2}\right)}} + \sqrt{J+\varrho}\sqrt{\log n}$, so that in the sparse setting

$$\mathbb{P}\left(\max_t S_{1,t} > 0\right) \leq n^{1-\frac{J}{2}-\varrho/2} \exp\left(\frac{(1+\vartheta)^2}{4\vartheta}\frac{1}{\sqrt{1+2\log d}}\right),$$

where here $\frac{\gamma\left(\frac{1}{2},\frac{\alpha}{2}\right)}{\Gamma\left(\frac{1}{2}\right)} \geq \vartheta > 0$.

As we have $\frac{\Gamma(s,x)}{\Gamma(s)} \leq e^{-x}\left(1+\frac{x}{s}\right)^{s-1}$ for $0 < s < 1$, we have that

$$\vartheta \leq 1 - \frac{1}{d\left(1+\log d\right)^{\frac{1}{2}}}$$

so that, for example, by taking $d = 2$, we may bound $\exp\left(\frac{(1+\vartheta)^2}{4\vartheta}\frac{1}{\sqrt{1+2\log d}}\right)$ above by an absolute constant $\forall d \geq 2$. For $\max_t S_{2,t}$, we use Lemma B.1.4, and the result for SUBSET in both settings follows as $S_t = \max_t\{S_{1,t}, S_{2,t}\}$. □

Proof of Corollary 4.3.3: This follows from the proof of Theorem 4.3.1 by taking a further Bonferroni correction in both the sparse and dense settings. □

Proof of Theorem 4.3.2: In the sparse setting, we may directly apply Lemma B.1.5, while in the dense setting we may directly apply Lemma B.1.6. Note that the condition in Lemma B.1.5 resolves to give the required statement by setting $K_{\mathcal{S}} = \beta + |\mathcal{S}|\alpha$. □

B.3 Post-Processing and Computational Discussion

As discussed in Section 4.3.4, a post-processing step is required in the SUBSET procedure. This ensures that masking between different changepoints present in the data does not cause misspecification in the estimates of the affected sets at each changepoint. We detail this post-processing procedure in Algorithm 6.

Algorithm 6 Post-processing step for the SUBSET procedure.

Data: A multivariate dataset, $\mathbf{y}_{1:n}$; a β and $\mathcal{C}(.)$ as for Algorithm 3; a set of candidates returned by Algorithm 3, $0 = \xi_0 < \xi_1 < \ldots < \xi_q < \xi_{q+1} = n$.

Result: An estimated set of changepoints $\hat{\tau}_1, \ldots, \hat{\tau}_{\hat{m}}$ and corresponding estimated affected sets $\hat{\mathcal{S}}_1, \ldots, \hat{\mathcal{S}}_{\hat{m}}$.

Step 0: Set $\hat{\mathcal{S}}_1 = \ldots = \hat{\mathcal{S}}_q = \emptyset$, $\hat{\tau} = NULL$;

for $i \in \{1, \ldots, d\}$ **do**
  $F = (-\beta, 0, \ldots, 0)$;
  for $j \in \{1, \ldots, q+1\}$ **do**
    $F[j+1] = \min_{1 \leq k \leq j} \left[F[k] + \mathcal{C}\left(y_{i,\xi_{k-1}:\xi_j}\right) + \beta\right]$;
    $r = \arg\min_{1 \leq k \leq j} \left[F[k] + \mathcal{C}\left(y_{i,\xi_{k-1}:\xi_j}\right) + \beta\right]$;
    $\hat{\mathcal{S}}_{r-1} = (\hat{\mathcal{S}}_{r-1}, \{i\})$
  end
end
for $j \in \{1, \ldots, q\}$ **do**
  if $\hat{\mathcal{S}}_j \neq \emptyset$ **then**
    $\hat{\tau} = (\hat{\tau}, \xi_j)$
  end
end

Note that this procedure, which closely parallels the Optimal Partitioning of Jackson et al. (2005), has complexity of $\mathcal{O}(q^2 d)$, where q is the number of candidate changepoint locations returned by SUBSET. Indeed, employing a pruning step as per the PELT procedure of Killick et al. (2012) results in an expected cost of $\mathcal{O}(qd)$. As shown in Tickle et al. (2018), this can be improved further to a worst-case cost of $\mathcal{O}(qd)$ using parallelisation. Therefore, the worst-case computational complexity of the post-processing step is $\mathcal{O}(nd)$.

Given that the SUBSET procedure uses an approach very similar to Wild Binary Segmentation 2, simulating M intervals at each stage, the worst-case computational cost of SUBSET is not dominated by the post-processing step, and is $\mathcal{O}(dn(\log n)^2)$;

see Fryzlewicz (2019) for details.

B.4 Simulation Study: Additional Materials

Here, we give further results from the simulation study in both the at most one change and multiple change settings. We begin by considering the power of the methods in the single change setting. This is done from a slightly different perspective to that discussed in Chapter 4. Tables B.1-B.3 give a snapshot into the Type II Errors in the single change in mean setting under Gaussian noise (again we take $\sigma^2 = 1$). The entries correspond to 'critical change magnitudes'. That is, in a system where all variates undergoing a change alter by the same amount, the minimum value of the change in mean required for the method to locate a changepoint at least 95% of the time. For these experiments, the location of the changepoint ($\theta = \tau/n$), the number of variates (d) and the density of the changepoint were all altered. However, the length of the series, n, was fixed at 1000.

We can infer several empirical properties of the methods from Tables B.1-B.3. For denser changes, where a higher proportion of variates are affected by the change, the method which seems to perform best is Mean. This confirms the intuition of Section 4.3.3 of Chapter 4. For sparser regimes, Max and Bin-Weight appear to have the best performance. This is again in line with the commentary of Section 4.3.3. We remark, however, that Inspect and, in particular, SUBSET shadow this best performance in both the sparse and dense settings very closely. SUBSET is regularly the best performing method behind Mean in the most dense cases, and a close third behind Max and Bin-Weight in cases of medium or high sparsity. This suggests that Inspect and SUBSET are most effective at giving a 'balanced' performance. In addition, we highlight the context of these results in light of the respective Type I Errors. The other methods each have a 5% Type I Error, while SUBSET has a negligible such error in all cases, given that the penalty values for SUBSET are not calculated empirically.

It was for these experiments that computation time was recorded. The results of

Critical $\Delta\mu$ Values		5 Variates			10 Variates		
Location (θ)	Method	100%	60%	20%	100%	50%	10%
0.050	SUBSET	0.51	0.62	1.03	0.41	0.54	1.07
	Mean	**0.33**	**0.46**	1.10	**0.28**	**0.41**	1.37
	Max	0.48	0.52	**0.75**	0.42	0.50	0.79
	BW	0.46	0.53	0.76	0.42	0.51	**0.78**
	Inspect	0.42	0.55	1.00	0.37	0.49	0.90
0.081	SUBSET	0.41	0.51	0.83	0.33	0.44	0.85
	Mean	**0.27**	0.37	0.86	**0.22**	**0.34**	1.07
	Max	0.38	**0.42**	**0.60**	0.33	0.40	**0.63**
	BW	0.38	**0.42**	**0.60**	0.33	0.41	**0.63**
	Inspect	0.30	0.45	0.65	0.30	0.36	0.75
0.184	SUBSET	0.29	0.35	0.61	0.23	0.31	0.59
	Mean	**0.20**	**0.26**	0.60	**0.15**	**0.24**	0.70
	Max	0.26	0.30	**0.44**	0.24	0.29	0.44
	BW	0.26	0.31	**0.44**	0.23	0.29	**0.43**
	Inspect	0.22	**0.26**	0.52	0.20	0.26	0.52
0.266	SUBSET	0.27	0.32	0.52	0.20	0.27	0.53
	Mean	**0.18**	**0.23**	0.52	**0.13**	**0.21**	0.66
	Max	0.22	0.26	**0.40**	0.21	0.25	**0.40**
	BW	0.23	0.26	**0.40**	0.20	0.26	**0.40**
	Inspect	0.21	**0.23**	0.44	0.16	0.22	0.50
0.383	SUBSET	0.23	0.28	0.47	0.18	0.25	0.48
	Mean	**0.16**	**0.21**	0.48	**0.12**	**0.18**	0.64
	Max	0.21	0.24	0.35	0.18	0.23	**0.35**
	BW	0.21	0.23	**0.34**	0.19	0.23	**0.35**
	Inspect	0.19	0.25	0.39	0.15	0.21	0.44

Table B.1: The critical (i.e. smallest observed) values for $\Delta\mu$ at which each of the methods exhibits a Type II Error of 0.05 or less. The percentages correspond to the density of the changes across the variates. Bold entries show the best performing algorithm. 200 repetitions were simulated in each case.

Critical $\Delta\mu$ Values		50 Variates				100 Variates				
Location (θ)	Method	100%	50%	10%	6%	100%	50%	10%	5%	1%
0.050	SUBSET	0.22	0.31	0.62	0.72	0.18	0.25	0.52	0.63	1.12
	Mean	**0.16**	**0.24**	0.65	0.96	**0.13**	**0.19**	0.52	0.86	3.89
	Max	0.35	0.41	0.59	**0.65**	0.34	0.37	0.53	0.58	**0.88**
	BW	0.91	0.94	1.11	1.17	0.25	0.31	**0.46**	**0.53**	0.89
	Inspect	0.25	0.33	**0.58**	0.66	0.23	0.30	0.51	0.68	1.32
0.081	SUBSET	0.18	0.26	0.51	0.58	0.14	0.20	0.42	0.51	0.88
	Mean	**0.13**	**0.19**	0.53	0.82	**0.11**	**0.16**	0.40	0.68	2.95
	Max	0.29	0.33	0.47	**0.52**	0.26	0.30	0.41	0.46	**0.68**
	BW	0.72	0.76	0.90	0.96	0.21	0.25	**0.36**	**0.44**	0.70
	Inspect	0.19	0.25	**0.46**	0.55	0.17	0.22	0.38	0.54	1.04
0.184	SUBSET	0.13	0.18	0.34	0.40	0.10	0.15	0.29	0.36	0.63
	Mean	**0.09**	**0.14**	0.39	0.57	**0.08**	**0.11**	0.28	0.46	1.98
	Max	0.19	0.23	**0.32**	**0.34**	0.18	0.21	0.30	0.34	**0.49**
	BW	0.50	0.54	0.62	0.65	0.14	0.17	**0.25**	**0.31**	0.52
	Inspect	0.12	0.15	0.33	0.37	0.10	0.13	**0.25**	0.32	0.71
0.266	SUBSET	0.11	0.16	0.31	0.35	0.09	0.13	0.25	0.30	0.53
	Mean	**0.08**	**0.12**	0.32	0.48	**0.07**	**0.09**	0.24	0.42	1.74
	Max	0.17	0.21	**0.28**	**0.30**	0.16	0.19	0.25	0.29	**0.43**
	BW_α	0.44	0.46	0.55	0.56	0.13	0.15	**0.22**	**0.26**	0.45
	Inspect	0.10	0.14	0.29	0.33	0.09	0.13	0.23	0.30	0.67
0.383	SUBSET	0.10	0.14	0.28	0.31	0.08	0.11	0.23	0.28	0.49
	Mean	**0.07**	**0.11**	0.30	0.45	**0.06**	**0.09**	0.23	0.37	1.59
	Max	0.16	0.18	**0.25**	**0.26**	0.14	0.17	0.23	0.26	**0.38**
	BW	0.40	0.43	0.49	0.51	0.11	0.14	0.21	**0.24**	**0.38**
	Inspect	0.08	0.12	**0.25**	0.33	0.07	**0.09**	**0.20**	0.25	0.56

Table B.2: The critical (i.e. smallest observed) values for $\Delta\mu$ at which each of the methods exhibits a Type II Error of 0.05 or less. The percentages correspond to the density of the changes across the variates. Bold entries show the best performing algorithm. 200 repetitions were simulated in each case.

Critical $\Delta\mu$ Values		500 Variates					1000 Variates					
Location (θ)	Method	100%	50%	10%	5%	1%	100%	50%	10%	5%	1%	0.5%
0.050	SUBSET	0.11	0.15	0.34	0.46	0.72	0.09	0.13	0.28	0.39	0.63	0.73
	Mean	**0.09**	**0.12**	**0.31**	0.47	1.69	**0.08**	**0.11**	**0.25**	0.37	1.30	2.46
	Max	0.29	0.33	0.43	0.48	0.74	0.28	0.36	0.42	0.46	0.62	0.70
	BW	0.18	0.23	0.34	**0.40**	**0.65**	0.16	0.20	0.30	**0.36**	**0.55**	**0.66**
	Inspect	0.16	0.21	0.39	0.47	0.91	0.15	0.22	0.35	0.45	0.76	1.10
0.081	SUBSET	0.09	0.12	0.27	0.37	0.58	0.07	0.10	0.22	0.31	0.51	0.59
	Mean	**0.07**	**0.10**	**0.24**	0.37	1.35	**0.06**	**0.08**	**0.19**	**0.29**	0.95	1.83
	Max	0.23	0.25	0.35	0.40	0.58	0.25	0.28	0.36	0.40	0.53	0.58
	BW	0.14	0.18	0.26	**0.32**	**0.51**	0.13	0.16	0.25	0.30	**0.45**	**0.53**
	Inspect	0.12	0.15	0.29	0.37	0.76	0.11	0.15	0.26	0.33	0.67	0.87
0.184	SUBSET	0.06	0.09	0.19	0.26	0.40	0.05	0.07	0.16	0.22	0.35	0.41
	Mean	**0.05**	**0.07**	**0.17**	0.26	0.96	**0.04**	**0.06**	**0.13**	**0.20**	0.69	1.27
	Max	0.15	0.18	0.25	0.28	0.39	0.17	0.20	0.25	0.28	0.37	0.41
	BW	0.10	0.12	0.19	**0.23**	**0.35**	0.09	0.11	0.17	**0.20**	**0.31**	**0.39**
	Inspect	0.07	0.10	**0.17**	**0.23**	0.45	0.06	0.08	0.16	0.21	0.42	0.63
0.266	SUBSET	0.06	0.08	0.16	0.22	0.36	0.05	0.06	0.14	0.19	0.32	0.36
	Mean	**0.05**	**0.06**	**0.14**	0.22	0.80	**0.04**	**0.05**	**0.12**	**0.17**	0.58	1.07
	Max	0.15	0.16	0.20	0.23	0.34	0.16	0.18	0.22	0.24	0.32	0.36
	BW	0.09	0.11	0.16	**0.20**	**0.31**	0.08	0.10	0.15	0.18	**0.28**	**0.33**
	Inspect	0.06	0.08	0.15	**0.20**	0.45	0.05	0.06	0.13	**0.17**	0.36	0.54
0.383	SUBSET	0.05	0.07	0.15	0.21	0.33	**0.04**	0.06	0.13	0.18	0.28	0.33
	Mean	**0.04**	**0.06**	**0.13**	0.20	0.77	**0.04**	**0.05**	**0.11**	0.16	0.54	1.03
	Max	0.13	0.15	0.19	0.22	0.31	0.14	0.16	0.21	0.23	0.28	0.34
	BW	0.08	0.10	0.14	**0.18**	**0.28**	0.07	0.09	0.14	0.16	**0.24**	**0.30**
	Inspect	0.05	**0.06**	**0.13**	**0.18**	0.40	**0.04**	0.06	0.12	**0.15**	0.33	0.47

Table B.3: The critical (i.e. smallest observed) values for $\Delta\mu$ at which each of the methods exhibits a Type II Error of 0.05 or less. The percentages correspond to the density of the changes across the variates. Bold entries show the best performing algorithm. 200 repetitions were simulated in each case.

this are shown in Table B.4 for the case when $\theta = 0.184$. In addition to the $n = 1000$ setting, we also examined the $n = 10000$ and $n = 100000$ cases for the purposes of exploring the scalability of each of these procedures.

The final additional simulation study results included here concern the detection of multiple changes in the negative binomial setting. In Section 4.4.3, we examined the missed change rate of each of the SUBSET methods when the over-dispersion parameter was kept fixed at 20. The full results for all five methods are detailed in Table B.5. Note that the SUBSET column is simply the set of results from Table 4.2.

We now give the equivalent results for an over-dispersion parameter of 3. These are summarised in Table B.6. Note that, unlike for the other simulations in Chapter 4 and in this appendix, we use an empirically calculated value for the β (and K) penalty for SUBSET. This is because the default Gaussian-based penalties lead to a higher false alarm rate in this instance, given that a lower over-dispersion rate corresponds to the noise departing more markedly from behaving in a sub-Gaussian fashion. We remark that this was not an issue in the single change setting, given the relatively low number of intervals drawn uniformly across the data sequence on which the SUBSET test statistic was then calculated. Table B.6 seems to suggest that SUBSET, Max and Inspect are roughly comparable in terms of locating the changes across these simulations. However, this is again in the context of the average number of false alarms which each method incorrectly places into the system.

Computation Time (s)		Method				
n	d	SUBSET	Mean	Max	BW	Inspect
1000	5	0.00114	0.00275	0.00106	0.00311	**0.00055**
	10	0.00148	0.00148	**0.00134**	0.00341	0.00199
	50	**0.00450**	0.00488	0.00857	0.00749	0.00740
	100	**0.00990**	0.01007	0.01895	0.01295	0.01913
	500	0.05116	0.04706	0.05153	**0.04421**	0.21236
	1000	0.08588	0.11421	**0.07538**	0.11082	0.77755
10000	5	**0.01153**	0.01222	0.02524	0.02254	0.00574
	10	**0.01474**	0.04726	0.03608	0.03042	0.01722
	50	0.05286	**0.04666**	0.09395	0.09434	0.07582
	100	0.10266	**0.08522**	0.09923	0.09680	0.14087
	500	0.45512	0.45858	**0.44463**	0.48023	0.79061
	1000	0.97533	**0.73940**	0.76509	0.83851	2.10629
100000	5	0.13211	0.13300	0.29013	**0.13022**	-
	10	**0.15759**	0.17302	0.17461	0.19873	-
	50	0.56481	**0.55088**	0.97380	0.58404	-
	100	0.95607	**0.92535**	0.99223	1.00181	-
	500	5.59921	**4.41211**	4.51795	5.09488	-
	1000	12.5877	**8.09231**	8.13885	8.69370	-

Table B.4: The average time taken (across 200 repetitions of the method) by each method, with the changepoint at proportionate temporal point 0.184, with $\Delta\mu = 1$, and 50% of variates undergoing a change (60% in the case of $d = 5$). The Inspect times for $n = 100000$ are not recorded here due to integer overflow preventing the method from running for these larger examples. Bold entries show the best performing algorithm. 200 repetitions were simulated in each case.

Average Number Missed (Average False Alarms)	Method				
Scenario	SUBSET	Mean	Max	BW	Inspect
F	0.07 (**0.02**)	0.01 (114)	0.01 (212)	**0.00** (387)	**0.00** (188)
G	0.10 (**0.01**)	0.09 (90.9)	0.02 (214)	0.02 (393)	**0.00** (151)
H	0.29 (**0.02**)	0.86 (41.7)	0.02 (179)	0.01 (394)	**0.00** (117)
I	0.16 (**0.01**)	1.25 (1.40)	0.02 (146)	**0.00** (394)	0.01 (0.79)
J	0.19 (**0.02**)	1.43 (1.37)	0.02 (142)	**0.01** (387)	0.03 (0.79)

Table B.5: The average number of changes missed by each of the methods in the negative binomial setting with an over-dispersion parameter of 20 for each variate; a starting success probability of 0.5 for each variate; $d = n = 1000$ fixed in all cases; and $\Delta p = 0.1$ for any variate undergoing a change. Each of the scenarios F, G, H, I and J has 3 changepoints, and the percentage of variates affected by each change in each scenario is discussed at the beginning of Section 4.4.2. Bold entries show the best performing algorithm. 200 repetitions were simulated in each case.

Average Number Missed (Average False Alarms)		Method				
Scenario	d	SUBSET	Mean	Max	BW	Inspect
F	1000	0.09 (**1.22**)	0.04 (95.2)	0.46 (70.4)	0.00 (403)	**0.00** (583)
G	1000	0.91 (**1.03**)	0.31 (54.7)	0.88 (38.8)	0.08 (404)	**0.00** (611)
H	1000	1.45 (**1.72**)	1.34 (24.1)	1.64 (12.9)	0.22 (412)	**0.00** (621)
I	1000	1.35 (1.72)	2.71 (**1.04**)	1.66 (3.60)	0.13 (409)	**0.00** (625)
J	1000	0.92 (3.00)	2.40 (**0.61**)	1.91 (2.56)	0.13 (405)	**0.01** (624)

Table B.6: The average number of changes missed by each of the methods in the negative binomial setting with an over-dispersion parameter of 3 for each variate; a starting success probability of 0.5 for each variate; $d = n = 1000$ fixed in all cases; and $\Delta p = 0.1$ for any variate undergoing a change. Each of the scenarios F, G, H, I and J has 3 changepoints, and the percentage of variates affected by each change in each scenario is discussed at the beginning of Section 4.4.2. Bold entries show the best performing algorithm. 200 repetitions were simulated in each case.

B.5 Additional Material on the Analysis of the Global Terrorism Database

This section contains many of the additional details on our analysis of the Global Terrorism Database (GTD). We begin with some basic visual representations of the data. Figure B.1 shows the world divided into the twelve regions as per the GTD. These regions are henceforth referred to as: Australasia & Oceania (Au & Oc), Central America & Caribbean (C.Am & C), Central Asia (C.As), East Asia (E.As), Eastern Europe (E.Eu), Middle East & North Africa (M.E. & N.Af), North America (N.Am), South America (S.Am), South Asia (S.As), Southeast Asia (SE.As), Sub-Saharan Africa (SS.Af) and Western Europe (W.Eu).

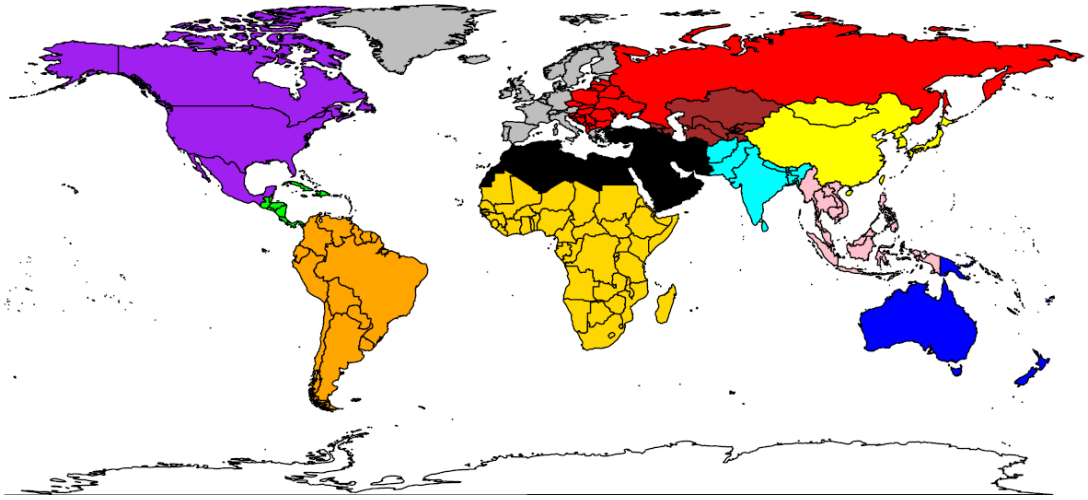

Figure B.1: Nations of the world divided into twelve geographical groups as per the Global Terrorism Database (GTD). Produced with the aid of the `maps` package of Becker and Wilks (2018). Political boundaries are correct as of 2015.

From Figure B.2, it is clear that there are points in time where abrupt changes occur in the terrorism incident rate for various series. It is less clear as to whether changes which share a common cause are present. As stated in Section 4.5, we assume a negative binomial likelihood for each of the twelve sequences within the time series. A changepoint in this context is therefore defined as a month in which the probability of a terrorist attack changes. We track the value of the over-dispersion parameters in each of the twelve regions using a method of moments estimator,

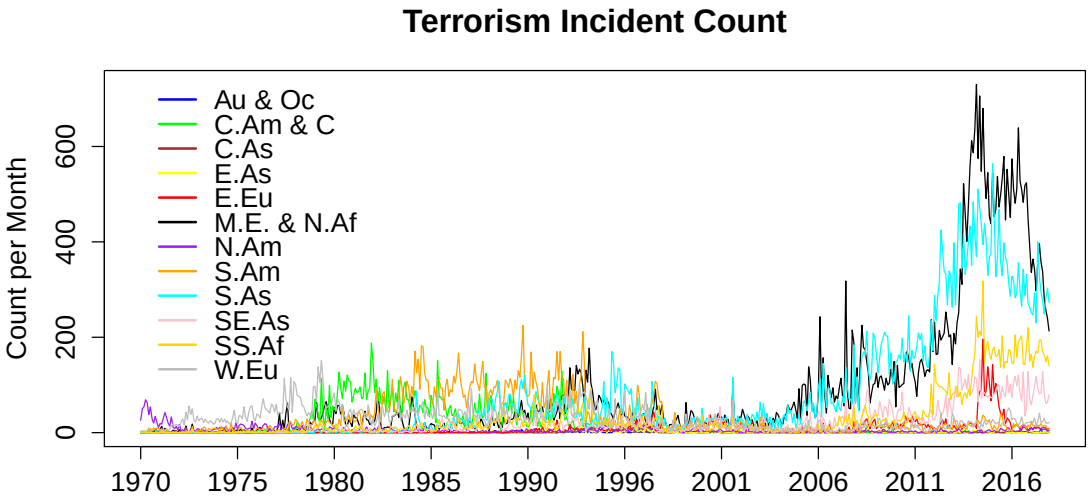

Figure B.2: Terrorism incident count per month for each of the 12 regions in Figure B.1. Note that the series' colours match those of the corresponding geographical regions in Figure B.1.

given the computational challenge of accurately computing the maximum likelihood estimators.

After running SUBSET through the time series, the estimated changepoints and the corresponding affected sets of regions were computed. These are shown in Table B.7. For an alternative visualisation, with the estimated changes for each region superimposed over the raw count data, see Figure B.3. By comparison, Figure B.4 shows the results of applying a univariate method to each series individually. In this case, the univariate method used is the minimisation of the penalised univariate negative binomial cost function using dynamic programming.

Several salient features of the dataset are revealed by this analysis. Firstly, we note that there are many similarities between the changes found by the univariate method and SUBSET. For several of the series (for example, Western Europe), the same number of changepoints are found, with broadly the same change locations. However, in general, we see that SUBSET is more parsimonious. In addition, by its nature, changepoints which occur in different series at the same time are more readily identified by SUBSET. For example, the most dense changepoint (following post-processing) located using SUBSET is that of January 1998. Note that this month corresponds to a change in the data collection methods for the GTD for the "GTD2"

Dates	Regions
Sep 1971	E.Eu, N.Am, W.Eu
Feb 1975	C.Am & C, M.E. & N.Af, SS.Af, W.Eu
Dec 1977	C.Am & C, E.As, S.Am, SE.As
Sep 1978	C.Am & C, E.As, M.E. & N.Af, S.As, SS.Af
Apr 1980	N.Am, S.Am, W.Eu
Mar 1984	Au & Oc, S.As, SE.As
Jan 1988	E.As, E.Eu, S.As, SS.Af
Mar 1990	E.Eu, M.E. & N.Af
Jan 1991	C.As
Feb 1992	C.Am & C
Jul 1994	N.Am, S.Am
May 1995	M.E. & N.Af
Apr 1996	E.Eu
Jan 1998	Au & Oc, C.Am & C, E.As, N.Am, S.Am, S.As, SS.Af, W.Eu
Mar 1999	C.As
Aug 2003	E.Eu, S.Am, W.Eu
Mar 2005	M.E. & N.Af, S.As, SE.As
Jun 2007	SS.Af
Apr 2008	E.Eu, S.Am
Jul 2011	S.As, SS.Af
Mar 2012	W.Eu
Jan 2013	E.As, M.E. & N.Af, SE.As
Jan 2014	Au & Oc, E.Eu
Sep 2015	E.As, E.Eu, N.Am

Table B.7: Changepoints found within the count data of terrorist incidents per month using the SUBSET procedure. The regions column corresponds to those areas which are said to be affected by the corresponding changepoint.

phase. Other changepoints of interest found by SUBSET include several "staggered" changepoints at the end of the 1980s and the beginning of the 1990s. These appear to correspond to the end of the Cold War. More recent changepoints seem to align with significant events in, for example, the Arab Spring uprising and the conflict in Ukraine.

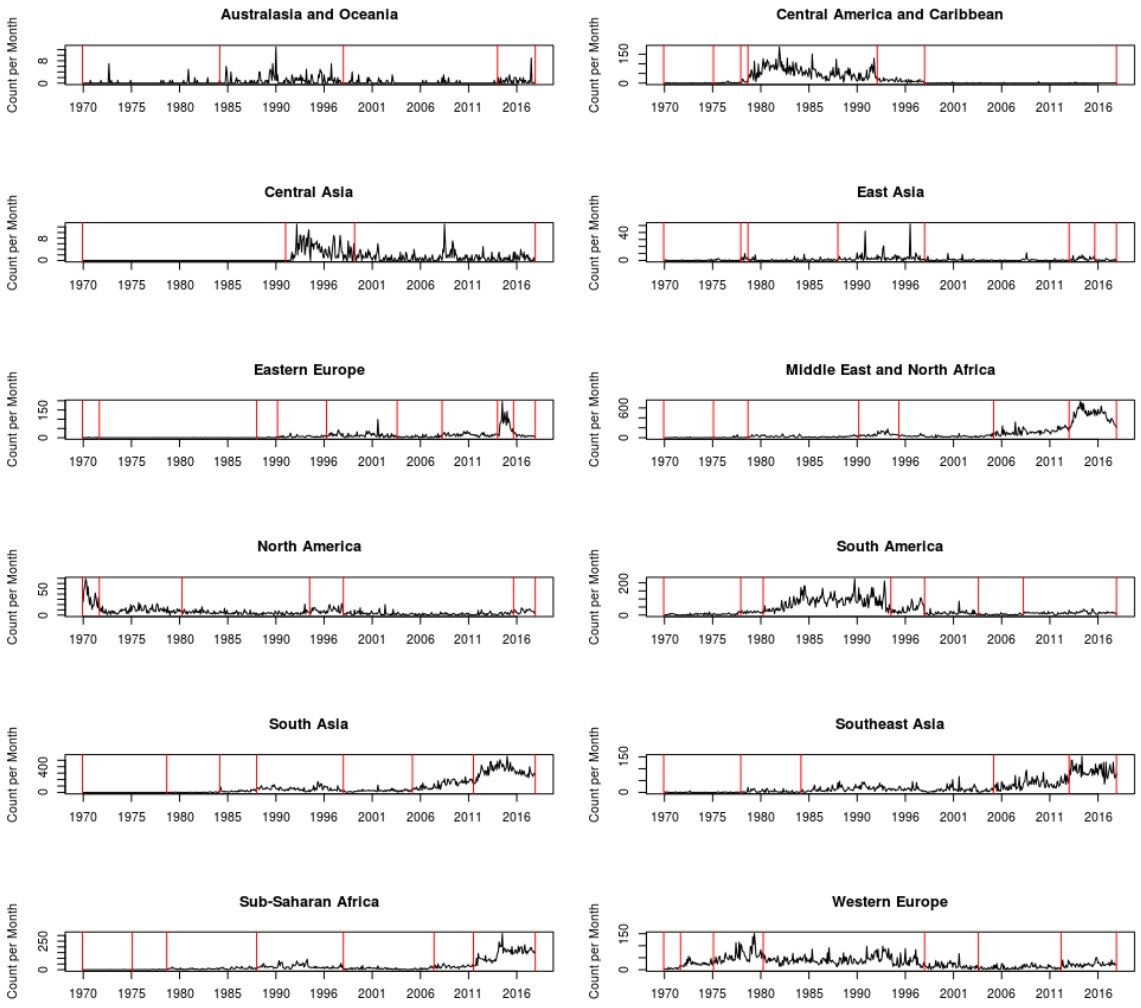

Figure B.3: Incident count for each region between 1970 and 2017, with changes found by the SUBSET method overlaid as red vertical lines.

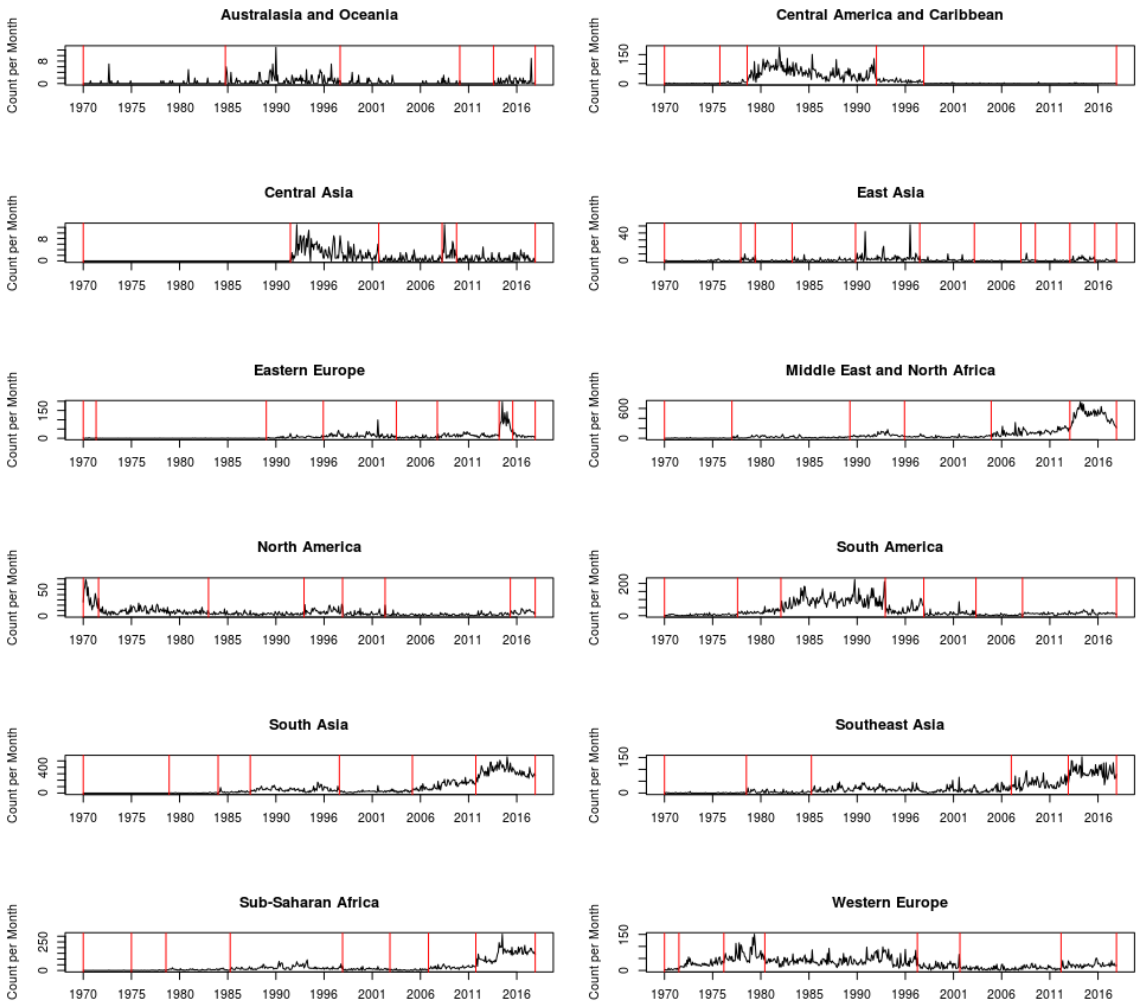

Figure B.4: Incident count for each region between 1970 and 2017, with changes for individual series found by a univariate method overlaid as red vertical lines.

Appendix C

OMEN

C.1 Proof of the False Alarm Result

Proof of Lemma 5.3.1: We first note that, whatever the behaviour of the stream following a changepoint, the transformed stream for each variate under Box-Muller is necessarily sub-Gaussian. We can therefore consider the problem of the method incorrectly labelling a changepoint from a standard normal stream to a non-standard normal stream.

Without loss of generality, we consider the i^{th} variate, such that the transformed streams of interest are

$$a_{i,t} = \sqrt{-2\log u_{i,t}}\cos(2\pi z_{i,t})$$
$$b_{i,t} = \sqrt{-2\log u_{i,t}}\sin(2\pi z_{i,t}),$$

for $t = 1, \ldots, N$, which we take as independently standard normal. Within a memory window of length ω with start time $l+1$ and end time $l+\omega$, the test statistic for a change in the a-sequence at time k is

$$S(k; a_{i,(l+1):(l+\omega)}) = \sum_{q=k+1}^{l+\omega} a_{i,q}^2 - (\omega + l - k)\log\left(\sum_{q=k+1}^{l+\omega} a_{i,q}^2 - \frac{1}{\omega + l - k}\left(\sum_{q=k+1}^{l+\omega} a_{i,q}\right)^2\right)$$
$$+ (\omega + l - k)\log(\omega + l - k) - (\omega + l - k).$$

By Fisch et al. (2019a), we have that

$$\mathbb{P}\left(S(k; a_{i,(l+1):(l+\omega)}) > \frac{\omega + l - k}{(\omega + l - k) - 1}\left(2 + 2r + 2\sqrt{2r}\right)\right) \leq ee^{-r}.$$

Note that this result follows by computing the moment generation function of $S(k; a_{i,(l+1):(l+\omega)})$, which can in turn be found by noting that

$$\sum_{q=k+1}^{l+\omega} a_{i,q}^2 = \sum_{q=k+1}^{l+\omega}\left(a_{i,q} - \frac{1}{\omega + l - k}\sum_{q=k+1}^{l+\omega} a_{i,q}\right)^2 + \frac{1}{\omega + l - k}\left(\sum_{q=k+1}^{l+\omega} a_{i,q}\right)^2,$$

where the two terms on the right hand side are independent. After bounding the moment generating function appropriately, we can then find the appropriate Chernoff bound to give the result.

Using the bound gives

$$\frac{\omega + l - k}{(\omega + l - k) - 1}\left(2 + 2r + 2\sqrt{2r}\right) = \beta'.$$

Labelling $f = \omega + l - k$ for brevity, we obtain that

$$r = \frac{f-1}{2f}\beta' - \sqrt{\frac{f-1}{f}\beta' - 1}, \text{ for } f \in \{2, \ldots, \omega - 1\},$$

such that applying a Bonferroni correction gives

$$\mathbb{P}\left(\sup_f S(\omega + l - f; a_{i,(l+1):(l+\omega)}) > \beta'\right) \leq \sum_{f=2}^{\omega-1} ee^{-\left(\frac{f-1}{2f}\beta' - \sqrt{\frac{f-1}{f}\beta' - 1}\right)}.$$

We note that

$$\frac{f-1}{2f}\beta' - \sqrt{\frac{f-1}{f}\beta' - 1} > \frac{f-1}{2f}\beta' - \sqrt{\frac{f-1}{f}\beta'},$$

so that

$$\begin{aligned}\mathbb{P}\left(\sup_{2\leq f\leq\omega-1} S(\omega + l - f; a_{i,(l+1):(l+\omega)}) > \beta'\right) &< (\omega - 2)\, ee^{-\left(\frac{1}{4} - \frac{1}{\sqrt{2\beta'}}\right)\beta'} \\ &= \frac{(\omega-2)e}{\Lambda^{\left(\frac{1}{2} - \sqrt{\frac{2}{\beta'}}\right)}}.\end{aligned}$$

In addition we have that

$$\mathbb{P}\left(\sup_{1\leq l\leq(n-\omega)}\sup_{1\leq i\leq d}\sup_{2\leq f\leq\omega-1} S(\omega + l - f; a_{i,(l+1):(l+\omega)})\right) < \frac{d(n-\omega)(\omega-2)}{\Lambda^{\left(\frac{1}{2} - \sqrt{\frac{2}{\beta}}\right)}}e,$$

as required. □

C.2 Further Simulations - Examining Different ω Values

We now examine the same scenarios under the same conditions as in Section 5.4, except we now compare the output of OMEN under three different values for the learning window length, ω. Note that the values reported here for $\omega = 30$ are the same as for those given in Tables 5.1-5.3 in Section 5.4, which we include for comparison with the two other values we examine, namely $\omega = 10$ and $\omega = 50$.

Table C.1 examines the average number of false alarms triggered by OMEN under each of the three learning windows. We note that while the performance of $\omega = 50$ appears to best minimise the average number of false alarms, the improvement over $\omega = 30$ is marginal in almost all cases. On the other hand, taking $\omega = 30$ gives a noticeably lower false alarm rate than $\omega = 10$ in those situations where the number of variates was relatively small. Given that a larger ω leads to a higher per-iteration computational cost, this supports our choice of $\omega = 30$ for the simulations of Section 5.4.

Table C.2 gives the average number of missed changes in each of the scenarios under each of the learning window lengths. Meanwhile, Table C.3 gives the average location error for the estimated changepoints which were not previously labelled as false alarms. The results in both tables give a more mixed picture of the performance for increasing ω, with some indication that the greater parsimony of the method for greater ω had a detrimental effect on detecting more subtle true changes. However, we remark that these results should be seen in the context of those of Table C.1. For example, in scenario 5, the competitive detection performance of OMEN for $\omega = 10$ is at the expense of a false alarm rate at least twice that seen for $\omega = 30$.

Average False Alarms	Method					
OMEN	5 Variates		10 Variates		100 Variates	
Scenario, ω	(D, D, D)	(M, M, M)	(D, D, D)	(M, M, M)	(D, D, D)	(M, M, M)
1, $\omega = 10$	0.15	0.15	0.03	0.03	**0.00**	**0.00**
1, $\omega = 30$	0.01	0.01	**0.00**	**0.00**	**0.00**	**0.00**
1, $\omega = 50$	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**
2, $\omega = 10$	0.12	0.12	0.06	0.06	**0.00**	**0.00**
2, $\omega = 30$	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**
2, $\omega = 50$	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**
3, $\omega = 10$	0.22	0.42	0.05	0.21	**0.00**	**0.00**
3, $\omega = 30$	**0.01**	**0.10**	**0.00**	0.10	**0.00**	**0.00**
3, $\omega = 50$	**0.01**	0.11	**0.00**	**0.09**	**0.00**	**0.00**
4, $\omega = 10$	1.05	1.50	0.80	0.87	**0.00**	**0.00**
4, $\omega = 30$	0.44	0.32	0.19	0.17	**0.00**	**0.00**
4, $\omega = 50$	**0.15**	**0.19**	**0.05**	**0.10**	**0.00**	**0.00**
5, $\omega = 10$	0.76	0.88	0.37	0.51	**0.00**	**0.00**
5, $\omega = 30$	0.37	0.36	0.18	0.14	**0.00**	**0.00**
5, $\omega = 50$	**0.19**	**0.19**	**0.08**	**0.07**	**0.00**	**0.00**
6, $\omega = 10$	0.14	0.15	0.05	0.06	**0.00**	**0.00**
6, $\omega = 30$	0.02	0.02	**0.01**	0.01	**0.00**	**0.00**
6, $\omega = 50$	**0.00**	**0.01**	0.02	**0.00**	**0.00**	**0.00**
7, $\omega = 10$	0.78	0.67	0.39	0.27	**0.00**	**0.00**
7, $\omega = 30$	**0.01**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**
7, $\omega = 50$	**0.01**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**

Table C.1: The average number of false alarms incurred by OMEN under each of the scenarios for three different values of ω. Bold entries show the best performing ω value. 200 repetitions were simulated in each case.

Average Num Missed OMEN	Method					
	5 Variates		10 Variates		100 Variates	
Scenario, ω	(D, D, D)	(M, M, M)	(D, D, D)	(M, M, M)	(D, D, D)	(M, M, M)
3, $\omega = 10$	0.08	0.24	0.01	0.19	**0.00**	**0.00**
3, $\omega = 30$	**0.01**	**0.12**	**0.00**	0.13	**0.00**	**0.00**
3, $\omega = 50$	**0.01**	**0.12**	**0.00**	**0.10**	**0.00**	**0.00**
4, $\omega = 10$	1.16	1.77	**0.95**	1.76	**0.99**	2.61
4, $\omega = 30$	1.15	1.27	1.05	1.26	1.00	1.14
4, $\omega = 50$	**1.04**	**1.19**	1.02	**1.14**	1.00	**1.07**
5, $\omega = 10$	**0.70**	**1.18**	**0.43**	**1.43**	0.99	2.61
5, $\omega = 30$	1.11	1.78	0.92	1.73	**0.34**	**1.50**
5, $\omega = 50$	1.54	1.97	1.28	1.94	0.63	1.89
6, $\omega = 10$	**2.79**	**2.81**	**2.64**	**2.79**	**1.83**	**2.30**
6, $\omega = 30$	2.95	2.96	2.93	2.96	2.77	2.93
6, $\omega = 50$	2.98	2.96	2.92	2.94	2.73	2.88
7, $\omega = 10$	**2.92**	**2.93**	**2.95**	**2.97**	**3.00**	**3.00**
7, $\omega = 30$	2.98	2.99	3.00	3.00	**3.00**	**3.00**
7, $\omega = 50$	3.00	3.00	3.00	3.00	**3.00**	**3.00**

Table C.2: The average number of changes missed by OMEN under each of the scenarios for three different values of ω. Bold entries show the best performing ω value. 200 repetitions were simulated in each case.

Average Location Error OMEN	Method					
	5 Variates		10 Variates		100 Variates	
Scenario, ω	(D, D, D)	(M, M, M)	(D, D, D)	(M, M, M)	(D, D, D)	(M, M, M)
3, $\omega = 10$	**1.49**	**1.42**	**1.48**	**0.89**	**1.33**	**0.27**
3, $\omega = 30$	8.22	6.37	7.61	3.59	3.90	1.28
3, $\omega = 50$	13.7	9.78	11.8	6.01	6.09	1.97
4, $\omega = 10$	**1.89**	**1.75**	**1.54**	**1.27**	**0.89**	**0.44**
4, $\omega = 30$	8.68	7.61	7.65	4.86	2.34	0.98
4, $\omega = 50$	16.4	14.1	13.3	8.96	2.96	1.53
5, $\omega = 10$	**1.71**	**1.44**	**1.52**	**0.88**	**1.02**	**0.36**
5, $\omega = 30$	7.71	5.75	5.92	3.25	2.79	0.71
5, $\omega = 50$	11.7	6.42	9.40	4.50	4.09	1.68
6, $\omega = 10$	**0.81**	**1.21**	**0.65**	**0.70**	**0.52**	**0.11**
6, $\omega = 30$	5.60	4.50	4.07	3.44	2.93	1.00
6, $\omega = 50$	10.5	13.6	8.87	10.4	4.65	2.00
7, $\omega = 10$	**4.00**	**5.21**	4.70	3.17	**0.00**	**0.00**
7, $\omega = 30$	11.7	20.0	**0.00**	21.0	**0.00**	**0.00**
7, $\omega = 50$	26.0	20.0	**0.00**	**0.00**	**0.00**	**0.00**

Table C.3: The average location error of the OMEN under each of the scenarios for three different values of ω. Bold entries show the best performing ω value. 200 repetitions were simulated in each case.

C.3 Application of OMEN to Running Paces Dataset

During the 2018-19 academic year, I have been commuting from home (close to the centre of the historic town of Lancaster) to the office (Lancaster University, located somewhat south of the town itself) and back by running the distance. This is approximately 5.1 km on the way in in the morning, and 5.7 km on the way back in the evening. This discrepancy in distance is due to the fact that my stopping point in the morning is the university gym, located at the north end of campus.

Starting from the evening of 11 January 2019, I began to record my running times using Strava on my mobile device. Strava uses GPS tracking to give an accurate breakdown of performance across the entire route. In particular, users are able to mark down 'segments' of various lengths over which people may then compare their performance with other users. After using Strava for some time, I discovered that my morning route covered seven such segments, and my afternoon route covered two. Strava had been automatically recording my pace in average number of minutes taken to run 1 km across each of the segments.

We here examine these segment paces using OMEN, looking at the morning and afternoon datasets separately for a total of 130 and 127 entries respectively. The series cover the period from 11 January - 2 September 2019. Note that the difference in the number of entries is due to the occasional time where an external factor forced me away from the office to a third location before I could run home. The results for three of the morning segments are given in Figure C.1. The two afternoon segments are given in Figure C.2. The changes found by OMEN are overlaid, using the standard penalties with $\omega = 10$ (corresponding to a memory window of 10 days, the number of times I typically run to and from the office in a fortnight).

Similar changes are found by OMEN for the morning and afternoon datasets. The 'morning changepoints' given by the method correspond to 15 February and 1 August, while the 'afternoon changepoints' returned by the method are 19 February and 5 August. Note that by using a window length of $\omega = 30$, as per the simulation

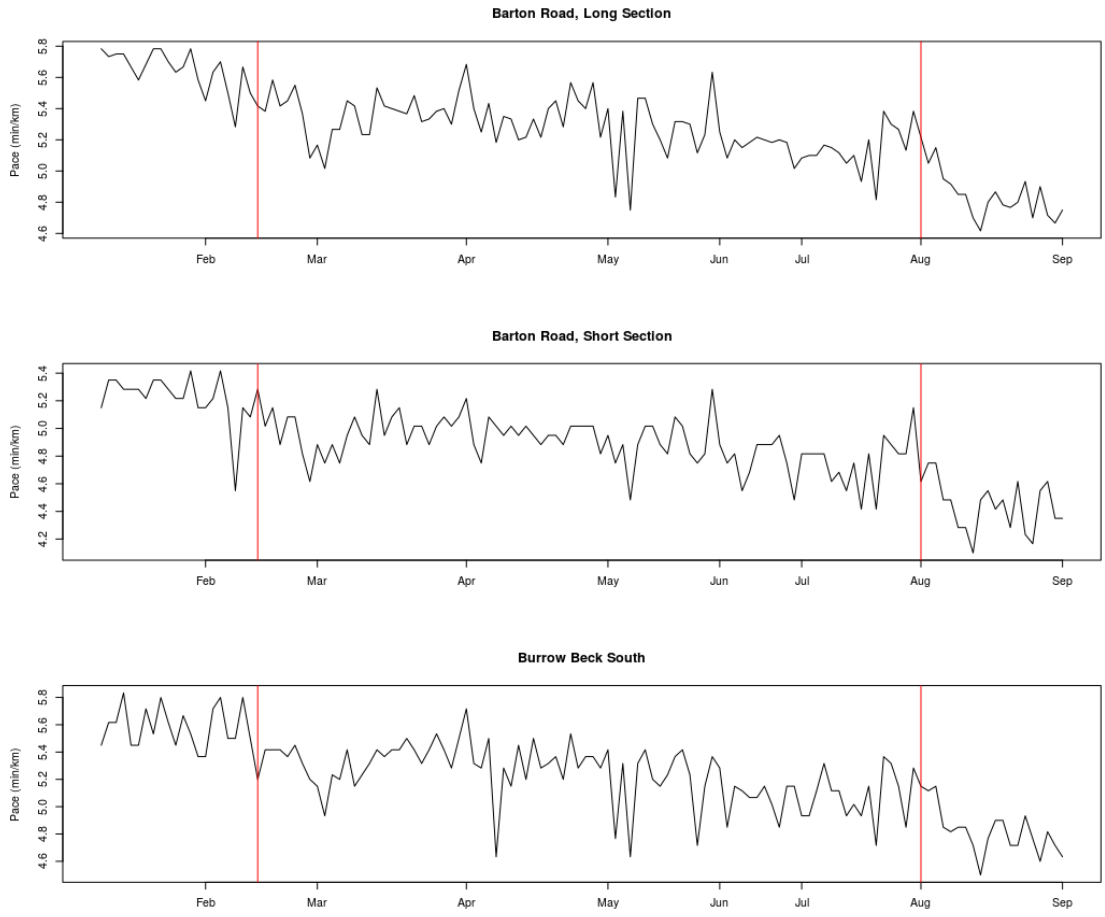

Figure C.1: Paces (in min/km) for three of the seven segments covered by the morning runs. Note that the months have far from an equal number of entries due to my presence at conferences away from Lancaster etc. Changes found by OMEN are overlaid as red vertical lines.

study, only the second change (in August) is found for either dataset. Interestingly, the changepoint in February corresponds roughly to the point at which my runs began to take place during daylight hours. Civil twilight in 2019 in Lancaster began at 6.55am on the morning of 15 February, and ended at 6.04pm on the evening of 19 February. As my morning runs typically begin around 6.45am, and I normally run home a little after 5.30pm, these dates almost exactly correspond with the first days in which visibility would have been suitable across the entire route for distinguishing objects. Meanwhile, the changepoint in August coincides with a decrease in temperature after an extremely humid summer.

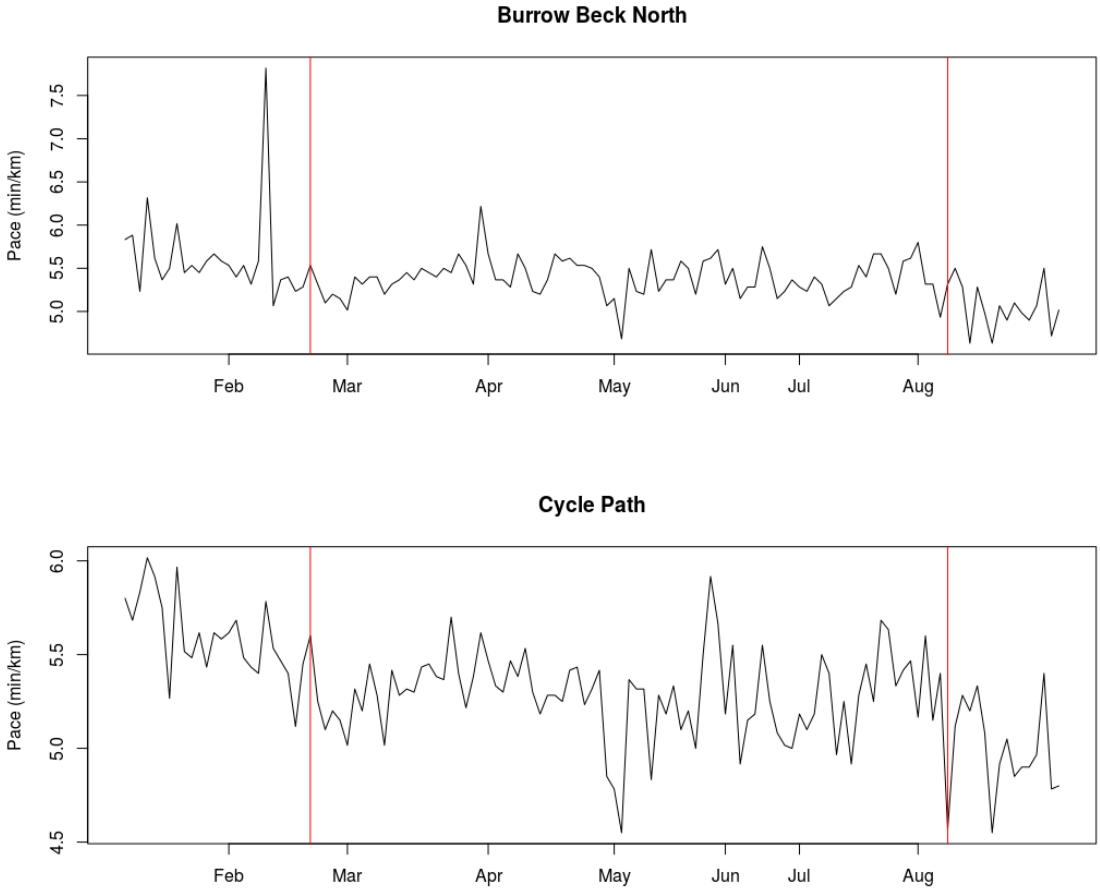

Figure C.2: Paces (in min/km) for the two segments covered by the afternoon runs. Note that the months have far from an equal number of entries due to my presence at conferences away from Lancaster etc. Changes found by OMEN are overlaid as red vertical lines.

Bibliography

U. A. Acar, A. Charguéraud, and M. Rainey. Scheduling parallel programs by work stealing with private deques. In *Proceedings of the 18th ACM SIGPLAN symposium on Principles and practice of parallel programming*, pages 219 – 228, Shenzhen, China, 2013. ACM.

R. P. Adams and D. J. C. MacKay. Bayesian online changepoint detection. *arXiv:0710.3742*, pages 1 – 7, 2007.

S. Ahmad, A. Lavin, S. Purdy, and Z. Agha. Unsupervised real-time anomaly detection for streaming data. *Neurocomputing*, 262:134 – 147, 2017.

H. Akaike. A new look at the statistical model identification. In E. Parzen, K. Tanabe, and G. Kitagawa, editors, *Selected Papers of Hirotugu Akaike. Springer Series in Statistics (Perspectives in Statistics)*, pages 215–222. Springer, New York, 1974.

E. Alba. *Parallel Metaheuristics*. John Wiley & Sons, Inc., Hoboken, New Jersey, United States of America, 2005.

P. F. Alcantarilla, S. Stent, G. Ros, R. Arroyo, and R. Gherardi. Street-view change detection with deconvolutional networks. *Autonomous Robots*, 42:1301 – 1322, 2018.

A. Anastasiou and P. Fryzlewicz. Detecting multiple generalized change-points by isolating single ones. *arXiv:1901.10852v1*, pages 1–44, 2019.

S. Arlot, A. Celisse, and Z. Harchaoui. Kernel change-point detection. *arXiv:1202.3878*, pages 1 – 26, 2012.

J. A. D. Aston and C. Kirch. Evaluating stationarity via change-point alternatives with applications to fmri data. *The Annals of Applied Statistics*, 6(4):1906 – 1948, 2012a.

J. A. D. Aston and C. Kirch. Detecting and estimating changes in dependent functional data. *Journal of Multivariate Analysis*, 109:204–220, 2012b.

A. Aue, L. Horváth, M. Hušková, and P. Kokoszka. Change-point monitoring in linear models. *The Econometrics Journal*, 9(3):373 – 403, 2006.

I. E. Auger and C. E. Lawrence. Algorithms for the optimal identification of segment neighborhoods. *Bulletin of Mathematical Biology*, 51:39 – 54, 1989.

L. Auret and C. Aldrich. Change point detection in time series data with random forests. *Control Engineering Practice*, 18:990 – 1002, 2010.

J. Bai. Least absolute deviation estimation of a shift. *Econometric Theory*, 11:403 – 436, 1995.

J. Bai. Estimation of multiple-regime regressions with least absolutes deviation. *Journal of Statistical Planning and Inference*, 74:103 – 134, 1998.

R. Baranowski and P. Fryzlewicz. *wbs: Wild Binary Segmentation for Multiple Change-Point Detection*, 2015. Version 1.3.

R. Baranowski, Y. Chen, and P. Fryzlewicz. Narrowest-over-threshold change-point detection. *arXiv:1609.00293v2*, pages 1–62, 2018.

J.-M. Bardet and C. Dion. Robust semi-parametric multiple change-points detection. *Signal Processing*, 156:145 – 155, 2019.

L. Bardwell and P. Fearnhead. Bayesian detection of abnormal segments in multiple time series. *Bayesian Analysis*, 12(1):193 – 218, 2017.

L. Bardwell, P. Fearnhead, I. A. Eckley, S. Smith, and M. Spott. Most recent changepoint detection in panel data. *Technometrics*, 61:88 – 98, 2019.

M. Barigozzi, H. Cho, and P. Fryzlewicz. Simultaneous multiple change-point and factor analysis for high-dimensional time series. *Journal of Econometrics*, 206:187 – 225, 2018.

I. Barnett and J.-P. Onnela. Change point detection in correlation networks. *Scientific Reports*, 6, 2016.

D. Barry and J. A. Hartigan. Product partition models for change point problems. *The Annals of Statistics*, 20(1):260–279, 1992.

A. Bücher, I. Kojadinovic, T. Rohmer, and J. Segers. Detecting changes in cross-sectional dependence in multivariate time series. *Journal of Multivariate Analysis*, 132:111–128, 2014.

R. A. Becker and A. R. Wilks. *maps: Draw Geographical Maps*, 2018. Version 3.3.0.

D. Beniaguev. Historical hourly weather data 2012-2017. https://www.kaggle.com/selfishgene/historical-hourly-weather-data, 2017. Accessed: 2019-09-05.

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

A. D. Bolton and N. A. Heard. Malware family discovery using reversible jump mcmc sampling of regimes. *Journal of the American Statistical Association*, 113:1490 – 1502, 2018.

M. M. Botezatu, I. Giurgiu, J. Bogojeska, and D. Wiesmann. Predicting disk replacement towards reliable data canters. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 39 – 48, San Francisco, California, United States of America, 2016. ACM.

S. Boucheron, G. Lugosi, and P. Massart. *Concentration Inequalities: A Nonasymptotic Theory of Independence*. Oxford University Press, Oxford, United Kingdom, 2013.

G. E. P. Box and M. E. Muller. A note on the generation of random normal deviates. *The Annals of Mathematical Statistics*, 29(2):610–611, 1958.

D. Brauckhoff, B. Tellenbach, A. Wagner, M. May, and A. Lakhina. Impact of packet sampling on anomaly detection metrics. In *Proceedings of the 6th ACM SIGCOMM conference on Internet measurement*, pages 159 – 164, Rio de Janeiro, Brazil, 2006. ACM.

K. Bulteel, E. Ceulemans, R. Thompson, C. Waugh, I. Gotlib, F. Tuerlinckx, and P. Kuppens. Decon: A tool to detect emotional concordance in multivariate time series data of emotional responding. *Biological Psychology*, 98(1):29 – 42, 2014.

J. Cabrieto, F. Tuerlinckx, P. Kuppens, M. Grassmann, and E. Ceulemans. Detecting correlation changes in multivariate time series: A comparison of four non-parametric change point detection methods. *Behaviour Research Methods*, 49:988 – 1005, 2017.

R. Calaway and S. Weston. *foreach: Provides Foreach Looping Construct for R*, 2017. Version 1.4.4.

R. Calaway, S. Weston, and D. Tenenbaum. *doParallel: Foreach Parallel Adaptor for the 'parallel' Package*, 2018. Version 1.0.14.

Y. Cao, L. Xie, Y. Xie, and H. Xie. Sequential change-point detection via online convex optimization. *Entropy*, 20(108), 2018.

E. Carlstein. Nonparametric change-point estimation. *The Annals of Statistics*, 16 (1):188–197, 1988.

F. Caron, A. Doucet, and R. Gottardo. On-line changepoint detection and parameter estimation with application to genomic data. *Statistics and Computing*, 22:579 – 595, 2012.

A. Celisse, G. Marot, M. Pierre-Jean, and G. J. Rigaill. New efficient algorithms for multiple change-point detection with reproducing kernels. *Computational Statistics and Data Analysis*, 128:200 – 220, 2018.

V. Chandola, S. R. Sukumar, and J. C. Schryver. Knowledge discovery from massive healthcare claims data. In *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1312 – 1320, Chicago, Illinois, United States of America, 2013. ACM.

N. Cheifetz, A. Samé, P. Aknin, and E. de Verdalle. A cusum approach for online change-point detection on curve sequences. In *Proceedings of the 2012 European Symposium on Artificial Neural Networks*, pages 399 – 404. ESANN, 2012.

F. Chen and S. Nkurunziza. On estimation of the change points in multivariate regression models with structural changes. *Communications in Statistics - Theory and Methods*, 46(14):7157 – 7173, 2017.

H. Chen. Change-point detection for multivariate and non-euclidean data with local dependency. *arXiv:1903.01598v1*, pages 1 – 33, 2019a.

H. Chen. Sequential change-point detection based on nearest neighbors. *The Annals of Statistics*, 47(3):1381 – 1407, 2019b.

H. Chen and L. Chu. *gStream: Graph-Based Sequential Change-Point Detection for Streaming Data*, 2019. Version 0.2.0.

J. Chen and A. K. Gupta. Testing and locating variance changepoints with application to stock prices. *Journal of the American Statistical Association*, 92(438):739–747, 1997.

J. Chen and A. K. Gupta. *Parametric Statistical Changepoint Analysis*. Birkhäuser, Boston, Massachusetts, United States of America, 2000.

K.-M. Chen, A. Cohen, and H. Sackrowitz. Consistent multiple testing for change points. *Journal of Multivariate Analysis*, 102:1339 – 1343, 2011.

N. Chen and K. L. Tsui. Condition monitoring and remaining useful life prediction using degradation signals: revisited. *Quality & Reliability Engineering*, 45:939 – 952, 2013.

D. T. Cheng, T. N. Mitchell, A. Zehir, R. H. Shah, R. Benayed, A. Syed, R. Chandramohan, Z. Y. Liu, H. H. Won, S. N. Scott, A. R. Brannon, C. O'Reilly, J. Sadowska, J. Casanova, A. Yannes, J. F. Hechtmann, J. Yao, W. Song, D. S. Ross, A. Oultache, S. Dogan, L. Borsu, M. Hameed, K. Nafa, M. E. Arcila, M. Ladanyi, and M. F. Berger. Memorial sloan kettering-integrated mutation profiling of actionable cancer targets (msk-impact). *The Journal of Molecular Diagnostics*, 17(3):251 – 264, 2015.

J. Cheng, X. Tang, and J. Yin. A change-point ddos attack detection method based on half interaction anomaly degree. *International Journal of Autonomous and Adaptive Communications Systems*, 10, 2017.

S. Cheon and J. Kim. Multiple change-point detection of multivariate mean vectors with the bayesian approach. *Computational Statistics and Data Analysis*, 54:406 – 415, 2010.

S. Chib. Estimation and comparison of multiple change-point models. *Journal of Econometrics*, 86:221 – 241, 1998.

H. Cho. Change-point detection in panel data via double cusum statistic. *Electronic Journal of Statistics*, 10(2):2000 – 2038, 2016.

H. Cho and P. Fryzlewicz. Multiscale and multilevel technique for consistent segmentation of nonstationary time series. *Statistica Sinica*, 22(1):207 – 229, 2012.

H. Cho and P. Fryzlewicz. Multiple-change-point detection for high dimensional time series via sparsified binary segmentation. *Journal of the Royal Statistical Society Series B*, 77(2):475–507, 2015.

M. F. R. Chowdhury, S.-A. Selouani, and D. O'Shaughnessy. Bayesian on-line spectral change point detection: a soft computing approach for on-line asr. *International Journal of Speech Technology*, 15:5 – 23, 2012.

A. Clauset and M. Young. Scale invariance in global terrorism. *arXiv:physics/0502014v2*, pages 1–6, 2005.

A. Cleynen, M. Koskas, E. Lebarbier, G. Rigaill, and S. Robin. Segmentor3isback: an r package for the fast and exact segmentation of seq-data. *Algorithms for Molecular Biology*, 9(6):1–11, 2014.

S. Aminikhanghahi D. J. Cook. A survey of methods for time series change point detection. *Knowledge and Information Systems*, 51:339 – 367, 2017.

M. Csörgő and Horváth. 20 nonparametric methods for changepoint problems. *Handbook of Statistics*, 7:403 – 425, 1988.

H. Dette and J. Gösmann. Relevant change points in high dimensional time series. *Electronic Journal of Statistics*, 12(2):2578 – 2636, 2018.

L. Dümbgen. The asymptotic behavior of some nonparametric change-point estimators. *The Annals of Statistics*, 19(3):1471 – 1495, 1991.

A. Dufays. Infinite-state markov-switching for dynamic volatility. *Journal of Financial Econometrics*, 14:418 – 460, 2016.

I. A. Eckley, P. Fearnhead, and R. Killick. Analysis of changepoint models. In D. Barber, T. Cemgil, and S. Chiappa, editors, *Bayesian Time Series Models*, pages 205 – 224. Cambridge University Press, Cambridge, 2011.

A. W. F. Edwards and L. L. Cavalli-Sforza. A method for cluster analysis. *Biometrics*, 21(2):362 – 375, 1965.

B. Eichinger and C. Kirch. A mosum procedure for the estimation of multiple random change points. *Bernoulli*, 24(1):526–564, 2018.

F. Enikeeva and Z. Harchaoui. High-dimensional change-point detection under sparse alternatives. *The Annals of Statistics*, 47(4):2051–2079, 2019.

J. Fan and R. Li. Variable selection via nonconcave penalized likelihood and its oracle properties. *Journal of the American Statistical Association*, 96:1348 – 1360, 2001.

P. Fearnhead. Exact and efficient bayesian inference for multiple changepoint problems. *Statistics and Computing*, 16:203 – 213, 2006.

P. Fearnhead and Z. Liu. On-line infgerence for multiple changepoint problems. *Journal of the Royal Statistical Society Series B*, 69(4):589 – 605, 2007.

P. Fearnhead and G. Rigaill. Changepoint detection in the presence of outliers. *Journal of the American Statistical Association*, 2017.

P. Fearnhead and G. Rigaill. Changepoint detection in the presence of outliers. *Journal of the American Statistical Association*, 114(525):169 – 183, 2019.

N. Fernando, S. W. Loke, and W. Rahayu. Computing with nearby mobile devices: A work sharing algorithm for mobile edge-clouds. *IEEE Transactions on Cloud Computing*, 7(2):329 – 343, 2019.

C. Fiandrino. Example: Network topology. `http://www.texample.net/tikz/examples/network-topology/`, 2014. Accessed: 2019-08-26.

J. D. Figueroa, R. M. Pfeiffer, D. A. Patel, L. Linville, L. A. Brinton, G. L. Gierach, X. R. Yang, D. Papathomas, D. Visscher, C. Mies, A. C. Degnim, W. F. Anderson, S. Hewitt, Z. G. Khodr, S. E. Clare, A. M. Storniolo, and M. E. Sherman. Terminal duct lobular unit involution of the normal breast: Implications for breast cancer etiology. *Journal of the National Cancer Institute*, 106, 2014.

A. T. M. Fisch, I. A. Eckley, and P. Fearnhead. A linear time method for the detection of point and collective anomalies. *arXiv:1806.01947v2*, pages 1–40, 2019a.

A. T. M. Fisch, I. A. Eckley, and P. Fearnhead. Subset multivariate collective and point anomaly detection. *arXiv:1909.01691v1*, pages 1–51, 2019b.

I. Frank and J. Friedman. A statistical view of some chemometrics regression tools (with discussion). *Technometrics*, 35(2):109 – 135, 1993.

K. Frick, A. Munk, and H. Sieling. Multiscale change point inference. *Journal of the Royal Statistical Society Series B*, 76:495 – 580, 2014.

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

P. Fryzlewicz. Tail-greedy bottom-up data decompositions and fast multiple change-point detection. *The Annals of Statistics*, 46(6B):3390 – 3421, 2018.

P. Fryzlewicz. Detecting possibly frequent change-points: Wild binary segmentation 2 and steepest-drop model selection. preprint on webpage at `http://stats.lse.ac.uk/fryzlewicz/wbs2/wbs2.pdf`, 2019.

W. J. Fu. Penalized regressions: The bridge versus the lasso. *Journal of Computational and Graphical Statistics*, 7:397 – 416, 1998.

E. Galceran, A. G. Cunningham, R. M. Eustice, and E. Olson. Multipolicy decision-making for autonomous driving via changepoint-based behavior prediction: Theory and experiment. *Autonomous Robots*, 41:1367 – 1382, 2017.

C. Gallagher, R. Lund, and M. Robbins. Changepoint detection in climate time series with long-term trends. *Journal of Climate*, 26:4994 – 5006, 2013.

D. Gamerman. Dynamic bayesian models for survival data. *Journal of the Royal Statistical Society Series C*, 40:63 – 79, 1991.

R. Garnett, M. A. Osborne, S. Reece, A. Rogers, and S. J. Roberts. Sequential bayesian prediction in the presence of changepoints and faults. *The Computer Journal*, 53:345 – 352, 2009.

F. G. Garre, A. H. Zwinderman, R. B. Geskuz, and Y. W. J. Sijpkens. A joint latent class changepoint model to improve the prediction of time to graft failure. *Journal of the Royal Statistical Society Series A*, 171:299 – 308, 2008.

D. Garreau and S. Arlot. Consistent change-point detection with kernels. *Electronic Journal of Statistics*, 12(2):4440 – 4486, 2018.

A. Gasull, J. A. López-Salcedo, and F. Utzet. Maxima of gamma random variables and other weibull-like distributions and the lambert w function. *TEST*, 24:714 – 733, 2015.

A. Gibberd and S. Roy. Multiple changepoint estimation in high-dimensional gaussian graphical models. *arXiv:1712.05786v1*, pages 1 – 39, 2017.

A. J. Gibberd and J. D. B. Nelson. High dimensional changepoint detection with a dynamic graphical lasso. In *2014 IEEE International Conference on Acoustics, Speech and Signal Processing*, Florence, Italy, 2014. IEEE.

J.J.J. Groen, G. Kapetanios, and S. Price. Multivariate methods for monitoring structural change. *Journal of Applied Econometrics*, 28:250–274, 2013.

W. Gu, J. Choi, M. Gu, H. Simon, and K. Wu. Fast change point detection for electricity market analysis. In *Proceedings of the 2013 IEEE International Conference on Big Data*, pages 50 – 57, Silicon Valley, California, United States of America, 2013. IEEE.

T. Guo, Z. Xu, X. Yao, H. Chen, K. Aberer, and K. Funaya. Robust online time series prediction with recurrent neural networks. In *Proceedings of the 2016 IEEE International Conference on Data Science and Advanced Analytics*, pages 816–825, Montreal, Quebec, Canada, 2016. IEEE.

A. K. Gupta and J. Chen. Detecting changes of mean in multidimensional normal sequences with applications to literature and geology. *Computational Statistics*, 11 (3):211 – 221, 1996.

Z. Harchaoui and O. Cappe. Retrospective mutiple change-point estimation with kernels. In *Proceedings of the IEEE/SP 14th Workshop on Statistical Signal Processing*, Madison, Wisconsin, United States of America, 2007. IEEE.

Z. Harchaoui and C. Lévy-Leduc. Multiple change-point estimation with a total variation penalty. *Journal of the American Statistical Association*, 105(492):1480 – 1493, 2010.

Z. Harchaoui, F. Bach, and E. Moulines. Kernel change-point analysis. In D. Koller, D. Schuurmans, and Y. Bengio, editors, *Proceedings of the Advances in Neural Information Processing Systems 21*, pages 609–616. Curran Associates, Inc., 2009.

M. Harel, K. Crammer, R. El-Yaniv, and S. Mannor. Concept drift detection through resampling. In E. P. Xing and T. Jebara, editors, *Proceedings of the 31st International Conference on International Conference on Machine Learning*, volume 32, pages II–1009 – II–1017, Beijing, China, 2014. ACM.

P. Harnish, B. Nelson, and G. Runger. Process partitions from time-ordered clusters. *Journal of Quality Technology*, 41:3 – 17, 2009.

J. A. Hartigan and M. A. Wong. Algorithm as 136: A k-means clustering algorithm. *Journal of the Royal Statistical Society: Series C*, 28(1):100 – 108, 1979.

D. M. Hawkins. Fitting multiple change-point models to data. *Computational Statistics & Data Analysis*, 37:323 – 341, 2001.

K. Haynes, I. A. Eckley, and P. Fearnhead. Computationally efficient changepoint detection for a range of penalties. *Journal of Computational and Graphical Statistics*, 26(1):134–143, 2017a.

K. Haynes, P. Fearnhead, and I. A. Eckley. A computationally efficient nonparametric approach for changepoint detection. *Statistics and Computing*, 27(5):1293–1305, 2017b.

R. Henderson. A problem with the likelihood ratio test for a change-point hazard rate model. *Biometrika*, 77(4):835–843, 1990.

N. Henze. A multivariate two-sample test based on the number of nearest neighbor type coincidences. *The Annals of Statistics*, 16(2):772 – 783, 1988.

F. J. Hernandez-Lopez and M. Rivera. Change detection by probabilistic segmentation from monocular view. *Machine Vision and Applications*, 25:1175 – 1195, 2014.

M. Höhle. Online change-point detection in categorical time series. In E. Parzen, K. Tanabe, and G. Kitagawa, editors, *Statistical Modelling and Regression Structures*, pages 377 – 397. Physica-Verlag HD, Berlin, Germany, 2010.

R. Hilborn, R. O. Amoroso, E. Bogazzi, O. P. Jensen, A. M. Parma, C. Szuwalski, and C. J. Walters. When does fishing forage species affect their predators? *Fisheries Research*, 191:211 – 221, 2017.

D. V. Hinkley. Inference about the change-point from cumulative sum tests. *Biometrika*, 58(3):509–523, 1971.

T. D. Hocking, G. Rigaill, P. Fearnhead, and G. Bourque. Generalized functional pruning optimal partitioning (gfpop) for constrained changepoint detection in genomic data. *arXiv:1810.00117v1*, pages 1–19, 2018.

S. R. Howard, A. Ramdas, J. McAuliffe, and J. Sekhon. Uniform, nonparametric, non-asymptotic confidence sequences. *arXiv:1810.08240v3*, pages 1 – 43, 2019.

D. A. Hsu. Tests for variance shift at an unknown time point. *Journal of the Royal Statistical Society Series C*, 26(3):279–284, 1977.

P. J. Huber. Robust statistics. In M. Lovric, editor, *International Encyclopedia of Statistical Science*. Springer, Berlin, Heidelberg, 2011.

M. Hušková. Robust change point analysis. In C. Becker, R. Fried, and S. Kuhnt, editors, *Robustness and Complex Data Structures*. Springer, Berlin, Heidelberg, 2013.

M. Hušková and A. Slabý. Permutation tests for multiple changes. *Kybernetika*, 37 (5):605 – 622, 2001.

T. Idé and K. Tsuda. Change-point detection using krylov subspace learning. In *Proceedings of the 2007 SIAM International Conference on Data Mining*, Minneapolis, Minnesota, United States of America, 2007. SIAM.

C. Inclan. Detection of multiple changes of variance using posterior odds. *Journal of Business & Economic Statistics*, 11(3):289 – 300, 1993.

C. Inclán and G. C. Tiao. Use of cumulative sums of squares for retrospective detection of changes of variance. *Journal of the American Statistical Association*, 89(427): 913–923, 1994.

D. B. Heras J. López-Fandiño, F. Argüello, and M. D. Mura. Gpu framework for change detection in multitemporal hyperspectral images. *International Journal of Parallel Programming*, 47:272 – 292, 2019.

B. Jackson, J.D. Scargle, D. Barnes, S. Arabhi, A. Alt, P. Gioumousis, E. Gwin, P. Sangtrakulcharoen, L. Tan, and T.T. Tsai. An algorithm for optimal partitioning of data on an interval. *IEEE Signal Processing*, 12(2):105–108, 2005.

M. Jackson. Update9 bt's national uk network suffers serious broadband outage. `https://www.ispreview.co.uk/index.php/2016/02/bts-national-uk-network-suffers-serious-broadband-outage.html`, 2016. Accessed: 2019-08-26.

B. James, K. L. James, and D. Siegmund. Tests for a change-point. *Biometrika*, 74 (1):71–83, 1987.

M. Jensen. The global terrorism database (gtd) [data file], 2018. National Consortium for the Study of Terrorism and Responses to Terrorism (START), University of Maryland. Retrieved from `https://www.start.umd.edu/gtd`.

J.-J. Jeon, J. H. Sung, and E.-S. Chung. Abrupt change point detection of annual maximum precipitation using fused lasso. *Journal of Hydrology*, 538:831 – 841, 2016.

S. Jewell, T. D. Hocking, P. Fearnhead, and D. Witten. Fast nonconvex deconvolution of calcium imaging data. *Biostatistics*, page To appear, 2019.

R. H. Jones and I. Dey. Determining one or more change points. *Chemistry and Physics of Lipids*, 76:1–6, 1995.

S. A. Julious. Inference and estimation in a changepoint regression problem. *Journal of the Royal Statistical Society Series D (The Statistician)*, 50(1):51–61, 2001.

M. Jutila. An adaptive edge router enabling internet of things. *IEEE Internet of Things Journal*, 3:1061 – 1069, 2016.

T. A. Kass-Hout, Z. Xu, P. McMurray, S. Park, D. L. Buckeridge, J. S. Brownstein, L. Finelli, and S. L. Groseclose. Application of change point analysis to daily influenza-like illness emergency department visits. *Journal of the American Medical Informatics Association*, 19:1075 – 1081, 2012.

E. Keogh, S. Chu, D. Hart, and M. Pazzani. An online algorithm for segmenting time series. In *Proceedings of the 2001 IEEE International Conference on Data Mining*, pages 289 – 296, San Jose, California, United States of America, 2001. IEEE.

A. Khaleghi and D. Ryabko. Locating changes in highly dependent data with unknown number of change points. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Proceedings of the Advances in Neural Information Processing Systems 25*, pages 3086 – 3094. Curran Associates, Inc., 2012.

N. Khomami. Bt apologises for broadband outage across much of uk. *The Guardian*, February 2016. URL `https://www.theguardian.com/business/2016/feb/02/bt-broadband-phone-network-down-uk-areas-birmingham-london-sheffield`.

R. Killick, P. Fearnhead, and I. A. Eckley. Optimal detection of changepoints with a linear computational cost. *Journal of the American Statistical Association*, 107 (500):1590–1598, 2012.

R. Killick, K. Haynes, I. Eckley, P. Fearnhead, and J. Lee. *changepoint: Methods for Changepoint Detection*, 2016. Version 2.2.2.

J. Knoblauch, J. E. Jewson, and T. Damoulas. Doubly robust bayesian inference for non-stationary streaming data with β-divergences. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Proceedings of the Neural Information Processing Systems 2018 Conference*, 2018.

S. I. M. Ko, T. T. L. Chong, and P. Ghosh. Dirichlet process hidden markov multiple change-point model. *Bayesian Analysis*, 10(2):275 – 296, 2015.

P. Kokoszka and R. Leipus. Change-point in the mean of dependent observations. *Statistics & Probability Letters*, 40:385–393, 1998.

M. Kolar and E. P. Xing. Estimating networks with jumps. *Electronic Journal of Statistics*, 6:2069 – 2106, 2012.

V. Kulkarni, R. Al-Rfou, B. Perozzi, and S. Skiena. Statistically significant detection of linguistic change. In *Proceedings of the 24th International Conference on the World Wide Web*, pages 625 – 635, Florence, Italy, 2015. ACM.

G. LaFree. The global terrorism database (gtd): Accomplishments and challenges. *Perspectives on Terrorism*, 4(1):24–46, 2010.

G. LaFree and L. Dugan. Introducing the global terrorism database. *Terrorism and Political Violence*, 19(2):181–204, 2007.

G. LaFree, L. Dugan, and E. Miller. *Putting Terrorism in Context: Lessons from the global terrorism database*. Routledge, London, United Kingdom, 2014.

B. Laurent and P. Massart. Adaptive estimation of a quadratic functional by model selection. *The Annals of Statistics*, 28(5):1302 – 1338, 2000.

M. Lavielle and G. Teyssiere. Detection of multiple change-points in multivariate time series. *Lithuanian Mathematical Journal*, 46:287 – 306, 2006.

E. Lebarbier. Detecting multiple change-points in the mean of gaussian process by model selection. *Signal Processing*, 85:717 – 736, 2005.

S. Lee, Y. Tokutsu, and K. Maekawa. The cusum test for parameter change regression models with arch errors. *Journal of the Japan Statistical Society*, 34:173 – 188, 2004.

S. Lee, Y. Nishiyama, and N. Yoshida. Test for parameter change in diffusion processes by cusum statistics based on one-step estimators. *Annals of the Institute of Statistical Mathematics*, 58:211 – 222, 2006.

J. Lei, J. Robins, and L. Wasserman. Distribution-free prediction sets. *Journal of the American Statistical Association*, 108:278 – 287, 2013.

J. Lei, M. G'Sell, A. Rinaldo, R. J. Tibshirani, and L. Wasserman. Distribution-free predictive inference for regression. *Journal of the American Statistical Association*, 113:1094 – 1111, 2018.

F. Leonardi and P. Bühlmann. Computationally efficient change point detection for high dimensional regression. *arXiv:1601.03704v1*, pages 1 – 32, 2016.

F. Liang. Annealing stochastic approximation monte carlo for neural network training. *Machine Learning*, 68:201 – 233, 2007.

F. Liang. Improving samc using smoothing methods: Theory and applications to bayesian model selection problems. *The Annals of Statistics*, 37(5B):2626 – 2654, 2009.

F. Lombard. Rank tests for changepoint problems. *Biometrika*, 74(3):615–624, 1987.

Z. Lu, M. Banerjee, and G. Michailidis. Intelligent sampling and inference for multiple change points in extremely long data sequences. *arXiv:1710.07420v2*, pages 1–43, 2018.

M. Ludkin, I. Eckley, and P. Neal. Dynamic stochastic block models: parameter estimation and detection of changes in community structure. *Statistics and Computing*, 28:1201 – 1213, 2018.

A. Lung-Yut-Fong, C. Lévy-Leduc, and O. Cappé. Homogeneity and change-point detection tests for multivariate data using rank statistics. *arXiv:1107.1971v3*, pages 1 – 27, 2012.

T.-M. Luong, Y. Rozenholc, and G. Nuel. Fast estimation of posterior probabilities in change-point analysis through a constrained hidden markov model. *Computational Statistics and Data Analysis*, 68:129 – 140, 2013.

J. M. Maheu and Q. Yang. An infinite hidden markov model for short-term interest rates. *Journal of Empirical Finance*, 38:202 – 220, 2016.

M. A. Mahmoud, P. A. Parker, W. H. Woodall, and D. M. Hawkins. A change point method for linear profile data. *Quality and Reliability Engineering International*, 23:247 – 268, 2007.

R. Maidstone, T. Hocking, G. Rigaill, and P. Fearnhead. On optimal multiple changepoint algorithms for large data. *Statistics and Computing*, 27(2):519–533, 2017.

R. Malladi, G. P. Kalamangalam, and B. Aazhang. Online bayesian change point detection for segmentation of epileptic activity. In *Proceedings of the 2013 Asilomar Conference on Signals, Systems and Computers*, Pacific Grove, California, United States of America, 2013. IEEE.

G. Manogaran and D. Lopez. Spatial cumulative sum algorithm with big data analytics for climate change detection. *Computers & Electrical Engineering*, 65: 207–221, 2018.

P. Massart. A non-asymptotic theory for model selection. In A. Laptev, editor, *Proceedings of the Fourth European Congress of Mathematics*, Stockholm, Sweden, 2004. European Mathematical Society.

N. Masuda and P. Holme. Detecting sequences of system states in temporal networks. *Scientific Reports*, 9, 2019.

D. S. Matteson and N. A. James. A nonparametric approach for multiple change point analysis of multivariate data. *Journal of the American Statistical Association*, 109 (505):334 – 345, 2014.

B. Merz, Z. W. Kundzewicz, J. Delgado, Y. Hundecha, and H. Kreibich. Detection and attribution of changes in flood hazard and risk. In Z. W. Kundzewicz, editor, *Changes in Flood Risk in Europe*, chapter 25. CRC Press, 2012.

M. Mezmaz, M. Melab, Y. Kessaci, Y.C. Lee, E.-G. Talbi, A.Y. Zomaya, and D. Tuyttnes. A parallel bi-objective hypbrid metaheuristic for energy-aware scheduling for cloud computing systems. *Journal of Parallel and Distributed Computing*, 71(11):1497–1508, 2011.

B. Q. Miao. Inference in a model with at most one slope-change point. *Multivariate Statistics and Probability*, pages 375–391, 1989.

V. Moskvina and A. Zhigljavsky. An algorithm based on singular spectrum analysis for change-point detection. *Communications in Statistics: Simulation and Computation*, 32:319 – 352, 2003.

G. Muniz-Terrera, A. van den Hout, and F. E. Matthews. Random change point models: investigating cognitive decline in the presence of missing data. *Journal of Applied Statistics*, 38:705 – 716, 2011.

D. H. Murray, M. Jahnel, J. Lauer, M. J. Avellaneda, N. Brouilly, A. Cezanne, H. Morales-Navarrete, E. D. Perini, C. Ferguson, A. N. Lupas, Y. Kalaidzidis, R. G. Parton, S. W. Grill, and M. Zerial. An endosomal tether undergoes an entropic collapse to bring vesicles together. *Nature*, 537:107 – 111, 2016.

C. T. Ng, W. Lee, and Y. Lee. Change-point estimators with true identification property. *Bernoulli*, 24(1):616 – 660, 2018.

S. Niekum, S. Osentoski, C. G. Atkeson, and A. G. Barto. Online bayesian changepoint detection for articulated motion models. In *Proceedings of the 2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1468 – 1475, Seattle, Washington, United States of America, 2015. IEEE.

V. Montes De Oca, D. R. Jeske, Q. Zhang, C. Rendon, and M. Marvasti. A cusum change-point detection algorithm for non-stationary sequences with application to data network surveillance. *Journal of Systems and Software*, 83:1288 – 1297, 2010.

A. B. Olshen, E.S. Venkatraman, R. Lucito, and M. Wigler. Circular binary

segmentation for the analysis of array-based dna copy number data. *Biostatistics*, 5(4):557–572, 2004.

J. Padmore. The analysis of stratigraphic data with particular reference to zonation problems. In O. Opitz, B. Lausen, and R. Klar, editors, *Information and Classification: Concepts, Methods and Applications*, pages 490–499, University of Dortmund, Dortmund, Germany, 1992. Springer.

E. S. Page. Continuous inspection schemes. *Biometrika*, 41(1/2):100–115, 1954.

V. Pedro and S. Abreu. Distributed work stealing for constraint solving. *arXiv:1009.3800v1*, pages 1 – 15, 2010.

L. Peel and A. Clauset. Detecting change points in the large-scale structure of evolving networks. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, pages 2914 – 2920, 2015.

F. Pein, H. Sieling, and A. Munk. Heterogeneous change point inference. *Journal of the Royal Statistical Society Series B*, 79:1207 – 1227, 2017.

S. Peluso, S. Chib, and A. Mira. Semiparametric multivariate and multiple change-point modelling. *Bayesian Analysis*, 14(3):727 – 751, 2019.

C. Petrov. Big data statistics 2019. `https://techjury.net/stats-about/big-data-statistics/`, 2019. Accessed: 2019-08-26.

A. N. Pettitt. A simple cumulative sum type statistic for the change-point problem with zero-one observations. *Biometrika*, 67:79 – 84, 1980.

B. Pickering. *Changepoint detection for acoustic sensing signals.* PhD thesis, Lancaster University, 2016.

J. Plasse and N. M. Adams. Multiple changepoint detection in categorical data streams. *Statistics and Computing*, pages 1 – 17, 2019.

W. Ploberger and W. Krämer. The cusum test with ols residuals. *Econometrica*, 60 (2):271 – 285, 1992.

S. Poddar, R. Kottath, V. Kumar, and A. Kumar. Adaptive sliding kalman filter using nonparametric change point detection. *Measurement*, 82:410 – 420, 2016.

W. Polonik. Minimum volume sets and generalized quantile processes. *Stochastic Processes and their Applications*, 69:1 – 24, 1997.

A. S. Polunchenko and A. G. Tartakovsky. On optimality of the shiryaev-roberts procedure for detecting a change in distribution. *The Annals of Statistics*, 38(6): 3445 – 3457, 2010.

G. Pranuthi, S. K. Dubey, S. K. Tripathi, and S. K. Chandniha. Trend and change point detection of precipitation in urbanizing districts of uttarakhand in india. *Indian Journal of Science and Technology*, 7(10):1573 – 1582, 2014.

A. E. Raftery and V. E. Akman. Bayesian analysis of a poisson process with a change-point. *Biometrika*, 73(1):85–89, 1986.

R. Rajaduray, S. Ovadia, and D. J. Blumenthal. Analysis of an edge router for span-constrained optical burst switched (obs) networks. *Journal of Lightwave Technology*, 22(11):2693 – 2705, 2004.

J. Reeves, J. Chen, X. L. Wang, R. Lund, and Q. Lu. A review and comparison of changepoint detection techniques for climate data. *Journal of Applied Meteorology and Climatology*, 46:900 – 915, 2007.

A. D. Richardson, K. Hufkens, T. Milliman, D. M. Aubrecht, M. Chen, J. M. Gray, M. R. Johnston, T. K. Keenan, S. T. Klosterman, M. Kosmala, E. K. Melaas, M. A. Friedl, and S. Frolking. Tracking vegetation phenology across diverse north american biomes using phenocam imagery. *Scientific Data*, 5, 2018.

S. De Ridder, B. Vandermarliere, and J. Ryckebusch. Detection and localization of change points in temporal networks with the aid of stochastic block models. *Journal of Statistical Mechanics: Theory and Experiment*, 2016(11), 2016.

G. Rigaill. Pruned dynamic programming for optimal multiple change-point detection. *arXiv:1004.0887v1*, pages 1–9, 2010.

G. Rigaill, E. Lebarbier, and S. Robin. Exact posterior distributions and model selection criteria for multiple change-point detection problems. *Statistics and Computing*, 22(4):917–929, 2012.

G. Rigaill, T. D. Hocking, F. Bach, and J. P. Vert. Learning sparse penalties for change-point detection using max margin interval regression. In S. Dasgupta and D. McAllester, editors, *Proceedings of the 30th International Conference on Machine Learning (ICML-13)*, 2013.

M. Robbins, C. Gallagher, R. Lund, and A. Aue. Mean shift testing in correlated data. *Journal of Time Series Analysis*, 32(5):498 – 511, 2011.

G. J. Ross, D. K. Tasoulis, and N. M. Adams. Nonparametric monitoring of data streams for changes in location and scale. *Technometrics*, 53(4):379 – 389, 2011.

S. Roy, Y. Atchadé, and G. Michailidis. Change point estimation in high dimensional markov random-field models. *Journal of the Royal Statistical Society Series B*, 79: 1187 – 1206, 2017.

F. Ruggeri and S. Sivaganesan. On modeling change points in non-homogeneous poisson processes. *Statistical Inference for Stochastic Processes*, 8:311 – 329, 2005.

E. Ruggieri and M. Antonellis. An exact approach to bayesian sequential change point detection. *Computational Statistics and Data Analysis*, 97:71 – 86, 2016.

C. Santifort, T. Sandler, and P. T. Brandt. Terrorist attack and target diversity: Changepoints and their drivers. *Journal of Peace Research*, 50(1):75 – 90, 2012.

P. Sarkar and W. Q. Meeker. A bayesian on-line change detection algorithm with process monitoring applications. *Quality Engineering*, 10:539 – 549, 1998.

V. Savani and A. A. Zhigljavsky. Efficient estimation of parameters of the negative binomial distribution. *Communications in Statistics - Theory and Methods*, 35:767 – 783, 2006.

M. F. Schilling. Multivariate two-sample tests based on nearest neighbors. *Journal of the American Statistical Association*, 81:799 – 806, 1986.

N. Schmid, C.D. Christ, M. Christen, A.P. Eichenberger, and W.F. van Gunsteren. Architecture, implementation and parallelisation of the gromos software for biomolecular simulation. *Computer Physics Communications*, 183(4):890–903, 2012.

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

A.J. Scott and M. Knott. A cluster analysis method for grouping means in the analysis of variance. *Biometrics*, 30(3):507–512, 1974.

A. K. Sen and M. S. Srivastava. On multivariate tests for detecting change in mean. *Sankhyā: The Indian Journal of Statistics, Series A (1961 - 2002)*, 35(2):173–186, 1973.

T. S. Sethi and M. Kantardzic. On the reliable detection of concept drift from streaming unlabeled data. *Expert Systems with Applications*, 82:77 – 99, 2017.

H. Shen. The detection and empirical study of variance change points on housing prices - taking wuhan city commodity prices as an example. *Journal of Mathematical Finance*, 6(5):699 – 710, 2016.

N. Simon, J. Friedman, T. Hastie, and R. Tibshirani. A sparse-group lasso. *Journal of Computational and Graphical Statistics*, 22:231 – 245, 2013.

A. F. M. Smith and D. G. Cook. Straight lines with a change-point: A bayesian analysis of some renal transplant data. *Journal of the Royal Statistical Society Series C*, 29:180 – 189, 1980.

M. S. Srivastava and K. J. Worsley. Likelihood ratio tests for a change in the multivariate normal mean. *Journal of the American Statistical Association*, 81: 199 – 204, 1986.

R. M. Steward, S. E. Rigdon, and R. Pan. A bayesian approach to diagnostics for multivariate control charts. *Journal of Quality Technology*, 48:303 – 325, 2016.

M. Steyvens and S. Brown. Locating changes in highly dependent data with unknown number of change points. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Proceedings of the 18th International Conference on Neural Information Processing Systems*, pages 1281 – 1288, Vancouver, British Columbia, Canada, 2005. MIT Press.

J. H. Sullivan. Detection of multiple change points from clustering individual observations. *Journal of Quality Technology*, 34:371 – 383, 2002.

S. Suparman, M. Doisy, and J.-Y. Tourneret. Changepoint detection using reversible jump mcmc methods. In *Proceedings of the 2002 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 1569 – 1572, Orlando, Florida, United States of America, 2002. IEEE.

G. J. Székely and M. L. Rizzo. Hierarchical clustering via joint between-within distances: Extending ward's minimum variance method. *Journal of Classification*, 22(2):151 – 183, 2005.

A. Tartakovsky, I. Nikiforov, and M. Basseville. *Sequential Analysis: Hypothesis Testing and Changepoint Detection*. CRC Press, Taylor and Francis Group, New York, United States of America, 2014.

A. G. Tartakovsky, A. S. Polunchenko, and G. Sokolov. Efficient computer network anomaly detection by changepoint detection methods. *IEEE Journal of Selected Topics in Signal Processing*, 7(1):4 – 11, 2013.

S. Thies and P. Molnár. Bayesian change point analysis of bitcoin returns. *Finance Research Letters*, 27:223 – 227, 2018.

R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B (Methodological)*, 58:267 – 288, 1996.

R. Tibshirani and P. Wang. Spatial smoothing and hot spot detection for cgh data using the fused lasso. *Biostatistics*, 9:18 – 29, 2008.

S. O. Tickle, I. A. Eckley, P. Fearnhead, and K. Haynes. Parallelisation of a common changepoint detection method. *arXiv:1810.03591v1*, pages 1–35, 2018.

D.-H. Tran. Automated change detection and reactive clustering in multivariate streaming data. In *Proceedings of the 2019 IEEE-RIVF International Conference on Computing and Communication Technologies*, Danang, Vietnam, 2019. IEEE.

C. Truong, L. Gudre, and N. Vayatis. Penalty learning for changepoint detection. In *Proceedings of the 2017 25th European Signal Processing Conference (EUSIPCO) in Kos, Greece*, 2017.

C. Truong, L. Oudre, and N. Vayatis. A review of changepoint detection methods. *arXiv:1801.00718*, pages 1–31, 2018.

C. Truong, L. Oudre, and N. Vayatis. Selective review of offline change point detection methods. *arXiv:1801.00718v2*, pages 1 – 46, 2019.

R. S. Tsay. Outliers, level shifts and variance changes in time series. *Journal of Forecasting*, 7(1):1 – 20, 1988.

G. Tsechpenakis, D. N. Metaxas, C. Neidle, and O. Hadjiliadis. Robust online change-point detection in video sequences. In P. Light, editor, *Proceedings of the 2006 Conference on Computer Vision and Pattern Recognition Workshop*, pages 155 – 161, New York, New York, United States of America, 2006. IEEE.

E. S. Venkatraman. *Consistency Results in Multiple Change-Point Problems*. PhD thesis, Stanford University, 1992.

E. S. Venkatraman and A. B. Olshen. A faster circular binary segmentation algorithm for the analysis of array cgh data. *Bioninformatics*, 23(6):657 – 663, 2007.

R. G. W. Verhaak, K. A. Hoadley, E. Purdom, V. Wang, Y. Qi, M. D. Wilkerson, C. R. Miller, L. Ding, T. Golub, J. P. Mesirov, G. Alexe, M. Lawrence, M. O'Kelly,

P. Tamayo, B. A. Weir, S. Gabriel, W. Winckler, S. Gupta, L. Jakkula, H. S. Feiler, J. G. Hodgson, C. D. James, J. N. Sarkaria, C. Brennan, A. Kahn, P. T. Spellman, R. K. Wilson, T. P. Speed, J. W. Gray, M. Meyerson, G. Getz, C. M. Perou, D. N. Hayes, and The Cancer Genome Atlas Research network. Integrated genomic analysis identifies clinically relevant subtypes of glioblastoma characterized by abnormalities in pdgfra, idh11, egfr, and nf1. *Cancer Cell*, 17:98 – 110, 2010.

D. Wang, Y. Yu, and A. Rinaldo. Optimal covariance change point localization in high dimensions. *arXiv:1712.09912v2*, pages 1 – 46, 2018.

D. Wang, Y. Yu, and A. Rinaldo. Univariate mean change point detection: Penalization, cusum and optimality. *arXiv:1810.09498v4*, pages 1 – 40, 2019a.

P. Wang, Y. Kim, J. Pollack, B. Narasimhan, and R. Tibshirani. A method for calling gains and losses in array cgh data. *Biostatistics*, 6:45 – 58, 2005.

S. Wang and M. R. Reynolds. A glr control chart for monitoring the mean vector of a multivariate normal process. *Journal of Quality Technology*, 45:18 – 33, 2013.

T. Wang and R. Samworth. *InspectChangepoint: High-Dimensional Changepoint Estimation via Sparse Projection*, 2016. Version 1.0.1.

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

X. Wang and D. B. Dunson. Parallelizing MCMC via weierstrass sampler. *arXiv:1312.4605v2*, pages 1–35, 2014.

X. L. Wang, Q. H. Wen, and Y. Wu. Penalized maximal t test for detecting undocumented mean change in climate data series. *Journal of Applied Meteorology and Climatology*, 46(6):916 – 931, 2007.

Y. Wang, Z. Wang, and X. Zi. Rank-based multiple change-point detection. *Communications in Statistics - Theory and Methods*, pages 1–17, 2019b.

P. Wessman. Some principles for surveillance adopted for multivariate processes with a common change point. *Communications and Statistics - Theory and Methods*, 27 (5):1143 – 1161, 1998.

B. Whitcher, P. Guttorp, and D. B. Percival. Multiscale detection and location of multiple variance changes in the presence of long memory. *Journal of Statistical Computation and Simulation*, 68:65 – 87, 2000.

D. W. Wichern, R. B. Miller, and D.-A. Hsu. Changes of variance in first-order autroregressive time series models - with an application. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 25:248 – 256, 1976.

P. Wills and F. G. Meyer. Change point detection in a dynamic stochastic blockmodel. In H. Cherifi, S. Gaito, J. Mendes, E. Moro, and L. Rocha, editors, *Complex Networks and Their Applications VIII*. Springer, Cham, 2020.

D. A. Wolfe and Y.-S. Chen. The changepoint problem in a multinomial sequence. *Communications in Statistics - Simulation and Computation*, 19(2):603 – 618, 1990.

D. A. Wolfe and E. Schechtman. Nonparametric statistical procedures for the changepoint problem. *Journal of Statistical Planning and Inference*, 9:389 – 396, 1984.

K. J. Worsley. On the likelihood ratio test for a shift in location of normal populations. *Journal of the American Statistical Association*, 74(366):365 – 367, 1979.

M. Xie, Q. P. Hu, Y. P. Wu, and S. H. Ng. A study of the modeling and analysis of software and fault-detection and fault-correction processes. *Quality and Reliability Engineering International*, 23:459 – 470, 2007.

Y. Xie, J. Huang, and R. Willett. Change-point detection for high-dimensional time series with missing data. *IEEE Journal of Selected Topics in Signal Processing*, 7: 12 – 27, 2013.

B. Xing, C. M. T. Greenwood, and S. B. Bull. A hierarchical clustering method for estimating copy number variation. *Biostatistics*, 8:632 – 653, 2007.

X. Xuan and K. Murphy. Modeling changing dependency structure in multivariate time series. In *Proceedings of the 24th international conference on Machine learning*, pages 1055 – 1062, Corvalis, Oregon, United States of America, 2007. ACM.

T. Y. Yang. Bayesian binary segmentation procedure for detecting streakiness in sports. *Journal of the Royal Statistics Society Series A*, 167:627 – 637, 2004.

Y.-C. Yao. Estimation of a noisy discrete-time step function: Bayes and empirical bayes approaches. *The Annals of Statistics*, 12(4):1434–1447, 1984.

Y.-C. Yao. Maximum likelihood estimation in hazard rate models with a change-point. *Communications in Statistics - Theory and Methods*, 15(8):2455–2466, 1986.

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

Y.-C. Yao and S. T. Au. Least-squares estimation of a step function. *Sankhya: The Indian Journal of Statistics, Series A*, 51(3):370–381, 1989.

E. Yudovina, M. Banerjee, and G. Michailidis. Changepoint inference for erdős–rényi random graphs. In A. Steland, E. Rafajłowicz, and K. Szajowski, editors, *Stochastic Models, Statistics and Their Applications*, pages 197 – 205. Springer, Cham, 2015.

T. I. Zack, S. E. Schumacher, S. L. Carter, A. D. Cherniack, G. Saksena, B. Tabak, M.S. Lawrence, C.-Z. Zhange, J. Wala, C. H. Mermel, C. Sougnez, S. B. Gabriel, B. Hernandez, H. Shen, P. W. Laird, G. Getz, M. Meyerson, and R. Beroukhim. Pan-cancer patterns of somatic copy number alteration. *Nature Genetics*, 45(10): 1134 – 1140, 2013.

K. D. Zamba and D. M. Hawkins. A multivariate change-point model for statistical process control. *Technometrics*, 48(4):539 – 549, 2006.

K. D. Zamba and D. M. Hawkins. A multivariate change-point model for change in mean vector and/or covariance structure. *Journal of Quality Technology*, 41(3):285 – 303, 2009.

J. Zdansky. Binseg: An efficient speaker-based segmentation technique. In *Proceedings on the Ninth International Conference on Spoken Language*, pages 2182 – 2185, Pittsburgh, United States of America, 2006.

B. Zhang, J. Geng, and L. Lai. Multiple change-points estimation in linear regression models via sparse group lasso. *IEEE Transactions on Signal Processing*, 63(9):2209 – 2224, 2015.

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

Z. Zhao, L. Chen, and L. Lin. Change-point detection in dynamic networks via graphon estimation. *arXiv:1908.01823v1*, pages 1 – 24, 2019.

C. Zheng, I. A. Eckley, and P. Fearnhead. Consistency of a range of penalised cost approaches for detecting multiple changepoints. *arXiv:1911.01716v1*, pages 1 – 54, 2019.

Q. Zhou, Y. Luo, and Z. Wang. A control chart based on likelihood ratio test for detecting patterned mean and variance shifts. *Computational Statistics & Data Analysis*, 54(6):1634 – 1645, 2010.

C. Zou, G. Yin, L. Feng, and Z. Wang. Nonparametric maximum likelihood approach to multiple change-point problems. *The Annals of Statistics*, 42(3):970 – 1002, 2014.

A. M. Zoubir and R. F. Brcich. Multiuser detection in heavy tailed noise. *Digital Signal Processing*, 12:262 – 273, 2002.