Change-Point Detection in an AR(1) Process via the Cross-Entropy Method

By

Lijing Ma

A thesis submitted to Macquarie University

for the degree of Master of Research

Department of Statistics

November 2017

MACQUARIE
University
SYDNEY·AUSTRALIA

Examiner's Copy

Typeset in LaTeX 2_ε.

Except where acknowledged in the customary manner, the material presented in this thesis is, to the best of my knowledge, original and has not been submitted in whole or part for a degree in any university.

Lijing Ma

Acknowledgements

I would sincerely like to express enormous thanks to my supervisor, Dr. Georgy Sofronov for his continuous guidance and motivation throughout the past year. My research would have been impossible without his aid and support. I would also like to thank the Department of Statistics and the Department of Mathematics, Macquarie University, for providing such a pleasant work environment. Lastly, heartfelt thanks go to my parents for their constant love, encouragement and support. Thanks for supporting every decision I made.

Abstract

In time series analysis, it is essential to check whether the observations are obtained by one or by several different mechanisms of data generation. In recent years, this problem, which is known as a change-point or break-point detection problem, has become a question of renewed interest for many researchers. Problems of this type arise in a wide range of applications, including financial time series analysis (e.g. changing volatility), signal processing (e.g. structural analysis of EEG signals), geology data analysis (e.g. analysis of volcanic eruption series) and environmental applications (e.g. detecting changes in ecological systems due to climatic conditions crossing some critical thresholds). This thesis focuses on detecting the changes in the mean level of autoregressive processes. We develop the Cross Entropy method for estimating the locations of change-points as well as parameters of the process in each segment. In order to identify the number of change-points, we use the Minimum Description Length information criterion. We apply the proposed method to simulated and real data to illustrate the usefulness of the approach.

Contents

List of Figures

List of Tables

1

Introduction

In practice, the statistical properties of time series data, such as mean or variance, abruptly change at unknown time point(s). This is known as a change-point problem, break-point problem or segmentation, and arises in wide range of scientific endeavours; for example, financial time series analysis ([36] [1]), econometrics ([5]), signal processing ([16] [46] [26]), genomics ([32] [18]), geology ([20] [31]) and environmental applications ([25] [28] [29] [10]). At the change points, observed data are segmented into several processes, with adjacent segments having different statistical properties. It is inappropriate to consider a time series a stationary process; applying the same statistical model to the whole set of observations would lead to inaccurate estimations, predictions and forecasting. Therefore, it is important to identify the number of change points, detect their locations and estimate the parameters of each segment.

Many methods for working with time series or sequences have been developed. We initially classify these methods, and then specify the research problem of this thesis. Generally,

there are two classes of change-point models: retrospective (off-line) and sequential (on-line). In the retrospective analysis, we apply change-point methods to past collected data, instead of at every step of data collection. The thesis does not present sequential-type methods; readers are referred to Brodsky and Darkhovsky [11] and Ross [40].

In addition, methods for time series segmentation may be divided into frequency-domain and time-domain methods, parametric and non-parametric methods, and in terms of linear and non-linear and univariate and multivariate time series. Frequency-domain methods and change-point detection literatures of multivariate time series are beyond the scope of this thesis but readers can refer to Preuss et al. [35] for literature related to frequency-domain method, Matteson and James [30] for multivariate time series segmentation. We will mainly review the approaches developed for univariate time series in time domain and focus on parametric methods.

There are essentially two branches of approaches for change-point estimation: the model selection method and hypothesis testing. The fundamental method of hypothesis testing is the likelihood ratio approach, originated to detect a single change point, which requires the maximum log-likelihood functions under both null (no change point) and alternative hypotheses (a single change point). The likelihood ratio approach for single change-point detection can be extended to the multiple case; for example, one extension is the binary segmentation (BS) algorithm ([43] [44]), the most popular method, used to detect the change in mean and variance. Olshen et al. [32] and Fryzlewicz [19] further improved the BS algorithm, proposing the circular BS (CBS) and wild BS approaches respectively. Moreover, Bai [4] proposed a consistent likelihood-ratio-type test for multiple structural changes.

The general idea of the model selection method is to minimise a penalised likelihood function by using optimisation methods. The model selection procedures based on the information criteria normally involve a penalty term selection problem. Akaike's information criterion (AIC) and the Bayesian information criterion (BIC) are well known penalty functions, which have been commonly used in change-point analysis, while the recently developed modified Bayesian information criterion (mBIC) (see Zhang and Siegmund [47]) and minimum description length (MDL) (see Davis et al. [16]) have been proposed to estimate the number of

change points. The frequently used optimisation algorithms search for change points based on the dynamic programming algorithm proposed by Auger and Lawrence [3] and Bai and Perron [6]. Jackson et al. [22] further explored this, developing the optimal partitioning method; Killick et al. [24] added pruning techniques, proposing the pruned exact linear time method to reduce computational cost. Moreover, genetic algorithms have been used to search for change points in time series by Doerr et al. [17]. Additionally, Evans et al. [18] proposed the cross-entropy method with maximum likelihood approaches to estimating change points in DNA sequences. Priyadarshana and Sofronov [36] applied a modified cross-entropy method with BIC to detect multiple break-points in stock price series. Further, Priyadarshana and Sofronov [38] strived to apply the cross-entropy method with modified BIC to detect multiple change points in genomes data. Another attempt is found in Bansal et al. [8], who applied the EM algorithm to the change-point problem.

Most of these change-point detection methods are based on the assumption of observations that are independent and identically distributed, from both Gaussian and non-Gaussian distributions. However, in reality, many time series show an inherent dependent structure; applying these methods to such data tends to cause false estimations. To date, there are few studies that pay attention to this problem. Bai [4] [7] made major contributions to detecting multiple change points in dependent structure of linear regression. Davis focused on autoregressive time series segmentation. The likelihood ratio test for an AR process with a single structural break was developed in 1995 [15]. Davis et al. [16](2006) considered the MDL information criterion as an objective function and used a genetic algorithm for optimisation; the procedure performed well for a piecewise AR model. In recent times, Yau and Zhao [46] (2016) proposed the likelihood ratio scan method with MDL information criterion to estimate multiple change points in piecewise stationary AR processes. Korkas and Fryzlewicz [26] (2017) paid attention to AR sequence segmentation, modifying the wild BS approach with CUSUM statistics. The research problem of this thesis concerns the fact that a stationary AR(1) model that presents moderate to strong autocorrelation is easily confused with structural breaks in the mean, and hence, may result in an overestimate problem if using common segmentation approaches. Our interest is to provide an effective computational method to estimate unknown multiple change points in a AR(1) process with breaks in the mean. For this particular problem, Chakar et al. [12] (2017) proposed a specific approach for estimating

change points in the mean of an AR(1) model.

Overall, in this thesis, we follow Davis et al. [16]'s study by using MDL information criterion to estimate the number of change points, the autocorrelation parameter and the mean shifts in an AR(1) model, and by using the cross-entropy method to locate and estimate the change points. We compare the accuracy of estimates obtained from the proposed method CE-MDL with CE-AIC, CE-BIC and some of the best performing techniques available in the literature. Numerical experiments illustrate the robustness of this approach. The main objective of this thesis is to present recent developments in model selection methods for retrospective time series segmentation, and to provide clear explanations of our proposed methodology and the details of the numerical experiments.

The thesis is structured as follows. Chapter 1 introduces the general background related to the study and objectives of the thesis. Chapter 2 provides the details of change-point analysis in time series, briefly presents the development of the time series segmentation methodology, separated into hypothesis testing and model selection, and lastly, introduces and describes our proposed method. Chapter 3 provides a detailed simulation study and presents the results of numerical experiments both for artificially generated and real data. Finally, Chapter 4 concludes the thesis with a general discussion and future research directions.

2

Methodology

2.1 The Change-Point Problem in Time Series

In many applications, data are collected over time. Often, the statistical properties, such as the mean and variance of the data or the coefficients of the regression model, change suddenly at unknown time-points. This is known as the change-point problem or segmentation in time series. Change-point tests can be classified by the distinction between posteriori, retrospective or historical (off-line) tests versus a priori, prospective or sequential (on-line) tests. The difference between these two methods lies in how data are collected. For the posteriori change-point tests, the process of data acquisition is completed at the moment the change-point test is applied, while for sequential structural break tests, the change point is checked simultaneously as new data come in. The sequential approach is particularly useful when a financial decision has to be made on-line, such as in risk management, asset allocation and portfolio selection; we refer readers to Andreou and Ghysels [1] for more details on sequential analysis in financial time series. In this thesis, we focus on retrospective

change-point methods and applications.

2.1.1 Popular Applications

In recent years, there has been an increasing interest to the change-point problem, mainly because it is widespread in many applications, including financial time series analysis (e.g., Andreou and Ghysels [1]), biomedical signal processing (e.g., Korkas and Fryzlewicz [26]), geology data analysis (e.g., Furlan [20]) and environmental applications (e.g., Beaulieu et al. [10]). Research effort has been concentrated in financial time series and climatological time series analysis, while there is increasing interest in other applications, such as signal processing, geology and environmental monitoring; further details are provided below.

Large volcanic eruptions are one of most catastrophic events on earth. There is an under-recording problem in the commonly used datasets recording the dates and magnitudes of eruptions over the last two millennia, with incomplete records before the 17th century. Furlan [20] take this recording bias into account and apply a change-point model to better fit the historical series of large magnitude volcanic activity. This paper gives an indication of how to apply a change-point model to censored count data.

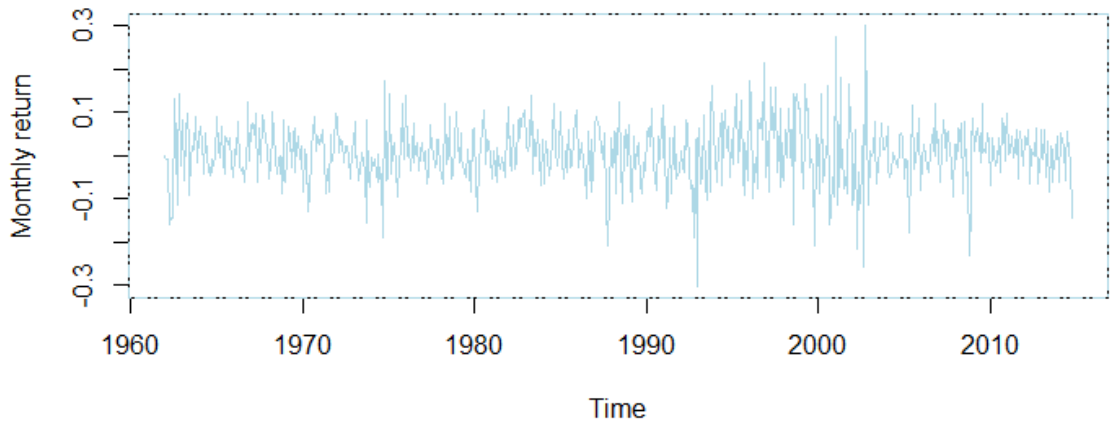

Figure 2.1: Plot of IBM monthly return

In the financial applications, efforts have been made to test multiple change points in financial asset returns and volatility, as well as in the distribution of financial time series. It is common

to describe the observations as non-stationary, with change points splitting the sample into piecewise stationary segments. The accurate prediction and correct estimation of parameters in non-stationary models only can be conducted after the detection of structural changes in data. For example, Bauwens et al. [9] adapted the AR-GARCH model to estimate multiple change points; the AR(0)-GARCH(1,1), AR(1)-GARCH(1,1) and AR(2)-GARCH(1,1) models were proposed. Andreou and Ghysels [1] pointed out that multivariate volatility models with ARCH or long memory type effects are largely unexplored, and continuous time stochastic volatility models with structural breaks attract less attention. Figure 2.1 shows stock monthly return data.

Change-point analysis of electroencephalogram (EEG) is a very popular area because of its complex structure and usefulness. An EEG signal is a recording of the electrical activity of the brain from the scalp. The different waveforms represent different cortical electrical activities (e.g., excited-drowsy-deep sleep). The large literature provides mathematical methods for analysing EEG, with most studies treating the EEG as a non-stationary process. Recently, articles have analysed electrocardiogram (ECG) data, used to check for problems with the heart's electrical activity. Figure 2.2 shows ECG data, which can be found in the R package *wavethresh*.

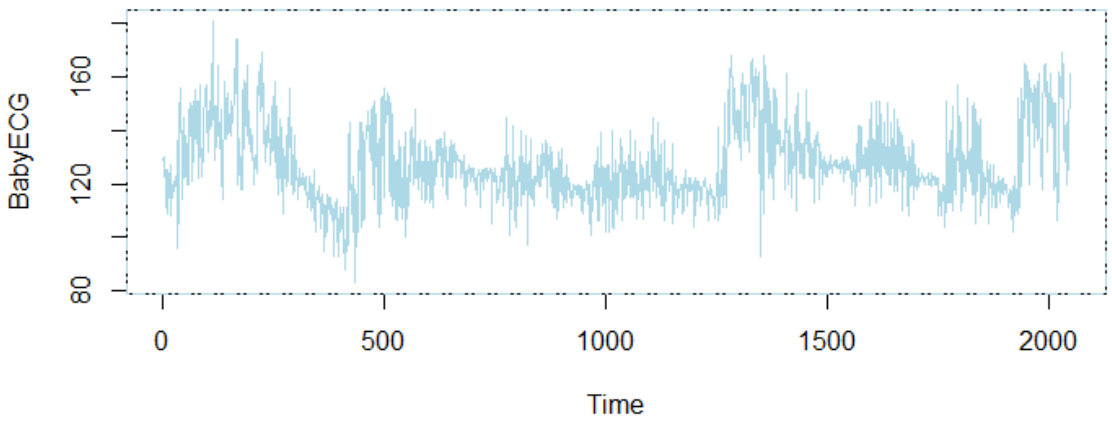

Figure 2.2: Plot of infant electrocardiogram data

In addition, change-point analysis in the area of climatology is popular, as there is much concern regarding environmental quality and climate change. Many multiple change-point

detection procedures for the climate segmentation problem are based on the assumption that the series is driven by independent and identically distributed errors. This is an unrealistic assumption in climate time series, where the data are known to present moderate to strong autocorrelation; in particular, as shown in figure 2.3, the periodicity of the monthly series makes detection difficult. To address this problem, Lu et al. [29] used a MDL approach, which demonstrated a superior empirical performance. The climate time series in figure 2.3 are built in the R package *iki.dataclim*.

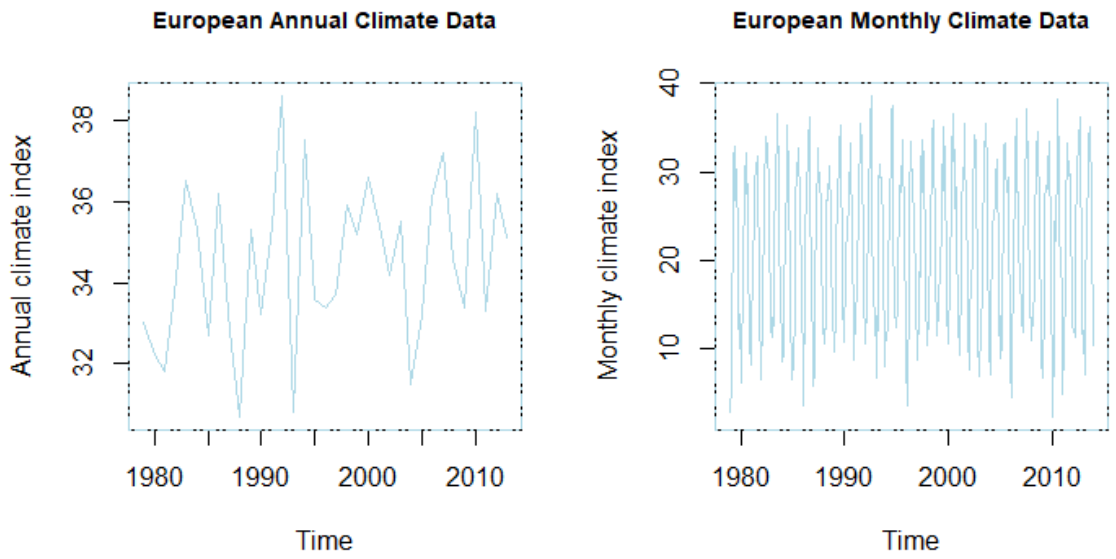

Figure 2.3: An example of annual and monthly climate time series

Overall, change-point analysis has a wide range of applications, and many examples of non-stationary time series exist (e.g., figure 2.2). With the unknown number and locations of change points, as well as the complex structure of time series (the monthly climate data in figure 2.3 represents an extreme example), it is very difficult to precisely estimate the change points. The conventional way of modelling non-stationary time series is to identify the structural breaks and segment the data into piecewise stationary processes. Davis et al. [16] (2006) proposed the valuable idea of using a piecewise AR model to fit the non-stationary process and developing the auto-PARM method to estimate the unknown number and locations of change points, as well as the orders of the respective AR processes. Cho and Fryzlewicz [14] (2012), Yau and Zhao [46](2016) and Korkas and Fryzlewicz [26] (2017) followed this idea to propose the MSML method, likelihood ratio scan method and WBSts method respectively. This research area is still very active and needs to be further explored.

2.1.2 Mean Shifts and Autocorrelation Structure

The change-point detection method was originally introduced in quality control by Page [34] [33] (1954, 1955) for detecting the change points suddenly occurring in the mean of independently and normally distributed time series. Since then, many methods have been developed, based on the assumption of normal and non-normal independent distributions, involving single and multiple change-point detection, in univariate and multivariate time series, in linear and non-linear sequences, with mean, variance and trend changes.

Figure 2.4 illustrates a single change point at time 50 in the (a) mean (b) variance only (c) the mean and variance (d) the intercept of a linear regression mode (e) both the intercept and the slope of a linear regression model (f) no change point in the AR(1) model with strong positive autocorrelation of 0.8.

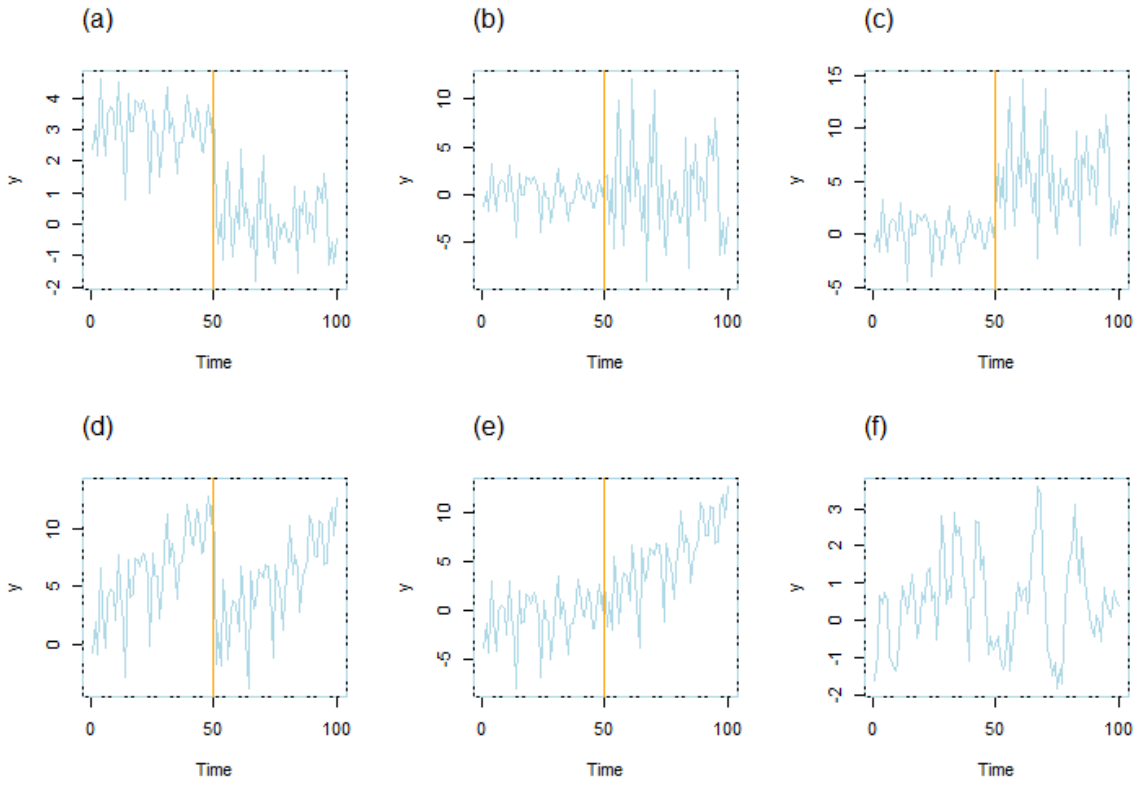

Figure 2.4: A change point in linear or non-linear time series [10]

As shown in figure 2.4 (f), the strong autocorrelation patterns in an AR(1) model can be easily confused with change points. We tend to perceive a variation in the mean in (f), even though there is none. As mentioned earlier, non-stationary time series can be segmented into

a piecewise stationary AR model, so it is important to accurately identify the change points in each AR segment. This is an interesting and challenging problem. Recently, Chakar et al. [12] (2017) proposed a new approach for estimating the number and locations of multiple change points in the mean of a Gaussian AR(1) process. In this thesis, we develop an innovative methodology to tackle this problem.

2.2 Hypothesis Testing Methods

Hypothesis testing was originally designed for at most one change, but can be easily extended to multiple change-point detection in posterior parametric change-point analysis. For single change-point testing, the null hypothesis H_0 is no change point in a time series; the alternative H_1 is that there is exactly one change, at an unknown location. Forecasting procedures may be accurately implemented as long as the observed data satisfy the underlying no-structural-break null hypothesis. In multiple change-point problems, the primary concern of both likelihood test statistics and the BS method is how to choose a threshold at which to stop the procedure.

2.2.1 Likelihood Ratio Test

In many cases, data are assumed to follow an independent normal distribution; the likelihood ratio approach for normal and non-normal distributions can be found in [13]. Davis et al. [15] (1995) focused on the case of a univariate segmented AR model, this specific example is provided below.

General Likelihood Ratio Approach

Let $x_1, x_2, \ldots, x_n$ be a sequence of random vectors with probability distribution functions $F_1, F_2, \ldots, F_n$, respectively. Then, in general, the change-point problem is to test the following null hypothesis:

$$H_0 : F_1 = F_2 = \cdots = F_n \tag{2.1}$$

versus the alternative hypothesis:

$$H_1 : F_1 = \cdots = F_{k_1} \neq F_{k_1+1} = \cdots = F_{k_2} \neq F_{k_2+1} = \cdots F_{k_q} \neq F_{k_q+1} = \cdots = F_n \tag{2.2}$$

where $1 < k_1 < k_2 < \cdots < k_q < n$, q is the unknown number of change points and $k_1, k_2, \ldots, k_q$ are their respective unknown positions, which must be estimated. If $q = 1$, specifically, this refers to the single change-point problem. Moreover, if the distributions $F_1, F_2, \ldots, F_n$ belong to a common parametric family F_θ, then the change-point problem is to test the null hypothesis regarding the population parameters $\theta_i, i = 1, \ldots, n$:

$$H_0 : \theta_1 = \theta_2 = \cdots = \theta_n \equiv \theta \tag{2.3}$$

where θ is unknown, and assumed to generate the observations, versus the alternative hypothesis:

$$H_1 : \theta_1 = \cdots = \theta_{k_1} \equiv \theta \neq \theta_{k_1+1} = \cdots = \theta_{k_2} \equiv \theta_1^* \neq \cdots \neq \theta_{k_q+1} = \cdots = \theta_n \equiv \theta_q^* \tag{2.4}$$

Thus, under the alternative hypothesis, the data are split into $q+1$ segments, $x_1, x_2, \ldots, x_{k_1} \mid x_{k_1+1}, x_{k_1+2}, \ldots, x_{k_2} \mid \cdots \mid x_{k_q+1}, x_{k_q+2} \ldots, x_{k_n}$, with segments have different generating parameters $\theta, \theta_1^*, \theta_2^*, \cdots, \theta_q^*$. The likelihood function under the alternative hypothesis is $L_{q+1}(\theta, \boldsymbol{\theta}^*)$, $\boldsymbol{\theta}^* = (\theta_1^*, \theta_2^*, \cdots, \theta_q^*)$, which can be compared to the likelihood $L_n(\theta)$ from the null hypothesis. The likelihood ratio is:

$$\Lambda_q = \frac{\max_\theta L_n(\theta)}{\max_{\theta, \boldsymbol{\theta}^*} L_{q+1}(\theta, \boldsymbol{\theta}^*)} = \frac{L_n(\hat{\theta_n})}{L_{q+1}(\hat{\theta_0}, \hat{\boldsymbol{\theta}}^*)} \tag{2.5}$$

$\hat{\theta_0}, \hat{\boldsymbol{\theta}}^*$ are the maximum likelihood estimators (MLE) for each subsample. If $\Lambda_q > \beta$, we do not reject H_0; if $\Lambda_q < \beta$, we reject H_0, where β is an arbitrary positive constant. Some authors [2] consider the test statistics, $Z_n = \max(-2 \log \Lambda_q)$, and if Z_n is large, reject the null hypothesis. For the likelihood-based methodology, the main problem is the selection of the threshold β.

Autoregressive Model Example

Davis et al. [15] (1995) applied the likelihood ratio test to a segmented autoregressive process with single change point, $\varepsilon_t \sim i.i.d.N(0, \sigma^2)$, and length of sequence T:

$$x_t = \begin{cases} \phi_0 + \phi_1 x_{t-1} + \cdots + \phi_p x_{t-p} + \varepsilon_t, & t = 1, \ldots, k \\ \phi_0^* + \phi_1^* x_{t-1} + \cdots + \phi_q^* x_{t-q} + \varepsilon_t, & t = k+1, \ldots, T \end{cases} \tag{2.6}$$

Assume that the change point is at lag k. Under the null hypothesis H_0, $k \geqslant n$, the data are generated by vector of parameters $\boldsymbol{\theta} = (\phi_0, \phi_1, \ldots, \phi_p, \sigma^2)$. The exact likelihood function of null hypothesis is:

$$\begin{aligned} L(x_t, x_{t-1}, \ldots, x_1; \boldsymbol{\theta}) = & f_{X_p, X_{p-1}, \ldots, X_1}(x_p, x_{p-1}, \ldots, x_1; \boldsymbol{\theta}) \\ & \times \prod_{t=p+1}^{T} f_{X_t | X_{t-1}, \ldots, X_{t-p}}(x_t | x_{t-1}, \ldots, x_{t-p}; \boldsymbol{\theta}) \end{aligned} \tag{2.7}$$

Since it is difficult to compute the exact likelihood function, particularly, the joint distribution, $f_{X_p, X_{p-1}, \ldots, X_1}(x_p, x_{p-1}, \ldots, x_1; \boldsymbol{\theta})$ may not easy to be derived, so the conditional likelihood

function method based on the first p observation is generally used. The log of the conditional likelihood function is:

$$LL(x_t, x_{t-1} \ldots, x_{p+1}|x_p, \ldots, x_1; \boldsymbol{\theta}) = -\frac{T-p}{2}\log(2\pi) - \frac{T-p}{2}\log(\sigma^2) - \sum_{t=p+1}^{T} \frac{(x_t - \phi_0 - \phi_1 x_{t-1} - \cdots - \phi_p x_{t-p})^2}{2\sigma^2} \tag{2.8}$$

Hence, the conditional MLE of $\boldsymbol{\theta} = (\phi_0, \phi_1, \ldots, \phi_p, \sigma^2)'$ can be obtained by minimising $\sum_{t=p+1}^{T}(x_t - \phi_0 - \phi_1 x_{t-1} - \cdots - \phi_p x_{t-p})^2$, the conditional sum-of-squares. Under the alternative hypothesis H_1, $1 \leq k < n$, the data are generated by vector of parameters $\boldsymbol{\theta} = (\phi_0, \phi_1, \ldots, \phi_p, \sigma^2)'$, $\boldsymbol{\theta}^* = (\phi_0^*, \phi_1^*, \ldots, \phi_q^*, \sigma^2)'$, and the exact likelihood function is:

$$\begin{aligned} L(x_t, x_{t-1}, \ldots, x_1; \boldsymbol{\theta}, \boldsymbol{\theta}^*) &= L_{X_k, X_{k-1}, \ldots, X_1}(x_k, x_{k-1}, \ldots, x_1; \boldsymbol{\theta}) \\ &\quad \times L_{X_t, X_{t-1}, \ldots, X_{k+1}}(x_t, x_{t-1}, \ldots, x_{k+1}; \boldsymbol{\theta}^*) \\ &= f_{X_p, X_{p-1}, \ldots, X_1}(x_p, x_{p-1}, \ldots, x_1; \boldsymbol{\theta}) \\ &\quad \times f_{X_{k+q+1}, X_{k+q}, \ldots, X_{k+1}}(x_{k+q+1}, x_{k+q}, \ldots, x_{k+1}; \boldsymbol{\theta}^*) \\ &\quad \times \prod_{t=p+1}^{k} f_{X_t|X_{t-1}, \ldots, X_{t-p}}(x_t|x_{t-1}, \ldots, x_{t-p}; \boldsymbol{\theta}) \\ &\quad \times \prod_{t=k+q+2}^{T} f_{X_t|X_{t-1}, \ldots, X_{t-q}}(x_t|x_{t-1}, \ldots, x_{t-q}; \boldsymbol{\theta}^*) \end{aligned} \tag{2.9}$$

Davis et al. [15] developed a likelihood ratio test statistics for this model. The main problem is the selection of threshold. At that time, the primary concern is the difficulty in computing the MLE of model parameters from exact likelihood function under alternative hypothesis, due to the complex joint distributions of sub-sequences. Nowadays, we can easily obtain the MLE from exact likelihood function by using R package *arima*. It is important to note that the likelihood ratio test is based on an assumption of known number of change-points.

2.2.2 Binary Segmentation Algorithm Family

The BS method used for detecting multiple change points is based on at most one change-point hypothesis test; thus, it can be considered one of the hypothesis testing methods. The BS method also involves the problem of developing the test statistic and selecting the appropriate threshold. The BS algorithm family includes the general BS proposed by [43], [44], CBS

[32] and the wild segmentation algorithm [19]. In this section, we analyse the pros and cons of BS algorithms.

General Binary Segmentation Algorithm

Consider a time series sequence $x_1, x_2, \ldots, x_n$, assuming the data are normally distributed with a known variance. BS uses the likelihood ratio statistic for testing the null hypothesis (no change point) against the alternative (exactly one change point). It is one of the most popular methods in change-point detection, and has the advantages of having low computational cost and being easy to code. The details of the technique are introduced in the following steps:

1. Testing the no change-point hypothesis (2.1) against the one change-point hypothesis (2.2), starting with the entire dataset. If H_0 is not rejected, stop - the output is no change point. If H_0 is rejected, a change point was detected, and we progress to step 2.

2. The change point found in step 1 partitions the data into two sub-sequences; apply the single change-point hypothesis test separately for each sub-sequence.

3. Iterate until no further sub-sequences have change points.

The idea behind the BS algorithm is simple and logical - it searches for a single change point at a time and never re-visits a segment that has been tested, which makes it computationally fast, with $O(n \log n)$ computational cost. However, the BS method tends to yield crude estimates of the number and locations of change-points. The reason for this is that, for example, after the first test, the data are segmented into two sub-sequences, with the length of each segment based on the location of the change point detected in the last procedure - this causes an underestimation problem because certain segments may contain multiple change points. Olshen et al. [32] proposed CBS to solve this problem. Fryzlewicz [19] argued that when the length between two change points is too small, less than $O(T^{3/4})$, T is the number of observations, the BS method has a tendency to underestimate the number of change points.

Circular Binary Segmentation Algorithm

Mutations in copy number are common in cancer and other diseases, so it is important to detect and identify the aberrant genomic regions. CBS targets analysing the change in the mean of array-based DNA copy-number data. The method aims to detect the clustered,

small structural breaks buried in the middle of a large dataset. The authors modified the BS method and added the permutation approach to render it more appropriate for array DNA sequences. The CBS method considers each segment or sub-segment as a circle and develops a new likelihood ratio statistic for hypothesis testing. Moreover, the CBS method reduces the computational cost to $O(n)$. However, with a known number of change points, there is no guarantee that the CBS method will obtain the optimal locations of change points.

Wild Binary Segmentation Algorithm

The motivation for the wild BS technique is to eliminate the weakness of the BS method and, at the same time, to inherit its strength. Fryzlewicz [19] proposed two new criteria to stop the wild BS procedure: one is based on the sub-sample CUSUM statistic, rather than the global CUSUM statistic; the other uses the strengthened Schwarz information criterion (sSIC). The authors introduced the model selection procedure in the BS method to improve its accuracy, as the model selection methods mostly use exact optimisation algorithms to obtain the global optimal. This contrasts with BS-like methods, as the solutions of model selection methods are more accurate but involve greater computational burden. In the next section, we provide further detail on model selection methods.

2.3 Model Selection Methods

In practice, the number of change points is always unknown, and a major challenge in change-point analysis is to determine the number of change points, as well as locating their positions. The model selection method set out below was developed to identify the unknown number of change points, and it promises superior performance relative to the likelihood ratio test method in this context. To help explain this idea, we reuse the segmented autoregressive model example demonstrated in the likelihood ratio test section. Here, the model under the null hypothesis with no change is:

$$x_t = \phi_0 + \phi_1 x_{t-1} + \cdots + \phi_p x_{t-p} + \varepsilon_t, t = 1, \ldots, T \tag{2.10}$$

We assume the number of change points is N, the location vector is $\tau = (\tau_1, \tau_2, \ldots, \tau_N)$, mathematically speaking, $1 \leq N \leq T - 1$; then the model with N change-points is:

$$x_t = \begin{cases} \phi_0 + \phi_1 x_{t-1} + \cdots + \phi_p x_{t-p} + \varepsilon_t, & t = 1, \ldots, \tau_1 \\ \phi_0^{(1)} + \phi_1^{(1)} x_{t-1} + \cdots + \phi_{p_1}^{(1)} x_{t-p_1} + \varepsilon_t, & t = \tau_1 + 1, \ldots, \tau_2 \\ \vdots & \\ \phi_0^{(N)} + \phi_1^{(N)} x_{t-1} + \cdots + \phi_{p_N}^{(N)} x_{t-p_N} + \varepsilon_t, & t = \tau_N + 1, \ldots, T \end{cases} \tag{2.11}$$

The example illustrates that the likelihood ratio test cannot be used without knowing the number of changes N, so it is important to make an inference about N first. Bai [4] proposed a conditional likelihood ratio test, based on the sum-of-square function for the unknown multiple structural breaks problem, and a new test statistic was built to determine N. As long as the N was determined, we obtain a specific model, and the change-point locations and other model parameters can be estimated from it. Overall, different values for N refer to models with different parameter dimensions; we need to select the best model, this is the model selection problem.

In this thesis, we focus on the information approach, which is a general model selection technique commonly used for detecting unknown multiple change points. Compared with hypothesis methods, there are two main advantages to the information approach. First, it is able to simultaneously estimate the number of changes, their locations and the other model parameters by optimising an information criterion. Second, it can be easily adapted to diverse

situations; not only to random variables based on assumptions of independence and normality, but also to the linear regression model with dependent structure and the autoregressive model. Our method is based on the information approach. In following subsections, we introduce the popular information criteria and optimisation algorithms available in change-point detection literatures.

2.3.1 Model Selection Criteria

Let $X = (X_1, X_2, \ldots, X_T)$ be a data sequence of length T. Then data are segmented by change points into $N + 1$ segments defined by N change points $\tau = (\tau_1, \tau_2, \ldots, \tau_N)$, $0 = \tau_0 < \tau_1 < \tau_2 < \cdots < \tau_N < \tau_{N+1} = T$, where the i-th segment includes observations $(X_{\tau_{i-1}+1}, \ldots, X_{\tau_i})$. In general, the problem of identifying unknown multiple change points can be considered equivalent to minimising the objective function [23]:

$$\sum_{i=0}^{N} \left[C(X_{(\tau_i+1):\tau_{i+1}}) \right] + \mathcal{P} \tag{2.12}$$

where the C is a cost function for a segment, and $\mathcal{P}$ represents a penalty term that takes model complexity into account. Logically, we choose the model with the lowest costs in 2.12 as the best model. In change-point analysis, the negative maximum log-likelihood, $-2 \log \mathcal{L}$ is the commonly used cost function; the cumulative sum is also used. In addition, Bai & Perron used the sum of squared residuals as the cost function; Zhang and Siegmund [47] took the length of the segment into account, combining this with the log-likelihood of each segment to form a cost function. More recently, Beaulieu et al. [10] extended the residual sum of squares cost function for segmenting an autoregressive AR(p) model.

Information criteria are widely used for selecting the best model from several competing statistical models. The main idea is to choose the simplest model with good fit. The popular information approaches to the change-point problem include the Akaike information criterion (AIC), Bayesian information criterion (BIC) and the minimum description length principle. The major difference between these information criteria is the amount of penalty added to each parameter of the model. The penalty terms for the AIC and the BIC are given below:

$$\diamond \text{ AIC: } \mathcal{P} = \theta_k * 2$$

$$\diamond \text{ BIC: } \mathcal{P} = \theta_k * \log(T)$$

where θ_k represents the total number of parameters in the model. Thus, each parameter is penalised by same amount. It is clear that the larger the number of model parameters, the greater the penalties attached.

Compared with the AIC and the BIC, the MDL method can tailor penalties for parameters having different natures, while the other information criteria treat every parameter as imposing the same penalty. MDL methods have been demonstrated to be useful in recent literature; for example, Davis et al. [16] proposed that the MDL method has a superior empirical performance for autoregressive models, and developed the penalty separately for the number of change points, each location, the order of each segment and the mean shift parameter. Li and Lund [28] developed an objective function in terms of log-normal distributions, to model annual precipitation data. Lu et al. [29] also used the MDL method to segment climatic time series. It is important to note that the penalty term $\mathcal{P}$ of the MDL method varies with the posterior distribution of the observed series, so there does not exist a general expression for the penalty term function of the MDL.

The MDL principle is rooted in information theories. We refer readers to [21] for an introduction to MDL. Briefly, MDL defines the best model as the one that can compress a large amount information by using less computer memory storage. In statistics, the amount of computer memory is also called code length, *CL*. Because of the flexibility of the MDL principle, there are multiple versions of the MDL penalty term. Here, we present the two-part MDL, and develop our objective function by following this version.

The general procedures of the MDL approach used for estimating the number of changes are described below:

1. **Decomposition of observed data $\{X_t\}$** . Decompose the data $\{X_t\}$ into two parts, the fitted model $\{\widehat{X}_t\}$ and the residuals model $\{\hat{\epsilon}_t\}$ conditional on the fitted model:

$$CL(\{X_t\}) = CL(\{\widehat{X}_t\}) + CL(\{\hat{\epsilon}_t\}) \tag{2.13}$$

   where the total code length consists of $CL(\{\widehat{X}_t\})$ and $CL(\{\hat{\epsilon}_t\})$. Taking into account the change-point problem, the data are segmented by change points into $N + 1$ segments,

using locations $\tau = (\tau_1, \tau_2, \ldots, \tau_N)$; the i-th segment is defined as η_i, and $\boldsymbol{\theta}$ denotes the other parameters of model.

$$CL(\{\widehat{X}_t\}) = CL(\{\hat{N}\}) + CL(\{\hat{\tau}\}) + CL(\{\hat{\boldsymbol{\theta}}\}) \tag{2.14}$$

$$CL(\{\widehat{X}_t\}) = CL(\{\hat{N}\}) + CL(\{\eta_1, \eta_2, \ldots, \eta_{N+1}\}) + CL(\{\hat{\boldsymbol{\theta}}\}) \tag{2.15}$$

Davis et al. [16] suggested that equation 2.14 and 2.15 are interchangeable, as when the locations of the change points are known, the information (length) of each segment is completely known. The main task is to find the computational expressions for code length.

2. **Calculate the total code length**. According to Shannon's information theory, $CL(\{\hat{\epsilon}_t\})$ is given by the negative log-likelihood of the fitted model. To derive the expression $CL(\{\widehat{X}_t\})$, we follow two basic principles, which are the essence of the MDL method. First, a real-valued parameter computed from a large number of observations, for example, the whole series T, takes up $\log_2(T)/2$ bits of memory, thus $\log(T)/2$ is used as the code length for matching the calculation of the log-likelihood. Second, an integer parameter bounded by Q that can be encoded with $\log_2(Q)$ bits, such as $\log(Q)$, is used for developing an objective function.

3. **Find the minimal code length**. From MDL's point of view, the best model is the one that produces minimal code length. After obtaining the calculable expression of total code length according to the principles from step 2, we can apply optimisation algorithms to obtain the minimised optimal code length.

2.3.2 Dynamic Programming Algorithms

Dynamic programming (DP) algorithms can be used as a mathematical optimisation technique in change-point segmentation. It is complementary to BS-type algorithms, with significant computational costs but the ability to output the global optimum. In contrast to BS algorithms, DP algorithms and derivations provide exact solutions. In the BS procedure, the current change-point locations are conditional on change points detected in the last step, whereas DP algorithms obtain the optimal segmentations by reusing all previous information of possible combinations. As the datasets become larger, the entire segmentation space and possible N increase, and this causes a heavy computational burden to minimise the cost function 2.12, this is the main concern regarding DP methods. Many methods have been developed to reduce computational complexity.

Segment Neighbourhood Algorithm

Auger and Lawrence [3] (1989) proposed the segment neighbourhood (SN) algorithm, by setting a maximum number of change points N_{max} to reduce the possible segments, with only candidate segmentations with change points between 0 to N_{max} involved in the computation. It identifies the optimal partitions for N_{max} change points by reusing the results calculated for $N_{max} - 1$. Further, the method allows an arbitrary penalty term. Hence, this exhaustive search technique has a slow computational speed, with $O(N_{max}T^2)$ computational cost.

Bai and Perron's Method

The authors [7] (2003) further explored the SN algorithm, aiming to improve computational speed by adding a step of setting the minimum segment length and using the DP algorithm to optimise the sum of least-squares objective function. The method allows for a dependent structure of linear regression observations, and both numerical and theoretical results suggest that the procedure has excellent empirical properties. Bai and Perron [6] (1998) solved the theoretical properties of using the sum of least-squares residuals as the cost function to detect the unknown multiple structural changes in linear models. Therefore, this is one of the most widely used methods. Weideman et al. [45] (2017) recently applied Bai and Perron's method to identify if there were structural breaks in the renewable energy market of South Africa in the period 1990 to 2010.

PELT Approach

The main problem of DP algorithms is computational complexity, especially when the number of change points linearly increases with the length of the dataset; for example, in analysing larger regions of genomes or recording financial time series over longer periods, in this context, the SN method will yield a computational cost of $O(T^3)$. To reduce computational cost, with as high a detection accuracy as possible, the pruned exact linear time (PELT) method was developed. The PELT method is based on the optimal partitioning (OP) method proposed by Jackson et al. [22]. The PELT method modified the OP algorithm by adding a pruning step, which aimed to remove the values that can never be minima from the optimization procedures, abandoning them in the next iteration, and hence, reducing computational cost to $O(T)$. However, both the OP and the PELT methods are limited by their assumptions: that data are independent within a segment, and that the penalty term used in 2.12 is a constant that does not depend on the number or location of change points.

2.3.3 Evolutionary Algorithms

Evolutionary Algorithms (EAs) have been implemented in a wide range of applications, with the dramatic development in technologies. EAs are designed to solve the difficult optimisation problem by using evolutionary computation techniques. It is natural to consider applying EAs for solving the multiple change-point detection problem by minimising an objective function 2.12. One of the EAs, the genetic algorithm (GA), has been studied by several authors in change-point analysis, such as [16], [28],[29] and [17]. Most have focused on models with complex dependent structures. In addition, few authors have considered the application of the EM algorithm to the change-point problem [27], [8]. Recently, the cross-entropy (CE) method has been considered for detecting break points in biological sequences [38], [18] and financial time series [36].

Compared with DP methods, EAs are able to provide exact solutions without making restrictive assumptions about the objective function; as mentioned above, the PELT method is based on the assumption of a constant penalty term. This is the main advantage of EAs. However, evolutionary type algorithms have the same problem as DP methods - high computational costs. We provide below a brief introduction of the GA, but focus on the CE method.

Genetic Algorithm

A GA is a stochastic optimisation method designed to follow the principle of 'Darwinian natural selection' and mimic the natural evolution process. The selection procedure guarantees that the objective function values of the offspring will gradually be improved over generations and approach the near-global optimum. The formal structure of the GA for minimising the objective function is displayed below:

1. Initially, generate a random population, which is called chromosomes.
2. Obtain the minimal values of the objective criterion, to determine the fitness of chromosomes.
3. Repeat
    (a) Choose healthy parent chromosomes from the initial population, the lower the

value of the objective function calculated from the population, the higher the possibility of being chosen.

(b) Perform crossover or mutation on the chosen parent chromosomes to produce offspring.

(c) Determine the fitness of child chromosomes.

4. Until the most fit individual is chosen.

Because of this unique evolutionary selection procedure, GA is indifferent to the expression of the objective function. This is the competing feature of GA. For example, as the MDL objective function of the segmented AR(p) process contains a large number of parameters and optimising MDL(N, $\tau_1, \tau_2, \ldots, \tau_N$, $p_1, p_2, \ldots, p_N$) is tedious, Davis et al. [16] proposed that GA promises a good result for minimising this complex objective function. Lu et al. [29] used GA to solve the optimisation problem for the periodic climate time series model.

Cross-Entropy Algorithm

The CE method is an evolutionary computing technique developed by Rubinstein and Kroese [41] to solve the complex optimisation problem in the multiple change-point detection problem. It is based on one of the fundamental concepts in modern information theory Kullback-Leibler information or cross-entropy [37]. Entropy is a fundamental quantity to measure total information-entropy of a message per bit multiplied by the length of that message. It is also defined as a functional of probability distributions in mathematics. CE measures the directed distance between the two probability distributions. The basic idea behind the CE method is summarised as a three-step iteration process: first, simulate a vector of candidate solutions for the problem, based on an appropriate statistical distribution. Next, calculate the performance function score for each candidate solution. Third, obtain the improved solution set in the next iteration by updating the parameters of the statistical distribution through minimising CE.

Compared with GAs or other stochastic optimisation methods, the CE method implements all its optimisation procedures by updating a parametrised probability distribution. The probabilistic approach enables the current best solution to have a greater probability of appearing in the next iteration. This distribution-based feature is the main advantage of the CE method compared with other optimisation approaches.

In addition, the CE method is an ideal methodology to address the unknown multiple change-point problem. First, as one of the best EAs, it does not require strict assumptions about the objective function, and performs well with an objective function containing a large number of uncertain parameters. Second, as mentioned earlier, in most applications, the number and locations of change points are not known in advance; by using the CE method, we can estimate the number of change points and their corresponding locations together. The details of this procedure are presented below:

Algorithm 1 The general CE algorithm for estimating both the number of change-points and their locations

1: Given the search space for the number of change-points $N_{min} \leq N \leq N_{max}$, and for each value of N, repeat,

   (1) Set the statistical distribution with the initial parameters $\boldsymbol{\theta}^{(0)}$, let the sample size be M, the iteration starts with $k = 1$.

   (2) Generate M sets of change-point locations vector $\boldsymbol{\tau}^{(1)}, \boldsymbol{\tau}^{(2)}, \ldots, \boldsymbol{\tau}^{(M)}$ from statistical distribution with parameters $\boldsymbol{\theta}^{(k-1)}$ from last iteration, where $\boldsymbol{\tau}^{(i)} = (\tau_1^{(i)}, \tau_2^{(i)}, \ldots, \tau_N^{(i)}), i = 1, 2, \ldots, M$.

   (3) For each sample i, order the simulated locations in ascending order and calculate the performance function score or the values of objective function based on 2.12, sort all M samples in increasing order.

   (4) Obtain the first M_{elite} samples from last step, $M_{elite} = \alpha \times M$, the elite samples can be used to update the parameter of statistical distribution in step 1.

   (5) Once the process has met a stopping criterion, we stop the process. The current solution is the estimates of locations.

2: In order to obtain the estimates of N, minimize the the performance function score with the estimates of locations from previous step.

3: Finally, the optimal estimates of N and its corresponding locations are acquired.

2.4 Detecting Change-points in AR(1) Structure

In this section, the detailed description of our methodology is provided. We utilize the MDL criterion to estimate the number of change-points, and apply the cross-entropy method to obtain the locations of change-points as well as estimating the parameters of each segment in a Gaussian AR(1) process. We call this new method the CE-MDL method. At the beginning of chapter two, we display many interesting problems in time series segmentation. In this thesis, we are interested in detecting the mean shifts in AR(1) process, and will not consider the problem of detecting change-points in an ambitious model.

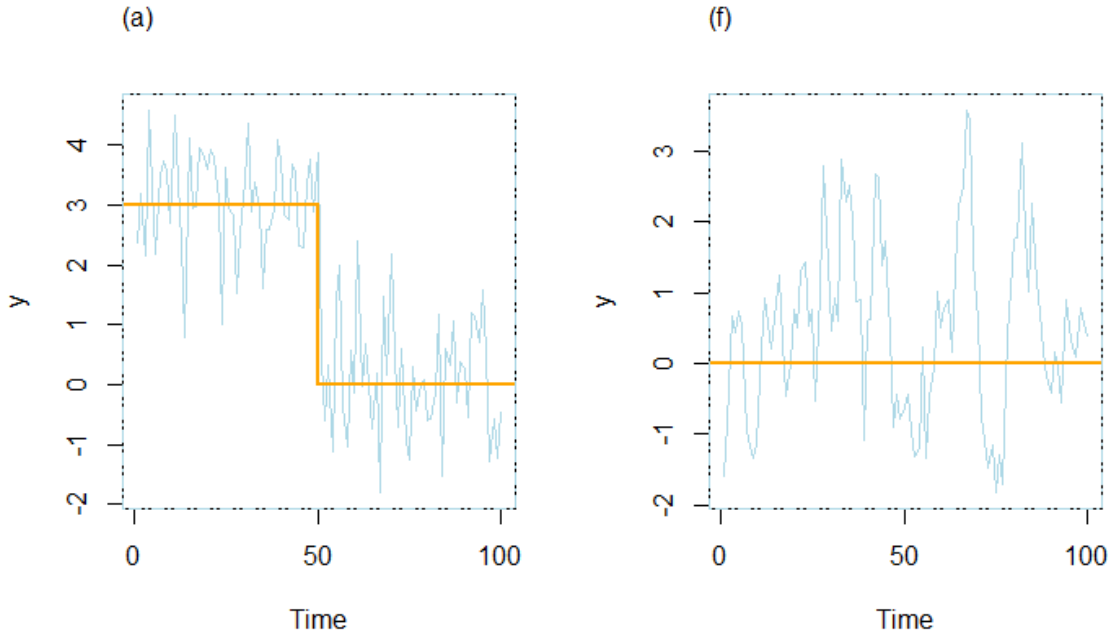

Figure 2.5: An example of true mean shift

We reuse the examples (a) and (f) shown in Figure 2.4 to display our research problem. There are several steep slopes in the AR(1) process which has a strong autocorrelation, they are likely to be wrongly identified as change-points when using most of existing change-point detect techniques. In other words, the inherent autocorrelation structure of AR(1) process is easily confused with true mean shifting. The CE-MDL method is developed to address this type of problem. We build an AR(1) model with changes in the mean.

2.4.1 Model Settings and Assumptions

Let $X = (X_1, X_2, \ldots, X_T)$ be a data sequence of length T. Suppose the number of change-points is N, then datasets are segmented into $N + 1$ segments, each segment is defined as η_i, $\tau = (\tau_1, \tau_2, \ldots, \tau_N)$ is the location vector, $0 = \tau_0 < \tau_1 < \tau_2 < \cdots < \tau_N < \tau_{N+1} = T$, where the i-th segment includes observations $(X_{\tau_{i-1}+1}, \ldots, X_{\tau_i})$. In this thesis, we consider an AR(1) model with mean shifts and assume that the autocorrelation coefficient ρ is same for each segment and σ^2 is homogeneous:

$$X_t = \rho X_{t-1} + \delta_i \mathcal{I}_{\lfloor \tau_{i-1}+1 \leqslant t \leqslant \tau_i \rfloor} + \varepsilon_t, \qquad t = 1, 2, \ldots, T \tag{2.16}$$

with independent and identically distributed $\varepsilon_t \sim N(0, \sigma^2)$, where $\mathcal{I} = 1$ for $\tau_{i-1} + 1 \leqslant t \leqslant \tau_i$ and 0 everywhere else, $1 \leqslant i \leqslant N + 1$, which allows for shifts in the mean level, $\delta = (\delta_1, \delta_2, \ldots, \delta_{N+1})$.

2.4.2 The MDL Setup

The first and foremost task is to develop an objective function based on information criterion in order to apply CE algorithm. Firstly, we need to derive the penalty term of the MDL based on the proposed model 2.16, the procedure is presented below:

1. **Decomposition of observed data $\{X_t\}$** . Decompose the data $\{X_t\}$ into two parts, the fitted model $\{\widehat{X}_t\}$ and the residuals model $\{\hat{\epsilon}_t\}$ conditional on the fitted model, in the expression of:

   $$CL(\{X_t\}) = CL(\{\widehat{X}_t\}) + CL(\{\hat{\epsilon}_t\}) \tag{2.17}$$

   The code length of the fitted model:

   $$CL(\{\widehat{X}_t\}) = CL(\{\hat{N}\}) + CL(\{\eta_1, \eta_2, \ldots, \eta_{N+1}\}) + CL(\{\hat{\rho}\}) + CL(\{\hat{\sigma^2}\}) \tag{2.18}$$

   $$CL(\{\widehat{X}_t\}) = CL(\{\hat{N}\}) + CL(\{\tau_1, \tau_2, \ldots, \tau_N\}) + CL(\{\hat{\rho}\}) + CL(\{\hat{\sigma^2}\}) \tag{2.19}$$

   Davis et al. [16] proposed that equation 2.18 and 2.19 are equal, since the locations of change-point are known, the length of each segment would also be completely known. So we developed two versions of the MDL penalty term based on 2.18 and 2.19 respectively. We refer them to MDL1 and MDL2.

2. **Calculate the total code length**. The MDL principle views the best model as the one that consumes less computer space with compressing relatively more information. In order to obtain the objective function, it is necessary to convert the computer language to statistical expressions. $CL(\{\hat{\epsilon}_t\})$ can be interpreted as $-2\mathcal{LL}(X \mid N, \tau, \rho, \delta, \sigma^2)$, the double negative maximum likelihood is given by the fitted model. In order to derive the expression of $CL(\{\widehat{X}_t\})$, we follow the two basic principles under real-valued parameters and integer parameters. For a real-valued parameter, if it is estimated from T observed data, it needs $\log_2 T/2$ bits memory; For an integer parameter, the code length is the space that its upper bound takes. The penalty term for each parameter is given as follows,

   (1) For the number N, the penalty terms is $\log(N)$. Since it can be considered as an integer parameter without being bounded, the code length is $\log_2(N)$.

   (2) For mean shift δ_j, the penalty term is $\log(\tau_{j+1} - \tau_j - 1)/2$, which is a real-valued parameter that needed to be estimated from piecewise $\tau_{j+1} - \tau_j - 1$, thus, it needs $\log_2(\tau_{j+1} - \tau_j - 1)/2$ computer space.

   (3) For the location τ_j, the penalty term is $\log(\tau_{j+1})$. Considering the integer parameter principle, because of $\log(\tau_j) < \log(\tau_{j+1})$, computing τ_j needs $\log_2(\tau_{j+1})$ bits memory.

   (4) For each segment η_j, the penalty term is $\log(T)$. Because maximum length for each segment can not beyond T, under the integer parameter principle, all η_j can be encoded with $\log_2(T)$ bits.

   (5) For model parameter ρ, the penalty term is $\log(T)/2$. As a real-valued parameter, it is estimated from whole datasets, so the code length is $\log_2(T)/2$.

   (6) For model parameter σ^2, the penalty term is $\log(T)/2$, so like ρ the code length is also $\log_2(T)/2$. .

After transforming the total code length to calculable expressions, we can get the objective functions based on AIC, BIC, MDL1 and MDL2 for the proposed model. $N + 3$ is the total parameters of model, δ is estimated from each segment, there are $N + 1$ segments in total, σ^2 and ρ are estimated from the whole datasets respectively.

$$\mathbf{F}_{AIC} = -2\mathcal{LL}(X \mid N, \tau, \rho, \delta, \sigma^2) + 2(N + 3), \tag{2.20}$$

$$\mathbf{F}_{BIC} = -2\mathcal{LL}(X \mid N, \tau, \rho, \delta, \sigma^2) + \log(T)(N + 3), \tag{2.21}$$

$$\begin{aligned}\mathbf{F}_{MDL1} &= -2\mathcal{LL}(X \mid N, \tau, \rho, \delta, \sigma^2) + 2(N+2)\log(T) + 2\log(N) \\ &+ \sum_{i=0}^{N} \log(\tau_{i+1} - \tau_i - 1). \qquad (2.22)\\ \mathbf{F}_{MDL2} &= -2\mathcal{LL}(X \mid N, \tau, \rho, \delta, \sigma^2) + 2\log(T) + 2\log(N) \\ &+ \sum_{i=0}^{N} \log(\tau_{i+1} - \tau_i - 1) + 2\sum_{i=0}^{N} \log(\tau_{i+1}). \qquad (2.23)\end{aligned}$$

2.4.3 Maximum Likelihood Framework

The next task is to obtain the MLE from proposed model. Even though our model is much simper than the segmented AR(p) process proposed by Davis et al. [16], the exact likelihood function is still complex, as seen below.

$$\begin{aligned}LL(X \mid N, \tau, \rho, \delta, \sigma^2) &= -\frac{N+1}{2}\ln\left(\frac{2\pi\sigma^2}{1-\rho^2}\right) - \sum_{i=0}^{N}\frac{\tau_{i+1}-\tau_i-1}{2}\ln(2\pi\sigma^2) \\ &- \sum_{i=0}^{N}\frac{(X_t - \delta_{i+1}/(1-\rho))^2}{2\sigma^2/(1-\rho^2)} - \sum_{i=0}^{N}\sum_{t=\tau_i+2}^{\tau_{i+1}}\frac{(X_t - \delta_{i+1} - \rho X_{t-1})^2}{2\sigma^2}.\end{aligned}$$

Since the number of change-points is unknown, the search space is quite large, applying CE method will cause extra computational burden. Therefore, we added one pruning procedure to the general CE algorithm in order to avoid computing the maximum likelihood function under the model with unknown multiple change-points. In algorithm 1, after finishing the step (5), we can acquire the estimates of change-point locations, this implies that the estimated mean $\hat{\delta}$ of each segment is given, by parallel shifting each segment, once the process return to a stationary state, we can easily calculate the maximum likelihood under the stationary AR(1) model by using R package, `arima`. In the next section, we explain this procedure in details.

2.4.4 The Cross-Entropy Algorithm for AR(1) Model

The CE method in Priyadarshana and Sofronov [38] and Priyadarshana and Sofronov [36] consider four-parameter beta distributions, whereas in this thesis, in order to detect the mean shifts, we use the normal distribution. The modified algorithm is summarized as:

Algorithm 2 The CE-AR(1) algorithm

1: Choose initial sets for $\mu^{(0)} = \left(\mu_1^{(0)}, \mu_2^{(0)}, \ldots, \mu_N^{(0)}, \mu_1^{(N+1)}, \mu_2^{(N+2)}, \ldots, \mu_N^{(2N+1)}\right)$ and $(\sigma^2)^{(0)} = \left((\sigma_1^2)^{(0)}, (\sigma_2^2)^{(0)}, \ldots, (\sigma_N^2)^{(0)}, (\sigma_{N+1}^2)^{(0)}, (\sigma_{N+2}^2)^{(0)}, \ldots, (\sigma_{2N+1}^2)^{(0)}\right)$. The length of both vectors is $2N+1$ with the first N components corresponding to N change-points and the last $N+1$ components corresponding to the vector δ. Set $k = 1$.

2: Generate a random sample $c^{(1)}, c^{(2)}, \ldots, c^{(M)}$ from the normal distributions with parameters $\left(\mu^{(k-1)}, (\sigma^2)^{(k-1)}\right)$, where $c^{(i)} = \left(c_1^{(i)}, c_2^{(i)}, \ldots, c_N^{(i)}, c_{N+1}^{(i)}, c_{N+2}^{(i)}, \ldots, c_{2N+1}^{(i)}\right)$, for $i = 1, 2, \ldots, M$, where the first N components $\left(c_1^{(i)}, c_2^{(i)}, \ldots, c_N^{(i)}\right)$ is a change-point vector.

3: For each $i = 1, 2, \ldots, M$, order $\left(c_1^{(i)}, c_2^{(i)}, \ldots, c_N^{(i)}\right)$ from smallest to biggest. After adjusting the mean levels in each segment, estimate the autocorrelation coefficient ρ using the entire data sequence.

4: Evaluate the objective function (either (2.20), (2.21) or (2.22), (2.23)) for each $c^{(1)}, c^{(2)}, \ldots, c^{(M)}$. Define the elite sample, which is the best performing combinations of the change-points. Let $M_{elite} = \beta M$ be the size of the elite sample.

5: For all $j = 1, 2, \ldots, 2N+1$, estimate the parameters $\mu_j^{(k)}$ and $(\sigma_j^2)^{(k)}$ using the elite sample and update the current parameter sets as follows:

$$\mu_j^{(k)} = \frac{\sum\limits_{i \in I} c_j^{(i)}}{M_{elite}}, \qquad \left(\sigma_j^2\right)^{(k)} = \frac{\sum\limits_{i \in I} \left(c_j^{(i)} - \mu_j^{(k)}\right)^2}{M_{elite}},$$

where I is the set of indices of the best performing samples.

6: Stopping criterion is $\max_j (\sigma_j^2)^{(k)} < \varepsilon$.

7: If the stopping criterion is met, then stop the process and identify the combination of the positions of change points and the values of mean levels for all segments that minimizes the objective function. Otherwise set $k = k + 1$ and iterate from step 2.

3

Numerical Results

In this section, we discuss two classes of numerical examples: artificial generated data and real data analysis. In the simulation study, we compare the performance of CE-AIC, CE-BIC, CE-MDL1, and CE-MDL2 against the best available methods implemented in R packages. Then we apply our method for a real data - United States inflation rate data from September of 2000 to December 2015, comparing with Bai & Perron's method.

3.1 Artificial Data Analysis

Following model (2.16), we generated 9 sets of 1000 AR(1) sequences with autocorrelation coefficient ranging from 0.1 to 0.9 and $\sigma = 1$ for each example. Example 1, deals with the single change-point problem, we generate a location at $\tau = 25$ and $\tau = 5$ respectively within 51 data, have the same mean shift equals 1. Then, in example 2 and 3, we introduced 2 and 3 abrupt change-points in 101 observations with positions $\tau = (\tau_1 = 30, \tau_2 = 60)$, mean shifts $\delta = (\delta_1 = 0, \delta_2 = -1, \delta_3 = 1)$ and $\tau = (\tau_1 = 10, \tau_2 = 30, \tau_3 = 60)$, and the mean levels

$\delta = (\delta_1 = 0, \delta_2 = -1, \delta_3 = 1, \delta_4 = 2)$.

We run the CE algorithm with the following simulation parameters: the elite proportion value $\beta = 0.05$, the sample size $M = 200$ and the stopping cut-off value $\epsilon = 0.01$. Let CE-AIC, CE-BIC, CE-MDL1 and CE-MDL2 denote the CE algorithms with the objective functions AIC, BIC and two version MDL, respectively. Since the limited space of the table, we use the lower case letter. All following tables compare the number of change-points obtained by the CE-AIC, CE-BIC, CE-MDL algorithms and the other best available methods that implemented in R package. It is important to note that we only consider the changes in the mean level, and limit the maximum number of change-points to five in all methods.

A list of all competing methods used is:

1. `changepoint` package [23], which implements the fast methods - BS method and the PELT method, we use the default BIC penalty.

2. `strucchange` implemented Bai & Perron's method for multiple change-points detection especially suitable for general regression problem.

3. `wbs` package implements the wild binary segmentation algorithm, we use sSIC penalty as model selection criterion.

4. `Segmentor3IsBack`, is based on a pruned dynamic programming algorithm developed by [39], which performs a fast exact segmentation on data and allows for use of various cost functions.

5. `AR1seg`, corresponds to the implementation of the robust approach for estimating change-points in the mean of an AR(1) Gaussian process by using the methodology described in the paper [12].

6. `breakpoint`, applies the CE method to multiple break-point detection mainly focus on independent sequences.

We will provide the summary of results for all methods in simulated data analysis at the end of this section, instead of describing the results separately.

3.1.1 Example 1: Simulated data with single change-point

Example 1.1: Simulated data with single change-point, the location is 25, mean shift equals 1

Table 3.1: Estimated number of single change-point for various competing methods over 1000 simulated sequences with location at $\tau_1 = 25$, mean shift $\delta_1 = 0, \delta_2 = 1$

Algorithm	ρ	$\hat{N}$					ρ	$\hat{N}$				
		0	1	2	3	$\geqslant 4$		0	1	2	3	$\geqslant 4$
BS-Bic		165	655	147	27	6		149	466	254	93	38
PELT-Bic		123	578	217	65	16		87	351	303	169	88
WBS-sSic		0	0	0	0	1000		0	0	1	0	999
B&P		0	745	205	43	7		0	541	285	130	39
Seg3IsB		23	164	323	85	405		55	199	280	108	358
bkpoint	0.3	19	529	328	94	30	0.5	0	318	386	193	103
AR1seg		558	264	93	44	41		544	217	107	72	60
CE-Aic		0	0	0	16	984		0	0	2	13	985
CE-Bic		0	31	62	122	784		0	28	64	118	790
CE-mdL1		0	916	70	9	5		0	875	93	19	13
CE-mdL2		0	780	130	39	51		0	729	154	52	65
BS-Bic		86	242	286	220	166		11	58	165	250	527
PELT-Bic		33	126	269	286	251		3	23	120	232	622
WBS-sSic		0	0	0	0	1000		0	0	0	0	1000
B&P		0	291	302	262	145		0	97	225	338	340
Seg3IsB		100	220	211	125	344		158	213	219	151	259
bkpoint	0.7	0	186	394	285	135	0.9	0	175	395	318	112
AR1seg		497	178	136	98	91		236	193	191	188	192
CE-Aic		0	0	2	6	992		0	0	1	16	983
CE-Bic		0	17	46	115	822		0	23	52	118	807
CE-mdL1		0	844	114	22	20		0	807	110	50	33
CE-mdL2		0	671	174	65	90		0	635	153	100	112

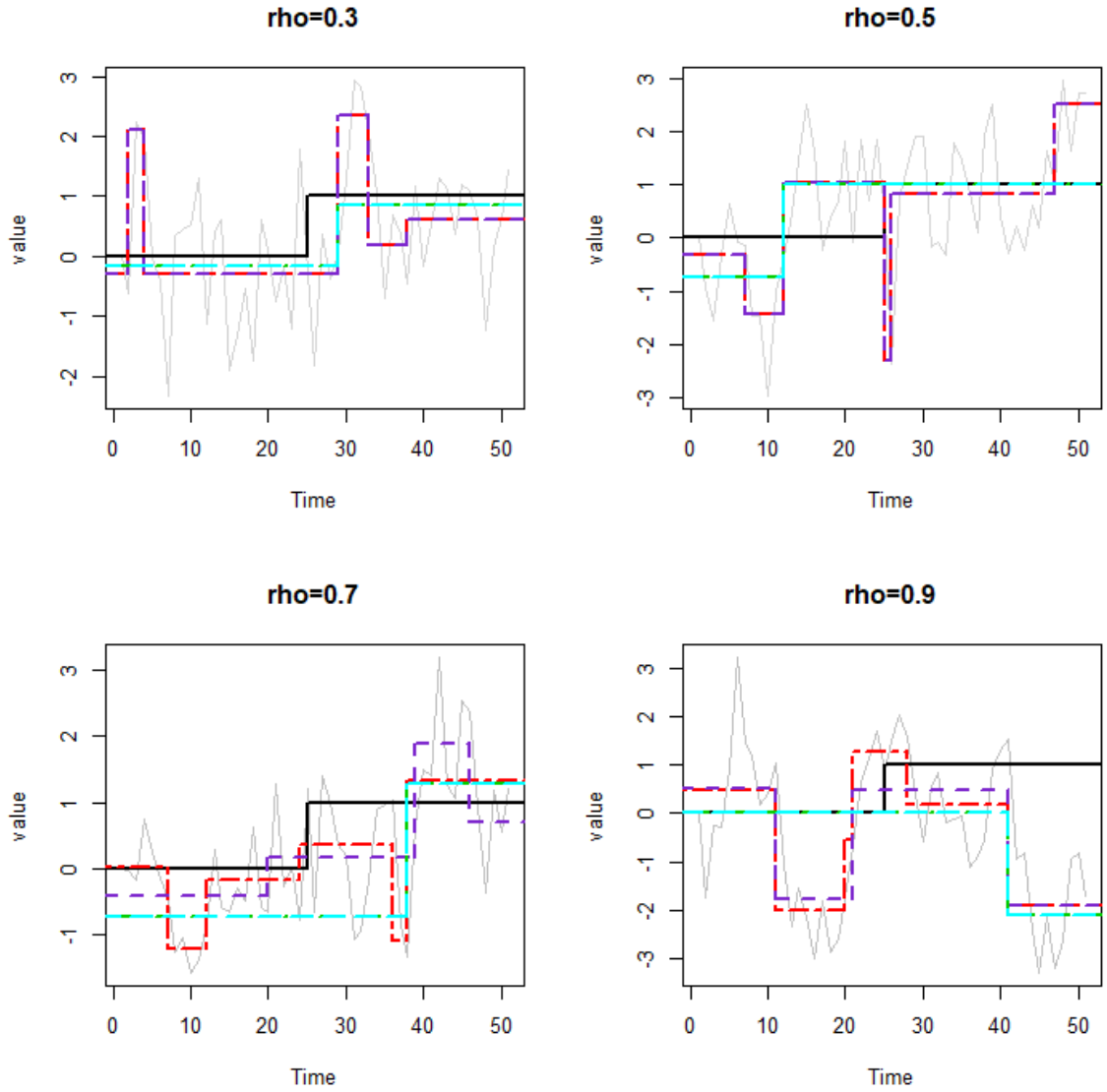

Figure 3.1: The true profile (black solid line) and the profile plots estimated by the CE-AIC algorithm (red twodash line), the CE-BIC algorithm (purple dashed line), the CE-MDL1 algorithm (green dotdash line) and CE-MDL2 algorithm (cyan longdash line)

Example 1.2: Single change-point, the location is 5, mean shift equals 1

Table 3.2: Estimated number of single change-point for various competing methods over 1000 simulated sequences with location at $\tau_1 = 5$, mean shift $\delta_1 = 0, \delta_2 = 1$

Algorithm	ρ	$\hat{N}$					ρ	$\hat{N}$				
		0	1	2	3	$\geqslant 4$		0	1	2	3	$\geqslant 4$
BS-Bic		397	477	102	20	4		289	354	224	100	33
PELT-Bic		314	415	175	73	21		175	259	283	178	97
WBS-sSic		0	1	0	0	999		0	01	0	1	998
B&P		0	782	147	63	8		0	562	254	147	33
Seg3IsB		9	342	299	58	292		16	351	268	94	271
bkpoint	0.3	65	371	323	197	44	0.5	25	192	381	273	129
AR1seg		694	154	79	39	34		637	148	98	66	51
CE-Aic		0	0	2	15	983		0	0	4	15	981
CE-Bic		0	42	54	129	775		0	21	66	112	801
CE-mdL1		0	904	69	17	10		0	862	102	23	13
CE-mdL2		0	773	127	44	56		0	710	156	64	70
BS-Bic		86	219	283	236	176		14	64	178	265	479
PELT-Bic		32	115	271	284	468		5	29	140	241	436
WBS-sSic		0	0	0	0	1000		0	0	0	0	1000
B&P		0	270	337	257	116		0	99	275	346	228
Seg3IsB		66	257	202	148	327		155	196	188	203	258
bkpoint	0.7	6	114	387	322	171	0.9	1	110	412	311	166
AR1seg		553	133	133	81	100		249	190	196	171	194
CE-Aic		0	0	1	8	991		0	0	1	17	982
CE-Bic		0	6	58	114	822		0	21	53	127	799
CE-mdL1		0	840	113	26	21		0	743	163	57	37
CE-mdL2		0	653	180	87	80		0	576	208	99	117

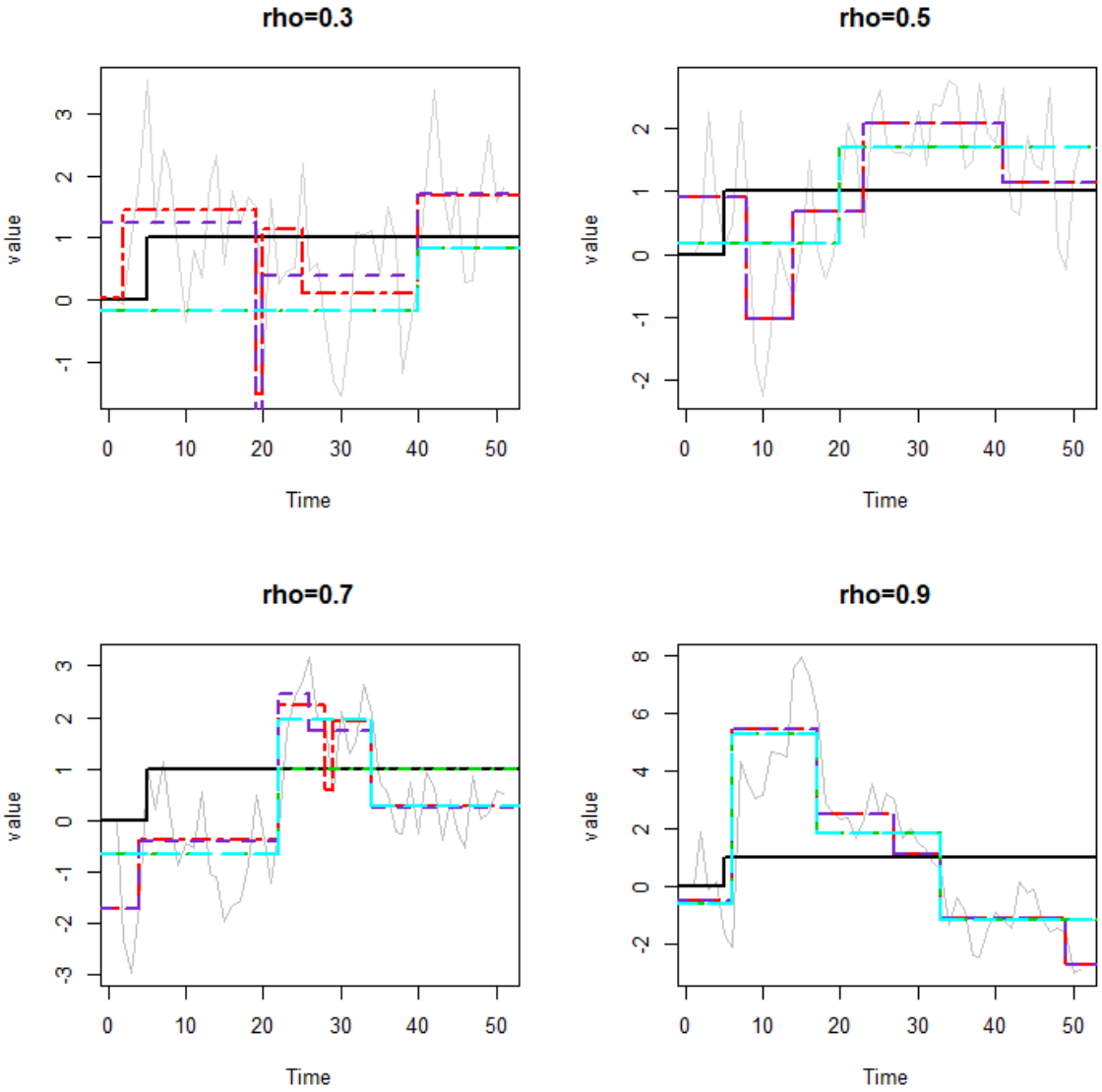

Figure 3.2: The true profile (black solid line) and the profile plots estimated by the CE-AIC algorithm (red twodash line), the CE-BIC algorithm (purple dashed line), the CE-MDL1 algorithm (green dotdash line) and CE-MDL2 algorithm (cyan longdash line)

3.1.2 Example 2: Simulated data with two change-points

Table 3.3: Estimated number of two change-points for various competing methods over 1000 simulated sequences with location at $\tau_1 = 30, \tau_2 = 60$, mean shift $\delta_1 = 0, \delta_2 = -1, \delta_3 = 1$

Algorithm	ρ	$\hat{N}$					ρ	$\hat{N}$				
		≤ 1	2	3	4	$\geqslant 5$		≤ 1	2	3	4	$\geqslant 5$
BS-Bic		113	605	226	44	12		120	293	301	169	117
PELT-Bic		67	530	260	111	26		40	208	286	240	226
WBS-sSic		103	471	181	134	111		70	206	187	151	386
B&P		149	733	116	2	0		184	536	248	28	0
Seg3IsB		252	246	135	367	0		272	212	122	394	0
bkpoint	0.3	61	631	223	68	17	0.5	58	347	316	177	102
AR1seg		572	293	77	58	0		647	177	107	69	0
CE-Aic		0	0	11	201	788		0	1	17	174	808
CE-Bic		11	219	180	278	312		20	141	188	264	387
CE-mdL1		631	344	21	4	0		761	209	23	5	2
CE-mdL2		431	506	41	16	6		600	320	50	21	9
BS-Bic		44	119	177	232	428		7	12	32	55	894
PELT-Bic		7	54	97	195	653		0	0	16	41	943
WBS-sSic		10	52	61	103	774		1	1	11	23	964
B&P		139	399	357	97	8		38	203	363	325	71
Seg3IsB		308	211	131	350	0		368	183	173	276	0
bkpoint	0.7	33	188	284	266	229	0.9	20	120	279	308	273
AR1seg		673	138	92	97	0		434	185	180	201	0
CE-Aic		0	1	5	161	833		1	2	22	206	769
CE-Bic		25	116	150	289	420		42	89	170	259	440
CE-mdL1		788	176	28	6	2		718	207	50	18	7
CE-mdL2		628	276	60	22	14		613	242	71	45	29

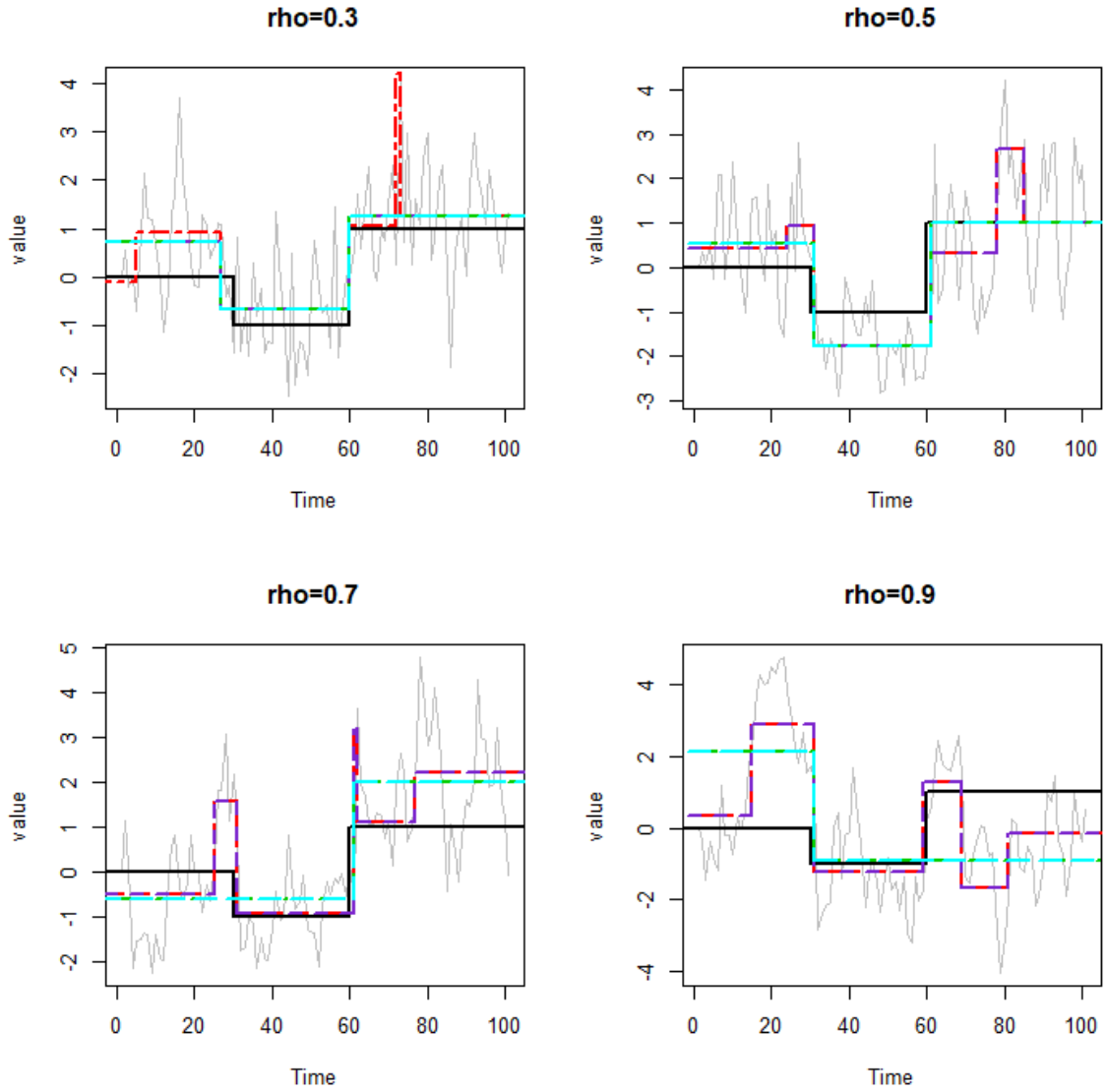

Figure 3.3: The true profile (black solid line) and the profile plots estimated by the CE-AIC algorithm (red twodash line), the CE-BIC algorithm (purple dashed line), the CE-MDL1 algorithm (green dotdash line) and CE-MDL2 algorithm (cyan longdash line)

3.1.3 Example 3: Simulated data with three change-points

Table 3.4: Estimated number of three change-points for various competing methods over 1000 simulated sequences with location at $\tau_1 = 10, \tau_2 = 30, \tau_3 = 60$, mean shift $\delta_1 = 0, \delta_2 = -1, \delta_3 = 1, \delta_4 = 2$

Algorithm	ρ	$\hat{N}$					ρ	$\hat{N}$				
		≤ 1	2	3	4	$\geqslant 5$		≤ 1	2	3	4	$\geqslant 5$
BS-Bic		113	325	385	138	39		103	234	261	228	274
PELT-Bic		74	254	418	173	64		50	121	266	272	291
WBS-sSic		114	247	294	160	185		84	113	172	164	467
B&P		180	576	225	18	0		220	428	286	61	5
Seg3IsB		2	408	48	542	0		3	455	52	490	0
bkpoint	0.3	130	447	350	62	11	0.5	126	311	359	153	51
AR1seg		610	161	148	81	0		712	115	89	84	0
CE-Aic		0	0	5	136	859		0	0	7	132	861
CE-Bic		20	55	145	309	471		29	57	130	284	500
CE-mdL1		774	157	60	7	2		896	57	36	10	1
CE-mdL2		653	181	115	38	13		773	85	82	49	11
BS-Bic		36	112	164	209	479		4	8	29	72	887
PELT-Bic		8	39	75	168	710		0	2	8	34	956
WBS-sSic		16	42	46	83	813		2	2	5	22	971
B&P		230	336	293	123	12		70	185	371	299	0
Seg3IsB		45	404	108	443	0		232	243	183	342	0
bkpoint	0.7	68	211	280	273	168	0.9	27	105	286	332	250
AR1seg		676	140	87	97	0		381	240	185	194	0
CE-Aic		0	1	12	148	839		0	2	18	183	797
CE-Bic		39	54	134	287	486		37	92	142	255	474
CE-mdL1		897	71	24	6	2		676	173	72	43	36
CE-mdL2		790	103	67	29	11		555	199	111	75	60

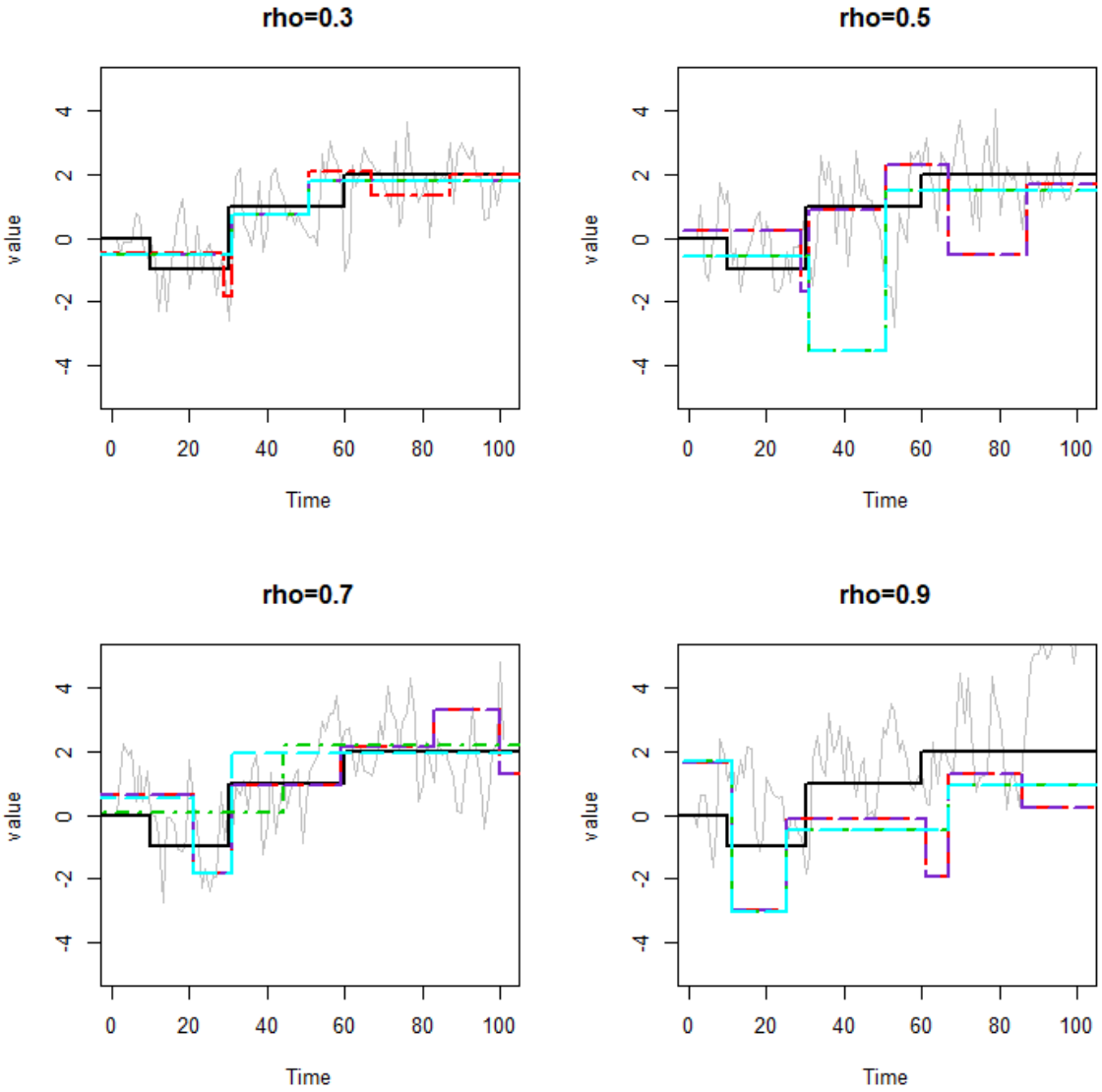

Figure 3.4: The true profile (black solid line) and the profile plots estimated by the CE-AIC algorithm (red twodash line), the CE-BIC algorithm (purple dashed line), the CE-MDL1 algorithm (green dotdash line) and CE-MDL2 algorithm (cyan longdash line)

3.1.4 Summary of Simulation Study

The tables provide the estimated number of change-points by applying extensive methods. We present several results below:

1. We use `changepoint` to implement BS method and the PELT method, both of them tend to heavily overestimate the change-points as the ρ get larger, especially when $\rho = 0.9$. The reason may be that BS method and the PELT method assume the observations are independently distributed.

2. `strucchange` implements Bai & Perron's method, which behave poorly as the auto-correlation gets stronger. In addition, it performs not well when the change occurred at 5.

3. `wbs` method with sSIC penalty clearly shows that it is not developed for detecting single change-points, the method performs well in multiple change-points situation.

4. `Segmentor3IsBack` does not perform well in the situation with three change-points. It displays a high computation efficiency.

5. `AR1seg` is based on the methodology described in the paper [12] which aims at detecting the mean shifts. It shows less accuracy in simulated data with three change-points comparing with the other examples.

6. `breakpoint` also has a tendency to overestimate the number of change-points, mainly because the method is designed for independent sequences.

We simulated the data sequence by using R package, `arima`. The mean of simulated data may largely diverge from theoretical mean, which may effect the accuracy of true profile. Therefore, it is necessary to consider the real data example.

3.2 Real Data Analysis

In this example, we analyse United States seasonally adjusted and annualized quarterly inflation rate data from September 2000 to December 2015. Figure 3.5 shows the estimated mean shifts obtained by the CE methods with AIC, BIC and MDL information criteria compared with Bai & Perron method that implemented in [42]. Since the actual number and the locations of change-points are unknown, we look for the agreement between the methods. The Bai & Perron method presents the 2 possible change-points in September 2003 and September 2007. The CE-AIC method estimates the 5 possible change-points in March 2004, March 2007, December 2007, September 2010 and March 2013, CE-BIC estimates 3 change-points occurred in March 2004, March 2006 and March 2008, while the CE-MDL1 and the CE-MDL2 methods present the different results, CE-MDL1 only estimates one change-point, while CE-MDL2 estimates March 2004 and March 2006. All methods have correctly identified the period from 2004 to 2007 with high annual inflation rate, in addition, the estimated segment mean of Bai & Perron method, CE-BIC and CE-MDL2 are very close.

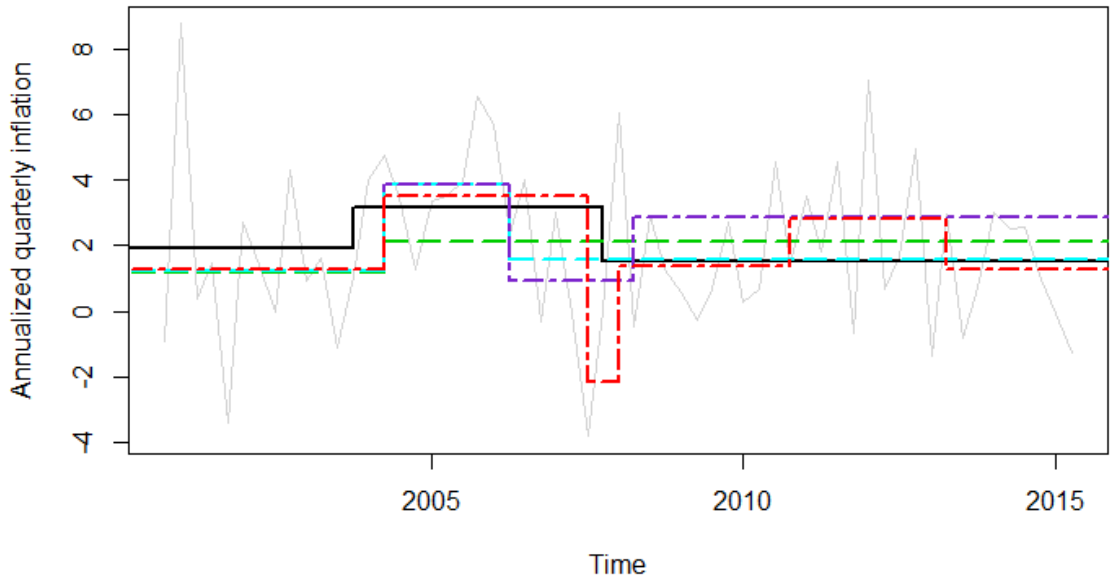

Figure 3.5: Annual inflation rate and profile plots of Bai & Perron method (black solid line) and the CE-AIC algorithm (red twodash line), the CE-BIC algorithm (purple dashed line), the CE-MDL1 algorithm (green dotdash line) and CE-MDL2 algorithm (cyan longdash line)

Data Information

The United States data are seasonally adjusted monthly and quarterly for the period March 1960 to June 2015. The United States national accounts data are from the National Income and Product Account (NIPA) tables from the United States of America, Bureau of Economic Analysis (BEA) and downloaded on 2 and 3 September 2015 except for Table 1.1.6 which was downloaded on 21 November. The data are available at www.BillRussell.info. Inflation defined as the log change in the price level (USqtlydata Database dlipfc)

4

Discussion and Future Directions

4.1 Discussion

In this thesis, we have proposed to use the cross-entropy method for identifying change-points in AR(1) sequence with shifts in the mean level. The important feature of the proposed method is that it allows us to estimate all parameters of the model such as the mean level at each segment, the autocorrelation for the whole data as well as the number and the locations of change-points. In order to estimate the correct number of change-points we use information criteria such as the AIC, the BIC and two versions of MDL. Comparing the performance of these information criteria attract less attention in change-point analysis. The numerical results in the thesis show that the standard information criteria, the AIC and the BIC have a strong tendency to overestimate the number of change-points mostly because both of them were designed for Gaussian independent data. Oppositely, the MDL criterion tends to underestimate the number of change-points. The data simulation process may be a factor to cause this problem, the sample means may heavily deviate from the theoretical means at

each segment, producing inaccurate estimates of the model parameters, which include the number and the positions of change-points. It is an inevitable problem of analysing artificially generated data. On the other hand, the underestimation problem of MDL may be due to the fact that we assume the autocorrelation coefficient is the same for all segments, which may contradict with the assumption that adjacent segments are independent. This suggests that the penalty function for the MDL may not accurate. Therefore, it is clear that we need to develop a new criterion or modify an existing one to address this problem, which can be considered as one of our future research.

4.2 Future Directions

Throughout the abundant literature on retrospective change-point detection problems, the common assumption is that univariate time series are either serially independent or short range dependent whereas fewer authors put emphasis on endogenous dependency of time series. In addition, for multivariate time series, it is usually assumed that observations are independent Gaussian. In this thesis, since we only focus on univariate time series, detecting change-points in multivariate data could be one of our future works.

The objective of our study is to develop an effective computational method to estimate unknown multiple change-points in the mean level of simple AR(1) model. We plan to develop new efficient methods to deal with the more complex models such as AR(p) model or the multiple change-point problems in discrete data. Moreover, we consider to develop new time series segmentation approach based on existing computational methods such as EM algorithm, Markov chain Monte Carlo and BS algorithms.

A

An Appendix

A.1 Algorithm 2: CE-AR(1)

```
#################################################################
library(MASS)
#chp.est is the vector of the segment length
all_fit=function(yy,chp.est,mu.est)
{
dd=length(chp.est)
mu_aux=NULL
for (ii in 1:dd)
{
mu_aux=c(mu_aux,rep(mu.est[ii],chp.est[ii]))
}
yy_aux=yy-mu_aux
```

```
ar.fit=
arima(yy_aux,order=c(1,0,0), method="ML",include.mean=FALSE,
optim.method = "Nelder-Mead")
}
#############################################################
T = 101
N_sim = 1000
D_max = 5

InfCrit1_all=array(0,dim=c(9,N_sim,6,D_max))
InfCrit2_all=array(0,dim=c(9,N_sim,6,D_max))
gamma_all1_all= array(0,dim=c(9,N_sim,1,D_max))
gamma_all2_all= array(0,dim=c(9,N_sim,1,D_max))
ChangePoints1_all= array(0,dim=c(9,N_sim,D_max,D_max))
ChangePoints2_all= array(0,dim=c(9,N_sim,D_max,D_max))
mu_est1_all =array(0,dim=c(9,N_sim,D_max,D_max+1))
mu_est2_all =array(0,dim=c(9,N_sim,D_max,D_max+1))

for (g in c(3,5,7,9))
{
yy <- read.table(paste("data_T",T,"_MS1","_gamma",g,".txt",sep=""),
sep=",",header = FALSE)
print("--------------------")
print("Gamma")
print(g)

for (nn in 1:N_sim)
{
y = as.numeric(yy[nn,])
print("Number Sim")
print(nn)
InfCrit1 = array(0,dim=c(6,D_max))
```

```
# 1 - LogLik
# 2 - AIC
# 3 - BIC
# 4 - mBIC
# 5 - MDL1
# 6 - MDL2
gamma_all1= array(0,dim=c(1,D_max))
ChangePoints1= array(0,dim=c(D_max,D_max))
mu_est1 =array(0,dim=c(D_max,D_max+1))

InfCrit2 = array(0,dim=c(6,D_max))
# 1 - LogLik
# 2 - AIC
# 3 - BIC
# 4 - mBIC
# 5 - MDL1
# 6 - MDL2
gamma_all2= array(0,dim=c(1,D_max))
ChangePoints2= array(0,dim=c(D_max,D_max))
mu_est2 =array(0,dim=c(D_max,D_max+1))
# Phase 1 #
N = 200
rho =0.05
Nelite =round(N*rho)
smooth=1

for (d in 1:D_max)
{
mu0=seq(from=T-(T-1)/(d+1),by=-(T-1)/(d+1),length.out=d)
sd0=rep((T-1)/(2*(d+1)),d)
t=1
mu_new=mu0
```

```
sd_new=sd0
mu_old=mu0
sd_old=sd0
#var_beta = (alpha*beta*(T-1)^2)/((alpha+beta)^2*(alpha+beta+1))
t_max=200
while ((max(sd_new)>0.1)&(t<=t_max))
{a=array(0,dim=c(N,d))
for (i in 1:d)
{
aa=rnorm(N,mean=mu_new[i],sd=sd_new[i])
b1=round(apply(a,1,sort))
if (d==1)
{
b1=t(as.matrix(b1))
}
if (d>1)
{
for (j in 1:N)
{
bb=b1[,j]
while (length(which(diff(bb)==0))>=1)
{
bb1=which(diff(bb)==0)
bb[bb1]=bb[bb1]+2*rbinom(length(bb1),1,0.5)-1
bb=sort(bb)
bb=ifelse(bb>(T-1),T-1,bb)
bb=ifelse(bb<2,2,bb)
}
b1[,j]=bb
}
}
```

```
b=array(0,dim=c(d+1,N))
b[1,]=b1[1,]
b[d+1,]=T-b1[d,]
if (d>1)
{
for (i in 2:d)
{
b[i,]=b1[i,]-b1[(i-1),]
}
}
a1=array(0,dim=c(d+1,N))
for (i in 1:(d+1))
{
if (i==1)
{
for (j in 1:N)
{
a1[1,j]=mean(y[1:b1[1,j]])
}
}
if ((i>1)&(i<(d+1)))
{
for (j in 1:N)
{
a1[i,j]=mean(y[(b1[(i-1),j]+1):b1[i,j]])
}
}
if (i==(d+1))
{
for (j in 1:N)
{
a1[(d+1),j]=mean(y[(b1[d,j]+1):T])
```

```
}
}
}
score=array(0,dim=c(1,N))
score_aux=array(0,dim=c(1,N))
score_sigma2=array(0,dim=c(1,N))
for (j in 1:N)
{
score[j]=all_fit(y,t(b[,j]),t(a1[,j]))$loglik
score_aux[,j]=all_fit(y,t(b[,j]),t(a1[,j]))$coef
score_sigma2[,j]=all_fit(y,t(b[,j]),t(a1[,j]))$sigma2
}
score_sorted=sort(score,decreasing=TRUE,index.return=TRUE)
#print(score_sorted$x)
elite_index=score_sorted$ix[1:Nelite]
elite_sample=b1[,elite_index]
#print(score_sorted$x[1])
if (d>1)
{
mu_new=apply(elite_sample,1,mean)
sd_new=apply(elite_sample,1,sd)
}
if (d==1)
{
mu_new=mean(elite_sample)
sd_new=sd(elite_sample)
}
mu_new=smooth*mu_new+(1-smooth)*mu_old
sd_new=smooth*sd_new+(1-smooth)*sd_old
mu_old=mu_new
sd_old=sd_new
#print(c(t,mu_new,sd_new))
```

```
t=t+1
}
estimate=mu_new
gamma_all1[d]=score_aux[,score_sorted$ix[1]]
LogLik=score_sorted$x[1]
Sigma2_est=score_sigma2[,score_sorted$ix[1]]
NumPar=d+3
# d+1 is the number of means, 1 is gamma, 1 is sigma^2
Pen_mBIC=sum(log(c(estimate[1],diff(estimate[1:d]),T-estimate[d])))
+(2*d-1)*log(T)
Pen_MDL=log(d)+(d+2)*log(T)
+sum(log(c(estimate[1],diff(estimate[1:d]),T-estimate[d])))*0.5
Pen_MDL1=log(d)+log(T)
+sum(log(c(estimate[1],diff(estimate[1:d]),T-estimate[d])))*0.5
+sum(log(estimate[1:d]))
#print(LogLik)

InfCrit1[1,d]=LogLik
InfCrit1[2,d]=-2*LogLik+2*NumPar # AIC
InfCrit1[3,d]=-2*LogLik+log(T)*NumPar # BIC
InfCrit1[4,d]=-2*LogLik+Pen_mBIC # mBIC
InfCrit1[5,d]=-2*LogLik+2*Pen_MDL # MDL1
InfCrit1[6,d]=-2*LogLik+2*Pen_MDL1 # MDL2

ChangePoints1[d,1:d]=estimate[1:d]
mu_est1[d,1:(d+1)]=a1[,score_sorted$ix[1]]

# Phase 2 #
N = 200
rho =0.05
Nelite =round(N*rho)
smooth=1
```

```
mu0=c(mu_new,a1[,score_sorted$ix[1]])
sd0=c(sd_new,rep(0.25,d+1))

t=1
mu_new=mu0
sd_new=sd0
mu_old=mu0
sd_old=sd0

t_max=200
while ((max(sd_new)>0.01)&(t<=t_max))
{
a=array(0,dim=c(N,d))
for (i in 1:d)
{
aa=ifelse(aa>(T-1),T-1,aa)
aa=ifelse(aa<2,2,aa)
a[,i]=aa
}
b1=round(apply(a,1,sort))
if (d==1)
{
b1=t(as.matrix(b1))
}
if (d>1)
{
for (j in 1:N)
{
bb=b1[,j]
while (length(which(diff(bb)==0))>=1)
{
bb1=which(diff(bb)==0)
```

```
bb[bb1]=bb[bb1]+2*rbinom(length(bb1),1,0.5)-1
bb=sort(bb)
bb=ifelse(bb>(T-1),T-1,bb)
bb=ifelse(bb<2,2,bb)
}
b1[,j]=bb
}
}
b=array(0,dim=c(d+1,N))
b[1,]=b1[1,]
b[d+1,]=T-b1[d,]
if (d>1)
{
for (i in 2:d)
{
b[i,]=b1[i,]-b1[(i-1),]
}
}
a1=array(0,dim=c(d+1,N))
for (i in 1:(d+1))
{
a1[i,]=rnorm(N,mean=mu_new[i+d],sd=sd_new[i+d])
}
score=array(0,dim=c(1,N))
score_aux=array(0,dim=c(1,N))
score_sigma2=array(0,dim=c(1,N))
for (j in 1:N)
{
score[j]=all_fit(y,t(b[,j]),t(a1[,j]))$loglik
score_aux[,j]=all_fit(y,t(b[,j]),t(a1[,j]))$coef
score_sigma2[,j]=all_fit(y,t(b[,j]),t(a1[,j]))$sigma2
}
```

```
score_sorted=sort(score,decreasing=TRUE,index.return=TRUE)
#print(score_sorted$x)
elite_index=score_sorted$ix[1:Nelite]
elite_sample=b1[,elite_index]
#print(score_sorted$x[1])
elite_sample1=a1[,elite_index]
if (d>1)
{
mu_new=apply(elite_sample,1,mean)
sd_new=apply(elite_sample,1,sd)
}
if (d==1)
{
mu_new=mean(elite_sample)
sd_new=sd(elite_sample)
}
mu_new1=apply(elite_sample1,1,mean)
sd_new1=apply(elite_sample1,1,sd)
mu_new=c(mu_new,mu_new1)
sd_new=c(sd_new,sd_new1)
mu_new=smooth*mu_new+(1-smooth)*mu_old
sd_new=smooth*sd_new+(1-smooth)*sd_old
mu_old=mu_new
sd_old=sd_new
t=t+1
}
estimate=mu_new
gamma_all2[d]=score_aux[,score_sorted$ix[1]]
LogLik=score_sorted$x[1]
Sigma2_est=score_sigma2[,score_sorted$ix[1]]
NumPar=d+3
# d+1 is the number of means, 1 is gamma, 1 is sigma^2
```

```
Pen_mBIC=sum(log(c(estimate[1],diff(estimate[1:d]),T-estimate[d])))
+(2*d-1)*log(T)
Pen_MDL=log(d)+(d+2)*log(T)
+sum(log(c(estimate[1],diff(estimate[1:d]),T-estimate[d])))*0.5
Pen_MDL1=log(d)+log(T)
+sum(log(c(estimate[1],diff(estimate[1:d]),T-estimate[d])))*0.5
+sum(log(estimate[1:d]))
#print(LogLik)
InfCrit2[1,d]=LogLik
InfCrit2[2,d]=-2*LogLik+2*NumPar # AIC
InfCrit2[3,d]=-2*LogLik+log(T)*NumPar # BIC
InfCrit2[4,d]=-2*LogLik+Pen_mBIC # mBIC
InfCrit2[5,d]=-2*LogLik+2*Pen_MDL # MDL1
InfCrit2[6,d]=-2*LogLik+2*Pen_MDL1 # MDL2
ChangePoints2[d,1:d]=estimate[1:d]
mu_est2[d,1:(d+1)]=estimate[-c(1:d)]
}

InfCrit1_all[g,nn,,]=InfCrit1
InfCrit2_all[g,nn,,]=InfCrit2
gamma_all1_all[g,nn,,]=gamma_all1
gamma_all2_all[g,nn,,]=gamma_all2
ChangePoints1_all[g,nn,,]=ChangePoints1
ChangePoints2_all[g,nn,,]=ChangePoints2
mu_est1_all[g,nn,,]=mu_est1
mu_est2_all[g,nn,,]=mu_est2
}
}
#############################################################
```

References

[1] Elena Andreou and Eric Ghysels. Structural breaks in financial time series. *Handbook of financial time series*, pages 839–870, 2009.

[2] Alexander Aue and Lajos Horváth. Structural breaks in time series. *Journal of Time Series Analysis*, 34(1):1–16, 2013.

[3] Ivan E Auger and Charles E Lawrence. Algorithms for the optimal identification of segment neighborhoods. *Bulletin of mathematical biology*, 51(1):39–54, 1989.

[4] Jushan Bai. Likelihood ratio tests for multiple structural changes. *Journal of Econometrics*, 91(2):299 – 323, 1999. ISSN 0304-4076. doi: https://doi.org/10.1016/S0304-4076(98)00079-7.

[5] Jushan Bai and Serena Ng. Evaluating latent and observed factors in macroeconomics and finance. *Journal of Econometrics*, 131(1):507–537, 2006.

[6] Jushan Bai and Pierre Perron. Estimating and testing linear models with multiple structural changes. *Econometrica*, pages 47–78, 1998.

[7] Jushan Bai and Pierre Perron. Computation and analysis of multiple structural change models. *Journal of applied econometrics*, 18(1):1–22, 2003.

[8] Naveen K Bansal, Hong Du, and G.G. Hamedani. An application of EM algorithm to a change-point problem. *Communications in Statistics-Theory and Methods*, 37(13): 2010–2021, 2008.

[9] Luc Bauwens, Arnaud Dufays, Bruno De Backer, et al. Estimating and forecasting structural breaks in financial time series. *Journal of Empirical Finance*, 2011.

[10] Claudie Beaulieu, Jie Chen, and Jorge L. Sarmiento. Change-point analysis as a tool to detect abrupt climate variations, philos. Technical report, TR Soc. A, 370, 1228–1249, 2012.

[11] B.E. Brodsky and B.S. Darkhovsky. *Non-Parametric Statistical Diagnosis: Problems and Methods*, volume 509. Springer Science & Business Media, 2000.

[12] Souhil Chakar, E Lebarbier, Céline Lévy-Leduc, and Stéphane Robin. A robust approach for estimating change-points in the mean of an AR(1) process. *Bernoulli*, 23(2):1408–1447, 2017.

[13] Jie Chen and Arjun K Gupta. *Parametric statistical change point analysis: with applications to genetics, medicine, and finance*. Springer Science & Business Media, 2011.

[14] Haeran Cho and Piotr Fryzlewicz. Multiscale and multilevel technique for consistent segmentation of nonstationary time series. *Statistica Sinica*, pages 207–229, 2012.

[15] Richard A Davis, Dawei Huang, and Yi-Ching Yao. Testing for a change in the parameter values and order of an autoregressive model. *The Annals of Statistics*, pages 282–304, 1995.

[16] Richard A Davis, Thomas C. M Lee, and Gabriel A Rodriguez-Yam. Structural break estimation for nonstationary time series models. *Journal of the American Statistical Association*, 101(473):223–239, 2006.

[17] Benjamin Doerr, Paul Fischer, Astrid Hilbert, and Carsten Witt. Detecting structural breaks in time series via genetic algorithms. *Soft Computing*, pages 1–14, 2016. doi: 10.1007/s00500-016-2079-0.

[18] Gareth E Evans, George Y Sofronov, Jonathan M Keith, and Dirk P Kroese. Estimating change-points in biological sequences via the cross-entropy method. *Annals of Operations Research*, 189(1):155–165, 2011.

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

[20] Claudia Furlan. Extreme value methods for modelling historical series of large volcanic magnitudes. *Statistical Modelling*, 10(2):113–132, 2010.

[21] Mark H Hansen and Bin Yu. Model selection and the principle of minimum description length. *Journal of the American Statistical Association*, 96(454):746–774, 2001.

[22] Brad Jackson, Jeffrey D Scargle, David Barnes, Sundararajan Arabhi, Alina Alt, Peter Gioumousis, Elyus Gwin, Paungkaew Sangtrakulcharoen, Linda Tan, and Tun Tao Tsai. An algorithm for optimal partitioning of data on an interval. *IEEE Signal Processing Letters*, 12(2):105–108, 2005.

[23] Rebecca Killick and Idris Eckley. changepoint: An R package for changepoint analysis. *Journal of Statistical Software, Articles*, 58(3):1–19, 2014. ISSN 1548-7660. doi: 10.18637/jss.v058.i03.

[24] Rebecca Killick, Paul Fearnhead, and Idris A Eckley. Optimal detection of changepoints with a linear computational cost. *Journal of the American Statistical Association*, 107 (500):1590–1598, 2012.

[25] Rebecca Killick, IA Eckley, Philip Jonathan, et al. A wavelet-based approach for detecting changes in second order structure within nonstationary time series. *Electronic Journal of Statistics*, 7:1167–1183, 2013.

[26] Karolos K Korkas and Piotr Fryzlewicz. Multiple change-point detection for non-stationary time series using wild binary segmentation. *Statistica Sinica*, 27(1):287–311, 2017.

[27] Estelle Kuhn and Marc Lavielle. Coupling a stochastic approximation version of EM with an MCMC procedure. *ESAIM: Probability and Statistics*, 8:115–131, 2004.

[28] Shanghong Li and Robert Lund. Multiple changepoint detection via genetic algorithms. *Journal of Climate*, 25(2):674–686, 2012.

[29] Qiqi Lu, Robert Lund, and Thomas C. M Lee. An MDL approach to the climate segmentation problem. *The Annals of Applied Statistics*, 4(1):299–319, 2010.

[30] David S Matteson and Nicholas A James. A nonparametric approach for multiple change point analysis of multivariate data. *Journal of the American Statistical Association*, 109 (505):334–345, 2014.

[31] Stuart Mead and Christina Magill. Determining change points in data completeness for the holocene eruption record. *Bulletin of Volcanology*, 76(11):874, 2014.

[32] Adam B Olshen, E.S. Venkatraman, Robert Lucito, and Michael Wigler. Circular binary segmentation for the analysis of array-based DNA copy number data. *Biostatistics*, 5 (4):557–572, 2004.

[33] ES Page. A test for a change in a parameter occurring at an unknown point. *Biometrika*, 42(3/4):523–527, 1955.

[34] Ewan S Page. Continuous inspection schemes. *Biometrika*, 41(1/2):100–115, 1954.

[35] Philip Preuss, Ruprecht Puchstein, and Holger Dette. Detection of multiple structural breaks in multivariate time series. *Journal of the American Statistical Association*, 110 (510):654–668, 2015.

[36] M. Priyadarshana and G.Y. Sofronov. A modified cross-entropy method for detecting change-points in the Sri-Lankan stock market. In *Proceedings of the IASTED International Conference on Engineering and Applied Science, EAS 2012*, pages 319–326, 2012. doi: 10.2316/P.2012.785-041.

[37] W.J.R.M Priyadarshana. *The cross-entropy method and multiple change-point detection in genomic sequences*. PhD thesis, Macquarie University, 2015.

[38] W.J.R.M Priyadarshana and Georgy Sofronov. Multiple break-points detection in array CGH data via the cross-entropy method. *IEEE/ACM Transactions on Computational Biology and Bioinformatics (TCBB)*, 12(2):487–498, 2015.

[39] Guillem Rigaill. A pruned dynamic programming algorithm to recover the best segmentations with 1 to K_{max} change-points. *ArXiv e-prints*, April 2010.

[40] Gordon J Ross. Sequential change detection in the presence of unknown parameters. *Statistics and Computing*, 24(6):1017–1030, 2014.

[41] Reuven Y Rubinstein and Dirk P Kroese. *The cross-entropy method: a unified approach to combinatorial optimization, Monte-Carlo simulation and machine learning*. Springer Science & Business Media, 2004.

[42] Bill Russell. ARCH and structural breaks in United States. *Applied Economics Letters*, 21(14):973–978, 2014.

[43] A. J. Scott and M. Knott. A cluster analysis method for grouping means in the analysis of variance. *Biometrics*, pages 507–512, 1974.

[44] Ashish Sen and Muni S Srivastava. On tests for detecting change in mean. *The Annals of statistics*, pages 98–108, 1975.

[45] Jaco Weideman, Roula Inglesi-Lotz, and J Van Heerden. Structural breaks in renewable energy in South Africa: A Bai & Perron break test application. *Renewable and Sustainable Energy Reviews*, 78:945–954, 2017.

[46] Chun Yip Yau and Zifeng Zhao. Inference for multiple change points in time series via likelihood ratio scan statistics. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 78(4):895–916, 2016.

[47] Nancy R Zhang and David O Siegmund. A modified Bayes information criterion with applications to the analysis of comparative genomic hybridization data. *Biometrics*, 63 (1):22–32, 2007.