Binary Segmentation Methods for Identifying Boundaries of Spatial Domains

By

Nishanthi Raveendran (44703376)

A thesis submitted to Macquarie University
for the degree of Master of Research
Department of Statistics
April 2017

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.

Nishanthi Raveendran (44703376)

Acknowledgements

I would sincerely like to express enormous thanks to my supervisor, Dr. Georgy Sofronov for taking me on as one of his students. I greatly appreciate his continuous guidance, support, motivation and generosity of time throughout the past year. I also owe a huge gratitude to Associate Professor Jun Ma for introducing me to Georgy, when I was searching for a supervisor. Also, I want to express my gratitude to my husband who gave me so much love and support. I would also like to acknowledge Macquarie University for providing me financial support through the International Macquarie University Research Excellence Scholarship (iMQRES). I really enjoyed every day I spent at statistic department.

Conference Participations and Publications

- Nishanthi Raveendran and Georgy Sofronov. *Binary segmentation methods for identifying boundaries of spatial domains.* (A full paper is in preparation to be submitted for the 10^{th} International workshop on Computational Optimization (WCO'17) which is scheduled to be held in Prague, Czech Republic from September 3 to September 6, 2017).

- Nishanthi Raveendran and Georgy Sofronov. *Identifying boundaries of domains in spatial binary data.* (Abstract is submitted for the International Conference on Robust Statistics(ICORS 2017). The conference is scheduled to be held at The University of Wollongong from July 3 to July 7, 2017).

- Nishanthi Raveendran and Georgy Sofronov. *Spatial clustering via binary segmentation.* (Poster presentation at AustMS2016. The conference was held at Australian National University, Canberra from December 5 to December 8, 2016).

- Travel grant to attend BioInfoSummer, from the Australian Mathematical Sciences Institute (AMSI), at The University of Adelaide from November 28 to December 2, 2016.

Abstract

Spatial clustering is an important component of spatial data analysis. The aim is to identify the boundaries of domains and their number. It is commonly used in disease surveillance, spatial epidemiology, population genetics, landscape ecology, crime analysis and many other fields. We focus on identifying homogeneous sub-regions in an ecology data set. We use binary data indicating the presence or absence of a certain plant species which are observed over a two-dimensional lattice. The problem of finding regional homogeneous domains is known as segmentation, partitioning or clustering. To solve this problem we propose to use change-point methodology. We develop new methods based on a binary segmentation algorithm which is a well-known multiple change-point detection method. The proposed algorithms are applied to artificially generated data to illustrate their usefulness. Our results show that the proposed methodologies are effective in identifying multiple domains and their boundaries in two dimensional spatial data.

Contents

List of Figures

List of Tables

1

Introduction

1.1 Background and Aims

Spatial statistics is one of the central topics in statistics. It is often the case that spatial data have pre-defined subdivisions of interest. For example, data is often collected on non-overlapping administrative or census districts and these districts are often irregular in shape; see, for example, Yang and Swartz [46]. As a part of modelling the spatial distribution, spatial clustering is also an important component of spatial data analysis. It is commonly used in disease surveillance, spatial epidemiology, population genetics, landscape ecology, crime analysis and many other fields. In general, the spatial data may be heterogeneous and difficult to understand. However, if we cluster the data into homogeneous clusters or domains, then it will be easier to construct appropriate statistical models for each cluster. The problem of finding regional homogeneous domains is known as segmentation, partitioning or

clustering.

There are two main problems in spatial clustering:

- Identifying the number of domains which will not be known in advance.
- Estimating the boundaries of such domains.

Many clustering algorithms have been developed in the literature, ranging from hierarchical methods such as bottom-up (or agglomerative) methods and top-down (or divisive) methods, to optimization methods such as the k-means algorithm [9]. The algorithms have numerous applications in pattern recognition, spatial data analysis, image processing and market research; see [40]. Spatial clustering covers enormous practical problems in many disciplines. For example, in epidemiological studies and public health research, it is known that the disease risk varies across the space and it is important to identify regions of safety and regions of risk. Methods based on hypothesis are commonly used for estimating disease rates or cluster risks. Gangnon and Clayton [17] presented a model for spatial clustering and used Bayesian approach to inference the parameters. Recently, Anderson *et al.* [2] proposed a two-stage Bayesian approach for estimating the spatial pattern in disease risk and identifying clusters which have high (or low) disease risks.

The homogeneity changes in space is an important research subject in ecology. In a large area, the spatial distribution of plant or animal species is never homogeneous. Studying these kinds of changes is important in several ways. For example, detecting early changes in vegetation improves productivity. Beckage *et al.* [4] introduced a class of Bayesian statistical models to identify thresholds and their locations in ecological data. Lopaz *et al.* [28] presented a method for estimating the distribution change-point between two patches of plants. Another important application for spatial clustering is weather forecasting. Investigations of weather and climatic systems at a global scale have become a prime area of research for a number of reasons among which the concern about global climatic change is a main one. Tripathi and Govindaraju [39] used Mann-Kendall trend test, Bayesian change point analysis and a hidden Markov model to find changes in the rainfall and temperature patterns over India. Nicholls and Nunn [29] proposed a Bayesian statistical framework for fitting

a spatio-temporal change point model for settlement and growth at Bourewa, Fiji Island. There has also been extensive literature on image recognition with some articles presenting statistical approaches to the boundary identification in statistical imaging. For example, Helterbrand *et al.* [19] presented a Markov chain Monte Carlo (MCMC) method to identify closed object boundaries in gray-scale images. Another application is to classify objects and background scenes in high spatial and low spectral resolution aerial images using circular growth technique. Change curve estimation problem is also referred as multidimensional detection problem or boundary estimation problem. Wang [44] proposed a wavelet method to estimate jumps and sharp curves in the plane. Even though there is a wide range of applications to spatial clustering, many statistical methods for detecting clusters have some limitations: either they detect the number of clusters and do not determine their locations, or they provide the inference with no clustering.

In this thesis, we are interested in identifying the boundaries of domains and their number with applications to an ecological landscape. In general, these problems are typically challenging due to the multivariate nature of the data which leads to complex and highly parameterized likelihoods. We use binary data indicating the presence or absence of plant species, which are observed over a two-dimensional lattice. Binary spatial data are commonly involved in various areas such as economics, social sciences, ecology, image analysis and epidemiology. Also, such data frequently occur in environmental and ecological research, for instance, when the data correspond to presence or absence of a certain invasive plant species at a location or, when the data happen to fall into one of two categories, say, two types of soil.

We approach our problem as change-point detection which is commonly used in statistics to detect changes and their locations. We develop a binary segmentation algorithm which is a well-known recursive partitioning tool in change-point literature and it leads to simple solutions for such problems. A binary segmentation procedure begins with searching a change-point for entire data. If a change-point is detected, the data are then split into two subsegments. The same procedure is then performed on both subsegments. The recursion on a given segment continues until a certain criterion is satisfied on it. In essence, the method

extends any single change-point method to multiple change-points by iteratively repeating the method on different subsets of the data. In spatial data, the change-point locations are the points which is used to draw a horizontal or vertical line to divide the domain into two homogeneous segments. In this thesis, we propose a binary segmentation approach which has an advantage on simplicity and less computational cost compare to other methods. Our aim is to develop statistical computational methods for identifying the number of spatial domains and their boundaries. The general overview of spatial data can be found in [12, 13, 41]. In the next section we shall discuss the change-point problem and its methods in detail.

1.2 Change-point Problem

The change-point problem is defined as the problem of detecting abrupt changes in the parameter(s) of interest at unknown times and estimating their corresponding positions in stochastic processes. There could be a single change or there could be many changes. Most of the statistical models assume that the parameter(s) of interest are at the same for the entire data. But, in the real applications, this assumption is violated for several reasons such as changes in mean, variance, amplitude or a combination of changes. Thus, it is important to be aware of these changes so that they are incorporated in a model. In statistics, the change-point is a location or time point such that the observations follow a particular distribution up to that point and follow a different distribution after that point. Therefore, the change-points (or break-points) divide the data into piece-wise homogeneous segments with respect to the parameter(s) of interest. Thus, it is important to identify such change-points when carrying out statistical analysis. Figure 1.1 illustrates four types of structural changes in a time series data [14].

The development of inference methods for change-point problems is by no means a recent phenomenon and it includes many early works (see, for example, [20, 31, 37]). Increasingly the ability to identify change-points accurately and quickly has become of interest to a broad range of applications. For example, in finance, change-point techniques are used to detect changes in the volatility of time series. In the last few decades the change-point problem has

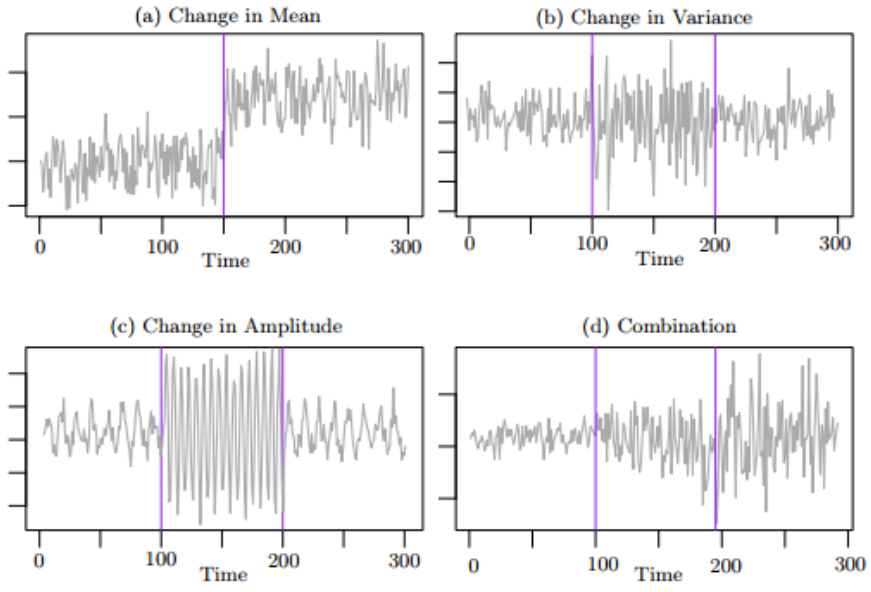

Figure 1.1: Four types of structural change.

received increasing attention in contemporary applications, which include bioinformatics, malware detection, network traffic analysis, finance, geology, climatology and oceanography [14]. The general overview of change-point problem can be found in [8, 14].

In the literature, there exists a number of methodologies to solve change-point problems. The methods are mainly based on likelihood ratio, non-parametric or Bayesian statistics. The change-point problems fall into two main categories: posterior (off-line or retrospective) class of change-point problems and prospective (on-line or sequential) class of change-point problems. The posterior change-point problems consider the entire data sequence which is already collected for inferences. However, in the prospective change-point problems, the future observations are not known and researches are interested in identifying the first time of a change in data which get updated continually; these problems are typically presented in statistical quality control, public health surveillance, and signal processing. In general, these problems consist focus on the detection of change-points in stochastic processes and the estimation of their corresponding locations. Change-point problems can also be formulated as a model selection problem. In this thesis, we are primarily interested in addressing the

posteriori multiple change-point problem with a special interest in detecting abrupt changes in spatial binary data.

1.3 Change-point Methods for Segmentation

There is a number of different models and methodologies for change-point problem: posterior and prospective, parametric and non-parametric, frequentist and Bayesian approaches. In this section, we discuss some of the main techniques used in change-point literature. Next, we shall provide a detailed review on change-point methods related to our work.

1.3.1 Sliding Window Analysis

Sliding window analysis is one of the techniques used in change-point literature to detect abrupt changes. It is a commonly used method for analysing the properties of biological sequences such as detecting variation in DNA copy number, identifying the changes in nucleotide counts and etc. Sliding window analysis begins with the choice of a window size which is a critical step in performing the analysis. This method provides two options by which to choose the window size: the user may assign a fixed window size or allow to search for the most informative window size using the computer program. The main step of this method is getting the average of the observations with respect to a pre-determined size of sliding window. For example, if the window size is 10, we obtain the first point by taking the average of observations 1-10, the second point by taking the average of observations 2-11, and so on, until the end of the data sequence. If the length of the observed data sequence is L and the window size is W, then the length of the average points sequence is $L - W + 1$. However, the choice of an optimal window size has been an open research question for more than two decades [33].

1.3.2 Dynamic Programming Methods

Dynamic programming (DP) is a general optimization method which provides a framework to solve a complex problem with the use of multistage optimization approach. Simply, it

transforms or splits the complex problem into a pool of simpler problems to obtain an optimal solution. In change-point literature, there are many variants of the DP methods developed to detect changes. Many multiple change-point problem methods are either fast and heuristic or exact and slow. The dynamic programming methods provide exact solutions but slow. The computational time for the DP is quadratic complexity which may not be acceptable for very large data sets. Recently, there are much works on developing pruned dynamic programming methods that are fast, exact and having linear complexity for computational time. For example, a modified DP algorithm called Pruned Exact Linear Time (PELT) [26] was proposed with linear computational cost under some conditions.

1.3.3 Evolutionary Algorithms

Evolutionary algorithms (EAs) are mainly designed to solve optimization-related search problems using evolutionary computation techniques. Recently, EAs has been used in variety of applications mainly due to the exponential growth in technologies. Multiple change-point problem may be considered as a mixture of optimization and estimation problems. Thus, naturally it is possible to utilize EA techniques in change-point problems. Many variants of EAs are proposed in the literature. The genetic algorithm is one of the EAs algorithms and it is studied by several authors in the change-point literature. Recently another EA procedure, called, Cross Entropy (CE) method which is a model-based stochastic optimization technique as an exact search method used to estimate both the number and locations of the break-points has been considered with applications in bioinformatics and economics. However, there is an issue on estimating the number of change-points [33].

1.3.4 Recursive Segmentation Methods

Recursive segmentation methods are commonly used in change-point literature. For example, binary segmentation, circular binary segmentation and wild binary segmentation are some popular recursive search methods. We shall discuss these methods in the next sections in detail since we propose to use recursive methods for our problem. These segmentation methods initially find the most significant change-point based on a statistical test and then it

is kept in the memory. The data set is split into two homogeneous domains according to the identified change-point. Then the same procedure is carried out until no more statistically significant change-points are found. These recursive methods are easy to implement and save lots of computational cost. Thus, these methods have obtained a significant attention in the change-point literature.

1.4 Posterior Class of Change-point Problems

The posteriori change-point problems have been of interest in different applications for many decades. Statistical inference about posterior (off-line) change-points has two aspects: firstly, to detect if there is any change in the sequence of observed random variables; secondly, to estimate the number of change-points and their corresponding locations. We shall define the single change-point and then multiple change-point problem as a posteriori change-point problem. This is due to the scope of this thesis, where in the context of analysing spatial data, we observed the entire data set already. Thus, the change-point analysis of spatial data can be primarily attributed to the posterior class of the change-point problem.

1.4.1 Single Change-point Problem

A single change-point model assumes only one change-point in the process. The general formulation of the single change-point problem can be given as follows. Let $y_n = (y_1, \ldots, y_n)$ be a sequence of observations of length n; $y_1, y_2, \ldots, y_n$ are independent random variables with the probability distribution functions $F_1, F_2, \ldots, F_n$. Let us assume there exists a change-point τ in the observed sequence. In general, the single change-point problem involves testing the following null hypothesis,

$$H_0 : F_1 = F_2 = \cdots = F_n$$

versus the alternatives

$$H_1 : F_1 = F_2 = \cdots = F_\tau \neq F_{\tau+1} = \cdots = F_n.$$

If the distribution $F_1, F_2 \ldots, F_n$ belong to a common parametric distribution family $\mathcal{F}(\theta)$, then the single change-point problem can be considered as the hypothesis testing about the population parameters $\theta_i, i = 1, \ldots, n$:

$$H_0 : \theta_1 = \theta_2 = \cdots = \theta_n = \theta \,(\text{unknown})$$

versus the alternatives

$$H_1 : \theta_1 = \theta_2 = \cdots = \theta_\tau \neq \theta_{\tau+1} = \cdots = \theta_n.$$

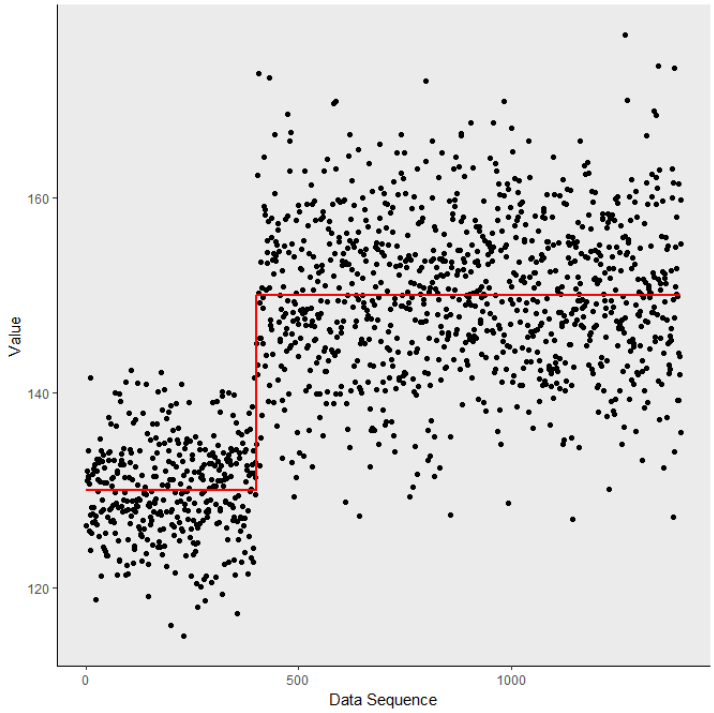

Figure 1.2: Single change-point problem.

Figure 1.2 shows an example of a single change-point process with a change-point in the mean of normally distributed random variables.

Likelihood Ratio Approach

There are several methods to make the statstical inference on the single change-point. It appears natural to consider the single change-point problem within the likelihood-based

framework. In this thesis, we propose to use the likelihood test for the single change-point problem. Thus, the hypothesis testing is performed by defining the null (H_0) and the alternative (H_1) hypotheses for a change as follows:

$$H_0\text{: No change-point} \qquad \text{vs} \qquad H_1\text{: A single change-point.}$$

The likelihood-ratio based method was first introduced by Hinkley in 1970 [20] for testing hypothesis about the change-point. He derived the asymptotic distribution of the likelihood ratio statistic for detecting a change in the mean of a normal distribution. Hinkley and Hinkley [21] have also considered inferences based on the maximum likelihood about the single change-point in a sequence of binomial variables. Later, this method has been extended to detect changes in mean for other distributions including gamma (see Hau [22]), exponential (see Haccou *et al.* [18]). Gupta and Tang [24] and Chen and Gupta [7] also proposed this approach to detect changes in the variance of normally distributed observations.

Let us consider a sequence of observations $y_1, \ldots, y_n$ of length n. A change-point $\tau \in \{1, \ldots, n-1\}$ divides the sequence into two segments such that the statistical properties of $(y_1, \ldots, y_\tau)$ and $(y_{\tau+1}, \ldots, y_n)$ differ.

To perform the likelihood ratio test, first we need to calculate the maximum log-likelihood values under both null and alternative hypothesis. Under the null hypothesis, the maximum log-likelihood is $\log p(y_{1:n}|\hat{\theta})$, where $\hat{\theta}$ is the maximum likelihood estimator of the parameter. Under the alternative hypothesis, the maximum log-likelihood for a given change-point τ (the profile log-likelihood for τ) is,

$$P(\tau) = \log p(y_{1:\tau}|\hat{\theta_1}) + \log p(y_{(\tau+1):n}|\hat{\theta_2}).$$

The maximum log-likelihood value under the alternative hypothesis is,

$$\max_{\tau} P(\tau).$$

This results in the test statistic:

$$\lambda(y_{1:n}) = 2[\max_{\tau} P(\tau) - \log(y_{1:n}|\hat{\theta})].$$

We then choose a threshold β such that we reject the null hypothesis if $\lambda > \beta$. In this case, a change-point is detected and its position $\hat{\tau}$ is estimated by the profile log-likelihood for the τ. The likelihood test statistic can be extended to multiple change-problem by simply summing likelihood for each of the segments.

Penalised Likelihood Approach

A penalised likelihood approach has also been used in the change-point literature. It is similar to the likelihood ratio statistic approach and can naturally be extended to the multiple change-point problem. It compares the maximum log-likelihoods of two models given in the hypothesis test for the single change-point problem and detects a change-point if the test statistic is greater than some threshold. The main difference between these two approaches is in the way how the thresholds are calculated. The general penalised likelihood for a model M_k with p_k parameters is given by

$$PL(M_k) = -2\log L_{\max}(\Theta_k) + p_k\phi(n),$$

where $L_{\max}(\Theta_k)$ is the maximum likelihood and $\phi(n)$ is a penalisation function, which is an non-decreasing function of the data of length n. There are different penalty functions discussed in the literature. It is important to select an appropriate penalty function corresponding to the model. Some popular penalty functions are given below.

- AIC: $\phi(n) = 2$
- BIC: $\phi(n) = \log n$
- Hannan-Quinn: $\phi(n) = 2\log\log n$

The penalty functions from the BIC and the Hannan-Quinn information crterion are commonly preferred as these methods asymptotically estimate the correct number of parameters rather than the AIC [14]. The model with the lowest $PL(M_k)$ is selected as the best model.

1.4.2 Multiple Change-point Problem

Whilst an approach for detecting a single change-point is very useful, in practice the assumption of only one change within the data may be unrealistic. It is likely that multiple changes exist within the data, particularly, when the amount of data collected is increasing. The multiple change-point problem can be seen as a natural extension of the single change-point problem. The problem can be formulated in the following way.

Let $y_n = (y_1, \ldots, y_n)$ be a sequence of observations of length n, $y_1, y_2, \ldots, y_n$ be independent random variables with the probability distribution functions $F_1, F_2, \ldots, F_n$. Let $\tau_1, \tau_2, \ldots, \tau_m$ be unknown positions of m change-points, where $\tau_1 < \tau_2 < \cdots < \tau_m$. We define $\tau_0 = 0$ and $\tau_{m+1} = n$. The sequence of observations is divided into $m + 1$ segments based on m change-points. In general, the multiple change-point problem involves testing the following null hypothesis,

$$H_0 : F_1 = F_2 = \cdots = F_n$$

versus the alternatives:

$$H_1 : F_1 = \cdots = F_{\tau_1} \neq F_{\tau_1+1} = \cdots = F_{\tau_2} \neq F_{\tau_2+1} = \cdots = F_{\tau_m} \neq F_{\tau_m+1} = \cdots = F_n.$$

Figure 1.3 shows an example of a multiple change-point process. It illustrates multiple change-points in the mean of normally distributed random variables.

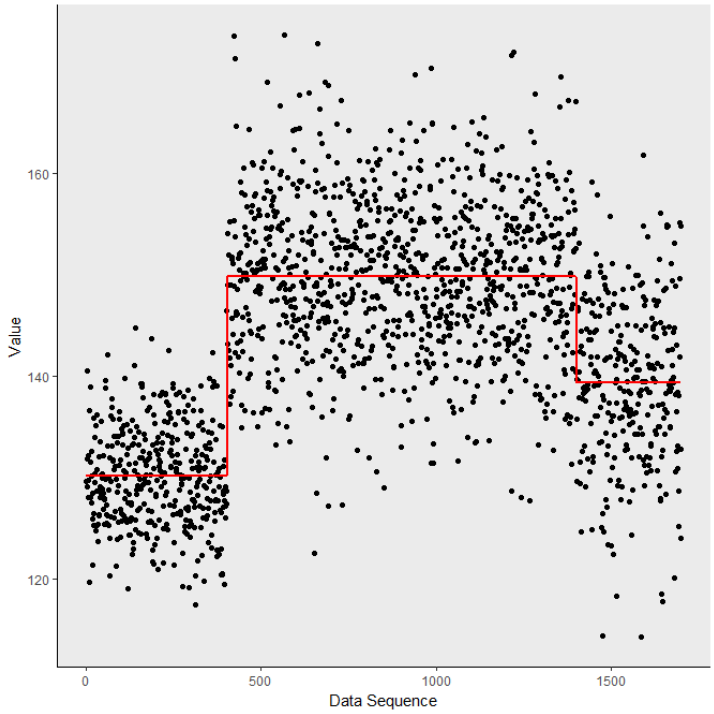

Figure 1.3: Multiple change-point problem.

Extension of Single Change-point Problem

Let us consider m change-points that split the data into $m+1$ segments with the i-th segment containing $y_{(\tau_{i-1}+1):\tau_i}$. Each segment i is summarized by parameters θ_i, ϕ_i where θ_i is a set of parameters that may contain changes and ϕ_i is a set of nuisance parameters [14].

The likelihood ratio test statistic can be extended to multiple change-point detection by summing the likelihood for the m data segments. However, testing all possible change-points locations is hardly practicable for large n as the number of possible partitions is 2^{n-1}. For example, with 1,500 data points there are 1,499 possible position of a single change-point, but $3 \cdot 10^{35}$ sets of possibilities for 15 change-points. Thus, the analysis of multiple change-point models is computationally much more challenging. Due to this, change-point detection search algorithms have been developed.

One way to detect multiple change-points is to minimize

$$\sum_{i=1}^{m+1}[C(y_{(\tau_{i-1}+1):\tau_i})] + \beta f(m), \tag{1.1}$$

where C is a cost function for a segment and $\beta f(m)$ is a penalty term in order to avoid over-fitting. The C could be negative log-likelihood which is commonly used in the literature [26].

There is more than one change-point algorithm that minimizes equation (1.1). Some of these methods minimize it exactly. For example, optimal partitioning which is an exact method and commonly used in change-point problems. The main drawback of these algorithms is higher computational cost. On the other hand, there are some approximation methods which are usually computationally quicker. It is clear that in many situations the number of change-points increases as we collect more data and the computational burden increases as well. Therefore, many authors are working on developing new algorithms which are fast and exact. In this thesis, we focus on binary segmentation, which is one of the approximation methods widely used in many scenarios. It can be seen as an approach to minimize (1.1) by iteratively deciding whether a change-point should be added or not. Binary segmentation is computationally efficient with $O(n \log n)$ where n is the number of data points. In the next section, we review and summarize both exact and approximation algorithms existing in the literature.

1.4.3 Exact Methods for Change-point Problem

Segment Neighbourhood Search

A segment neighbourhood search algorithm is one of the exact search algorithms for the change-point problem. It was introduced by Braun and Muller [5]. It is also called as a global segmentation algorithm. The pivotal step of this algorithm is defining some measure of data fit, say, $R(\cdot)$. The segment neighbourhood algorithm uses a dynamic programming technique to obtain best segmentation of the data into $m+1$ segments for $m = 0, \ldots, M-1$, M is the maximum number of segments corresponding to at most $M-1$ change-points.

The best partition is achieved by minimizing the cost function

$$\sum_{i=0}^{m} R(y_{(\tau_i+1):\tau_{i+1}})$$

for a partition with change-points at locations $\tau_1, \tau_2, \ldots, \tau_m$. The segment neighbourhood search algorithm originally follows a dynamic programming approach described by Auger and Lawrence [3]. The main disadvantage of this procedure comes due to its computational cost. It has an $O(n^2)$ computation complexity which is far from $O(n)$ for binary segmentation.

Optimal Partitioning

Optimal partioning is one of the exact search methods used in the change-point literature. It was first introduced by Yao [47]. Recently, Jackson *et al.* [23] proposed an optimal partitioning method to minimize equation (1.1) with linear penalty term, i.e., $f(m) = m$. This method solves the minimization problem using a dynamic programming technique. This algorithm performs by optimizing at each time step using the optimal solution from all previous steps. The cost of the partition is obtained by the cost of the optimal partition prior to the last change-point plus the cost for the segment from the last change-point to the end of the data. The following optimization is solved at each time step, t.

$$F(t) = \min_{\tau^*}[F(\tau^*) + C(y_{(\tau^*+1):t}) + \beta]. \tag{1.2}$$

By setting $F(0) = 0$ and selecting an appropriate value for β, $F(t)$ can be calculated for all values of t. Then the optimal cost $F(n)$ is obtained by storing the values of τ which minimize the above equation (1.2). This allows to identify the change-point. It is noted that optimal partitioning improves the computational efficacy of the segment neighbourhood search method. But, it is still far from being competitive computationally with binary segmentation. To make this algorithm more efficient, the best way is pruning the set of candidates. Thus, several authors have recently been interested in developing pruned dynamic programming methods, which are fast and exact.

Pruned Exact Linear Time (PELT)

This exact search method was introduced by Killick *et al.* in 2012 [26] to improve the computational cost and accuracy properties whilst still ensuring that the method finds a global minimum of the cost function (1.1). The computational efficacy is gained by removing unwanted solution paths which are known not to lead to the optimal solution. The theoreoms, proofs and some assumptions for removing the solution paths are clearly defined in [26]. The PELT algorithm is $O(n)$ under certain assumptions such as the number of true change-points being linear with the data length. Still this method has $O(n^2)$ complexity at the worst case.

1.4.4 Approximate Methods for Change-point Problems

Binary segmentation

Binary segmentation (BS) is a popular fast algorithm used within the change-point literature. We propose to use this approach in our work. It starts with applying the single change-point approach to the whole data set. If we detect a change-point τ, it satisfies,

$$C(y_{1:\tau}) + C(y_{(\tau+1):n}) + \beta < C(y_{1:n}). \tag{1.3}$$

If (1.3) is not satisfied, no change-point is detected and the algorithm stops. The full information on this algorithm including pseudo-code and examples are given in Chapter 2. The binary segmentation method can be seen as an approach with $f(m) = m$ to minimize (1.1) iteratively deciding whether a change-point should be added or not. BS is simple to code and computationally efficient with an $O(n \log n)$ computational cost even though it is an approximate method.

Circular Binary Segmentation

Circular binary segmentation (CBS) is a modification of the binary segmentation algorithm. It was introduced by Olshen *et al.* [30] to detect DNA copy number changes in the mean change-point model. It follows the same basic structure as binary segmentation. However, at each iteration it allows to identify one change-point or two change-points.

A special multiple change-point problem is the epidemic change-point problem which is defined by testing the following null hypothesis

$$H_0 : \theta_1 = \theta_2 = \cdots = \theta_n = \theta \,(\text{unknown})$$

versus the alternatives

$$H_1 : \theta_1 = \cdots = \theta_k = \alpha \neq \theta_{k+1} = \cdots = \theta_t = \beta \neq \theta_{t+1} = \cdots = \theta_n = \alpha,$$

where $1 \leq k < t \leq n$, and α and β are unknown. The epidemic change-point problem is of a great practical interest, especially in quality control and medical studies. The epidemic alternative means that we have two change-points which split the data into three segments, and the first and last segments follow the same distribution. For example, a temporal structural change happens in time series data and then the previous structure is resumed. The same situation happens in the DNA copy number data.

Circular binary segmentation performs hypothesis testing at each iteration. It is given by:

$$H_0 : \text{No change-point}$$

versus the alternatives

$$H_1 : \text{Single change-point or epidemic alternative.}$$

Let $y_n = (y_1, \ldots, y_n)$ be a sequence of observations of length n and assume we have two change-points τ_1, τ_2 for parameter θ. It joins the two ends to form a circle,

$$\theta_1 = \cdots = \theta_{\tau_1} = \theta_{\tau_2+1} = \cdots = \theta_n \quad \text{and} \quad \theta_{\tau_1+1} = \cdots = \theta_{\tau_2},$$

for $1 < \tau_1 < \tau_2 \leq n$.

It then tests the hypothesis that the arc from $\tau_1 + 1$ to τ_2, for $1 < \tau_1 < \tau_2 \leq n$, and its complement have different means. A single change-point can be detected when $\tau_2 = n$. CBS fist performs hypothesis testing for the entire data set and continues the same procedure as

binary segmentation.

Under the alternative hypothesis, the maximum log-likelihood for two change-point τ_1, τ_2 where $1 < \tau_1 < \tau_2 \leq n$ is given by

$$P(\tau_1, \tau_2) = \log(y_{E(\tau_1,\tau_2)}|\hat{\theta}_1) + \log(y_{(\tau+1):\tau_2}|\hat{\theta}_2),$$

where $E(\tau_1, \tau_2)$ is the set $E(\tau_1, \tau_2) = (1, \ldots, \tau_1) \cup (\tau_2 + 1, \ldots, n)$. Hence, the maximum log-likelihood under H_1 is

$$\max_{1<\tau_1<\tau_2\leq n} P(\tau_1, \tau_2).$$

The test statistic is

$$\lambda_{E(y_{1:n})} = 2[\max_{1<\tau_1<\tau_2\leq n} P(\tau_1, \tau_2) - \log(y_{1:n}|\hat{\theta})].$$

We choose a threshold β such that we reject the null hypothesis if $\lambda_{E(y_{1:n})} > \beta$ and the position of the change-points is given by,

$$(\hat{\tau}_1, \hat{\tau}_2) = \arg \max_{1<\tau_1<\tau_2\leq n} P(\tau_1, \tau_2).$$

For the application of the CBS method to the analysis of change-point problems see Olshen *et al.* [30] and Venkatraman and Olshen [30]. In this thesis, we are unable to use the CBS method since epidemic alternative is impossible, we cannot consider our spatial data in the form of a circle (or a cylinder). There is no guarantee that the first and the last segments follow the same distribution. Thus, we propose another algorithm which is suitable to our data and we shall describe it in the next chapter.

Wild binary segmentation

A wild binary segmentation (WBS) procedure proposed by Fryzlewicz in 2014 [15] is another modification of the binary segmentation algorithm. It is developed to estimate the number and corresponding locations of multiple change-points. It follows the same concept as binary segmentation. In contrast to BS, which is commonly performed using the CUSUM

(Cumulative Sum), WBS avoids using a global CUSUM statistic, which is based on the entire data $y_1, y_2 \dots y_n$. Wild binary segmentation draws a number of subsamples, say, vectors $Y_s, Y_{s+1}, \dots, Y_t$, where s, t are integers such that $1 \leq s < t \leq n$. The CUSUM statistic is computed for each subsample. Then the largest statistic among all CUSUMs is selected, which is considered as the first change-point candidate. Finally, the obtained change-point is tested against a certain threshold value. If it is significant, the same procedure is then carried out recursively to the right and left parts of it. Two stopping criteria based on threshold and strengthened Schwarz information criteria for WBS are proposed. The wild binary segmentation procedure is suggested as simple, easy to code, consistent and computationally fast method.

1.4.5 Thesis Structure

The structure of this thesis is as follows. Chapter 1 includes an introduction to the study. Chapter 2 describes the methodology followed in this study. First, it begins with an introduction to the general binary segmentation and its applications in the change-point literature. Then, it moves to describe the proposed method to spatial data and introduces the model we focus on. Further, it describes the algorithms in detail and provides an example to illustrate the effectiveness of the proposed method. In the next sections, model selection criteria, stopping rules and the likelihood ratio test are provided in detail. Chapter 3 gives all numerical results. Chapter 4, where discussion and future directions are provided, concludes the thesis.

2

Methodology

2.1 Introduction

Detection of optimal homogeneous sub-regions is one of the challenging issues in ecology. The most difficult part is identifying their boundaries. We begin with binary data indicating the presence and absence of particular type of plant which are observed over a two dimensional lattice, in an ecological landscape see [38]. To identify such domains and their boundaries, new algorithms based on binary segmentation method are introduced. In this section of the thesis, we provide detailed information on both general binary segmentation method and new algorithms. We further discuss model selection criteria used for determining the number of domains which is one of the challenges in this study. We use the maximum likelihood ratio test to prove the homogeneity of the obtained domains. We give examples to illustrate the usefulness of our methods.

Figure 2.1: Alchemilla Gracilis.

Figure 2.2: Study region.

Figure 2.1 and Figure 2.2 represent a particular type of plant *Alchemilla Gracilis* and its study region, respectively. At this stage we only consider artificial data sets.

2.2 The General Binary Segmentation Method

The binary segmentation is a well-known multiple change-point method and has been studied by various authors. It was first introduced by Scott and Knott in the context of cluster analysis [35]. Sen and Srivastava [36] proposed the concept of binary segmentation in detecting changes in mean . Later, Vostrikova [43] extended this procedure to detect the number of change-points in a multidimensional random process and proved the consistency of the estimates produced by binary segmentation under mild conditions, the first of which is based on the minimal distance between change-points . Similar results when the change-points are allowed to approach one another are achieved by Venkatraman [42].

Recent studies include many applications. Chen and Gupta [7] proposed methods based on binary segmentation to divide the normal data into homogeneous segments. Subsequently, Braun and Muller [5] used this procedure for locating change-points in DNA sequence segmentation, Yang and Kuo [48] and Yang [45] proposed Bayesian binary segmentation for homogeneous Poisson process and sporting performance. This method is also proposed for estimating spatial density by Yang and Swartz [46] but circular growth cluster technique

is used instead of segmenting the areas. Recently, Killick *et al.* [25] developed binary segmentation for detecting changes within an oceanographic time series. Fryzleicz and Subba Rao [16] and Cho and Fryzlewicz [10] proposed binary segmentation procedure for univariate time series segmentation; for multivariate high dimensional time series segmentation refer to [11]. Thus, this method is now the most understood and widely cited search algorithm used within the multiple change-point literature. The generic binary segmentation algorithm is given below [14].

Algorithm 1 The Generic Binary Segmentation Algorithm

Input: A set of data of the form, $(y_1, y_2, \ldots, y_n)$.

A test statistic $\lambda(\cdot)$, which depends on the data.

An estimator of change-point position $\hat{\tau}(\cdot)$.

A rejection threshold β.

Initialise: Let $C = \emptyset$, and $S = [1, n]$.

Iterate: while $S \neq \emptyset$

1. Choose an element of S; denote this element as $[s, t]$.
2. If $\lambda(y_{s:t}) < \beta$, remove $[s, t]$.
3. If $\lambda(y_{s:t}) \geq \beta$:
   (a) remove $[s, t]$ from S;
   (b) calculate $r = \hat{\tau}(y_{s:t}) + s - 1$, and add r to c;
   (c) if $r \neq s$ add $[s, r]$ to S;
   (d) if $r \neq t - 1$ add $[r + 1, t]$ to S.

Output: The set of change-points recorded C.

The above algorithm defines the general work-flow of the binary segmentation method in estimating the locations of the change-points with respect to the number of change-points.

The generic binary segmentation algorithm is given in Algorithm 1. It mainly concludes a general test statistic $\lambda(\cdot)$ depending on the data such as the likelihood ratio statistic, an estimator of the change-point position $\hat{\tau}(\cdot)$ and rejection threshold β. In this code, S denotes a set of segments of the data to be tested for the change-points and C denotes a set of detected change-points. A change-point is detected in the segment $y_{s:t}$, if $\lambda(y_{s:t}) > \beta$. And the corresponding estimate of its position is $\hat{\tau}(y_{s:t})$. At each iteration, this algorithm selects one segment from S. If there is no change-point within this particular segment, it is removed from S. Otherwise, the detected change-point is added to C and the original segment is split into two separate segments for detecting more change-points and replaced in S. The position of the change-point in the original data, r, is calculated from $\hat{\tau}(y_{s:t})$, the position of the change-point in the segment $[s, t]$. We add a new segment to S if the segment contains at least 2 observations. Otherwise, it is assumed that there are no more change-points in this segment.

A modified algorithm of binary segmentation called circular binary segmentation (CBS) is introduced by Olshen *et al.* in 2004 [30] and it is now used as one of the gold standards in biological sequences especially in micro-array data for locating change-points [33].

Binary segmentation can be used to extend any single change-point method to multiple change-points. As we discussed in Chapter 1, there are number of methodologies available to solve the single change-point problem using frequentist as well as Bayesian approaches. In the early works, binary segmentation was performed using a simple CUSUM test. It starts with applying the chosen single change-point detection method to the entire data set say $y_{1:n}$, sequence of observations of length n. If no change-point is found, then the algorithm stops. If a change-point is detected, say τ, then the data set is split into two separate segments, $y_{1:\tau}$ and $y_{\tau+1:n}$. The single change-point method is applied to two segments and the procedure is repeated iteratively. Finally, we stop when no more change-points are detected.

The binary segmentation has the merits of detecting the number of change-points and their locations simultaneously. It is a fast algorithm and saving a lots of computational time and it can be implemented with the computational complexity $O(n)$.

2.3 Binary Segmentation for Spatial Data

We develop new algorithms based on binary segmentation for spatial binary data. We approach single the change-point problem within the maximum likelihood framework. This procedure involves a sequence of nested hypothesis tests of a single domain versus a pair of distinct domains. Under the null hypothesis, single domain implies that the data within region come from a common density. For the alternative hypothesis we split the data into two homogeneous domains and assume distinct densities for each.

2.3.1 Model and Notation

Assume we have independent binary observations on an $n \times m$ lattice. For simplicity we assume the observations at each cell is uni-variate. We consider only one particular type of plant or animal species though extensions to multivariate data are straightforward. A generalized model for multiple species is discussed in Chapter 4. Our model has a number of domains, M, and $p = (p_1, \ldots, p_M)$ are the parameters of Bernoulli distribution for the domains.

1	1	1	1	0	1	1	0	1	1	1	1
1	1	1	1	1	0	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	0	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1
1	0	1	1	0	0	1	0	1	1	1	1
1	1	1	1	1	1	0	1	1	1	1	1
0	0	0	0	0	0	0	0	0	0	0	0
0	0	1	0	0	1	1	0	0	0	0	0
1	0	0	0	1	0	0	1	0	0	0	1

FIGURE 2.3: Binary lattice data.

The likelihood function is given by:

$$L(X,p) = \prod_{j=1}^{M} p_j^{I_{D_j}} (1-p_j)^{O_{D_j}}, \quad j = 1, 2, \ldots, M,$$

X is the data (a matrix of zeroes and ones),

M is the number of domains,

D_j is the j-th domain,

$p = (p_1, \ldots, p_M)$ is the vector of probabilities,

I_{D_j} is the number of ones in D_j,

O_{D_j} is the number of zeroes in D_j.

We maximize the log-likelihood function

$$\log(X,p) = \sum_{j=1}^{M} I_{D_j} \log p_j + O_{D_j} \log(1-p_j).$$

2.3.2 Maximum Likelihood Framework

Let X is an $n \times m$ matrix. A natural approach to split a domain into homogeneous sub-domains is to view it as performing a hypothesis test.

$$H_0 : \text{No sub-domain}, M = 1; \quad \text{Vs} \quad H_1 : \text{Two domains}, M = 2.$$

Under the null hypothesis, the log-likelihood function for the entire domain is given as

$$\log(X|\hat{p}).$$

Under the H_1, the log-likelihood function (given a change-point c, which divides the whole domain into two homogeneous domains say D_1 and D_2) is,

$$P(c) = \log(D_1|\hat{p_1}) + \log(D_2|\hat{p_2}),$$

where $\hat{p}_1$ and $\hat{p}_2$ are the maximum log-likelihood estimates of the parameters for the first and the second domain respectively. To estimate the location of the change-point, the log-likelihood function under H_1 is maximized.

The test statistic:

$$\lambda(X) = 2[\max_{c} P(c) - \log(X|\hat{p})],$$

where a threshold β is chosen such that if $\lambda(X) > \beta$, the null hypothesis is rejected.

The threshold could be based on the use of an information criterion: AIC $\beta = 2p$ and SIC $\beta = p \log n$, where p is the number of extra parameters as a result of adding another domain.

2.3.3 Algorithm 2 (Main Algorithm)

Our proposed algorithms use the maximum likelihood test as described in the previous section. It searches every column and row to detect the change-point and selects the maximum test statistic for the optimal cut. If the test statistic greater than the threshold value, it splits the domain according to the index (row or column) and stores the obtained domains. Otherwise, the algorithm stops. This procedure is repeated a stopping criterion is met. At this stage, we only consider rectangle shaped domains. We also propose two more algorithms with modifications. In general, our method can be summarized by a three-step iterative procedure.

Step 1: Given the data, search the change point column-wise and find the optimal cut which maximizes the test statistic. Repeat this procedure row-wise.

Step 2: Select the maximum of the two test statistics for the optimal column and row cuts and compare with the threshold value. If the test statistic is greater than the threshold value, then split the data in two domains.

Step 3: Repeat steps 1 and 2 for each domain until no new domains are identified.

2.3.4 Model Selection

Our objective is to estimate both the number of domains and their boundaries. Thus, it can be formulated as a model selection problem. Usually model selection is done by using

a specific criterion. In the literature, there are several popular model selection criteria that have been proposed in different contexts. The model selection criteria is mainly used for two different purposes: the first is to choose a model that well approximates the true model and the second is to find the true model in a list of candidate models [27]. In this thesis, we use the later approach. The Akaike's Information Criterion (AIC) [1] and the Bayesian Information Criterion (BIC) [34] are well-known criteria used for model selection. There are also modified AIC (mAIC) and modified BIC (mBIC) used in particular cases when the general AIC and BIC do not work well because of irregularities in the likelihood function. Recently, Pan and Chen [32] proposed a new information criterion named the modified information criterion (MIC) for studying change-point models. Further, Zhang and Siegmund [49] proposed a modified BIC (mBIC) for identifying change-points in a comparative genomic hybridization data.

The AIC and the BIC for our model is described as below:

$$\mathrm{AIC}(k) = -2\log L(\hat{\Theta}_k) + 2k, \quad k = 1, 2, \ldots, M,$$

where $L(\hat{\Theta}_k)$ is the maximum likelihood for a model with k parameters, as a measure of model evaluation. A model that minimizes the AIC (minimum AIC estimate) is considered to be the most appropriate model.

The BIC can be expressed as:

$$\mathrm{BIC}(k) = -2\log L(\hat{\Theta}_k) + k\log n, \quad k = 1, 2, \ldots, M,$$

where k is the number of parameters and n is the sample size. The difference between the AIC and the BIC is in the penalty term: instead of $2k$, it is $k\log n$. The BIC gives an asymptotically consistent estimate of the number of parameters of the true model. The BIC has been applied to change-point analysis for different underlying models by many authors in the literature [8].

According to [6] a version of the modified BIC for the change-point problems can be expressed as:

$$\mathrm{BIC}(k) = -2 \log L(\hat{\Theta}_k) + 2(k+1) \log n, \quad k = 1, 2, \ldots, M,$$

2.3.5 Stopping Criteria

In the binary segmentation one has to define a stopping criterion to terminate the iterative procedure. We use one of two methods, which can be described as follows.

1. The algorithm is reiterated while we have significant cuts based on the results of a hypothesis testing. Let us define that number of cuts $c = C$, the process is stopped and the corresponding solution is considered as the optimal solution for the problem.

2. The decision to stop the algorithm is based on an information criterion.

2.3.6 Motivating Example

We generate a (10×12) matrix with three domains having the parameters of Bernoulli distribution $p_1 = 0.97$, $p_2 = 0.20$, $p_3 = 0.63$. Our interest is to estimate the number of domains and their boundaries using our proposed method.

1	1	1	0	1	0	0	0	0	0	0	1
1	1	1	0	1	0	0	0	1	0	1	1
1	1	1	0	1	0	0	0	0	0	1	1
1	1	1	0	0	0	0	0	1	0	0	1
1	1	1	0	0	0	0	1	1	1	0	1
1	1	1	1	0	1	0	0	1	1	1	1
1	1	0	0	0	0	1	0	1	1	0	0
1	1	1	0	0	0	0	0	1	1	1	1
1	0	1	1	0	0	1	0	0	0	0	0
1	1	1	0	0	0	1	0	1	1	1	1

FIGURE 2.4: Motivating example.

Figure 2.4 shows our data matrix. We applied our algorithm to the data and obtained three exactly the same domains in two iterations. The results are given in the following table:

Table 2.1: Results for motivating example

No of Domains	RMSE	AIC	BIC
2	0.1828909	133.05342	133.2117825
3	0.0000000	117.73398	117.9715237

Table 2.1 represents that the proposed method correctly found the number of domains and their boundaries. The RMSE = 0 means that the algorithm found the same domains that we expected. It gives low AIC and BIC values when number of domains is three.

2.3.7 Algorithm 3

We introduce a modified algorithm of the above proposed Algorithm 2. It follows the same structure but at each iteration it identifies two change-points and three domains as the circular binary segmentation algorithm described in Chapter 1. Here, all three segments have different means. In this study, we consider six different cases at each iteration.

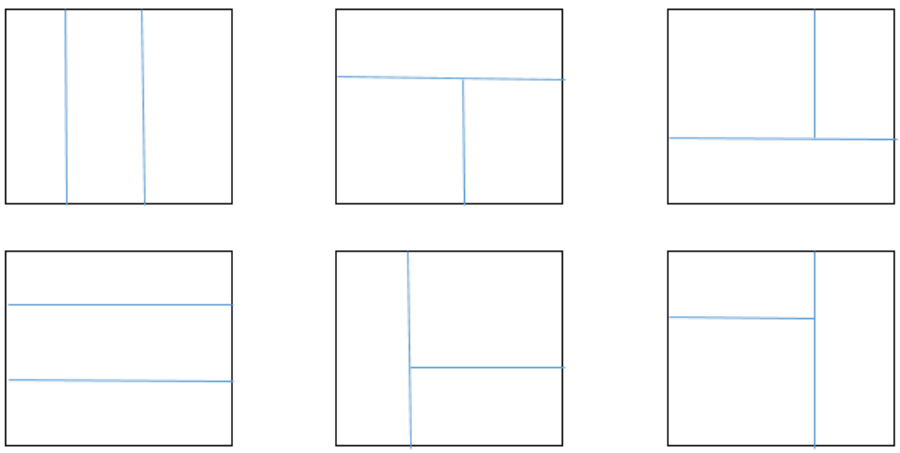

Figure 2.5: Six types of cuts.

At each iteration, Algorithm 3 performs a hypothesis testing with the null and alternative hypothesis given by:

$$H_0 : \text{No sub-domain or one domain.}$$

versus the alternatives

$$H_1 : \text{Two or three domains.}$$

Under the null hypothesis, the log-likelihood function for entire domain is given as:

$$\log(X|\hat{p}).$$

Under the H_1, the log-likelihood function (given change-points c_1 and c_2, which divide the whole domain into three homogeneous domains, say, D_1 , D_2 and D_3) is,

$$P(c) = \log(D_1|\hat{p_1}) + \log(D_2|\hat{p_2}) + \log(D_3|\hat{p_3}),$$

where $\hat{p}_1$, $\hat{p}_1$ and $\hat{p}_2$ are the maximum log-likelihood estimates of the parameters for the first, second and third domains, respectively. To estimate the locations of the change-points, the log-likelihood function under H_1 is maximized.

The test statistic:

$$\lambda(X) = 2[\max_{c} P(c) - \log(X|\hat{p})],$$

where a threshold β is chosen such that if $\lambda(X) > \beta$, the null hypothesis is rejected. Two change-points can be identified when index of c_1 or c_2 is n (number of rows) or m (number of columns).

2.3.8 Algorithm 4

Algorithm 4 is a modified version of Algorithm 2. The main difference is that at each iteration it selects the bigger domain for next iteration. We assume that the bigger domain has a higher chance to be split at the next iteration. Here, "the bigger" means the area of the rectangle. This algorithm performs well compare to Algorithm 2 and Algorithm 3. The great advantage of this algorithm is that it performs faster because at each iteration

it selects only one domain to split. But in Algorithm 2, at each iteration it considers two segments in parallel. This algorithm would be useful when we need to split the data into major domains (few number of domains).

2.4 Likelihood Ratio Test

The likelihood ratio test is a hypothesis test that is used to choose the best model between two nested models. Here, one model is a special case of other model. If we know the log-likelihood functions for the two models, the test statistic is relatively easy to calculate as the ratio between the log-likelihood of the simpler model, $L_1(\hat{\theta})$, to the log-likelihood of the larger model, $L_2(\hat{\theta})$. Thus, the likelihood ratio test statistic is given as:

$$\text{LRT} = -2\log\left(\frac{L_1(\hat{\theta})}{L_2(\hat{\theta})}\right).$$

The likelihood ratio test statistic approximately follows a chi-square distribution. The number of degrees of freedom for the test is equal to the difference in the number of parameters for the two models.

2.4.1 Likelihood Test for Spatial Clustering

In this study, we use likelihood ratio test to check whether the domains which are obtained by the proposed algorithms are homogeneous or not. The null hypothesis for this model is given as:

$$H_0 : \text{domain 1 and domain 2 are homogeneous.}$$

The alternative hypothesis is

$$H_1 : \text{domain 1 and domain 2 are not homogeneous.}$$

The test statistic is:

$$\text{LRT} = -2\log\left(\frac{\text{likelihood for null model}}{\text{likelihood for alternative model}}\right).$$

Figure 2.6 shows a matrix with two domains: domain 1 and domain 2.

1	1	1	0	1	0	0	0	0	0	1
1	1	1	0	1	0	0	0	1	0	1	1
1	1	1	0	1	0	0	0	0	0	1	1
1	1	1	0	0	0	0	0	1	0	0	1
1	1	1	0	0	0	0	1	1	1	0	1
1	1	1	1	0	1	0	0	1	1	1	1
1	1	0	0	0	0	1	0	1	1	0	0
1	1	1	0	0	0	0	0	1	1	1	1
1	0	1	1	0	0	1	0	0	0	0	0
1	1	1	0	0	0	1	0	1	1	1	1

FIGURE 2.6: Model with two domains.

After the algorithm finishes, we obtain several homogeneous domains. The next step is to perform a multiple comparison test for all combinations of the domains. Further, we consider the Bonferroni correction, which is used to control the family-wise error rate when conducting multiple hypothesis tests. The Bonferroni correction adjusts p-values when several statistical tests are being performed simultaneously on a single data set. To perform the Bonferroni correction, divide the critical p-value (α) by the number of comparisons or the number of hypothesis being made. For example, if we have M domains for our data set and have to perform N comparisons, then the Bonferroni correction would test each individual hypothesis at α/N. Here we do not need to perform all comparisons since we consider only rectangle shaped domains in this study.

3

Numerical Results

3.1 Introduction

This chapter examines the numerical results to illustrate and validate the proposed algorithms. In the first section, a simulation study was carried out to demonstrate the properties of our algorithms and to analyse their segmentation capabilities. In the next section, we present an example to illustrate the usefulness of our method. Finally, we compare our three algorithms using the Root Mean Square Error (RMSE) and information criteria. All proposed algorithms have been implemented using the statistical software R. The relevant R codes are attached in the appendix.

3.2 Simulation Results

To perform simulation study, we generate artificial matrices using Bernoulli distribution. We apply the binary segmentation Algorithm 2, record the position of the optimal cut and estimate the parameter of Bernoulli distribution for each domain. Each time we calculate the RMSE and plot a kernel density estimation curve to analyze the effectiveness of the algorithm. We focus on the following aspects.

1. Testing the performance of binary segmentation algorithm in order to detect the expected number of domains.

2. Reporting the RMSE on the size of data to test how the size of data affects the performance of the proposed algorithms.

3. Analysing the RMSE depending on the parameters of the domains.

The RMSE measures the differences between the actual values of the model and their estimates. It is commonly given as:

$$\text{RMSE} = \sqrt{\frac{\sum_{i=1}^{N}(e_i - \text{true}_i)^2}{N}},$$

where e_i, true_i denotes the estimated and the true values, respectively; N is the size of the data. In this study, we use the parameter of Bernoulli distribution to calculate the RMSE. We consider the values for each cell of the matrix. The estimated values for each cell within a particular domain remains the same. It is given as follows:

$$\text{RMSE} = \sqrt{\frac{\sum_{i=1}^{n}\sum_{j=1}^{m}(e_{ij} - t_{ij})^2}{N}},$$

where e_{ij}, t_{ij} denotes the estimated and the true values, respectively, for each cell of the matrix; i, j indicates the corresponding rows and columns; $n \times m$ is the size of the matrix.

3.2.1 Testing the Performance of Binary Segmentation Algorithm

Our aim is to find the optimal number of domains for a particular data set. We generate artificial data with four domains and run our algorithm 1000 times. We record the number

of domains identified by Algorithm 2.

Table 3.1: The number of domains with frequencies

No of Domains	1	2	3	4	5	5+
Frequency	0	0	0	316	529	155

Table 3.1 shows that the algorithm correctly found four domains in 316 simulations (out of 1000). However, the algorithm tends to overestimate the number of domains.

3.2.2 Reporting the RMSE on the Parameters of the Domains

We analyse how the RMSE depends on the parameters of domains, that is, the probability of "1". In this study, we generate artificial data with two domains, where p_1 and p_2 are the parameters of Bernoulli distributions for the domains.

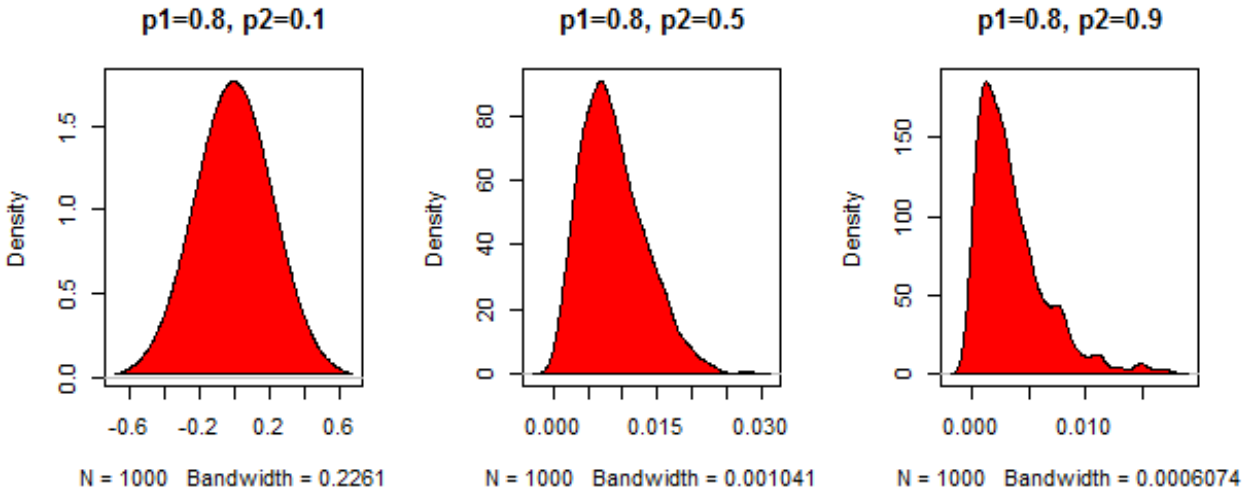

Figure 3.1: Kernel density estimation of the RMSE when $p_1 = 0.8$.

The Algorithm 2 performs well in identifying the correct position of the cut when the difference between p_1 and p_2 are rather high (for example, $p_1 = 0.8$ and $p_2 = 0.2$). Note that even if the difference is getting smaller, the algorithm works quite well; in this situation the RMSE is slightly larger compare to the high difference of the probabilities. We consider

kernel density estimation for three different cases. In all cases we fix p_1 and we change p_2 from 0.1 to 0.9. Figure 3.1 shows a kernel density estimation for $p_1 = 0.8$. In the second case (Figure 3.2), we fix $p_1 = 0.5$ and in the third case (Figure 3.3), $p_1 = 0.2$. It is clear that the effectiveness of the algorithm depends on the difference of the probabilities.

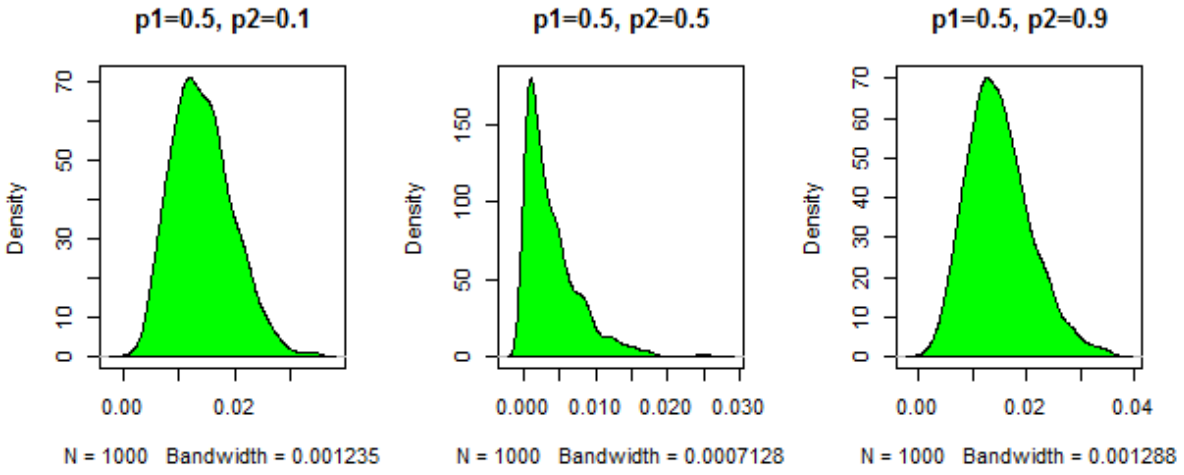

FIGURE 3.2: Kernel density estimation of the RMSE when $p_1 = 0.5$.

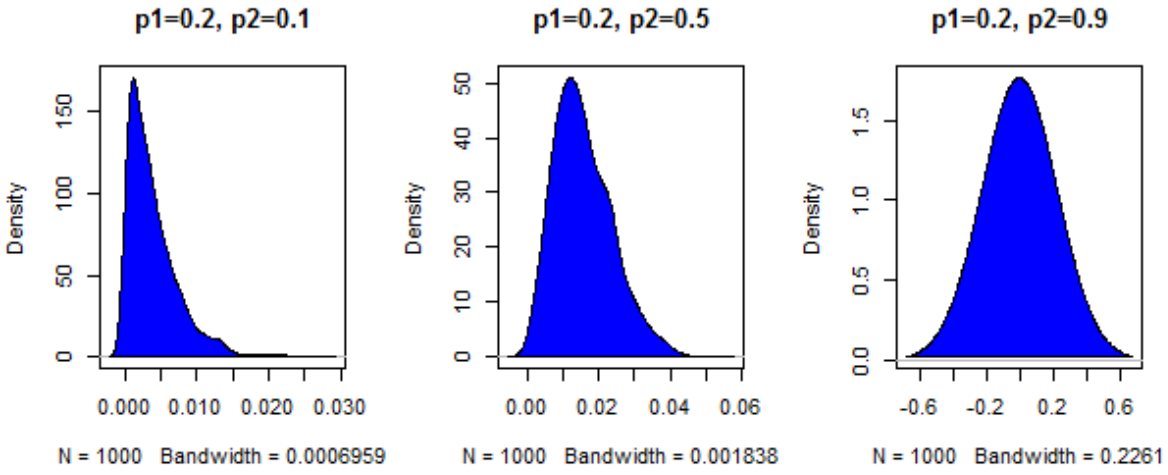

FIGURE 3.3: Kernel density estimation of the RMSE when $p_1 = 0.2$.

3.2.3 Reporting the RMSE on the Size of the Data

In this section, we analyse how the RMSE changes depending on the size of the data. It is important to test our algorithms for different sizes. We generate matrices of different sizes (100×100, 200×200, 400×400) but with the same probabilities p_1 and p_2 for domain 1 and domain 2, respectively. Here we restrict the number of domains to two.

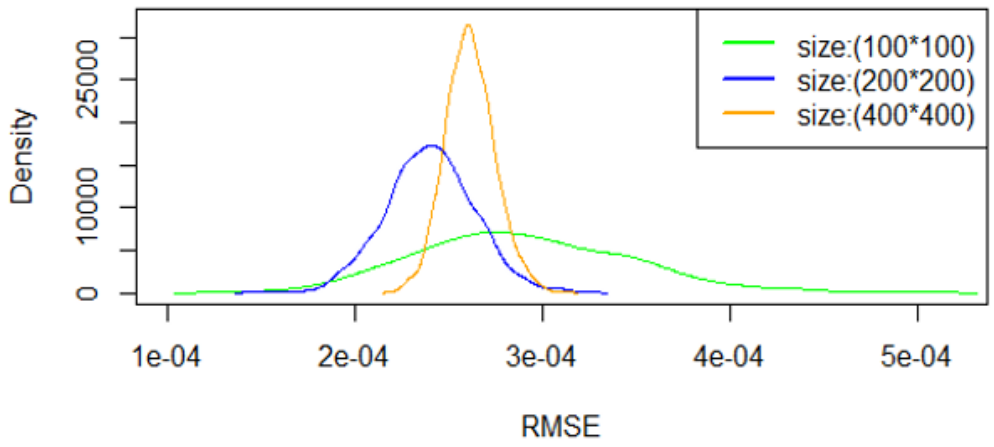

Figure 3.4: Kernel density estimation of the RMSE for different sizes.

Figure 3.4 shows the plot of kernel density estimation, which illustrates that the average value of the RMSE is not significantly influenced by the size of the data, whereas it is clear that the variability the RMSE is getting smaller when the data size is becoming larger.

3.3 Illustrative Example for Binary Segmentation

We generate a (100×100) matrix using Bernoulli distributions with four domains (all are vertical cuts); the parameters are given in Table 3.2. We apply our binary segmentation algorithms, record the positions of the optimal cuts and estimate the parameters of Bernoulli distributions at each iterations. Each time we calculate the RMSE, AIC, BIC and mBIC.

TABLE 3.2: The parameters of the generated data matrix

Domains	**Coordinates (top left to bottom right)**	**Probability (p_i)**
Domain 1	(1,1) — (100,20)	$p_1 = 0.1$
Domain 2	(1,20) — (100,60)	$p_2 = 0.5$
Domain 3	(1,60) — (100,90)	$p_3 = 0.9$
Domain 4	(1,90) — (100,100)	$p_4 = 0.2$

3.3.1 Results on Algorithm 2

We applied our binary segmentation algorithm to the data generated above (Table 3.2). Table 3.3 shows that the algorithm run up to four iterations and at the end identified seven domains.

TABLE 3.3: Results on Algorithm 2

No of Iterations	**No of Domains**	**RMSE**	**AIC**	**BIC**	**mBIC**
1	2	0.2173865	11,987.274	11,989.274	12,007.274
2	3	0.1646841	11,141.564	11, 144.564	11,167.567
3	5	0.0072969	9,789.830	9, 794.830	9,815.830
4	7	0.0190475	9,807.992	9, 814.992	9,857.992

An important step is to determine the optimal number of domains. The RMSE value attains its minimum at the third iteration (Number of domains = 5), which coincides with the results given by the information criteria AIC, BIC and mBIC.

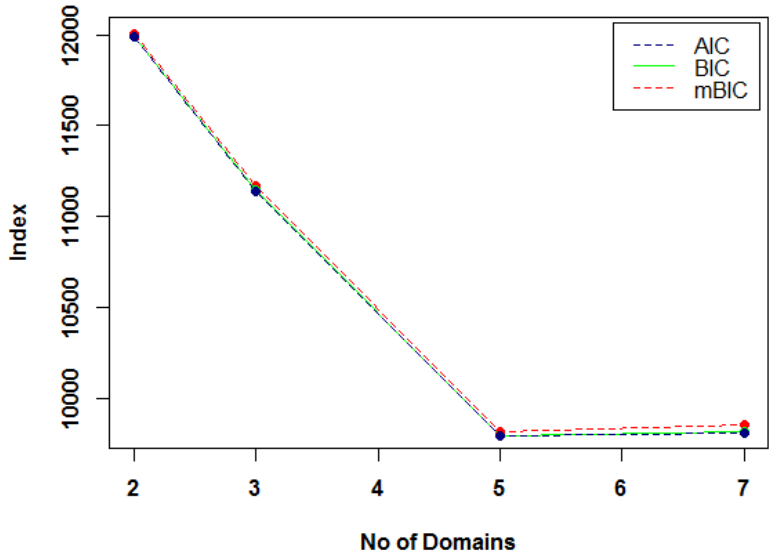

Figure 3.5: Comparisons of AIC, BIC and mBIC for Algorithm 2.

Figure 3.5 plots the values of the information criteria versus the number of domains; it shows that the minimal values for all three criteria correspond to five domains.

Now we examine the obtained domains for their heterogeneity using the likelihood ratio test. Here, we consider only rectangle shaped domains so we do not need to check all possible comparisons. Therefore, in this example, we perform only four comparisons. The results are given below.

Table 3.4: Obtained domains for Algorithm 2

Domains	**Coordinates (top left to bottom right)**
D1	(1,1) — (100,20)
D2	(1,20) — (100,60)
D3	(1,60) — (100,90)
D4	(1,90) — (6,100)
D5	(6,90) — (100,100)

TABLE 3.5: Likelihood ratio test for Algorithm 2

Domain combinations	p**-value**	**Results**
D1 and D2	< 0.00001	Significant
D2 and D3	< 0.00001	Significant
D4 and D5	0.077242	Not significant

Table 3.4 shows the obtained domains for this example. Table 3.5 illustrates the results of the likelihood ratio test. According to Table 3.5, domain 4 and domain 5 can be considered as homogeneous. Thus, we combined those two domains into one domain. Finally, we obtained the same domains as in our generated data matrix (see Table 3.2).

3.3.2 Results on Algorithm 3

We applied Algorithm 3 for the same example described in the previous section and, as before, we recorded the positions of cuts at each iterations. The detailed results on the RMSE, the information criteria and the likelihood ratio test are given below.

TABLE 3.6: Results on Algorithm 3

No of Iterations	**No of Domains**	**RMSE**	**AIC**	**BIC**	**mBIC**
1	3	0.1466984	10,838.12	10,859.75	10,864.120
2	9	0.0048789	9,974.526	10,039.42	10,036.526
3	17	0.03071665	18,840.482	18,857.482	18,950.482

It is clear from both Table 3.6 and Figure 3.6 that the RMSE, AIC, BIC and mBIC values are lowest for the case when the number of domains is equal to nine. Thus, Algorithm 3 identified nine domains for the same example illustrated above.

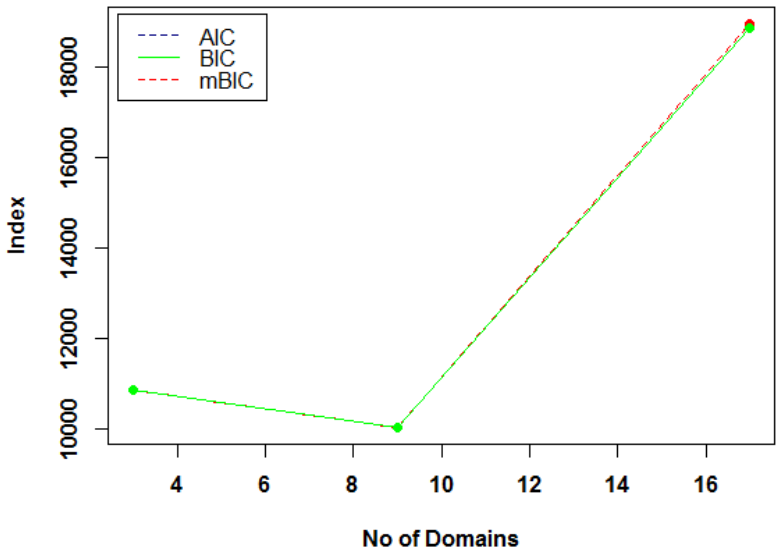

Figure 3.6: Comparisons of AIC, BIC and mBIC for Algorithm 3

Table 3.7: Likelihood ratio test for Algorithm 3

Domain Combinations	p**-value**	**Results**
D1 and D2	< 0.00001	Significant
D2 and D3	< 0.00001	Significant
D4 and D7	< 0.00001	Significant
D5 and D6	0.000512	Significant
D8 and D9	0.002597	Significant

Table 3.7 illustrates that all obtained domains identified by Algorithm 3 are significantly different.

3.3.3 Results on Algorithm 4

In this section, we applied Algorithm 4 for the same example described in above section. We recorded the positions of cuts at each iterations. The detailed results on the RMSE, the information criteria and the likelihood ratio test are given in the tables below.

TABLE 3.8: Results on Algorithm 4

No of iterations	No of Domains	RMSE	AIC	BIC	mBIC
1	2	0.2173865	11,987.274	11,989.274	12,007.274
2	3	0.1646841	11,141.564	11,144.564	11,167.564
3	4	0.0000000	9,790.952	9,794.952	9,822.952
4	5	0.0175100	9,880.598	9,885.598	9,918.598

Table 3.8 shows that the algorithm found five domains in four iterations. The RMSE, AIC, BIC and mBIC values are lowest for the case when number of domains equals four. Thus, Algorithm 4 identified four domains (the same domains as we expected) in three iterations.

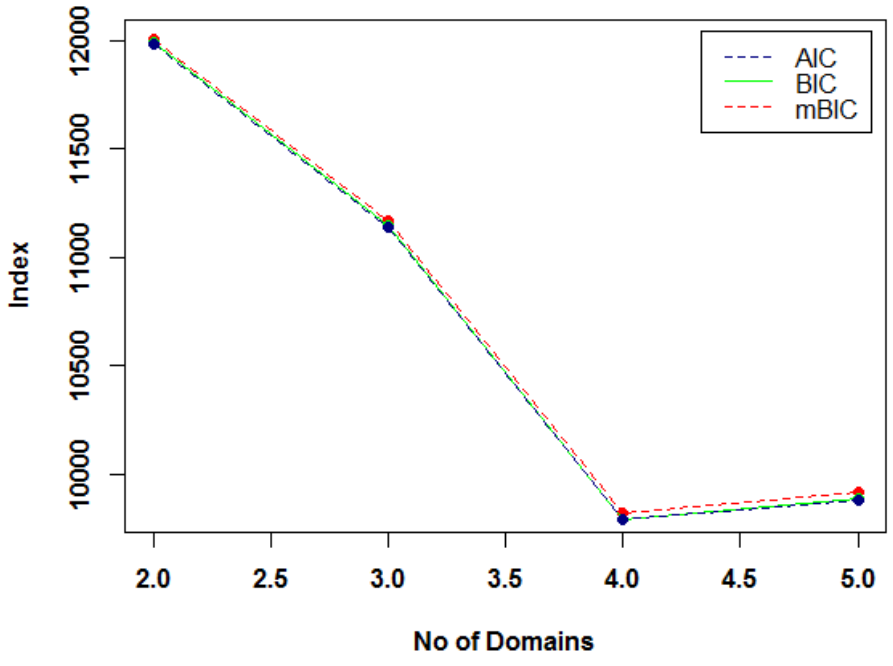

FIGURE 3.7: Comparisons of AIC, BIC and mBIC for algorithm 4.

Figure 3.7 shows that the AIC, BIC and mBIC values are lowest when the number of domains is equal to four. Table 3.9 illustrates that all obtained domains identified by Algorithm 4 are statistically significant.

Table 3.9: Likelihood ratio Test for Algorithm 4

Domain combinations	p**-value**	**Results**
D1 and D2	< 0.00001	Significant
D2 and D3	< 0.00001	Significant
D3 and D4	< 0.00001	Significant

3.3.4 Comparison of the Algorithms

In this section, we compare our all algorithms. Final results of all three algorithms in the form of the RMSE, AIC, BIC and mBIC are given below.

Table 3.10: Comparison of all three algorithms

Algorithm	**Iterations**	**Domains**	**RMSE**	**AIC**	**BIC**	**mBIC**
2	3	5	0.0072969	9,789.83	9,794.83	9,815.83
3	2	9	0.0048789	9,974.526	10,039.42	10,036.526
4	3	4	0.0000000	9,790.952	9,794.952	9,822.952

Our results show that the algorithms based on binary segmentation work well in identifying correct number of domains and their boundaries. Algorithm 3 finds more domains which are buried within larger domains. Algorithm 4 is fast and it is accurate in identifying major domains but overestimates the total number of domains.

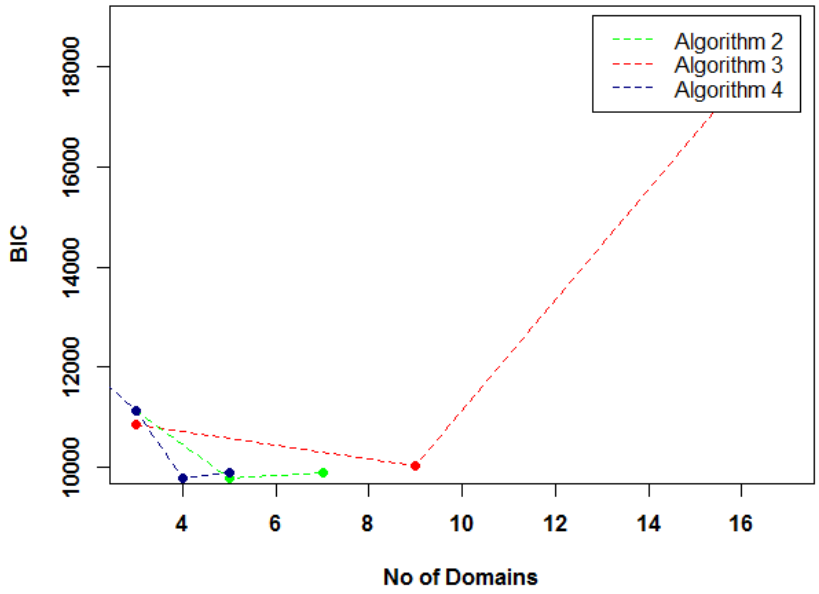

FIGURE 3.8: Comparisons of the BIC for all three algorithms.

4

Discussion and Future Directions

4.1 Summary

This thesis is divided into four chapters. In Chapter 1, an introduction to the scope of the thesis is given. It starts with discussing spatial clustering and its importance in wide range of applications. It introduces the problem of identifying spatial domains and their boundaries and gives detailed information on previous works in the literature. We propose to apply change-point methodologies in estimating the boundaries of spatial domains. The chapter introduces the change-point problem in detail. It further provides an overview of the main branches of segmentation methods in the literature. Moreover, it gives more details on both the single change-point problem and the multiple change-point problem for the posterior class of change-point problems. It further provides some existing methods on recursive segmentation related to our method. Lastly, it introduces the main concepts of

binary segmentation and its modified algorithms.

In Chapter 2, we describe the methods that are used in this thesis. First, we discuss in detail the general binary segmentation procedure and its characteristics, the use of the binary segmentation method in detecting multiple change-points and applicability to spatial data. The next section of the chapter describes binary segmentation for spatial data as well as our proposed model, algorithms, stopping criteria and motivating example. Since the change-point detection can be considered as a model selection problem, we further explain information criteria which are used to select the number of domains. Finally, the chapter briefly introduces the likelihood ratio test, which is used to check whether the obtained domains are heterogeneous or not.

Chapter 3 contains the numerical results of simulation study. The first section includes the simulation study, which is performed to illustrate the usefulness of binary segmentation in three different aspects: performance of binary segmentation in identifying the correct number of domains, results on the RMSE for different values of the parameters of domains and results on the RMSE for various sizes of data. The last two sections of the chapter compare the three proposed algorithms in terms of the RMSE and the number of obtained domains.

4.2 Discussion

There have been very few studies in the existing literature that focus on the development of statistical segmentation methods for spatial data. To address this issue, we have generalised the binary segmentation method, a well-known multiple change-point detection method, for identifying the number of homogeneous domains and their boundaries in spatial data. In particular, we have applied the modified versions of the binary segmentation algorithms to binary spatial data indicating the presence or absence of a certain plant species, which are observed over a two dimensional lattice. To carry out an extensive simulation study, all proposed algorithms have been implemented using the statistical software R. The numerical

results have illustrated that the algorithms work well under different scenarios; they accurately identify both the expected number of domains and their boundaries in few iterations.

Binary segmentation is described as "arguably the most widely used change-point search method" [26] and it is used for multidimensional data sequence. The benefits of binary segmentation include low computational complexity (typically of order $O(n)$), conceptual simplicity, the fact that it is usually easy to code, even in more complex models, and at each stage it involves one-dimensional rather than multi-dimensional optimization. On the other hand, the method is a "greedy" procedure in the sense that it is performed sequentially, with each stage depending on the previous ones, which are never re-visited.

Analysing literature on binary segmentation, we have found out that it has been never discussed with respect to identifying both number of domains and boundaries in spatial data. To fill this gap, we develop effective procedures for estimating both the number of domains and their locations in spatial data by modifying the binary segmentation method. The applications of the proposed procedures are not limited to analysing ecological data. They can be easily extended and applied to other spatial data. For instance, it can be applied to epidemiological and economic data.

4.3 Future Directions

This study is planned to be accomplished in two stages: the MRes and the Doctorate stages. The MRes thesis has mainly focused on literature review, problem formulation and the study of methodology while the remaining work will be expanded and accomplished during the Doctorate study.

Over the last decades, spatial statistical models have been studied by many authors from different angles and the spatial clustering problem is one of main topics in spatial statistics. However, the problem has not been considered as a change-point detection problem. In this thesis, we have demonstrated how spatial clusters can be identified by using a new

approach based on binary segmentation. At this stage, we have considered a simple model which assumes that observations are independent. However, statistical models that involve spatial dependence are more realistic. Extension to dependent data is considered as one of our future works. Moreover, we have only considered rectangular shaped domains and we plan to extend it to other more complex shapes in the future.

In this work, we have used univariate binary data. It is possible to consider multivariate case (for example, for several species) and other types of data such as count or continuous data as well. Furthermore, we have assumed that data is observed over a regular shaped lattice but it is also possible to consider a set of random points on a plane. The problem that we consider can be seen as a model selection problem and one of the major challenges is to determine the optimal number of domains. In this thesis, we have used well-known information criteria such as the AIC, BIC and modified BIC (developed for change-point problems). The criteria may not work well for spatial cluster models because of irregularities in their likelihood functions. Our intention is to develop new modified information criteria particularly for specific spatial segmentation problems under different assumptions.

In this thesis, we have focused on constructing binary segmentation methods because of their simplicity and low computation cost. We plan to develop new spatial segmentations algorithms bases on well-known statistical computational methods such as Cross Entropy (CE), Markov chain Monte Carlo (MCMC) and Sequentially Importance Sampling (SIS) methods.

A

An Appendix

This appendix contains the R code for the proposed algorithms.

A.1 Algorithm 2(Main Algorithm)

```
#Input data matrix for splitting into homogeneous domains.
Binary_segmentation<-function(D)
{
     N=length(D) # Total observations
     n=nrow(D)   # No of rows
     m=ncol(D)   # No of columns
     p=mean(D)   # Mean for entire data matrix
     N1=p*N      # No of ones in entire data
```

```
    N0=N-N1     # No of zeros in entire data
    H0=(N1*log(p))+(N0*log(1-p)) # Null hypothesis

# Case 1: Vertical split(Search by columns)
##########################################
#Storing all values for alternative hypothesis
H1_Vertical=c()
#Search all columns
for(i in c(2:(m-1)))
  {
    N11=mean(D[,c(1:i)])*(i*n)    # No of ones in Domain 1
    N01=(i*n)-N11                 # No of zeros in Domain 1
    N12=mean(D[,c((i+1):m)])*((m-i)*n)   # No of ones in Domain 2
    N02=((m-i)*n)-N12        # No of zeros in Domain 2
    p1=N11/(i*n)             # Mean of Domain 1
    p2=N12/((m-i)*n)         # Mean of Domain 2
    #Values for alternative hypothesis
    H1_Vertical[i]= N11*log(p1)+N01*log(1-p1)+N12*log(p2)+N02*log(1-p2)
  }
#Finding the index corresponding to the maximum value
index_Vertical=which.max(H1_Vertical)
H1_V=H1_Vertical[index_Vertical]
#Calculating Test Statistic
T_V=2*(H1_V-H0)

# Case 2: Horizontal split(Search by rows)
##########################################
H1_Horizontal=c()
for(i in c(2:(n-1)))
  {
```

```
     N11=mean(D[c(1:i),])*(i*m)     # No of ones in Domain 1
     N01=(i*m)-N11                  # No of zeros in Domain 1
     N12=mean(D[c((i+1):n),])*((n-i)*m)   # No of ones in Domain 2
     N02=((n-i)*m)-N12                    # No of zeros in Domain 2
     p1=N11/(i*m)          # mean of Domain 1
     p2=N12/((n-i)*m)      # mean of Domain 2
     #Values for alternative hypothesis
     H1_Horizontal[i]= N11*log(p1)+N01*log(1-p1)+N12*log(p2)+N02*log(1-p2)
   }
index_Horizontal=which.max(H1_Horizontal)
H1_H=H1_Horizontal[index_Horizontal]
T_H=2*(H1_H-H0)
if((T_H >= T_V) & (T_H > 2))
  {
    cut==‘Horizontal’
    index=index_Horizontal
    Domain1=D[(1:index_Horizontal),]
    Domain2=D[((index_Horizontal+1):n),]
    print(index)
    print("Horizontal")
    }else if((T_H <= T_V) & (T_V > 2))
      {
         cut==‘Vertical’
         index=index_Vertical
         Domain1=D[,(1:index_Vertical)]
         Domain2=D[,((index_Vertical+1):m)]
         print(index)
         print("Vertical")
   }else
      {
```

```
        print("No split for this case")
      }
         return(Binary_segmentation(Domain1))
         return(Binary_segmentation(Domain2))
}
```

A.2 Algorithm 3

```
# Case 1: Vertical split
#########################
Algorithm3<-Function(D)
{
df<-data.frame(case=character(),Test=numeric(),firstCut=integer(),
               secondCut=integer())
x<-c() # storing H11 values for all cases
H11=matrix(-Inf,(m-2),m) # creating a matrix to store all H11 values
for(i in c(2:(m-2)))
   {
     for(j in c((i+1):(m-1)))
       {
          N11=(mean(D[,c(1:i)])*(i*n))+(mean(D[,c((j+1):m)])*((m-j)*n))
          N01=((i*n)+((m-j)*n))-N11
          N12=mean(D[,c((i+1):j)])*((j-i)*n)
          N02=((j-i)*n)-N12
          p1=N11/((i*n)+((m-j)*n))
          p2=N12/((j-i)*n)
          H11[i,j]=N11*log(p1)+N01*log(1-p1)+N12*log(p2)+N02*log(1-p2)
       }
   }
j=m;
```

```
for(i in c(2:(m-2)))
  {
    N11=(mean(D[,c(1:i)])*(i*n))
    N01=(i*n)-N11
    N12=mean(D[,c((i+1):j)])*((j-i)*n)
    N02=((j-i)*n)-N12
    p1=N11/(i*n)
    p2=N12/((j-i)*n)}
    H11[i,j]=N11*log(p1)+N01*log(1-p1)+N12*log(p2)+N02*log(1-p2)
  }
H11[!is.finite(H11)]<-(-Inf)
a1<-which(H11==max(H11),arr.ind=TRUE)
H1max=H11[a1[1,1],a1[1,2]]
T1=2*(H1max-H0)

if(T1>2)
  {
    x[1]=T1
    df<-c("case1",T1,a1[1,1],a1[1,2])
  }else
     {
        x[1]=0
     }
# Case 2: Horizontal split
##########################
H12=matrix(-Inf,(n-2),n) # H1 values for case 2
for(i in c(1:(n-2)))
  {
    for(j in c((i+1):(n-1)))
    {
```

```
          N11=(mean(D[c(1:i),])*(i*m))+(mean(D[c((j+1):n),])*((n-j)*m))
          N01=((i*m)+((n-j)*m))-N11
          N12=mean(D1[c((i+1):j),])*((j-i)*m)
          N02=((j-i)*m)-N12
          p1=N11/((i*m)+((n-j)*m))
          p2=N12/((j-i)*m)
          H12[i,j]=N11*log(p1)+N01*log(1-p1)+N12*log(p2)+N02*log(1-p2)
    }
 }
j=n;
for(i in c(2:(n-2)))
   {
        N11=(mean(D[c(1:i),])*(i*m))
        N01=(i*m)-N11
        N12=mean(D1[c((i+1):j),])*((j-i)*m)
        N02=((j-i)*m)-N12
        p1=N11/(i*m)
        p2=N12/((j-i)*m)
        H12[i,j]=N11*log(p1)+N01*log(1-p1)+N12*log(p2)+N02*log(1-p2)
   }
H12[!is.finite(H12)]<-(-Inf)
a2<-which(H12==max(H12),arr.ind=TRUE)
H1max=H12[a2[1,1],a2[1,2]]
T2=2*(H1max-H0)
if(T2>2)
  {
     x[2]=T2
     df<-rbind(df,c("case2",T2,a2[1,1],a2[1,2]))
  }else
    {
```

```
      x[2]=0
    }
#Case 3: First horizontal then vertical cut
##########################################
H13=matrix(-Inf,(n-1),m)
for(i in c(2:(n-1)))
  {

  for(j in c(2:(m-1)))
    {
         E=D[c(i:n),]
         N11=(mean(D[c(1:i),])*(i*m))+(mean(E[,c((j+1):m)])*((m-j)*(n-i)))
         N01=((i*m)+((m-j)*(n-i)))-N11
         N12=mean(E[,c(1:j)])*(j*(n-i))
         N02=(j*(n-i))-N12
         p1=N11/((i*m)+((m-j)*(n-i)))
         p2=N12/(j*(n-i))
         H13[i,j]=N11*log(p1)+N01*log(1-p1)+N12*log(p2)+N02*log(1-p2)
    }
  }
j=m;
    for(i in c(2:(n-1)))
      {
         N11=(mean(D[c(1:i),])*(i*m))
         N01=(i*m)-N11
         N12=mean(D[c((i+1):n),])*(m*(n-i))
         N02=(m*(n-i))-N12
         p1=N11/(i*m)
         p2=N12/(m*(n-i))
         H13[i,j]=N11*log(p1)+N01*log(1-p1)+N12*log(p2)+N02*log(1-p2)
```

```
      }
H13[!is.finite(H13)]<-(-Inf)
a3<-which(H13==max(H13),arr.ind=TRUE)
H1max=H13[a3[1],a3[2]]
T3=2*(H1max-H0)
    if(T3>2)
      {
         x[3]=T3
         df<-rbind(df,c("case3",T3,a3[1,1],a3[1,2]))
      }else
        {
            x[3]=0
        }
# Case 4: First horizontal then vertical(upper cut)
#####################################################
H14=matrix(-Inf,(n-1),m)
for(i in c(2:(n-1)))
  {
  for(j in c(2:(m-1)))
    {
           E=D[c(1:i),]
           N11=(mean(D[c((i+1):n),])*((n-i)*m))+(mean(E[,c((j+1):m)])*((m-j)*i))
           N01=(((n-i)*m)+((m-j)*i))-N11
           N12=mean(E[,c(1:j)])*(j*i)
           N02=(j*i)-N12
           p1=N11/(((n-i)*m)+((m-j)*i))
           p2=N12/(j*i)
           H14[i,j]=N11*log(p1)+N01*log(1-p1)+N12*log(p2)+N02*log(1-p2)
     }
  }
```

```
j=m;
    for(i in c(2:(n-1)))
      {
             N11=(mean(D[c((i+1):n),])*((n-i)*m))
             N01=((n-i)*m)-N11
             N12=mean(D[c(1:i),])*(m*i)
             N02=(m*i)-N12
             p1=N11/((n-i)*m)
             p2=N12/(m*i)
             H14[i,j]=N11*log(p1)+N01*log(1-p1)+N12*log(p2)+N02*log(1-p2)
      }
H14[!is.finite(H14)]<-(-Inf)
a4<-which(H14==max(H14),arr.ind=TRUE)
T4=2*(H1max-H0)
if(T4>2)
  {
      x[4]=T4
      df<-rbind(df,c("case4",T4,a4[1,1],a4[1,2]))
  }else
    {
       x[4]=0
    }
# Case 5: First vertical then Horizontal
########################################
H15=matrix(-Inf,(m-2),n)
for(i in c(1:(m-2)))
  {
     for(j in c(2:(n-1)))
       {
          E=D[,c((i+1):m)]
```

```
          N11=(mean(D[,c(1:i)])*(i*n))+(mean(E[c((j+1):n),])*((n-j)*(m-i)))
          N01=((i*n)+((n-j)*(m-i)))-N11
          N12=mean(E[c(1:j),])*(j*(m-i))
          N02=(j*(m-i))-N12
          p1=N11/((i*n)+((n-j)*(m-i)))
          p2=N12/(j*(m-i))
          H15[i,j]=N11*log(p1)+N01*log(1-p1)+N12*log(p2)+N02*log(1-p2)
       }
  }
j=n;
    for(i in c(1:(m-2)))
      {
          N11=(mean(D[,c(1:i)])*(i*n))
          N01=(i*n)-N11
          N12=mean(D[,c((i+1):m)])*(n*(m-i))
          N02=(n*(m-i))-N12
          p1=N11/(i*n)
          p2=N12/(n*(m-i))
          H15[i,j]=N11*log(p1)+N01*log(1-p1)+N12*log(p2)+N02*log(1-p2)
      }
H15[!is.finite(H15)]<-(-Inf)
a5<-which(H15==max(H15),arr.ind=TRUE)
H1max=H15[a5[1],a5[2]]
T5=2*(H1max-H0)
    if(T5>2)
      {
           x[5]=T5
           df<-rbind(df,c("case5",T5,a5[1,1],a5[1,2]))
      }else
        {
```

```
          x[5]=0
        }
#Case 6: First vertical then Horizontal
########################################
H16=matrix(-Inf,(m),n)
for(i in c(2:(m-1)))
  {
     for(j in c(2:(n-1)))
       {
            E=D[,c(1:i)]
            N11=(mean(D1[,c((i+1):m)])*((m-i)*n))+(mean(E[c((j+1):n),])*((n-j)*i))
            N01=(((m-i)*n)+((n-j)*i))-N11
            N12=mean(E[c(1:j),])*(j*i)
            N02=(j*i)-N12
            p1=N11/(((m-i)*n)+((n-j)*i))
            p2=N12/(j*i)
            H16[i,j]=N11*log(p1)+N01*log(1-p1)+N12*log(p2)+N02*log(1-p2)
       }
  }
j=n;
    for(i in c(2:(m-1)))
      {
             N11=(mean(D1[,c((i+1):m)])*((m-i)*n))
             N01=((m-i)*n)-N11
             N12=mean(D1[,c(1:i)])*(n*i)
             N02=(n*i)-N12
             p1=N11/((m-i)*n)
             p2=N12/(n*i)
             H16[i,j]=N11*log(p1)+N01*log(1-p1)+N12*log(p2)+N02*log(1-p2)
      }
```

```
H16[!is.finite(H16)]<-(-Inf)
i=m;
   for(j in c(2:(n-1)))
     {
              N11=(mean(D1[c((j+1):n),])*((n-j)*m))
              N01=((n-j)*m)-N11
              N12=mean(D1[c(1:j),])*(m*j)
              N02=(m*j)-N12
              p1=N11/((n-j)*m)
              p2=N12/(m*j)
              H16[i,j]=N11*log(p1)+N01*log(1-p1)+N12*log(p2)+N02*log(1-p2)
      }
H16[!is.finite(H16)]<-(-Inf)
a6<-which(H16==max(H16),arr.ind=TRUE)
H1max=H16[a6[1],a6[2]]
T6=2*(H1max-H0)
   if(T6>2)
     {
          x[6]=T6
          df<-rbind(df,c("case6",T6,a6[1,1],a6[1,2]))
     }else
       {
          x[6]=0
       }
}
```

References

[1] Akaike, H. (1974). A new look at the statistical model identification. *IEEE transactions on automatic control*, 19(6):716–723.

[2] Anderson, C., Lee, D., and Dean, N. (2016). Bayesian cluster detection via adjacency modelling. *Spatial and spatio-temporal epidemiology*, 16:11–20.

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

[4] Beckage, B., Joseph, L., Belisle, P., Wolfson, D. B., and Platt, W. J. (2007). Bayesian change-point analyses in ecology. *New Phytologist*, 174(2):456–467.

[5] Braun, J. V. and Muller, H.-G. (1998). Statistical methods for DNA sequence segmentation. *Statistical Science*, 13(2):142–162.

[6] Chen, J., Gupta, A., and Pan, J. (2006). Information criterion and change point problem for regular models. *Sankhyā: The Indian Journal of Statistics*, pages 252–282.

[7] Chen, J. and Gupta, A. K. (1997). Testing and locating variance changepoints with application to stock prices. *Journal of the American Statistical association*, 92(438):739–747.

[8] Chen, J. and Gupta, A. K. (2011). *Parametric statistical change point analysis: with applications to genetics, medicine, and finance*. Springer Science & Business Media.

[9] Chen, S. S. and Gopalakrishnan, P. S. (1998). Clustering via the bayesian information criterion with applications in speech recognition. In *Acoustics, Speech and Signal Processing, 1998. Proceedings of the 1998 IEEE International Conference on*, volume 2, pages 645–648. IEEE.

[10] Cho, H. and Fryzlewicz, P. (2012). Multiscale and multilevel technique for consistent segmentation of nonstationary time series. *Statistica Sinica*, pages 207–229.

[11] Cho, H. and Fryzlewicz, P. (2015). Multiple-change-point detection for high dimensional time series via sparsified binary segmentation. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 77(2):475–507.

[12] Cliff, A. D. and Ord, J. K. (1981). *Spatial processes: models & applications*. Taylor & Francis.

[13] Cressie, N. (2015). *Statistics for spatial data*. John Wiley & Sons.

[14] Eckley, I. A., Fearnhead, P., and Killick, R. (2011). Analysis of changepoint models. *Bayesian Time Series Models*, pages 205–224.

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

[16] Fryzlewicz, P. and Subba Rao, S. (2014). Multiple-change-point detection for auto-regressive conditional heteroscedastic processes. *Journal of the Royal Statistical Society: series B (statistical methodology)*, 76(5):903–924.

[17] Gangnon, R. E. and Clayton, M. K. (2000). Bayesian detection and modeling of spatial disease clustering. *Biometrics*, 56(3):922–935.

[18] Haccou, P., Meelis, E., and Van De Geer, S. (1987). The likelihood ratio test for the change point problem for exponentially distributed random variables. *Stochastic processes and their applications*, 27:121–139.

[19] Helterbrand, J. D., Cressie, N., and Davidson, J. L. (1994). A statistical approach to identifying closed object boundaries in images. *Advances in applied probability*, 26(04):831–854.

[20] Hinkley, D. V. (1970). Inference about the change-point in a sequence of random variables. *Biometrika*, 57(1):1–17.

[21] Hinkley, D. V. and Hinkley, E. A. (1970). Inference about the change-point in a sequence of binomial variables. *Biometrika*, 57(3):477–488.

[22] Hsu, D. (1979). Detecting shifts of parameter in gamma sequences with applications to stock price and air traffic flow analysis. *Journal of the American Statistical Association*, 74(365):31–40.

[23] Jackson, B., Scargle, J. D., Barnes, D., Arabhi, S., Alt, A., Gioumousis, P., Gwin, E., Sangtrakulcharoen, P., Tan, L., and Tsai, T. T. (2005). An algorithm for optimal partitioning of data on an interval. *IEEE Signal Processing Letters*, 12(2):105–108.

[24] Jen, T. and Gupta, A. (1987). On testing homogeneity of variances for gaussian models. *Journal of Statistical Computation and Simulation*, 27(2):155–173.

[25] Killick, R., Eckley, I. A., Jonathan, P., and Chester, U. (2011). Efficient detection of multiple changepoints within an oceano-graphic time series. In *Proceedings of the 58th World Science Congress of ISI.*

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

[27] Li, W. (2001). Dna segmentation as a model selection process. In *Proceedings of the fifth annual international conference on Computational biology*, pages 204–210. ACM.

[28] López, I., Gámez, M., Garay, J., Standovár, T., and Varga, Z. (2010). Application of change-point problem to the detection of plant patches. *Acta biotheoretica*, 58(1):51–63.

[29] Nicholls, G. K. and Nunn, P. D. (2010). On building and fitting a spatio-temporal change-point model for settlement and growth at bourewa, fiji islands. *arXiv preprint arXiv:1006.5575*.

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

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

[32] Pan, J. and Chen, J. (2006). Application of modified information criterion to multiple change point problems. *Journal of multivariate analysis*, 97(10):2221–2241.

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

[34] Schwarz, G. et al. (1978). Estimating the dimension of a model. *The annals of statistics*, 6(2):461–464.

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

[36] Sen, A. and Srivastava, M. S. (1975). On tests for detecting change in mean. *The Annals of statistics*, 3(1):98–108.

[37] Shiryaev, A. N. (1963). On optimum methods in quickest detection problems. *Theory of Probability & Its Applications*, 8(1):22–46.

[38] Sofronov, G. Y., Glotov, N. V., Zhukova, O. V., et al. (2015). Statistical analysis of spatial distribution in populations of microspecies of Alchemilla l. 2:259–262.

[39] Tripathi, S. and Govindaraju, R. S. (2009). Change detection in rainfall and temperature patterns over India. In *Proceedings of the Third International Workshop on Knowledge Discovery from Sensor Data*, pages 133–141. ACM.

[40] Tung, A. K., Hou, J., and Han, J. (2001). Spatial clustering in the presence of obstacles. In *Data Engineering, 2001. Proceedings. 17th International Conference on*, pages 359–367. IEEE.

[41] Upton, G. J. and Fingleton, B. (1985). Spatial data analysis by example. vol. 1: Point pattern and quantitative data. *Chichester: Wiley, 1985*, 1.

[42] Venkatraman, E. S. (1992). *Consistency results in multiple change-point problems*. PhD thesis, to the Department of Statistics.Stanford University.

[43] Vostrikova, L. (1981). Detection of the disorder in multidimensional random-processes. *Doklady Akademii Nauk SSSR*, 259(2):270–274.

[44] Wang, Y. (1998). Change curve estimation via wavelets. *Journal of the American Statistical Association*, 93(441):163–172.

[45] Yang, T. Y. (2004). Bayesian binary segmentation procedure for detecting streakiness in sports. *Journal of the Royal Statistical Society: Series A (Statistics in Society)*, 167(4):627–637.

[46] Yang, T. Y. and Swartz, T. B. (2005). Applications of binary segmentation to the estimation of quantal response curves and spatial intensity. *Biometrical journal*, 47(4):489–501.

[47] Yao, Y.-C. (1988). Estimating the number of change-points via schwarz'criterion. *Statistics & Probability Letters*, 6(3):181–189.

[48] Young Yang, T. and Kuo, L. (2001). Bayesian binary segmentation procedure for a poisson process with multiple changepoints. *Journal of Computational and Graphical Statistics*, 10(4):772–785.

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