STATISTICAL MODELING AND INFERENCE FOR MULTIPLE TEMPORAL OR SPATIAL CLUSTER DETECTION

BY QIANKUN SUN

A dissertation submitted to the

Graduate School—New Brunswick

Rutgers, The State University of New Jersey

in partial fulfillment of the requirements

for the degree of

Doctor of Philosophy

Graduate Program in Statistics

Written under the direction of

Minge Xie

and approved by

_____

_____

_____

_____

New Brunswick, New Jersey

Oct, 2008

ABSTRACT OF THE DISSERTATION

Statistical Modeling and Inference for Multiple Temporal or Spatial Cluster Detection

by Qiankun Sun

Dissertation Director: Minge Xie

This thesis develops a latent modeling framework and likelihood based inference tool to detect multiple temporal or spatial clusters.

Cluster detection is important to researchers from various fields. Practical applications include: biological studies of DNA sequencing, environmental researches, epidemiological studies and surveillance for biological terrorism. The traditional scan statistics procedures have technical difficulties to detect multiple clusters of varying sizes. Some Bayesian approaches have to limit the potential clusters in cell divisions. A recently proposed stepwise regression method tends to be inefficient in some cases. We utilize some probability distributions to model the latent clusters and mimic the sample data generation process. With model selection techniques, we can obtain an optimal number of total potential clusters. Based on a Monte-Carlo EM algorithm and likelihood based inference, we are able to estimate the associated model parameters, detect significant clusters and identify their locations and sizes. Compared with other procedures, this new approach is intuitive and simple. It is also more efficient and flexible for further extensions.

Preface

This thesis develops a general latent modeling framework and likelihood based inference tool to detect clustering events within temporal or spatial samples.

Multiple cluster detection in temporal or spatial data has become more and more important ever since last century. Practical applications include: in biological studies of DNA sequencing, the detection of unusual clusters of specific patterns can be used to allocate lab resources and help find "biologically important origins of diseases". In environmental studies, people living near factories that generate pollution may have an increased chance of certain diseases. It is important to detect and monitor such clusters. In epidemiological studies, when the "etiology of diseases" has not been well established, it is often required to obtain evidence of temporal or spatial clusters. In surveillance for biological terrorism, it is essential to provide early warnings of terrorist attacks.

A large number of approaches have been proposed for cluster detection. Among them, the traditional scan statistics procedures, based on a "hypothesis testing framework", have technical difficulties to detect multiple clusters of varying sizes. Some Bayesian approaches, based on a "disease mapping framework", have to limit the potential clusters in cell divisions. A recently proposed stepwise regression method, based on a "distance measuring framework", tends to be inefficient in some cases.

We utilize some probability distributions to model the latent clusters and mimic the data generation processes. This modeling framework is intuitive and allows multiple cluster detection. With model selection techniques, we can obtain an optimal number of potential clusters. Based on likelihood inference and a Monte-Carlo EM algorithm, we are able to estimate the associated model parameters, detect significant clusters and identify their locations and sizes. Compared with other methods, this new approach has

several distinct advantages: (i) It does not limit potential choices of clusters in a finite set. (ii) It is more efficient than those that rely on pseudo-likelihood, weighted least squares or empirically weighted likelihood ratio tests. (iii) It is flexible for extensions that include baseline function incorporation, temporal-spatial or higher dimensional cluster detection. Though it requires some programming efforts, this method is comprehensible and can be easily implemented. A new and efficient tool to solve cluster detection problems is developed.

This thesis is organized as follows. An introduction of "cluster" definition, its applications and importance, and some existing methods are reviewed in chapter 1. Those who are familiar with this information can skip this part. Chapter 2 provides detailed procedures for multiple temporal cluster detection. Some probability distributions are utilized to model the locations and sizes of the latent cluster intervals. A piecewise uniform distribution is then introduced to mimic the sample data generation process. When the number of clusters is assumed to be given, a Monte-Carlo EM algorithm is developed to estimate the associated model parameters. Wald and LRT tests are then used to test the significance of the latent clusters. When the results are significant, robust methods are proposed to estimate the locations and sizes of the latent cluster intervals. With the number of clusters unknown, AIC or BIC model selection criterion is used to determine the optimal number of total potential clusters. Lastly, simulation studies and real data analysis are performed to illustrate the efficiency of this new method. The whole procedure is extended in chapter 3 for multiple spatial cluster detection. Chapter 4 discusses our findings and outlines some open questions. Detailed equations for the spatial case are derived in Appendix A. Appendix B provides the two real data sets we analyze in the temporal case. Finally, the R and C program codes for the temporal case are presented in Appendix C.

Acknowledgements

I'd like to acknowledge those who have made this thesis possible. First of all, I want to thank my advisor Professor Minge Xie for his stimulating instructions, patient guidance and helpful support over the years. Secondly, I want to thank Professor Joseph Naus, who is both one of the joint workers and my thesis committee members, for his wonderful instructions on this project. I also want to thank other committee members, Professor John Kolassa and Dr. Jun Zhao, for their precious time, continued guidance and encouragement. Moreover, I am very grateful to Professor Cunhui Zhang, who has provided me the great opportunity of studying in Rutgers and has been so patient in providing advices whenever needed. In addition, I want to say thanks to all my friends from Rutgers University, for being the surrogate family during these years. Finally, I am forever indebted to my father Dezhen Sun, my mother Yanchun Ye, my elder sister Zhikun Sun, my younger sister Yukun Sun and my husband Yun Wang for their understanding, endless encouragement and support when it is most required.

Dedication

To my family who are supporting me all the time.

Table of Contents

List of Tables

List of Figures

Chapter 1

Background and Literature Review

This chapter provides background information about cluster detection and reviews some existing methods. In section 1.1, two commonly used "cluster" definitions are introduced. In section 1.2, the importance and some applications of cluster detection are provided. Section 1.3 reviews some existing methods. Our contributions are briefly presented in section 1.4. Those who are familiar with this information can directly skip to the next chapter.

1.1 What is "cluster"?

There are two commonly used equivalent definitions of "cluster". The first one is that clusters are temporal intervals or spatial regions within which an incidence of interest is much more/less likely to happen (i.e., with a much greater/smaller probability to happen per unit time or area) than that outside these temporal intervals or spatial regions. The second one is that cases inside these temporal intervals or spatial regions are closer to/farther from each other than cases outside these temporal intervals or spatial regions [15]. Our approach uses the first definition.

1.2 Applications and importance of cluster detection

In the past 50 years, researchers have been investigating different types of clusters in time and space. Some applications look for an unusually large number of events within small clusters, or patterns that suggest clumping over the entire study period or area. Other applications are concerned with unusually large clusters within a small region of time, space or location in a sequence. In some cases, focus lies on a specific region, for example, a region with heavy pollution. In other cases, researchers scan the entire

study area and seek to locate unusually high likelihood of clustering. Practical examples cover a wide range of fields over various disciplines. For instance:

In biological studies of DNA sequencing, the detection of clusters of unusual pattern can be used to allocate lab resources and help find "biologically important origins of diseases". For example, Leung et al. uses palindrome clustering in DNA to locate the origin of replication of viruses [28].

In environmental studies, people living near factories that generate pollution may have an increased chance of certain diseases. It is of interest to detect and monitor such clusters. The AEGISS (Ascertainment and Enhancement of Gastrointestinal Infection Surveillance and Statistics) project is such an example. This project aims to identify anomalies in the space and time distribution of non-specific gastrointestinal infections in the Southampton area in England [10].

In epidemiological studies, when the "etiology of diseases" has not been well established, it is often required to study the data to obtain evidence of temporal or spatial clusters [30, 6]. Establishing an etiologic link with exposure both provides clues to the cause of disease and helps target periods or areas when or where health care needs improvement for preventive measures [22].

In surveillance for biological terrorism, it is essential to provide early warnings of terrorist attacks. Syndrome surveillance for biological terrorism requires statistical methods that detect "relatively abrupt increase in incidence" [53].

Other areas of application are: agriculture, archaeology, botany, criminology, demography, ecology, economics, engineering, entomology, forestry, genetics, geography, health management, history, neurology, physics, sociology, veterinary medicine, zoology, cosmology with spatial clustering of galaxies, agronomy and more.

1.3 Existing methods for cluster detection

A large number of statistical methods have been proposed for cluster detection. Some methods are based on a "hypothesis testing framework". They usually test a null hypothesis of a common disease rate against a "clustering" alternative, for example, the

scan statistics procedures [22, 23]. Some are based on a "disease mapping framework". These methods usually use Bayes or empirical Bayes approaches to produce smoothed estimates of cell-specific disease rates suitable for mapping, for example, the Bayesian procedure developed by Gangnon and Clayton [13]. Others adopt a "distance measuring framework". For these methods, statistics that measure average distance between cases are usually used, for example, the stepwise regression methods [30, 6]. These three kinds of methods are respectively reviewed in more detail in the following three subsections.

1.3.1 Scan statistics procedures

A traditional statistical method to detect clustering of events is via *Scan Statistics*. It was first studied in 1965 by Naus, who looked at the cluster detection problem in both one and two dimensions [32, 33]. Afterwards, the same statistician along with others have published many papers to develop the statistical theory. Exact distributions for statistical inference are derived [32, 33, 34, 35, 36, 37, 38, 39]. For the one-dimensional problem, the most commonly used scan statistic is the maximum number of cases in a fixed size moving-window that scans through the whole study period [32, 33]. The test based on this scan statistic has been shown to be a generalized likelihood ratio test for a uniform null against a pulse alternative [35]. A related scan statistic is the diameter of the smallest window that contains a fixed number of cases. For both statistics, the exact distributions are only known in special situations. Asymptotic and approximate expressions are used instead for general statistical inference [37, 18] and power evaluation [52, 38, 39, 8, 47]. In higher dimensions, theory becomes more complex. Only distributional bounds are obtained for a two dimensional scan statistic with a rectangular window of fixed size on a square [33]. More detailed information about a variety of these statistics and their applications can be obtained in four recent books [3, 12, 19, 20].

Since the exact distribution of this statistic can not be determined, starting 1995, an extended scan statistic that uses a range of fixed window sizes or fixed number of cases is proposed [22, 23, 24, 45]. This statistic uses Monte Carlo simulation [11] to perform the hypothesis testing. An underlying intensity that generates sample events under

the null hypothesis is first assumed. The usual probability distribution used is either Poisson or Bernoulli. A regular or irregular grid of centroids that cover the whole study region is then created. Around each centroid, an infinite number of circles are produced. The radii of these circles range from zero up to a maximum so that at most 50 percent of the population is included inside. For each circle, actual and expected number of cases inside and outside are obtained and the likelihood function is calculated. The scan statistic is defined as the maximum likelihood ratio test statistic. The circle with the highest likelihood function is picked as the most likely cluster. For statistical inference, random replicas of the data set are generated under the null hypothesis of no clusters via Monte Carlo sampling. The scan statistics in real and random data sets are then calculated and ordered. If the scan statistic from the real data set is ranked in the highest β percent, the null hypothesis is then rejected at β percent significance level.

The scan statistics procedures have been very successful in detecting a single significant cluster. They also have some success in detecting multiple clusters of fixed sizes. Therefore, it is not surprising that together with a developed free software, SatScan, the extended spatial scan statistic methodology has become a popular method for cluster detection. However, they have technical difficulties to detect multiple clusters of varying sizes and make inference for related problems. Though the SatScan software is currently being extended to detect elliptical-shaped clusters [25], scan statistics have low powers to detect irregularly shaped clusters due to the use of rectangular or circular shaped scan windows. Moreover, these statistics require a set of potential clusters to be specified in advance.

1.3.2 Bayesian approaches

In addition to classical statistical methods, Bayesian approaches that compute posterior probabilities of potential clusters have also been proposed for cluster detection problems. One natural scan statistics extended "Bayesian spatial scan statistic" is proposed for spatial cluster detection by Neill et. al [40]. This method uses a conjugate Gamma-Poisson model instead of Poisson model for the model assumption. Compared with the standard frequentist methods, this Bayesian method can not only incorporate prior

information about the size and shape of a cluster, but also the impact of the cluster on the monitored data stream. Moreover, since closed form for likelihood functions can be obtained, randomization testing becomes unnecessary. Both make this method a faster algorithm. However, just as with the standard frequentist scan statistics methods, the potential spatial clusters are limited to a finite set of specific choices.

Gangnon and Clayton develop a "weighted average likelihood ratio statistic (WALR)" with a Bayesian interpretation. This statistic approximates posterior probabilities of a cell being part of the cluster, and in turn helps locate the cluster [14]. The set of cells is usually pre-fixed and finite, which leads to a finite choice of potential clusters. They note in a later paper [16] that the posterior probabilities may not specify a region of clustering. This is a disadvantage of the WALR statistic. To both achieve the advantage of the specificity of scan statistics and correct for the cluster size bias, they further develop two other scan type statistics: a "weighted average likelihood ratio scan statistic (WALRS)" and a "penalized scan statistic" [16]. These approaches can be viewed as generalized scan statistics approaches.

Some Bayesian methods are based on the "disease mapping framework" [9, 13, 15, 17, 21, 26, 27, 54]. Among them, the Bayesian approach proposed in Gangnon [13] incorporates ideas from image analysis, Bayesian model averaging and model selection. It analyzes a study region that is first divided into N subregions, or cells. For each cell i, the events number O_i and the population at risk n_i are observed. The interest lies on the underlying event rates r_i, $i = 1, \cdots, N$. A cluster model with k clusters is identified by a vector of cluster memberships $\mathbf{c} = (c_1, \cdots, c_N)$, where $c_i = 0$ if cell i belongs to the background and $c_i = j$ if it belongs to cluster $j, j = 1, \cdots, k$. With the assumption that O_i is Poisson$(r_i n_i)$ distributed and r_i is hierarchical gamma prior distributed and assuming $\mathbf{c}$ is known, the conditional posterior distribution of $(r_j \mid \mathbf{c,\ O})$ is found to be gamma distributed. Starting with a saturated model with N-1 clusters, a randomized model search algorithm similar to the backwards elimination is proposed by repeatedly merging adjacent components to produce models with high posterior densities.

Even though the above methods can both allow for multiple cluster detection and produce estimates for disease rates, the potential spatial clusters are limited to the

cell divisions. In addition, disease rates are estimated conditional on the estimated clusters. Such conditional estimation may not accurately reflect the uncertainty about the composition of the cluster. Meanwhile, the choice of priors is always challenging.

1.3.3 Distance measuring approaches

One distance measuring approach is by Molinari et. al [30]. This method uses a stepwise regression and model selection procedures to locate and determine the number of unusually high clustering regions in temporal data. For a given number of clusters, the location of the clusters are determined by a weighted least square method where the observed response values are the inter-arrival times (gaps) between events. To make inference, they rely on bootstrap methodology and the weighted least square formulation. Since the responses used in their model are usually non-normally distributed, the weighted least square method, however, may not be efficient. Meanwhile, the bootstrap simulation could be computationally expensive. To overcome the difficulty of using bootstrap simulation, based on Beinstein's inequality, Demattei et. al propose a new method for testing Molinari et. al's stepwise regression approach [4, 5]. This method is conservative in terms of declaring significant clusters (thus loss of power), which is inherited from the inequality.

Recently, Demattei et. al extend the Molinari et. al's method to detect arbitrarily shaped multiple spatial clusters [6, 7]. This new approach deals with precise events within R^2, such as spatial coordinates for the occurrence of disease cases or the geographical positions of individuals. Once pre-selected points have been taken into account, a selection order and the distance from the nearest neighbor are attributed to each point. These distances are weighted by the expected distance under the uniform distribution hypothesis. For a given number of clusters, these ordered weighted distances are then used to structure a stepwise regression model. The cluster bounds ("breaks") are determined by a constrained least square method. The disc-based wrap method is then used to visualize the cluster zone. The best model containing one or several portions (potential clusters) is selected using the double maximum test. The final potential clusters are determined after a final union of the portions (two or more)

that have a non-empty wrap intersection. Finally, a p-value is obtained for each potential cluster. This method is shown not to be influenced by the choice of the first ordered point via a simulation study. With this method, multiple clusters of any shape can be detected. One limiting point of this method is that the trajectory may leave the cluster before going through all the cluster points. The remaining cluster points will then be detected as a second cluster. The union of the two clusters could thus build a new larger cluster. Moreover, since the disc-based wrap method depends on the ratio of $\frac{N(\text{ total number of background population })}{n(\text{ total number of disease cases })}$, the final sizes of the estimated cluster zones may not be precise. Another issue is that this method can only adjust for an underlying population inhomogeneity. The adjustment for other covariates such as age or gender is not possible yet.

1.4 Our method

Our approach is different from the above methods. We first mimic the processes and mechanisms that generate the clusters and develop a latent structure model. This model allows us to use the standard or constrained likelihood inference to detect multiple clusters in a given window or region. Unlike the scan statistics procedures, we emphasize detecting multiple clusters of varying sizes simultaneously. Therefore, it can not only detect multiple clusters all together but also identify the most significant single cluster. In our approach, the likelihood functions can be fully specified and computed. We can answer a variety of inference questions related to our goal. The likelihood based approach is more efficient than the stepwise regression method (SR) [30], which is illustrated via simulation studies. Furthermore, the latent model approach is flexible and can incorporate various extensions.

The rest of this thesis is organized as follows. Chapter 2 provides detailed procedures to detect multiple temporal clusters. A model structure for the latent clusters is proposed in Section 2.1 followed by a piecewise uniform distribution to mimic the sample data generation process. In Section 2.2, the detailed model inference is developed with given number of clusters. Likelihood functions of observed data are first derived in Subsection 2.2.1. An EM/MCMC algorithm is then developed in Subsection 2.2.2

to estimate the associated parameters. Wald and LRT statistics are used in Subsection 2.2.3 for the significant testings of the latent clusters. Robust methods to estimate the cluster intervals are proposed in Subsection 2.2.4 when the tests show significance. With the number of clusters unknown, model selection criteria are presented in Section 2.3 to determine the optimal number of total potential clusters. Simulation studies and real data analysis are performed in Section 2.4 and 2.5 to illustrate and evaluate the proposed method. Chapter 3 extends the approach for spatial cluster detection. Chapter 4 discusses our findings and outlines some open questions. Equations for the spatial case are derived in more detail in Appendix A. Appendix B provides the two real data sets we analyze in the temporal case. Finally, the R and C program codes for the temporal case are presented in Appendix C.

Chapter 2

Multiple Temporal Cluster Detection

This chapter presents detailed procedures to detect multiple temporal clusters of varying sizes within a given time period. It is organized as follows. Section 2.1 proposes a general latent model for multiple temporal clusters. Some probability distributions are assumed for the locations and lengths of the latent cluster intervals. A piecewise uniform distribution is then used to mimic the typical sample data generation process. With a more generalized model, we are able to include a known background function that can adjust for available inhomogeneous background information. Section 2.2 develops detailed model inference procedures assuming the number of clusters is known. The likelihood functions for observed data are derived in section 2.2.1. A Monte-Carlo EM algorithm procedure to estimate the associated model parameters is described in section 2.2.2. Section 2.2.3 introduces Wald and LRT tests for the significant testings of the latent clusters. Robust methods to estimate cluster intervals are proposed in section 2.2.4 when the tests show significance. Since in reality, the number of clusters is rarely known, in section 2.3, AIC and BIC model selection criteria are used to determine the optimal number of total potential clusters. A comprehensive simulation study is provided in section 2.4 to illustrate and evaluate the proposed methodology. The comparison results with the stepwise regression method developed by Molinari et al. [30] are also presented. Section 2.5 contains two real data analysis examples. We reanalyze the Hospital Hemoptysis Admission data studied by Molinari et. al [30]. We also implement our method to monitor potential abrupt increase in brucellosis incidence using data collected by CDC (Centers for Disease Control and Prevention). Further comments and discussions can be found in section 2.6.

2.1 A latent multiple temporal cluster model

Suppose in a given time window $(0, T)$, there are k clusters. Here, k is a known or unknown fixed integer. A temporal latent cluster model is specified in Figure 2.1. Starting from time 0, we wait b_1 units of time for the first cluster which lasts c_1 units of time. After the first cluster, we wait b_2 units of time for the second cluster which lasts c_2 units of time, and so on until the kth cluster appears which lasts c_k units of time. After the kth cluster, b_{k+1} is the waiting period until the next cluster, which occurs after the endpoint T. For simplicity, we assume in Figure 2.1 that the first cluster appears after the starting point 0 and the kth cluster ends before the endpoint T. It is possible that there are clusters before the starting time of the study, and the waiting time between the last cluster before time 0, and the first cluster after time 0 is longer than b_1. In the exponential case (with the memoryless property), the existence of clusters before time 0 does not change the results. In other cases, we may need to model b_1 separately by a truncated $\psi_b(t)$ distribution. It is also possible that either time 0 or T falls within a cluster interval. These changes only affect some formula calculations and the general developments remain the same. We illustrate our methodology with the model in Figure 2.1 to simplify the presentation.

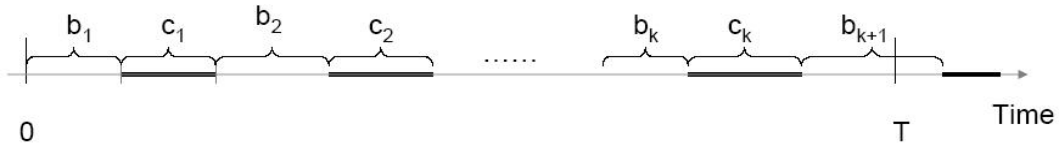

Figure 2.1: An illustrative example of a latent multiple temporal cluster model

We assume that the waiting time periods $b_1, b_2, \ldots, b_{k+1}$ are i.i.d. random samples from a distribution with density function $\psi_b(t) = \psi_b(t; \lambda_b)$ and the cluster interval lengths $c_1, c_2, \ldots, c_k$ are i.i.d. random samples from a distribution with density function $\psi_c(t) = \psi_c(t; \lambda_c)$. Here, λ_b and λ_c are unknown parameters. ψ_b and ψ_c may or may not be from the same distribution family. One simple example that we use later is both $\psi_b(t)$ and $\psi_c(t)$ are exponential density functions with means equal to $1/\lambda_b$ and $1/\lambda_c$ respectively. Denote $\mathbf{b} = (b_1, b_2, \ldots, b_{k+1})'$ and $\mathbf{c} = (c_1, c_2, \ldots, c_k)'$. For convenience, we introduce a random number δ so that $\{\delta = k\}$ is the event that exactly k clusters

occur in the time window $(0, T)$. Clearly, $\{\delta = k\}$ is equivalent to event $\{\sum_{j=1}^{k}(b_j + c_j) + b_{k+1} \geq T$ and $\sum_{j=1}^{k}(b_j + c_j) \leq T\}$. Meanwhile, from Figure 2.1, it is easy to see that $I_j = [\sum_{l=1}^{j}(b_l + c_l) - c_j, \sum_{l=1}^{j}(b_l + c_l)]$ is the jth cluster interval, $j = 1, \cdots, k$.

The latent variables $\mathbf{b}$ and $\mathbf{c}$ are not observed. What we can observe in this model setting are only the time points $y_1, y_2, \ldots, y_n$ when incidences of interest occur. We assume that the observations $y_1, y_2, \ldots, y_n$ are i.i.d. samples from the following piecewise uniform function,

$$f_\theta(y|\mathbf{b}, \mathbf{c}, k) = \begin{cases} \frac{\alpha_1}{T+\sum_{j=1}^{k}(\alpha_j-1)c_j}, & \text{if } y \in I_1 \\ \cdots\cdots & \\ \frac{\alpha_k}{T+\sum_{j=1}^{k}(\alpha_j-1)c_j}, & \text{if } y \in I_k \\ \frac{1}{T+\sum_{j=1}^{k}(\alpha_j-1)c_j}, & \text{if } y \notin \cup_{j=1}^{k} I_j \end{cases} \tag{2.1}$$

where $\theta = (\alpha, \lambda)$ is the collection of all unknown parameters, including the parameters $\alpha = (\alpha_1, \ldots, \alpha_k)'$ and $\lambda = (\lambda_b, \lambda_c)'$ that are associated with random variables b_i's and c_i's. When $k = 1$, the piecewise uniform density function (2.1) becomes the single piecewise uniform density function used for the single cluster case; See, e.g., Chapter 14 of Glaz, et al. [20]. The parameters $\alpha_j \geq 0$ for each $j = 1, 2, ..., k$, and may or may not be the same across the k clusters. Under the density assumption (2.1), the incidence is α_j times more likely to happen inside the jth cluster than that outside the clusters. The case with $\alpha_j > 1$ corresponds to a denser cluster of more incidences. The case with $\alpha_j < 1$ corresponds to a sparser cluster of less incidences and the case with $\alpha_j = 1$ corresponds to non cluster. When k = 1, the piecewise uniform density function (2.1) becomes the single piecewise uniform density function used for a single cluster case in Naus [32], Nargarwalla [31] and many others. If we want to see whether there are any significant clusters in the data, we can test a hypothesis H_0: $\alpha_1 = \alpha_2 = ... = \alpha_k = 1$ versus H_1: at least one $\alpha_j \neq 1$.

The proposed model can be alternatively expressed in terms of Poisson models,

similar to those used in Gangnon and Clayton [16] and others. We use the current formulation of a piecewise uniform function in order to highlight the interpretation of the parameters α_i's. The proposed model is also closely related to a Bayesian model. In particular, if we further assume that the parameters α_i's are random variables with a proper assumed prior distribution, the proposed model would correspond to a Bayesian hierarchical model. Here, we use the frequentist formulation, since it can utilize the fully developed likelihood inference and avoid choosing priors. Although we illustrate our latent model for temporal data, the model also covers other types of data, for example, patterns or events in a sequence such as the DNA data studied by Leung et. al [28].

Sometimes, we want to analyze the observed data with the adjustment of the inhomogeneous background information. As mentioned in Molinari et. al [30] and Wallenstein and Naus [53], the background value, such as seasonal patterns or population sizes, may not be the same across the time window (0,T). We realize this via a more generalized piecewise uniform function, which includes a known background function $W(t)$. The known background function $W(t)$ is usually assessed from separated sources. It can be easily incorporated into model (2.1). In this case, we replace the density function (2.1) by

$$f_\theta(y|\mathbf{b},\mathbf{c},k) = \begin{cases} \frac{\alpha_1 W(y)}{\tilde{T}+\sum_{j=1}^{k}(\alpha_j-1)\tilde{c}_j}, & \text{if } y \in I_1 \\ \qquad\qquad \cdots\cdots & \\ \frac{\alpha_k W(y)}{\tilde{T}+\sum_{j=1}^{k}(\alpha_j-1)\tilde{c}_j}, & \text{if } y \in I_k \\ \frac{W(y)}{\tilde{T}+\sum_{j=1}^{k}(\alpha_j-1)\tilde{c}_j}, & \text{if } y \notin \cup_{j=1}^{k} I_j \end{cases} \tag{2.2}$$

where $\tilde{T} = \int_0^T W(t)dt$ and $\tilde{c}_j = \int_{I_j} W(t)dt$, for $j = 1, 2, \ldots, k$. This generalized model is equivalent to model (2.1) when $W(t) \propto 1$. Fitting model (2.2) is exactly the same as fitting model (2.1), except that T and c_j need to be replaced by $\tilde{T}$ and $\tilde{c}_j$. To simplify our presentation, in the next few sections, we develop a Monte-Carlo EM algorithm and likelihood inference for data generated from model (2.1).

2.2 Model inference for given number of clusters

In order to avoid an over fitting problem of too many parameters, the number of clusters k needs to be bounded away from n. We will apply model selection techniques to determine this number later in section 2.3. Currently, we assume k is known. Our concern in this section is to make model inference for given number of clusters.

2.2.1 Likelihood function of observed data

In the latent model illustrated in Figure 2.1, the probability of the event that k clusters exist in the time window $[0, T]$ can be computed by

$$\begin{aligned} P_\lambda(\delta = k) &= P_\lambda\{\sum_{j=1}^{k}(b_j + c_j) + b_{k+1} \geq T \text{ and } \sum_{j=1}^{k}(b_j + c_j) \leq T\} \\ &= \int_0^T \int_{T-s}^{\infty} \psi_b(t)\psi_{bc}^{[k]}(s)dtds. \end{aligned} \tag{2.3}$$

where $\psi_{bc}^{[k]}(s)$ is the density function of $\sum_{j=1}^{k}(b_j + c_j)$. We have $\psi_{bc}^{[k]} = \psi_b^{[k]} * \psi_c^{[k]}$, $\psi_b^{[k]} = \psi_b * \ldots * \psi_b$ and $\psi_c^{[k]} = \psi_c * \ldots * \psi_c$, where $*$ means convolution, and both $\psi_b^{[k]}$ and $\psi_c^{[k]}$ involve a total of k functions. Conditional on $\delta = k$, the joint conditional likelihood function of $(\mathbf{b}, \mathbf{c})$ is

$$f_\theta(\mathbf{b}, \mathbf{c}|k) = \frac{\prod_{j=1}^{k}\{\psi_b(b_j)\psi_c(c_j)\}\psi_b(b_{k+1})\mathbf{1}_{\{\delta=k\}}}{P_\lambda(\delta = k)}. \tag{2.4}$$

Here, $\mathbf{1}_{\{C\}}$ equals 1 if set C is true and 0 otherwise.

In the special case of ψ_b and ψ_c being density functions of exponential distributions $\text{Exp}(\lambda_b)$ and $\text{Exp}(\lambda_c)$, we have $\psi_b^{[k]}(t) = \lambda_b^k t^{k-1}e^{-\lambda_b t}/\Gamma(k)$ and

$$\psi_{bc}^{[k]}(t) = \frac{(\lambda_b\lambda_c)^k e^{-\lambda_c t}\int_0^t\{z(t-z)\}^{k-1}e^{-(\lambda_b-\lambda_c)z}dz}{\{\Gamma(k)\}^2}.$$

Therefore, in this case,

$$P_\lambda(\delta = k) = \int_0^T e^{-\lambda_b(T-s)}\psi_{bc}^{[k]}(s)ds = \frac{(\lambda_b\lambda_c)^k e^{-\lambda_b T}}{k\{\Gamma(k)\}^2}G_k(\lambda_b - \lambda_c) \tag{2.5}$$

and

$$f_\theta(\mathbf{b}, \mathbf{c}|k) = \frac{k\{\Gamma(k)\}^2\lambda_b e^{-\lambda_b(\sum_{j=1}^{k+1} b_j - T) - \lambda_c\sum_{j=1}^{k} c_j}\mathbf{1}_{\{\delta=k\}}}{G_k(\lambda_b - \lambda_c)}, \tag{2.6}$$

where the function $G_k(u) = T^{2k}\tilde{G}_k(Tu)$ with $\tilde{G}_k(u) = \int_0^1 (1-t)^k t^{k-1} e^{ut} dt$.

From model (2.1), the conditional joint density function of the data observations $\mathbf{y} = (y_1, y_2, \ldots, y_n)$, conditional on $\mathbf{b}$, $\mathbf{c}$ and $\delta = k$, is

$$f_\theta(\mathbf{y}|\mathbf{b}, \mathbf{c}, k) = \prod_{i=1}^{n} f(y_i|\mathbf{b}, \mathbf{c}, k) = e^{\sum_{j=1}^{k}(\log \alpha_j) Z_j - n \log\{T + \sum_{j=1}^{k}(\alpha_j - 1)c_j\}}, \tag{2.7}$$

where $Z_j = Z_j(\mathbf{y}, \mathbf{b}, \mathbf{c}) = \sum_{i=1}^{n} \mathbf{1}_{\{y_i \in I_j\}}$ is the number of incidences that occur within the jth cluster interval. Thus, the joint density function of $\mathbf{y}$ and $\delta = k$, is

$$\begin{aligned} f_\theta(\mathbf{y}, k) &= \int\int f_\theta(\mathbf{y}, \mathbf{b}, \mathbf{c}|k) P_\lambda(\delta = k) d\mathbf{b} d\mathbf{c} \\ &= \int\int f_\theta(\mathbf{y}|\mathbf{b}, \mathbf{c}, k) f_\theta(\mathbf{b}, \mathbf{c}|k) P_\lambda(\delta = k) d\mathbf{b} d\mathbf{c} \end{aligned} \tag{2.8}$$

and the log-likelihood function of observing $\mathbf{y}$ and $\delta = k$ is

$$\ell_k(\theta|\mathbf{y}) = \log\{f_\theta(\mathbf{y}, k)\}. \tag{2.9}$$

Since form (2.9) involves multiple integrations, it is complicated to directly compute the log-likelihood function $\ell_k(\theta|\mathbf{y})$, its first and second derivatives. As a result, it is hard to obtain the maximum likelihood estimates by directly maximizing the likelihood function. We instead develop a Monte-Carlo EM algorithm to estimate the model parameters in the next subsection.

2.2.2 Monte-Carlo EM algorithm for model estimation

We note that the joint density function of $(\mathbf{y}, \mathbf{b}, \mathbf{c}, \delta = k)$ is explicit,

$$\begin{aligned} & f_\theta(\mathbf{y}, \mathbf{b}, \mathbf{c}, k) = f_\theta(\mathbf{y}|\mathbf{b}, \mathbf{c}, k) f_\theta(\mathbf{b}, \mathbf{c}|k) P_\lambda(\delta = k) \\ = \; & e^{\sum_{j=1}^{k}(\log \alpha_j) Z_j - n \log[T + \sum_{j=1}^{k}(\alpha_j - 1)c_j]} \prod_{j=1}^{k} \{\psi_b(b_j)\psi_c(c_j)\} \psi_b(b_{k+1}) \mathbf{1}_{\{\delta = k\}} \end{aligned} \tag{2.10}$$

Instead of directly maximizing the log-likelihood function $\ell_k(\theta|\mathbf{y})$, we propose an EM algorithm to solve the estimation problem, in which we treat $(\mathbf{y}, \mathbf{b}, \mathbf{c}, \delta = k)$ as the complete responses and $(\mathbf{y}, \delta = k)$ as the observed responses.

From the standard procedure of deriving EM algorithm, we have the following EM algorithm to obtain parameter estimates.

Step 1. Select a starting point $\theta^{(0)} = (\alpha^{(0)}, \lambda^{(0)})$ of $\theta = (\alpha, \lambda)$.

Step 2. (E-step) For a given $\theta^{(s)}$ at the sth iteration, $s = 0, 1, 2, \ldots$, calculate

$$Q(\theta|\theta^{(s)}) = Q_1(\alpha|\theta^{(s)}) + Q_2(\lambda|\theta^{(s)}), \tag{2.11}$$

where

$$Q_1(\alpha|\theta^{(s)}) = \sum_{j=1}^{k} \mathrm{E}(Z_j|\mathbf{y}, k, \theta^{(s)}) \log \alpha_j - n\mathrm{E}\{\log[T + \sum_{j=1}^{k}(\alpha_j - 1)c_j]|\mathbf{y}, k, \theta^{(s)}\}, \tag{2.12}$$

$$Q_2(\lambda|\theta^{(s)}) = \sum_{j=1}^{k+1} \mathrm{E}\{\log \psi_b(b_j)|\mathbf{y}, k, \theta^{(s)}\} + \sum_{j=1}^{k} \mathrm{E}\{\log \psi_c(c_j)|\mathbf{y}, k, \theta^{(s)}\}. \tag{2.13}$$

Step 3. (M-step) For each $s = 0, 1, 2, \ldots$, update the parameter estimates, $\theta^{(s+1)} = (\alpha^{(s+1)}, \lambda^{(s+1)})$, by maximizing the following functions,

$$\alpha^{(s+1)} = \operatorname{argmax} Q_1(\alpha|\theta^{(s)}), \quad \text{and} \quad \lambda^{(s+1)} = \operatorname{argmax} Q_2(\lambda|\theta^{(s)}). \tag{2.14}$$

In the case with ψ_b and ψ_c being density functions of exponential distributions $\mathrm{Exp}(\lambda_b)$ and $\mathrm{Exp}(\lambda_c)$, the updating formula of $\lambda^{(s+1)}$ is simply $\lambda_b^{(s+1)} = (k+1)/\sum_{j=1}^{k+1} \mathrm{E}(b_j|\mathbf{y}, k, \theta^{(s)})$ and $\lambda_c^{(s+1)} = k/\sum_{j=1}^{k} \mathrm{E}(c_j|\mathbf{y}, k, \theta^{(s)})$.

Step 4. Repeat steps 2 and 3 until $\|\theta^{(s+1)} - \theta^{(s)}\|$ is very small; that is, until the algorithm numerically converges.

The conditional expectations in the E-step do not usually have explicit form. We show below how to simulate from the (fully) conditional distributions of b_j or c_j given the rest of b's and c's. The conditional expectations in the E-step can then be computed by a Gibbs sampling approach.

Suppose $\mathbf{b}^* = (b_1^*, \ldots, b_{k+1}^*)'$ and $\mathbf{c}^* = (c_1^*, \ldots, c_k^*)'$ are a set of Gibbs samples from $f(\mathbf{b}, \mathbf{c}|\mathbf{y}, k, \theta^{(s)})$. They are generated many times by cycling through simulations from the fully conditional distributions of b_j or c_j given all the other b's and c's until the Gibbs sampling chain is "burn-in". Repeat the Gibbs sampling chain a large number of times to get M sets of Gibbs samples. The four conditional expectations in the E-step of the EM-algorithm can be evaluated by $\frac{1}{M}\sum_* Z_j^*$, $\frac{1}{M}\sum_* \log\{T + \sum_{j=1}^{k}(\alpha_j - 1)c_j^*\}$, $\frac{1}{M}\sum_* \log\{\psi(b_j^*)\}$, and $\frac{1}{M}\sum_* \log\{\psi(c_j^*)\}$, respectively, where, $\sum_*$ is the summation over the M sets of Gibbs samples $\mathbf{b}^*$ and $\mathbf{c}^*$. Z_j^* is the total number of incidences computed

with b_j^* and c_j^* in each of the Gibbs sample sets. In the exponential case, the last two expectations are just $\frac{1}{M}\sum_* b_j^*$, and $\frac{1}{M}\sum_* c_j^*$.

To carry out the EM computation, the only remaining question is how to simulate a b_j or c_j from the fully conditional distributions given the rest of b_j's and c_j's. By ignoring unwanted terms, it is easy to see that, for $j = 1, 2, \ldots, k$,

$$f(b_j|b_l, l = 1, 2, \ldots, k+1, l \neq j, \mathbf{c}, \mathbf{y}, k) \propto f(\mathbf{b}, \mathbf{c}, \mathbf{y}|k) \propto e^{\sum_{s=j}^{k} Z_s(\log \alpha_s)} \psi_b(b_j) \mathbf{1}_{(\delta=k)}, \tag{2.15}$$

and $f(b_{k+1}) \propto \psi_b(b_{k+1})\mathbf{1}_{(\delta=k)}$. Similarly, we have

$$f(c_j|c_l, l = 1, 2, \ldots, k, l \neq j, \mathbf{b}, \mathbf{y}, k) \propto \frac{e^{\sum_{s=j}^{k} Z_s(\log \alpha_s)}}{[T + \sum_{s=1}^{k}(\alpha_s - 1)c_s]^n} \psi_c(c_j)\mathbf{1}_{(\delta=k)}. \tag{2.16}$$

Thus, given a set of parameters $\theta = (\alpha, \lambda)$, we can use the following importance sampling method to simulate a b_j:

Step A. Simulate a large number of random deviates $e_1, e_2, \ldots, e_N$ from a candidate distribution $\tilde{\psi}_b(b_j)$. Then, compute weight $w_l = e^{\sum_{s=j}^{k} Z_s^{[l]}(\log \alpha_s)} \frac{\psi_b(e_l)}{\tilde{\psi}_b(e_l)} \mathbf{1}_{(\delta^{[l]}=k)}$, for $l = 1, 2, \ldots, N$, where $Z_s^{[l]}$ is the total number of incidence in sth cluster and $\{\delta^{[l]} = k\}$ is the constraint of having k clusters with b_j replaced by e_l and the rest of b's and c's the same. In the case of simulating b_{k+1} given the rest b's and c's, the weight can be simplified to $w_l = \{\psi_b(e_l)/\tilde{\psi}_b(e_l)\}\mathbf{1}_{(\delta^{[l]}=k)}$.

Step B. Simulate b_j from one of the N values $\{e_1, e_2, \ldots, e_N\}$ with respective probabilities $(p_1, p_2, \ldots, p_N)$, where $p_l = w_l / \sum_{s=1}^{N} w_s$.

Similarly, we can simulate a c_j from the fully conditional probability $f(c_j|c_l, l = 1, 2, \ldots, k, l \neq j, \mathbf{b}, \mathbf{y}, \delta = k)$. In this case, with simulating random derivates $e_1, e_2, \ldots, e_N$ from a candidate distribution $\tilde{\psi}_c(c_j)$, the weight

$$w_l = \frac{e^{\sum_{s=j}^{k} Z_s^{[l]}(\log \alpha_s)}}{[T + \sum_{s \neq j}(\alpha_s - 1)c_s + (\alpha_j - 1)e_l]^n} \frac{\psi_c(e_l)}{\tilde{\psi}_c(e_l)} \mathbf{1}_{(\delta^{[l]}=k)}.$$

Here, again, $Z_s^{[l]}$ and $\{\delta^{[l]} = k\}$ are computed with given b and c values and the c_j is replaced by e_l.

In the special exponential case, $\psi_b(b_j) \sim \text{Exp}(\lambda_b)$ and $\psi_c(c_j) \sim \text{Exp}(\lambda_c)$. Since it is easy to directly simulate from a truncated exponential distribution, we suggest to pick $\tilde{\psi}_b(b_j) \propto \text{Exp}(\lambda_b)\mathbf{1}(\delta = k)$ and $\tilde{\psi}_c(c_j) \propto \text{Exp}(\lambda_c)\mathbf{1}(\delta = k)$.

The above EM algorithm does not provide the variance-covariance matrix calculation for the parameter estimators. To obtain an estimator of the variance-covariance matrix, we use the missing information principle and Louis's method [48]. In particular, the information matrix is,

$$\begin{aligned} H_n &\stackrel{d}{=} -\left\{\frac{\partial^2}{\partial\theta^2}\ell_k(\theta|\mathbf{y})\right\} \\ &= -E\left\{\frac{\partial^2}{\partial\theta^2}\ell_k(\theta|\mathbf{b},\mathbf{c},\mathbf{y})|\mathbf{y},\delta=k\right\} - \mathrm{Var}\left\{\frac{\partial}{\partial\theta}\ell_k(\theta|\mathbf{b},\mathbf{c},\mathbf{y})|\mathbf{y},\delta=k\right\} \quad (2.17) \end{aligned}$$

where $\ell_k(\theta|\mathbf{b},\mathbf{c},\mathbf{y}) = \log\{f_\theta(\mathbf{y},\mathbf{b},\mathbf{c},k)\}$. The formula for the observed information matrix does not have an explicit form. However, it can be numerically estimated by [48]

$$\begin{aligned} \hat{H}_n &= -\frac{1}{M}\sum\frac{\partial^2}{\partial\theta^2}\ell_k(\theta|\mathbf{b}^*,\mathbf{c}^*,\mathbf{y}) \\ &\quad -\left[\frac{1}{M}\sum\left\{\frac{\partial}{\partial\theta}\ell_k(\theta|\mathbf{b}^*,\mathbf{c}^*,\mathbf{y})\right\}\left\{\frac{\partial}{\partial\theta}\ell_k(\theta|\mathbf{b}^*,\mathbf{c}^*,\mathbf{y})\right\}'\right. \\ &\quad \left.-\left\{\frac{1}{M}\sum\frac{\partial}{\partial\theta}\ell_k(\theta|\mathbf{b}^*,\mathbf{c}^*,\mathbf{y})\right\}\left\{\frac{1}{M}\sum\frac{\partial}{\partial\theta}\ell_k(\theta|\mathbf{b}^*,\mathbf{c}^*,\mathbf{y})\right\}'\right] \quad (2.18) \end{aligned}$$

where the summations are over the M sets of Gibbs samples $\mathbf{b}^*$ and $\mathbf{c}^*$ obtained in the final round of the EM algorithm.

2.2.3 Likelihood inference for significance testings

In this subsection, we illustrate inference for two sided tests related to α's. Detailed comments on extensions to one sided tests, sometimes straightforward and sometimes more complex, can be found in Chapter 4.

Let us first consider to test a single (jth) cluster and see whether it is significant or not, i.e., $H_0 : \alpha_j = 1$ versus $H_1 : \alpha_j \neq 1$. Let $\hat{\alpha}_j$ be the estimator of the parameter α_j. From the observed information matrix, we can get an estimator of the standard deviation of $\hat{\alpha}_j$, $se(\hat{\alpha}_j)$. We can construct a Wald-type t statistic $t = (\hat{\alpha_j} - 1)/se(\hat{\alpha_j})$ for this test. When n is large, t is asymptotically normally distributed and we can use a two-sided z test to justify whether $\alpha_1 = 1$ or not.

Another interesting problem is to check whether at least one significant cluster exists or not among the k potential clusters, i.e., we are interested in the test $H_0 : \alpha_1 = \alpha_2 =$

... $= \alpha_k = 1$ versus H_1 : at least one $\alpha_j \neq 1$. We propose to use a likelihood ratio test for this problem. According to likelihood inference, twice log likelihood ratio test statistic is

$$R = 2\log\{\frac{\max_{H_1} f_\theta(\mathbf{y},k)}{\max_{H_0} f_\theta(\mathbf{y},k)}\} = 2\ell_k(\hat{\theta}|\mathbf{y})|_{\theta=\hat{\theta}} + 2n\log(T) - 2\max_{\lambda}\log P_\lambda(\delta=k) \tag{2.19}$$

$$= 2\{\log\int\int f_{\hat{\theta}}(\mathbf{y}|\mathbf{b},\mathbf{c},k) f_{\hat{\theta}}(\mathbf{b},\mathbf{c}|k)d\mathbf{b}d\mathbf{c} + \log P_{\hat{\lambda}}(\delta=k) + n\log(T) - \max_{\lambda}\log P_\lambda(\delta=k)\}$$

where $\hat{\theta} = (\hat{\alpha},\hat{\lambda})$ are the estimates of the parameters obtained from the aforementioned EM algorithm under H_1. Suppose for a moment, we know how to simulate $\mathbf{b}^{**} = (b_1^{**},\ldots,b_{k+1}^{**})$ and $\mathbf{c}^{**} = (c_1^{**},\ldots,c_k^{**})$ from $f(\mathbf{b},\mathbf{c}|k)$ when $\theta = \hat{\theta}$ and we have M sets of such simulated $\mathbf{b}^{**}$ and $\mathbf{c}^{**}$ samples. By Monte-Carlo approximation, the test statistic R can be approximated by

$$R^{**} = 2\Bigg[\log\{\frac{1}{M}\sum_{**} f(\mathbf{y}|\mathbf{b}^{**},\mathbf{c}^{**},k)\} + \log P_{\hat{\lambda}}(\delta=k) + n\log(T) - \max_{\lambda}\log P_\lambda(\delta=k)\Bigg], \tag{2.20}$$

where $\sum_{**}$ is the summation over the M sets of $\mathbf{b}^{**}$ and $\mathbf{c}^{**}$ samples. Based on likelihood inference, we know that R is asymptotically χ^2 distributed with k degrees of freedom. Comparing R^{**} with χ^2_k distribution, we can perform a formal test for $H_0 : \alpha_1 = \alpha_2 = ... = \alpha_k = 1$ versus H_1 : at least one $\alpha_j \neq 1$.

The question that remains is how to simulate $\mathbf{b}^{**} = (b_1^{**},\ldots,b_{k+1}^{**})$ and $\mathbf{c}^{**} = (c_1^{**},\ldots,c_k^{**})$ from $f(\mathbf{b},\mathbf{c}|k)$ with any set of given parameter values θ. Again, we turn to the Gibbs sampling approach method. The Gibbs sampling procedure suggests that when we cycle through simulating b_j or c_j from $f_\theta(b_j|b_l, l=1,2,\ldots,k+1, l\neq j,\mathbf{c},\delta=k)$ or $f_\theta(c_j|c_l, l=1,2,\ldots,k, l\neq j,\mathbf{b},\delta=k)$ many times, we can get a set of $\mathbf{b}^{**}$ and $\mathbf{c}^{**}$ from $f(\mathbf{b},\mathbf{c}|k)$. Repeat the procedure M-times to get M sets of simulated $\mathbf{b}^{**}$ and $\mathbf{c}^{**}$ samples. Note that,

$$f_\theta(b_j|b_l, l=1,2,\ldots,k+1, l\neq j,\mathbf{c},\delta=k) \propto \psi_b(b_j)\mathbf{1}_{(\delta=k)} \tag{2.21}$$

and

$$f_\theta(c_j|c_l, l=1,2,\ldots,k, l\neq j,\mathbf{b},\delta=k) \propto \psi_c(c_j)\mathbf{1}_{(\delta=k)}. \tag{2.22}$$

They are truncated distributions. In most cases, for example $\psi_b(b_j)$ and $\psi_c(c_j)$ being

exponential distributions, the truncated distribution can be directly simulated. Otherwise, we can use an importance sampling algorithm to obtain $\mathbf{b}^{**}$ and $\mathbf{c}^{**}$ samples, assuming we know how to simulate from $\psi_b(b_j)$ and $\psi_c(c_j)$.

2.2.4 Identification of cluster intervals

If a cluster is significant (i.e. $\alpha_j \neq 1$), we usually want to determine the location and size of the cluster. Note that the lower and upper bounds of the jth cluster interval I_j are respectively $\mathrm{L}_j = \sum_{l=1}^{j}(b_l + c_l) - c_j$ and $\mathrm{U}_j = \sum_{l=1}^{j}(b_l + c_l)$. Their conditional expectations given $\mathbf{y}$ and k ("posterior mean") are $\mathrm{E}\{\mathrm{L}_j|\mathbf{y}, k\} = \sum_{l=1}^{j}\mathrm{E}(b_l|\mathbf{y}, k) + \sum_{l=1}^{j}\mathrm{E}(c_l|\mathbf{y}, k) - \mathrm{E}(c_j|\mathbf{y}, k)$ and $\mathrm{E}\{\mathrm{U}_j|\mathbf{y}, k\} = \sum_{l=1}^{j}\mathrm{E}(b_l|\mathbf{y}, k) + \sum_{l=1}^{j}\mathrm{E}(c_l|\mathbf{y}, k)$. The bounds L_j and U_j can be simply estimated by $\frac{1}{M}\sum_{l=1}^{j}\sum_{*} b_l^* + \frac{1}{M}\sum_{l=1}^{j}\sum_{*} c_l^* - \frac{1}{M}\sum_{*} c_j^*$ and $\frac{1}{M}\sum_{l=1}^{j}\sum_{*} b_l^* + \frac{1}{M}\sum_{l=1}^{j}\sum_{*} c_l^*$ respectively, where $\sum_{*}$ is the summation over the M sets of Gibbs samples in the last iteration of the EM algorithm.

An alternative approach is to compute $\mathrm{L}_j^* = \sum_{l=1}^{j}(b_l^* + c_l^*) - c_j^*$ and $\mathrm{U}_j^* = \sum_{l=1}^{j}(b_l^* + c_l^*)$ for each set of Gibbs sample set. The medians of the M values of L_j^* and U_j^* can be used to estimate L_j and U_j, respectively. Note that since the distribution may not be symmetric, this median method may provide more accurate estimators.

We can also obtain confidence intervals for the bounds L_j and U_j. We increasingly sort these M values. For $0 < \beta < 1$, the $M\beta/2$th and $M(1-\beta/2)$th values forms a $(1-\beta)\%$ confidence interval for L_j. The same method applies for the upper bound U_j.

We are able to assess the performance of these cluster interval estimators only in simulation studies. In the simulation studies presented in Section 2.4, we employ four empirical statistics to access the accuracy of the estimated cluster intervals. They are sensitivity, specificity, positive predictive value (PPV) and negative predictive value (NPV). Here, sensitivity is the proportion of the event points (y's) inside the true clusters, are inside the estimated clusters. Specificity is the proportion of the event points (y's) outside the true clusters, are outside the estimated clusters. PPV is the proportion of the event points (y's) inside the estimated clusters, are inside the true clusters. NPV is the proportion of the event points (y's) outside the estimated clusters, are outside the true clusters. Table 2.1 shows the relations of these four measures. The

closer they are to one, the more accurate the estimated cluster intervals are.

Table 2.1: Definitions of Sensitivity, Specificity, PPV and NPV

		True Condition		
		In(True)	*Out(False)*	
Estimate	*In(Positive)*	True Positive(TP)	False Positive(FP)	PPV=$\frac{TP}{\text{Positive}}$
Outcome	*Out(Negative)*	False Negative(FN)	True Negative(TN)	NPV=$\frac{TN}{\text{Negative}}$
		Sensitivity=$\frac{TP}{\text{True}}$	Specificity=$\frac{TN}{\text{False}}$	

2.3 Determination of the unknown number of clusters

In the previous section, we assume that the number of clusters is known. It is rarely true in reality. We now describe a model selection approach to determine the number of clusters from the observed data. We propose to use both AIC and BIC criteria. The AIC criterion [2] is a commonly used model selection method developed based on the Kullback-Leibler information between the candidate models and the true model. Schwarz [46] obtains the BIC procedure by using Bayes estimators and a fixed penalty for choosing the wrong dimension. Both criteria minimize an expression that consists of a term that measures model fit plus a term that penalizes model complexity. In our context, a direct application of the AIC and BIC rules yields

$$\begin{aligned} & \mathrm{AIC}(k) = -2\log f_\theta(\mathbf{y}, k) + 2k \\ = \; & -2\log\{\int\int f_\theta(\mathbf{y}|\mathbf{b},\mathbf{c},k) f_\theta(\mathbf{b},\mathbf{c}|k) d\mathbf{b} d\mathbf{c}\} - 2\log P_\lambda(\delta = k) + 2k \end{aligned} \quad (2.23)$$

and

$$\begin{aligned} & \mathrm{BIC}(k) = -2\log f_\theta(\mathbf{y}, k) + k\log(n) \\ = \; & -2\log\{\int\int f_\theta(\mathbf{y}|\mathbf{b},\mathbf{c},k) f_\theta(\mathbf{b},\mathbf{c}|k) d\mathbf{b} d\mathbf{c}\} - 2\log P_\lambda(\delta = k) + k\log(n) \end{aligned} \quad (2.24)$$

Often $n > e^2 = 7.389$, the BIC method places more penalty against a large number of clusters than the AIC method.

The parameters θ are unknown. To compute the criteria, these parameters should be replaced by their estimators $\hat{\theta} = \hat{\theta}(k)$ that can be obtained by the Monte-Carlo

EM algorithm proposed in the previous section. Furthermore, the formula involves integrations that do not have an explicit form. We numerically evaluate their values. From the previous section, we know how to simulate $\mathbf{b}^{**} = (b_1^{**}, \ldots, b_{k+1}^{**})$ and $\mathbf{c}^{**} = (c_1^{**}, \ldots, c_k^{**})$ from $f(\mathbf{b}, \mathbf{c}|k)$ when $\theta = \hat{\theta}$. By Monte-Carlo approximation, the AIC(k) criterion can be approximated by

$$\widehat{\text{AIC}}(k) = -2\log\Big\{\frac{1}{M}\sum_{**} f(\mathbf{y}|\mathbf{b}^{**}, \mathbf{c}^{**}, k)\Big\} - 2\log P_{\hat{\lambda}}(\delta = k) + 2k, \tag{2.25}$$

and BIC(k) criterion can be approximated by

$$\widehat{\text{BIC}}(k) = -2\log\Big\{\frac{1}{M}\sum_{**} f(\mathbf{y}|\mathbf{b}^{**}, \mathbf{c}^{**}, k)\Big\} - 2\log P_{\hat{\lambda}}(\delta = k) + k\log(n), \tag{2.26}$$

where $\sum_{**}$ is the summation over M sets of repeatedly simulated b^{**}'s and c^{**}'s from $f_{\hat{\theta}}(\mathbf{b}, \mathbf{c}|k)$. The k selected is the one with the smallest corresponding $\widehat{\text{AIC}}(k)$ or $\widehat{\text{BIC}}(k)$ value.

Denote $\mathcal{K}$ as a pre-selected set of k's. We want this set small for computing purpose but large enough to cover all potential choices of the correct number of clusters. Based on Subsections 2.2.1-2.2.4, a practical approach to detect clusters emerges:

- For each fixed $k \in \mathcal{K}$, apply the Monte-Carlo EM algorithm in Subsection 2.2.2 to obtain the parameter estimates. Then use either AIC or BIC rule to determine the optimal number of clusters k.

- For the chosen k, use the results in Subsections 2.2.3-2.2.4 to detect and determine the cluster intervals.

2.4 Simulation studies

In this section, we perform simulation studies in two designed settings. In the first setting, the cluster intervals are fixed. Only the observations (the time records of incidences y's) are randomly generated. In the second setting, the cluster intervals are randomly simulated from latent exponential distribution models. Both cluster intervals and the observations are randomly generated. We demonstrate through the simulation results the performance of the proposed method under these two settings. We also compare the results obtained by our method with those obtained by the stepwise regression

method [30]. Without loss of generality, all the simulation studies are done within the time window $(0, 1)$.

2.4.1 Setting I

Let us first consider the single cluster case ($k = 1$) in the setting of fixed cluster interval. In particular, we fix $\mathbf{b} = (b_1, b_2)' = (.258, 1.209)', \mathbf{c} = c_1 = .236$ and thus the single cluster at $[.258, .494]$. With an $\alpha = 3$, we simulate $n = 100$ independently identically distributed time points $y_1, y_2, \ldots, y_{100}$ according to model (2.1). Assume we only know these 100 y values and apply the model inference procedures developed in Section 2.2 with $k = 1$. We can get a set of parameter estimates, test whether this potential cluster is significant or not, and identify its location and size. This simulation exercise is repeated 300 times.

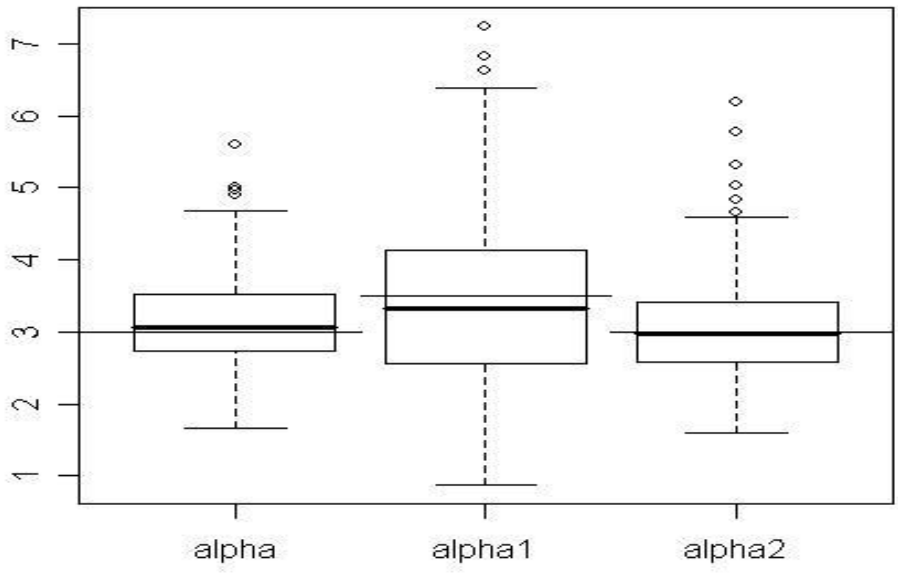

Figure 2.2: Box Plots for α_s estimates in setting I

We also consider the two clusters case ($k = 2$) in the setting of fixed cluster intervals. In particular, with $\mathbf{b} = (b_1, b_2, b_3)' = (0.244, 0.159, 1.119)', \mathbf{c} = (c_1, c_2)' = (0.075, 0.272)'$, we fix the cluster intervals at $[0.244, 0.319]$ and $[0.478, 0.750]$. We then

choose $\alpha_1 = 3.5, \alpha_2 = 3$, and simulate $n = 150$ independently identically distributed time points $y_1, y_2, \ldots, y_{150}$ according to model (2.1). Assuming we only know these 150 y values and given $k = 2$, we can get a set of parameter estimates, test whether a significant cluster exists or not, and identify the locations and sizes of the potential clusters. Again, this simulation exercise is repeated 300 times.

Figure 2.2 provides the box plots of the estimates of the main parameters α's in the two simulation cases. The first one is for the single cluster ($k = 1$) case and the other two are for α_1 and α_2 in the two clusters ($k = 2$) case. Clearly, the centers (medians) of the 300 parameter estimates are all near the respective targets $\alpha = 3.0$, and $(\alpha_1, \alpha_2) = (3.5, 3.0)$, respectively. This indicates that the Monte-Carlo EM algorithm can provide reasonable estimates for the parameters α.

To assess the performance of the likelihood based tests, including both the Wald and likelihood ratio tests, on cluster detection, we examine the powers and type I errors of their level 0.05 tests. For power evaluation, we use the above results. We list in Table 2.2, for 300 repeated simulations, the percentage of times our method succeeds (rejects H_0) and fails (accepts H_0) to detect clusters. To evaluate the type I errors, we simulate another 300 data sets of 100 time points from Uniform(0,1) distribution (in the $k = 1$ case) and 300 data sets of 150 time points from the Uniform(0,1) distribution (in the $k = 2$ case). We apply the same estimation and testing procedures to these two groups of 300 data sets with $k = 1$ and $k = 2$ respectively. Then, we record in Table 2.2 the percentage of times the method claims to find significant clusters. In the single cluster case, both the Wald and LRT tests have over 99% powers to detect the cluster. The type I error of the Wald test is inflated at .093 and the LRT test is conservative at .03.

Table 2.2: Power and Type I Error evaluation in setting I

$k = 1$						$k = 2$							
Power			Type I Error			Power				Type I Error			
Wald	LRT	SR	Wald	LRT	SR	Wald		LRT	SR	Wald		LRT	SR
						1	2			1	2		
99.7%	99.3%	91.0%	.093	.03	.167	86%	99%	99%	64.3%	.03	.053	.01	.26

In the two clusters case, the powers for Wald test are 86% and 99% with controlled type I errors .03 and .053. The LRT test for whether there exists at least one significant cluster has a high power 99% with conservative type I error .01.

Also included in Table 2.2 are the the powers and type I errors of the stepwise regression method [4, 30]. The same $2 \times 2 \times 300 = 1,200$ simulated data sets are analyzed via the R codes released to us by Dr. Molinari. In the single cluster case, the stepwise regression method has a power 91.0% with an inflated type I error .167. In the two clusters case, the power to detect one or more clusters is about 64.3% with an inflated type I error .26. It appears that for the particular type of data from model (2.1), the model based likelihood test approach has higher powers and lower type I errors to detect significant clusters.

Table 2.3: Cluster interval estimates evaluation in setting I (%)

	Method	Statistics	Min	1^{st} QT	Median	3^{rd} QT	Max	Mean
$k = 1$	Mean based	Sensitivity	58.33	92.94	98.11	100	100	95.60
		Specificity	43.40	94.12	98.00	100	100	95.35
		PVP	61.04	94.12	98.00	100	100	95.49
		NPV	79.10	94.06	98.08	100	100	96.44
	Median based	Sensitivity	69.77	94.12	98.15	100	100	96.19
		Specificity	43.40	94.06	98.04	100	100	95.21
		PPV	61.04	94.29	98.11	100	100	95.44
		NPV	79.69	95.06	98.15	100	100	96.93
	SR	Sensitivity	62.22	85.10	96.08	98.20	100	91.62
		Specificity	39.66	82.07	96.08	100.00	100	90.18
		PPV	48.15	81.59	96.08	100.00	100	90.24
		NPV	67.24	87.61	96.30	98.23	100	92.44
$k = 2$	Mean based	Sensitivity	39.77	93.62	97.14	99.02	100	95.13
		Specificity	39.06	85.00	93.10	98.02	100	89.57
		PPV	62.16	91.11	95.79	98.87	100	93.83
		NPV	49.52	89.47	95.00	98.33	100	93.11
	Median based	Sensitivity	34.09	94.38	97.78	100.00	100	95.18
		Specificity	35.94	86.00	92.73	98.25	100	89.33
		PPV	67.46	91.37	95.83	98.96	100	93.71
		NPV	48.67	90.65	95.83	100.00	100	93.36
	SR	Sensitivity	0.00	59.89	97.78	100.00	100	74.71
		Specificity	9.23	55.65	75.00	90.81	100	70.79
		PPV	0.00	70.37	84.00	94.15	100	73.06
		NPV	6.82	59.22	96.00	100.00	100	76.38

We can also examine the accuracy of our method to determine the cluster locations and sizes. Table 2.3 lists the summary statistics of the four measures "sensitivity, specificity, PPV and NPV" obtained by our method described in section 2.2.4 as well as by the stepwise regression method [30, 4]. In their approach, the cluster intervals are estimated by the outputting "break points" obtained from their codes. The majority of these measurements by the mean based and median based cluster estimation methods are over 90%. These indicate both methods can identify cluster intervals accurately. Based on the above simulated 2×300 data sets, the least square based stepwise regression method also appears reasonable. However, it has a higher chance to misidentify the significant clusters than our methods (compare the values listed in the columns of "Min" and "1^{st} QT").

In reality, we rarely know the true number of clusters. In order to determine the optimal number of clusters from a data set that only consists of the event time records, Section 2.3 has proposed to use the AIC or BIC criteria. To study the performance of the proposed model selection criteria, we define $\mathcal{K} = 1, 2, 3, 4$ as the pre-selected set of ks. For each fixed $k \in \mathcal{K}$, we estimate the values of $\widehat{\text{AIC}}(k)$ and $\widehat{\text{BIC}}(k)$. We then pick the $\hat{k}$s which maximize $\widehat{\text{AIC}}(k)$ and $\widehat{\text{BIC}}(k)$ respectively to estimate the true number of clusters. Table 2.4 summarizes the model selection results via AIC and BIC criteria. In the single cluster case, 94% times the AIC criterion and 99.7% times the BIC criterion choose the true cluster number $k = 1$. In the two clusters case, 67.7% times the AIC method and 61.3% times the BIC method choose the true cluster number $k = 2$. These numbers are consistent with the percentages reported in the model selection literature [42]. Among those misidentified by the BIC criterion in the two clusters case, most of

Table 2.4: Model selection evaluation in setting I (%)

	AIC				BIC			
	Estimated				Estimated			
	$k=1$	$k=2$	$k=3$	$k=4$	$k=1$	$k=2$	$k=3$	$k=4$
True $k=1$	94	3	1.3	1.7	99.7	.3	0	0
$k=2$	15.3	67.7	12.7	4.3	36.7	61.3	1.7	.3

them have misidentified $k = 1$. Since the BIC penalty term is larger, the BIC criterion is more conservative and tends to select a smaller number of clusters than the AIC criterion.

2.4.2 Setting II

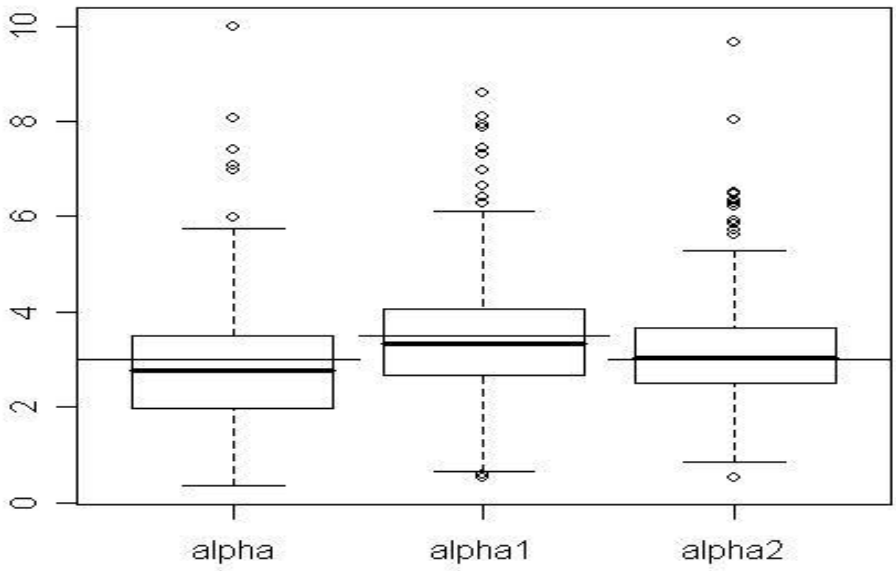

Figure 2.3: Box Plots for α_s estimates in setting II

We also perform simulation studies with the same analysis plan under the setting that the clusters are randomly simulated in each exercise. The parameters used are $\theta = (\alpha_1, \lambda_b, \lambda_c) = (3.0, 1.2, 5.2)$ for single cluster case and $\alpha_1 = 3.5, \alpha_2 = 3.0, \lambda_b = 4, \lambda_c = 3$ for $k = 2$ case. As in subsection 2.4.1, for $k = 1$ case, we generate 300 data sets of $n = 100$ time points (y's), and for $k = 2$ case, we generate 300 data sets of $n = 150$ time points (y's). In particular, a set of cluster intervals within the time interval $(0, 1)$ is first simulated with the given λ_b, λ_c and k. Then, given α_s, one data set with n event points is simulated according to model (2.1). Repeat 300 times each for $k = 1$ and $k = 2$ cases and obtain $2 \times 300 = 600$ simulated data sets of y's. The underlying cluster intervals are different in these 600 data sets. Sometimes by random chance, the simulated cluster

or clusters are too small to be detected by analyzing a data set of size only 100 or 150. It is not surprising that the simulation results will not be as good as those in the first simulation setting. This simulation study demonstrates the performance of our method to deal with general clusters generated from the latent cluster model described in this thesis.

Table 2.5: Power and Type I Error evaluation in setting II

	$k=1$				$k=2$					
	Power		Size		Power			Size		
Method	Wald	LRT	Wald	LRT	Wald		LRT	Wald		LRT
					1	2		1	2	
Reject H_0	74.3%	67.7%	.100	.017	86.0%	89.7%	95.7%	.063	.100	.017

Figure 2.3 shows the box plots for α estimates when we analyze the data sets with $k = 1$ and $k = 2$. For both cases, we can see the α_s estimates are also quite good with most of them around their respective true values at $\alpha = 3.0$ and $(\alpha_1, \alpha_2) = (3.5, 3.0)$. Table 2.5 shows the powers and type I errors for Wald and LRT tests. In the single cluster case, the Wald test has 74.3% power with inflated type I error .100 and the LRT test has 67.7% power with conservative type I error .017. In the two clusters case, the powers for the Wald test are 86.0% and 89.7% with type I errors .063 and .100 separately. The LRT test has a high power 95.7% with conservative type I error .017. Compared with the first simulation study using fixed clusters, both tests have similar type I errors and slightly lower but still reasonable powers. The results of cluster interval estimation are summarized in Table 2.6. The measurements are good with most medians around 95% or higher, although they are not as good as those in the first simulation setting.

We again examine the performance of the proposed AIC and BIC criteria in this simulation setting. Table 2.7 lists the model selection results. In the single cluster case, the AIC criterion gives 90.7% correct selection and the BIC gives 98.0% correct selection. However, in the two clusters case, the selection accuracy is only in the thirties with the AIC method performing a little better than the BIC method. Most of

Table 2.6: Cluster interval estimates evaluation in setting II (%)

	Method	Statistics	Min	1^{st} QT	Median	3^{rd} QT	Max	Mean
$k=1$	Mean	Sensitivity	0	59.41	94.87	100	100	73.21
		Specificity	0	88.71	95.96	100	100	88.14
		PPV	0	21.62	94.00	100	100	67.17
		NPV	0	90.53	96.24	100	100	90.16
	Median	Sensitivity	0	80.00	96.15	100	100	78.25
		Specificity	0	88.88	96.94	100	100	87.95
		PPV	0	42.71	94.12	100	100	70.85
		NPV	0	92.51	98.04	100	100	91.12
$k=2$	Mean	Sensitivity	0.00	88.21	95.16	98.76	100	90.65
		Specificity	14.29	73.75	87.10	96.49	100	82.35
		PPV	0.00	89.69	95.50	98.37	100	91.33
		NPV	8.33	69.39	88.00	96.95	100	80.82
	Median	Sensitivity	0.00	89.66	96.19	99.08	100	91.53
		Specificity	29.27	73.53	90.11	98.15	100	84.10
		PPV	0.00	90.30	95.83	99.12	100	91.12
		NPV	8.60	76.00	89.29	97.50	100	84.28

the misses are in the estimated $k=1$ category. Although the percentages are low, they are not surprising. We have traced down quite a few misidentified cases. Close to two thirds of them have at least one small cluster, within which only a few and sometimes no events (y's) occur. Without increasing the sample size n, few methods can detect such small clusters. The theory behind is shown as follows. In single cluster case, the probability of having at least $s(s \neq 0)$ events inside the cluster I_1, given n and α_1, is

$$P(Z_1 \geq s) = \sum_{j=s}^{n} \binom{n}{j} p_1^j (1-p_1)^{n-j}, \text{ where } p_1 = \frac{\alpha_1 c_1}{1 - c_1 + \alpha_1 c_1}.$$

When c_1 is very small, p_1 is close to 0 and this probability becomes small. In this situation, the small cluster is unlikely to be declared as one. In the case of multiple clusters,

Table 2.7: Model selection evaluation in setting II (%)

	AIC				BIC			
	Estimated				Estimated			
	$k=1$	$k=2$	$k=3$	$k=4$	$k=1$	$k=2$	$k=3$	$k=4$
True $k=1$	90.7	7.0	1.7	.6	98.0	2.0	0	0
$k=2$	45.3	37.7	12.7	4.3	64.7	31.7	2	1.6

the principle is similar with more complicated calculation, which is not presented in this thesis since it is out of the interest.

2.4.3 Results

Summarized from the above simulation results, our method can provide both consistent α_s and accurate cluster intervals estimators. It has high powers to detect significant clusters at least in the cases of clusters within which an event is about three times as likely to happen per time unit than that outside. The Wald test has slightly inflated type I errors and the LRT test is a little conservative. However, they are not too far from the targets. In addition, the AIC criterion performs well for fixed cluster location structure and the BIC criterion tends to favor models with smaller number of clusters. In general, our proposed methodology performs well.

2.5 Real data analysis

We implement our modeling procedure to analyze two real data sets both with and without background adjustment in this section. The first one is the hospital hemoptysis admission data studied by Molinari et.al [30]. The investigation can not only adapt admission conditions and predisposed patients treatment during a favorable period but also point out potential climatic factors that influence the disease occurrence. The second one is the brucellosis data collected by the CDC during 1997-2004. The study is useful in detecting surges in illness [29], particularly when these increases are abrupt, as might occur during a biologic attack. Both data sets can be found in Appendix B.

2.5.1 Hospital hemoptysis admission data

This data set consists of 62 spontaneous hemoptysis admissions (pulmonary disease) at Nice (a southern French city) hospital from January 1 to December 31, 1995. Previously, it was analyzed by the stepwise regression method [30] to detect clusters of minimum size 6 events (the number was chosen according to the Nice hospital pneumologic team). By applying the bootstrap model selection procedure, both single-cluster model with a

winter cluster [58, 87] (February 27-March 28) and two-cluster model with a cluster [58, 126] & a summer cluster [187, 201] (July 6-20) could be selected against the non-cluster model. Nevertheless, since Nice is a tourist city located on the Mediterranean coast, each summer, lots of tourists ($\sim 15\%$) increase the population at risk. After adjusting an estimated

$$R(t) = 1 + \frac{72t}{10,000 \times 365} + \frac{55,000}{355,000} \times \mathbf{1}_{[182,244]}(t), t \in [1, 365] \tag{2.27}$$

function (2.27) for the population at risk, the two-cluster model was no longer selected. The fact that more events occurred between day 187 and 201 was explained due to the presence of tourists. Therefore, only the winter cluster [58, 87] was detected significant in the end. However, in a recent paper [4], the bootstrap based testing method is found not reliable. This data set is reanalyzed and the same potential cluster [58, 87] is no longer detected significant with an associated p-value .13.

Here, we analyze this data set using our method with a set of k's $\kappa = \{1, 2, 3, 4\}$. First, the data is analyzed without background function adjustment (i.e. with model (2.1)). With $k = 1$, an estimated $\hat{\alpha} = 1.961$ is provided via our Monte-Carlo EM algorithm in Subsection 2.2.2. This α is not detected significant different from 1 with either Wald (P-value=.141) or LRT test (P-value=.262) presented in Subsection 2.2.3. The estimated cluster is [58, 108] (February 27-April 18) with our median method in Subsection 2.2.4. When analyzed with $k = 2$, an estimated $\hat{\alpha} = (\hat{\alpha_1}, \hat{\alpha_2})' = (1.935, 1.117)'$ is provided. Neither α is tested significant with either test method (the P-values for Wald test are .107 and .837, for LRT test is .330). The estimated clusters are [58, 108] and [198, 235] (July 17-August 23). Both AIC and BIC criteria described in Section 2.3 select the single-cluster model.

When analyzing using model (2.2) with background function $W(y) \equiv R(t)$, the estimates of α_s increase a little with $\hat{\alpha} = 2.082$ when $k = 1$ and $\hat{\alpha} = (\hat{\alpha_1}, \hat{\alpha_2})' = (2.067, 1.128)'$ when $k = 2$. The corresponding P-values become .083 for Wald and .205 for LRT when $k = 1$, .065 and .813 for Wald and .282 for LRT when $k = 2$. The estimated cluster intervals remain the same. The single-cluster model is selected a second time with both model selection criteria.

Therefore, we conclude, for this data set, that we have one potential non-significant cluster [58, 108]. Compared with Molinari et. al's and Demattei and Molinari's [30, 4] results, our cluster estimator covers a longer period. Both our conclusion and that of Demattei and Molinari point out one potential non-significant cluster. Moreover, the scan statistic also leads to the same conclusion with p-value=.29.

2.5.2 Brucellosis data

Brucellosis (Malta fever) is a infectious disease transmitted from animals to humans. It is caused by bacteria of the genus Brucella and is one "critical biologic agent reported to NNDSS (National Notifiable Disease Surveillance System)" [1]. The data considered here is weekly incidences across the US, collected every year by CDC. Here, we analyze the 2004 data with the weekly number of brucellosis averaged over year 1997 to 2003 as the background function.

Since the data is provided as weekly counts, to avoid ties, we uniformly display the cases in each week and then analyze the transformed data. The background function in the discrete form remains the same. The same procedure used for the hospital data is performed. When analyzed without background function, with $k = 1$, the estimated $\hat{\alpha} = 5.337$; this is significantly different from 1 with both Wald and LRT test (both P-values$\ll .001$). The estimated cluster is between week 44 and 46. When analyzed with $k = 2$, the estimates $\hat{\alpha} = (\hat{\alpha}_1, \hat{\alpha}_2)' = (1.540, 5.938)'$. The LRT test gives significant result (P-value$\ll .001$) and α_2 is detected significant with Wald test (P-value$\ll .001$) while α_1 just misses significance (P-value=.051). The estimated clusters are week [22, 30] and [44, 46]. The AIC criterion selects the two-cluster model and BIC selects the one-cluster model.

When analyzed with the background function, the estimates of α change to $\hat{\alpha} = 6.574$ when $k = 1$ and $\hat{\alpha} = (\hat{\alpha}_1, \hat{\alpha}_2)' = (1.088, 6.715)'$ when $k = 2$. The testing results remain the same. The p-value corresponding to the test $H_0 : \alpha_1 = 1$ vs $H_1 : \alpha_1 \neq 1$ in $k = 2$ case is .078, and the p-values $\ll .001$ for the rest tests. The estimated cluster for single-cluster model remains to be weeks [44, 46], while the estimated clusters for the two-cluster model become weeks [20, 24] and [44, 46]. Both AIC and BIC criteria select

the single-cluster model this time. For this data set, we conclude that there exists one significant cluster between week 44 and 46.

2.6 Conclusion

We develop in this chapter a new approach to detect multiple temporal clusters of varying sizes. With the latent model proposed in section 2.1, we are able to detect multiple clusters simultaneously without limiting potential choices of clusters in a finite set. With the model selection techniques presented in section 2.3, we can obtain an optimal number of potential clusters within the given time window. Based on the likelihood inference and Monte-Carlo EM algorithm developed in section 2.2, we are able to estimate the associated model parameters, detect the significance of the estimated clusters and identify their locations and sizes. Via the simulation studies performed in section 2.4, this new procedure is more efficient than the stepwise regression method that relies on weighted least squares method. In addition, the real data analysis results in section 2.5 show its flexibility for extensions that include regression inclusion etc. We will also extend the whole approach for spatial cluster detection in next chapter.

Chapter 3

Multiple Spatial Cluster Detection

In this chapter, we extend the modeling and inference framework presented in Chapter 2 to detect multiple clusters in spatial data. A different latent model for spatial clusters is proposed in Section 3.1. Some probability distributions are utilized to model the locations and sizes of the latent clusters. A similar piecewise uniform distribution is used to mimic the spatial sample data generation process. The generalized piecewise uniform distribution can be used to adjust for available inhomogeneous background information. Section 3.2 develops similar model inference procedures assuming the number of clusters is known. Section 3.3 demonstrates the AIC and BIC model selection criteria to determine the optimal number of total potential clusters. A simulation study is provided in section 3.4.1 to illustrate and evaluate the proposed methodology for spatial cluster detection. The comparison results with the stepwise regression method developed by Demattei et al. [6] are also presented. Section 3.4.2 contains one real data analysis example. We reanalyze the Pharmacy data studied by Demattei et. al [6]. Further comments and discussions can be found in section 3.5.

3.1 A latent multiple spatial cluster model

Suppose in a given spatial region, for example, $I = (0, T_1) \times (0, T_2)$, there are k clusters. The region does not need to be a rectangle. We assume that these clusters do not overlap with each other and some of them can cross over the boundary of the spatial region as shown in Figure 3.1. For our approach, we assume the clusters are elliptically shaped with centers located at $\mathbf{o}_1 = (o_1^{(1)}, o_1^{(2)})', \mathbf{o}_2 = (o_2^{(1)}, o_2^{(2)})', \ldots, \mathbf{o}_k = (o_k^{(1)}, o_k^{(2)})'$, orientations rotated by angles $\phi_1, \phi_2, \ldots, \phi_k$, semi-major axes equal to $a_1, a_2, \ldots, a_k$ and semi-minor axes equal to $b_1, b_2, \ldots, b_k$ respectively. The center $\mathbf{o}_j$ and the angle ϕ_j

determine the location of the cluster. The axes a_j and b_j determine the shape and size of the cluster. Different from the temporal case where we can order these k clusters according to their temporal occurrences, the spatial clusters can not be ordered in the two-dimensional space. Because of this orderless property, spatial cluster detection is always more complex than the temporal case. In order to make the model structure simpler, we order these k clusters by the first coordinates of their centers, i.e, we assume $o_1^{(1)} \leq o_2^{(1)} \leq \ldots \leq o_k^{(1)}$.

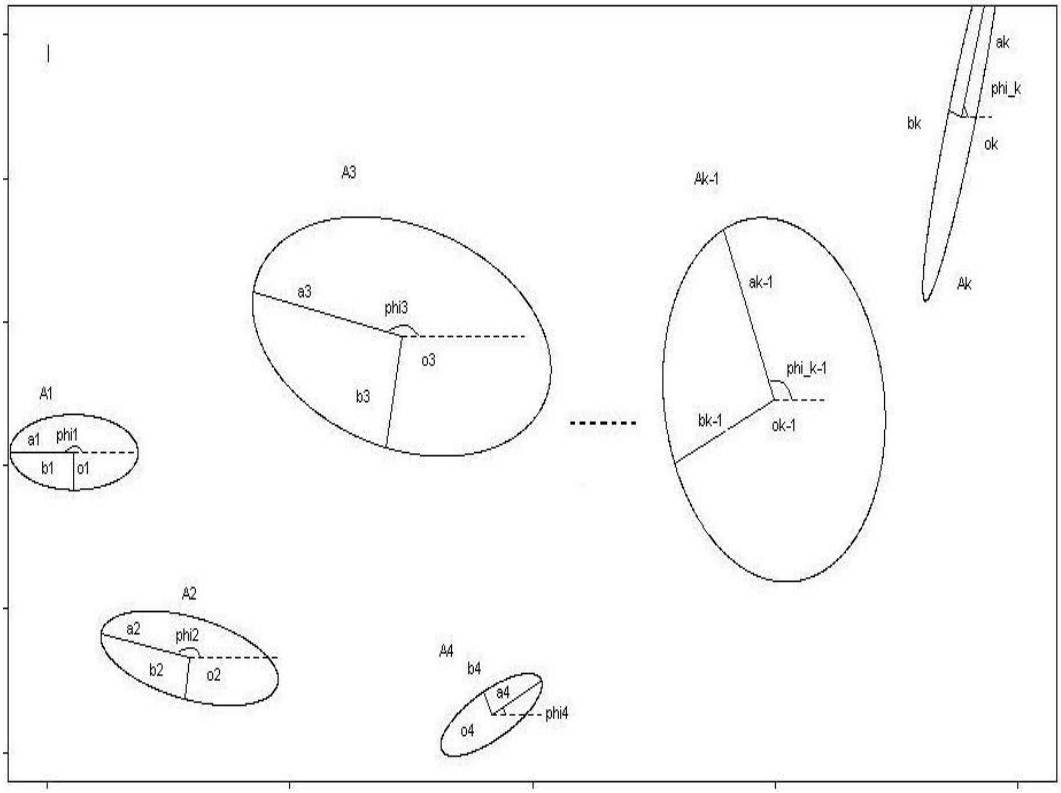

Figure 3.1: An illustrative example of a latent multiple spatial cluster model

To complete the model specification, we assume that the cluster centers $\mathbf{o}_1, \mathbf{o}_2, \ldots, \mathbf{o}_k$ are latent random vectors (ordered by the first coordinates) that are distributed according to a two dimensional distribution with a density function $\psi_{\mathbf{o}}(\mathbf{x}) = \psi_{\mathbf{o}}(\mathbf{x}; \lambda_o)$. The angles are nonnegative random variables with a density function $\psi_\phi(x) = \psi_\phi(x; \lambda_\phi)$. Note the constraint that the semi-major axis $a_j \geq b_j$ (the semi-minor axis) should hold for every cluster. To simplify the model structure, we assume that the cluster axes a_j and b_j are positive ordered random variables on R^+ from the same density function

$\psi_r(x) = \psi_r(x;\lambda_r)$ for all $j = 1, 2, \ldots, k$. Here, λ_o, λ_ϕ and λ_r are unknown parameters. If the cluster centers $\mathbf{o}_1, \mathbf{o}_2, \ldots, \mathbf{o}_k$ are inside the region I, one simple example we use later is that $\psi_{\mathbf{o}}(\mathbf{x};\lambda_o)$ is the two-dimensional Uniform distribution on I. In other cases, we may also assume $\psi_{\mathbf{o}}(\mathbf{x};\lambda_o)$ to have some mass outside the region I. The simple example for $\psi_\phi(x;\lambda_\phi)$ we use later is Uniform$[0,\pi]$. The common choice of $\psi_r(x)$ is among (truncated) exponential, inverse Gamma, or log-normal distributions.

Write $\mathbf{O} = (\mathbf{o}_1, \mathbf{o}_2, \ldots, \mathbf{o}_k)$, $\phi = (\phi_1, \phi_2, \ldots, \phi_k)'$, $\mathbf{a} = (a_1, a_2, \ldots, a_k)'$, $\mathbf{b} = (b_1, b_2, \ldots, b_k)'$, and denote A_j as the j_{th} cluster. We use the same expression $\{\delta = k\}$ to denote the event that k non overlapping clusters occur in spatial region I. Because of the natural orderless property of the clusters in the spatial case, the event $\{\delta = k\}$ is defined differently from the temporal case. When the true cluster number is one, i.e. only one cluster exists in I, we assume the cluster center $\mathbf{o}_1$ falls inside the study region and the cluster is no bigger than that able to cover the whole region I. In other words, $\{\delta = 1\}$ is equivalent to $\{\mathbf{o}_1 \in I \text{ and } A_1 \cap I \subset I\}$. When $k > 1$, $\{\delta = k\}$ means k cluster centers fall inside I and the clusters are not overlapping with each other, i.e. $\{\mathbf{o}_i \in I \text{ and } A_i \cap A_j = \varnothing, i, j = 1, \cdots, k, j > i\}$.

Many methodologies have defined their clusters through circles with centers inside the region under consideration [13, 14, 15, 16, 22, 40]. Extension to elliptical-shaped clusters has not been that successful. A straightforward extended elliptic spatial scan statistic is described in Kulldorff et. al [25]. This statistic has similar properties as the circular-shaped scan statistics. For all these methods, the potential choices of the clusters are limited to a finite set. We have a more general assumption. In our latent model, circular shaped clusters are only a special case of the elliptical-shaped clusters by fixing the k angles as 0 and forcing the k semi-major axes equal to the corresponding semi-minor axes. The latent clusters are then determined by the centers $\mathbf{O} = (\mathbf{o}_1, \mathbf{o}_2, \ldots, \mathbf{o}_k)$ and radii $\mathbf{r} = (r_1, r_2, \ldots, r_k)'$. The circular shaped cluster model assumption is simpler than the elliptical-shaped cluster model.

The latent variables $\mathbf{O}, \phi, \mathbf{a}$ and $\mathbf{b}$ are not observed. What we can observe in this model setting are only the spatial data points $y_1, y_2, \ldots, y_n$ where incidences of interest occur. We assume that the observations $y_1, y_2, \ldots, y_n$ are i.i.d. samples from the

piecewise uniform density function,

$$f_\theta(y|\mathbf{O},\phi,\mathbf{a},\mathbf{b},k)=\begin{cases}\frac{\alpha_1}{E+\sum_{j=1}^{k}(\alpha_j-1)D_j}, & \text{if } y\in A_1\\ \cdots\cdots & \\ \frac{\alpha_k}{E+\sum_{j=1}^{k}(\alpha_j-1)D_j}, & \text{if } y\in A_k\\ \frac{1}{E+\sum_{j=1}^{k}(\alpha_j-1)D_j}, & \text{if } y\notin \cup_{j=1}^{k}A_j\end{cases} \tag{3.1}$$

where $E = Area(I)$ is the total area of the study region I, $D_j = Area(A_j \bigcap I)$ is the area of the j_{th} cluster inside the study region. $\theta = (\alpha, \lambda)$ is the collection of all unknown parameters, including the parameters $\alpha = (\alpha_1, \ldots, \alpha_k)'$ and $\lambda = (\lambda_o, \lambda_\phi, \lambda_r)'$ that are associated with random vectors $\mathbf{o}_i$'s, random variables ϕ_i's, a_i's and b_i's separately. The parameters α have the same meanings as in the temporal case.

Similar to model (2.1), we can also incorporate a baseline function $B(x^{(1)}, x^{(2)})$ in model (3.1) and have the following generalized model (3.2)

$$f_\theta(y|\mathbf{O},\phi,\mathbf{a},\mathbf{b},k)=\begin{cases}\frac{\alpha_1 B(y^{(1)},y^{(2)})}{\tilde{E}+\sum_{j=1}^{k}(\alpha_j-1)\tilde{D}_j}, & \text{if } y\in A_1\\ \cdots\cdots & \\ \frac{\alpha_k B(y^{(1)},y^{(2)})}{\tilde{E}+\sum_{j=1}^{k}(\alpha_j-1)\tilde{D}_j}, & \text{if } y\in A_k\\ \frac{B(y^{(1)},y^{(2)})}{\tilde{E}+\sum_{j=1}^{k}(\alpha_j-1)\tilde{D}_j}, & \text{if } y\notin \cup_{j=1}^{k}A_j\end{cases} \tag{3.2}$$

where $\tilde{E} = \int\int_I B(x^{(1)}, x^{(2)})dx^{(1)}dx^{(2)}$ and $\tilde{D}_j = \int\int_{A_j \bigcap I} B(x^{(1)}, x^{(2)})dx^{(1)}dx^{(2)}$, for $j = 1, 2, \ldots, k$. This model is equivalent to model (3.1) when $B(x^{(1)}, x^{(2)}) \propto 1$. Similarly as in the last chapter, we develop methodology for data from the less complicated model (3.1). Appendix A derives all the formulas for the generalized model (3.2).

3.2 Model inference for given number of clusters

The model inference for given number of clusters is slightly different from the temporal case. We state the whole procedures in this section. In section 3.3, we discuss the model selection procedure.

3.2.1 Likelihood function of observed data

The definition of event $\{\delta = k\}$ makes the calculation of the explicit formula $P_\lambda(\delta = k)$ difficult for the spatial case. We instead propose a general computational method to approximate this probability as the following: when $k = 1$,

Step a. Simulate one set of $\mathbf{o}_1 \sim \psi_{\mathbf{o}}(\mathbf{x}; \lambda_o)$, $\phi_1 \sim \psi_\phi(x; \lambda_\phi)$, and ordered samples a_1 & $b_1 \sim \psi_r(x; \lambda_r)$. Locate the left and right foci of the elliptical cluster.

Step b. Calculate the maximum sum of the distances between all the points on the border of I and the left and right foci of the elliptical cluster. Denote it as d_1 and check if $d_1 > 2a_1$.

Step c. Repeat Step a and b for a large number of times (M).

Step d. The probability of event $\{\delta = 1\}$ can be approximated by

$$\frac{\sum_{m=1}^{M} \mathbf{1}_{\{d_1^{(m)} > 2a_1^{(m)}\}}}{M}$$

when $k > 1$,

Step a. Simulate one set of ordered vectors $\mathbf{o}_1, \mathbf{o}_2, \ldots, \mathbf{o}_k, \sim \psi_{\mathbf{o}}(\mathbf{x}; \lambda_o)$, $\phi_1, \phi_2, \ldots, \phi_k, \sim \psi_\phi(x; \lambda_\phi)$, and ordered $(a_1, b_1), (a_2, b_2), \ldots, (a_k, b_k), \sim \psi_r(x; \lambda_r)$

Step b. Check whether $A_i \cap A_j = \varnothing$ for all $i, j = 1, \cdots, k, j > i$

Step c. Repeat Step a and b for M times

Step d. The probability of event $\{\delta = k\}$ can be approximated by

$$\frac{\sum_{m=1}^{M} \mathbf{1}_{\{A_i^{(m)} \cap A_j^{(m)} = \varnothing, i,j=1,2,\cdots,k, j>i\}}}{M}$$

Given two arbitrary ellipses in two-dimensional space, it is not straightforward to detect their non-overlapping condition. In the paper published by Zheng and Palffy-Muhoray [55], an analytic expression for the distance between two ellipses' centers

when they are tangent ("distance of closest approach") is derived as a function of their orientation relative to the line joining their centers. Initially, if the two ellipses A_1 and A_2 are not tangent, one ellipse is then translated toward (if they are non-overlapping), or away from (if they are overlapping) the other along the line joining their centers until they are tangent. An anisotropic scaling matrix is then introduced to transform them into a unit circle A_1' and an ellipsis A_2', that remain tangent to each other. The distance of closest approach of the transformed A_1' and A_2' can be determined analytically. Finally, the distance of closest approach of the original two ellipses A_1 and A_2 is determined by the inverse scaling transformation.

Denote d_{ij} as the distance of closest approach of i_{th} and j_{th} clusters, $i, j = 1, \cdots k, j > i$. By comparing d_{ij} with the true distance between the two centers $\|\mathbf{o}_i - \mathbf{o}_j\|$, we are able to check the non-overlapping status of these two ellipses. If $\|\mathbf{o}_i - \mathbf{o}_j\| > d_{ij}$, i_{th} and j_{th} clusters are non-overlapping with each other. If $\|\mathbf{o}_i - \mathbf{o}_j\| = d_{ij}$, they are tangent to each other. Otherwise, they are overlapping. For circular shaped clusters, the non-overlapping condition can be checked easily by comparing the sum of the two radius $r_i + r_j$ with $\|\mathbf{o}_i - \mathbf{o}_j\|$.

Conditional on $\{\delta = k\}$, the joint conditional likelihood function of $(\mathbf{O}, \phi, \mathbf{a}, \mathbf{b})$, similar as 2.4, becomes

$$f_\theta(\mathbf{O}, \phi, \mathbf{a}, \mathbf{b}|k) = \frac{k! \prod_{j=1}^{k} \{\psi_{\mathbf{o}}(\mathbf{o}_j)\psi_\phi(\phi_j)2!\psi_r(a_j)\psi_r(b_j)\}\mathbf{1}_{\{\delta=k\}}}{P_\lambda(\delta = k)}. \tag{3.3}$$

In the special case with ψ_o being density function of uniform distribution on I, ψ_ϕ being Uniform$[0, \pi]$ and ψ_r being exponential distribution Exp(λ_r), we have

$$f_\theta(\mathbf{O}, \phi, \mathbf{a}, \mathbf{b}|k) = \frac{k!(\frac{1}{E\pi})^k \prod_{j=1}^{k} [\mathbf{1}_{\{o_j \in I\}}\mathbf{1}_{\{\phi_j \in [0,\pi]\}}](2!)^k \lambda_r^{2k} e^{-\lambda_r \sum_{j=1}^{k}(a_j+b_j)}\mathbf{1}_{\{\delta=k\}}}{P_\lambda(\delta = k)}, \tag{3.4}$$

From model (3.1), the conditional joint density function of $\mathbf{y} = (y_1, y_2, \ldots, y_n)$, conditional on $\mathbf{O}, \phi, \mathbf{a}, \mathbf{b}$, and $\delta = k$, is

$$f_\theta(\mathbf{y}|\mathbf{O}, \phi, \mathbf{a}, \mathbf{b}, k) = \prod_{i=1}^{n} f(y_i|\mathbf{O}, \phi, \mathbf{a}, \mathbf{b}, k) = \frac{\prod_{j=1}^{k} \alpha_j^{Z_j}}{\{E + \sum_{j=1}^{k}(\alpha_j - 1)D_j\}^n}, \tag{3.5}$$

where $Z_j = Z_j(\mathbf{y}, \mathbf{O}, \phi, \mathbf{a}, \mathbf{b}) = \sum_{i=1}^{n} \mathbf{1}_{\{y_i \in A_j\}}$ is the number of incidences that occur

inside the jth cluster. Thus, the joint density function of $\mathbf{y}$ and $\delta = k$, is

$$\begin{aligned} f_\theta(\mathbf{y}, k) &= \int \cdots \int f_\theta(\mathbf{y}, \mathbf{O}, \phi, \mathbf{a}, \mathbf{b}|k) P_\lambda(\delta = k) d\mathbf{O} d\phi d\mathbf{a} d\mathbf{b} \\ &= \int \cdots \int f_\theta(\mathbf{y}|\mathbf{O}, \phi, \mathbf{a}, \mathbf{b}, k) f_\theta(\mathbf{O}, \phi, \mathbf{a}, \mathbf{b}|k) P_\lambda(\delta = k) d\mathbf{O} d\phi d\mathbf{a} d\mathbf{b} \quad (3.6) \end{aligned}$$

and the log-likelihood function of observing $\mathbf{y}$ and $\delta = k$ is

$$\ell_k(\theta|\mathbf{y}) = \log\{f_\theta(\mathbf{y}, k)\}. \quad (3.7)$$

All of the above formulas are more complicated than the temporal case. We develop similar Monte-Carlo EM algorithm procedures in the next subsection to estimate the spatial model parameters.

3.2.2 Monte-Carlo EM algorithm for model estimation

Similarly to Chapter 2, $(\mathbf{y}, \mathbf{O}, \phi, \mathbf{a}, \mathbf{b}, \delta = k)$ are treated as the complete responses and $(\mathbf{y}, \delta = k)$ as the observed responses. The joint density function of $(\mathbf{y}, \mathbf{O}, \phi, \mathbf{a}, \mathbf{b}, \delta = k)$ becomes,

$$\begin{aligned} & f_\theta(\mathbf{y}, \mathbf{O}, \phi, \mathbf{a}, \mathbf{b}, k) = f_\theta(\mathbf{y}|\mathbf{O}, \phi, \mathbf{a}, \mathbf{b}, k) f_\theta(\mathbf{O}, \phi, \mathbf{a}, \mathbf{b}|k) P_\lambda(\delta = k) \\ = & \frac{\prod_{j=1}^k \alpha_j^{Z_j}}{\{E + \sum_{j=1}^k (\alpha_j - 1) D_j\}^n} k! \prod_{j=1}^k \{\psi_{\mathbf{o}}(\mathbf{o}_j) \psi_\phi(\phi_j) 2! \psi_r(a_j) \psi_r(b_j)\} \mathbf{1}_{\{\delta = k\}} \quad (3.8) \end{aligned}$$

The same EM procedures proposed in chapter 2 can be used here for spatial data. Note, equations (2.12) and (2.13) become

$$\begin{aligned} Q_1(\alpha|\theta^{(s)}) &= \sum_{j=1}^k \mathrm{E}(Z_j|\mathbf{y}, k, \theta^{(s)}) \log \alpha_j - n\mathrm{E}[\log\{E + \sum_{j=1}^k (\alpha_j - 1) D_j\}|\mathbf{y}, k, \theta^{(s)}] \quad (3.9) \\ Q_2(\lambda|\theta^{(s)}) &= \sum_{j=1}^k \mathrm{E}[\log \psi_{\mathbf{o}}(\mathbf{o}_j) + \log \psi_\phi(\phi_j) + \log \psi_r(a_j) + \log \psi_r(b_j)|\mathbf{y}, k, \theta^{(s)}] \quad (3.10) \end{aligned}$$

For each $s = 0, 1, \ldots$ the parameter estimates $\theta^{(s+1)} = (\alpha^{(s+1)}, \lambda^{(s+1)})$ are updated by

$$\alpha^{(s+1)} = \operatorname{argmax} Q_1(\alpha|\theta^{(s)}), \quad \text{and} \quad \lambda^{(s+1)} = \operatorname{argmax} Q_2(\lambda|\theta^{(s)}). \quad (3.11)$$

In the case with ψ_o, ψ_ϕ and ψ_r being density functions of uniform distribution on I, Uniform$[0, \pi]$ and exponential distribution Exp(λ_r), the updating formula of $\lambda^{(s+1)}$ is simply $\lambda_r^{(s+1)} = 2k / \sum_{j=1}^k [\mathrm{E}(a_j|\mathbf{y}, k, \theta^{(s)}) + \mathrm{E}(b_j|\mathbf{y}, k, \theta^{(s)})]$.

Similarly to Chapter 2, the equations (3.9) and (3.10) do not have explicit form. We need to use Monto-Carlo simulation to estimate them. The Gibbs sampling approach including the importance sampling method as in chapter 2 can be used to simulate $\mathbf{O}^*$, ϕ^*, $\mathbf{a}^*$ and $\mathbf{b}^*$ from $f(\mathbf{O}, \phi, \mathbf{a}, \mathbf{b}|\mathbf{y}, k, \theta^{(s)})$. Once we have M sets of Gibbs samples $\mathbf{O}^*$, ϕ^*, $\mathbf{a}^*$ and $\mathbf{b}^*$, the six conditional expectations in the two equations can be evaluated by $\frac{1}{M}\sum_* Z_j^*$, $\frac{1}{M}\sum_* \log\{E + \sum_{j=1}^k (\alpha_j - 1)D_j^*\}$, $\frac{1}{M}\sum_* \log\{\psi(o_j^*)\}$, $\frac{1}{M}\sum_* \log\{\psi(\phi_j^*)\}$, $\frac{1}{M}\sum_* \log\{\psi(a_j^*)\}$, and $\frac{1}{M}\sum_* \log\{\psi(b_j^*)\}$, respectively. In the special case, the last four expectations just become $\frac{1}{M}\sum_*(a_j^* + b_j^*)$.

To carry out the EM computation, the only remaining question is how to simulate a set of $\mathbf{o_j}, \phi_j, a_j$ and b_j, i.e, a j^{th} cluster A_j, from the fully conditional distributions given the rest of other $\mathbf{o_j}, \phi_j, a_j$ and b_j's, i.e, all the other clusters A_j's. By ignoring the unwanted terms, it is easy to see that, for $j = 1, 2, \ldots, k$,

$$f(A_j|A_l, l = 1, 2, \ldots, k, l \neq j, \mathbf{y}, k) \propto f(\mathbf{O}, \phi, \mathbf{a}, \mathbf{b}, \mathbf{y}|k) \tag{3.12}$$
$$\propto \frac{\alpha_j^{Z_j}}{\{E + \sum_{l\neq j}(\alpha_l - 1)D_l + (\alpha_j - 1)D_j\}^n} \psi_{\mathbf{o}}(\mathbf{o_j})\psi_\phi(\phi_j)\psi_r(a_j)\psi_r(b_j)\mathbf{1}_{(\delta=k)},$$

Thus, given a set of parameters $\theta = (\alpha, \lambda)$, we can use the following importance sampling method to simulate a A_j:

Step A. Simulate a large number of random deviates $A_j^{[1]}, A_j^{[2]}, \ldots, A_j^{[S]}$ from a candidate distribution $\widetilde{\psi_{A_j}(A_j)}$. Then, compute weight

$$w_l = \frac{\alpha_j^{Z_j^{[s]}}}{\{E + \sum_{l\neq j}(\alpha_l - 1)D_l + (\alpha_j - 1)D_j^{[s]}\}^n} \frac{\psi_{A_j}(A_j^{(s)})}{\widetilde{\psi_{A_j}(A_j^{(s)})}} \mathbf{1}_{(\delta^{[s]}=k)}$$

for $s = 1, 2, \ldots, S$. Note, here, $\psi_{A_j}(A_j) = \psi_{\mathbf{o}}(\mathbf{o_j})\psi_\phi(\phi_j)\psi_r(a_j)\psi_r(b_j)$, $Z_j^{[s]}$ is the total number of incidences in jth cluster and $\{\delta^{[s]} = k\}$ is the constraint of having k non-overlapping clusters with A_j replaced by $A_j^{[s]}$ and the rest of A_l's the same.

Step B. Simulate one A_j from the S values $A_j^{[1]}, A_j^{[2]}, \ldots, A_j^{[S]}$ with respective probabilities $(p_1, p_2, \ldots, p_S)$; Here, $p_s = w_s / \sum_{s=1}^S w_s$.

Different from the temporal case, even in the special case that $\psi_{\mathbf{o}} \sim Uniform(I), \psi_\phi \sim Uniform[0, \pi]$ and $\psi_r \sim Exp(\lambda_r)$, it is not easy to simulate a $A_j^{[s]}$ from a candidate

distribution $\widetilde{\psi_{A_j}(A_j)}$. The following is a suggested way: let

$$\widetilde{\psi_{A_j}(A_j)} = \psi_{\mathbf{o}}(\mathbf{o_j})\psi_{\phi}(\phi_j)\psi_r(a_j)\psi_r(b_j)\mathbf{1}_{(\delta=k)}$$

when $k = 1$,

Step a. Simulate a $\mathbf{o}_1^{[s]} \sim Uniform(I)$

Step b. Simulate a $\phi_1^{[s]} \sim Uniform[0, \pi]$.

Step c. Use importance sampling algorithm to obtain a set of ordered $a_1^{[s]}$ and $b_1^{[s]} \sim Exp(\lambda_r)$ so that $A_1^{[s]}$ is not big enough to cover the whole study region I.

when $k > 1$,

Step a. Use importance sampling algorithm to obtain a $\mathbf{o}_j^{[s]} \sim Uniform(I)$ outside of all the other clusters $A_l's, l \neq j$. With the constraint that $\mathbf{o}_1^{(1)} \leq \mathbf{o}_2^{(1)} \leq \ldots \leq \mathbf{o}_k^{(1)}$, we simulate $\mathbf{o}_j^{[s](1)} \sim Uniform[\mathbf{o}_{(j-1)}^{[s](1)}, \mathbf{o}_{(j+1)}^{[s](1)}]$. Here, $\mathbf{o}_0^{[s](1)}$ is defined as the smallest x coordinate of all the points inside the region I and $\mathbf{o}_{(k+1)}^{[s](1)}$ is the largest x coordinate.

Step b. Simulate a $\phi_j^{[s]} \sim Uniform[0, \pi]$.

Step c. Use importance sampling algorithm to obtain a set of ordered $a_j^{[s]}$ and $b_j^{[s]} \sim Exp(\lambda_r)$ so that $A_j^{[s]}$ is non overlapping with all the other clusters $A_l's, l \neq j$.

In other more complicated cases, assuming we know how to simulate from $\psi_{\mathbf{o}}(\mathbf{o}_j), \psi_{\phi}(\phi_j)$ and $\psi_r(r_j)$, we can use importance sampling method to simulate from the truncated distributions.

After the Monte-Carlo EM procedure, similar estimator of the variance-covariance matrix as in Chapter 2 can be obtained. In particular, equation (2.17) becomes,

$$\begin{aligned} H_n &\overset{d}{=} -\left\{\frac{\partial^2}{\partial\theta^2}\ell_k(\theta|\mathbf{y})\right\} \\ &= -E\left\{\frac{\partial^2}{\partial\theta^2}\ell(\theta)|\mathbf{y}, \delta = k\right\} - \mathrm{Var}\left\{\frac{\partial}{\partial\theta}\ell(\theta)|\mathbf{y}, \delta = k\right\} \end{aligned} \tag{3.13}$$

where $\ell(\theta) = \log\{f_\theta(\mathbf{y}, \mathbf{O}, \mathbf{r}, k)\}$ is the complete log-likelihood function. The information matrix is numerically estimated by

$$H_n = -\frac{1}{M}\sum_{*}\frac{\partial^2}{\partial\theta^2}\ell(\theta) \tag{3.14}$$

$$-\left\{\frac{1}{M}\sum_{*}\left[\frac{\partial}{\partial\theta}\ell(\theta)\right]\left[\frac{\partial}{\partial\theta}\ell(\theta)\right]' - \left[\frac{1}{M}\sum_{*}\frac{\partial}{\partial\theta}\ell(\theta)\right]\left[\frac{1}{M}\sum_{*}\frac{\partial}{\partial\theta}\ell(\theta)\right]'\right\} \tag{3.15}$$

where the summations are over the M sets of Gibbs samples $\mathbf{O}^*$, ϕ^*, $\mathbf{a}^*$ and $\mathbf{b}^*$ in the final round of the EM algorithm. One can also derive these two formulas by making the baseline function equal to 1 in the formulas (A.9) and (A.10) in Appendix A.

3.2.3 Likelihood inference for tests related to α's

In this subsection, We illustrate likelihood inference for two sided tests related to α's.

For the significant test of a single (jth) cluster, i.e., $H_0 : \alpha_j = 1$ versus $H_1 : \alpha_j \neq 1$, the Wald test procedure proposed in Section 2.2.3 can be performed here exactly the same way. For the testing problem whether there exists at least one significant cluster among the k clusters, i.e., $H_0 : \alpha_1 = \alpha_2 = ... = \alpha_k = 1$ versus H_1 : at least one $\alpha_j \neq 1$, the LRT statistic is slightly different because of different definition of the event $\delta = k$. Now, the probability $P_\lambda(\delta = k) \to 1$ with $\lambda_r \to \inf$. Therefore, the twice log likelihood ratio test statistic from formula (2.19) becomes

$$\begin{aligned} & R = 2\log\{\frac{\max_{H_1} f_\theta(\mathbf{y}, k)}{\max_{H_0} f_\theta(\mathbf{y}, k)}\} \\ = \;& 2\{\log \int\int f_{\hat{\theta}}(\mathbf{y}|\mathbf{O}, \phi, \mathbf{a}, \mathbf{b}, k) f_{\hat{\theta}}(\mathbf{O}, \phi, \mathbf{a}, \mathbf{b}|k) d\mathbf{O} d\phi d\mathbf{a} d\mathbf{b} + \log P_{\hat{\lambda}}(\delta = k) + n\log(E)\}, \end{aligned}$$

where $\hat{\theta} = (\hat{\alpha}', \hat{\lambda}')'$ are the estimates of the parameters obtained from the aforementioned EM algorithm under H_1. This statistic is numerically estimated by,

$$R^{**} = 2\bigg[\log\{\frac{1}{M}\sum_{**} f(\mathbf{y}|\mathbf{O}^{**}, \phi^{**}, \mathbf{a}^{**}, \mathbf{b}^{**}, k)\} + \log P_{\hat{\lambda}}(\delta = k) + n\log(E)\bigg], \quad (3.16)$$

where the summation is over the M sets of samples $\mathbf{O}^{**} = (\mathbf{o}_1^{**}, \ldots, \mathbf{o}_k^{**}), \phi^{**} = (\phi_1^{**}, \ldots, \phi_k^{**}), \mathbf{a}^{**} = (a_1^{**}, \ldots, a_k^{**})$ and $\mathbf{b}^{**} = (b_1^{**}, \ldots, b_k^{**})$ simulated from $f_{\hat{\theta}}(\mathbf{O}, \phi, \mathbf{a}, \mathbf{b}|k)$ using the similar Gibbs sampling approach method in Section 2.2.3. The test for $H_0 : \alpha_1 = \alpha_2 = ... = \alpha_k = 1$ versus H_1 : at least one $\alpha_j \neq 1$ is performed by comparing R^{**} with the χ^2_k distribution since R is also asymptotically χ^2 distributed with k degrees of freedom.

Note that, the previous truncated distributions given in formula (2.21) and (2.22) become

$$f_\theta(A_j|A_l, l = 1, 2, \ldots, k, l \neq j, \phi, \mathbf{a}, \mathbf{b}, \delta = k) \propto \psi_{\mathbf{o}}(\mathbf{o}_j)\psi_\phi(\phi_j)\psi_r(a_j)\psi_r(b_j)\mathbf{1}_{(\delta=k)} \quad (3.17)$$

In the special case, one computation method has been suggested in the last subsection 3.2.2 to simulate a $A_j^{[s]}$ from the above truncated distribution.

3.2.4 Identification of cluster regions

If a cluster is significant (i.e. $\alpha_j \neq 1$), we often want to determine the cluster region. Note that the jth cluster A_j is determined by the center $\mathbf{o}_j$, the rotated angle ϕ_j, the semi-major axis a_j and the semi-minor axis b_j. Their conditional expectations given $\mathbf{y}$ and k ("posterior means") are $\mathrm{E}\{\mathrm{o}_j|\mathbf{y},k\}|_{\theta=\hat{\theta}}$, $\mathrm{E}\{\phi_j|\mathbf{y},k\}|_{\theta=\hat{\theta}}$, $\mathrm{E}\{\mathrm{a}_j|\mathbf{y},k\}|_{\theta=\hat{\theta}}$ and $\mathrm{E}\{\mathrm{b}_j|\mathbf{y},k\}|_{\theta=\hat{\theta}}$. The cluster center $\mathbf{o}_j$, the rotated angle ϕ_j, the semi-major axis a_j and the semi-minor axis b_j can be simply estimated by $\frac{1}{M}\sum_* \mathbf{o}_j^*$, $\frac{1}{M}\sum_* \phi_j^*$, $\frac{1}{M}\sum_* a_j^*$ and $\frac{1}{M}\sum_* b_j^*$ respectively. Here $\sum_*$ is the summation over the M sets of Gibbs samples in the last iteration of the EM algorithm.

An alternative approach is to use the medians of the M sets of o_j^*, ϕ_j^*, a_j^* and b_j^* to estimate $\mathrm{o}_j, \phi_j, \mathrm{a}_j$ and b_j, respectively. Since the distribution may not be symmetric, this median method may provide more accurate estimators.

We can also use the same four empirical statistics sensitivity, specificity, PPV and NPV introduced in Subsection 2.2.4 to assess the performance of these cluster region estimators for simulation studies. The closer they are to one, the more accurate the estimated cluster regions are.

3.3 Determination of the unknown number of clusters

Since the number of clusters is rarely known in practice, we use the same AIC and BIC criteria as in Section 2.3 to determine the number of clusters from the observed data. Note, the AIC and BIC criteria statistics given in equations (2.23) and (2.24) become,

$$\begin{aligned} & \mathrm{AIC}(k) = -2\log f_\theta(\mathbf{y},k) + 2k \qquad (3.18) \\ = \; & -2\log[\int\int f_\theta(\mathbf{y}|\mathbf{O},\phi,\mathbf{a},\mathbf{b},k) f_\theta(\mathbf{O},\phi,\mathbf{a},\mathbf{b}|k) d\mathbf{O} d\phi d\mathbf{a} d\mathbf{b}] - 2\log P_\lambda(\delta = k) + 2k \end{aligned}$$

and

$$\begin{aligned} & \mathrm{BIC}(k) = -2\log f_\theta(\mathbf{y}, k) + k\log(n) \qquad (3.19) \\ = \; & -2\log[\int\int f_\theta(\mathbf{y}|\mathbf{O}, \mathbf{r}, k) f_\theta(\mathbf{O}, \phi, \mathbf{a}, \mathbf{b}|k) d\mathbf{O} d\phi d\mathbf{a} d\mathbf{b}] - 2\log P_\lambda(\delta = k) + k\log(n) \end{aligned}$$

They are approximated by (3.20) and (3.21) respectively using the M sets of $\mathbf{O}^{**}$ and $\mathbf{r}^{**}$ simulated in Subsection 3.2.3.

$$\widehat{\mathrm{AIC}}(k) = -2\log\big[\frac{1}{M}\sum_{**} f(\mathbf{y}|\mathbf{O}^{**}, \phi^{**}, \mathbf{a}^{**}, \mathbf{b}^{**}, k)\big] - 2\log P_{\hat{\lambda}}(\delta = k) + 2k, \qquad (3.20)$$

and $\mathrm{BIC}(k)$ criterion can be approximated by

$$\widehat{\mathrm{BIC}}(k) = -2\log\big[\frac{1}{M}\sum_{**} f(\mathbf{y}|\mathbf{O}^{**}, \phi^{**}, \mathbf{a}^{**}, \mathbf{b}^{**}, k)\big] - 2\log P_{\hat{\lambda}}(\delta = k) + k\log(n), \quad (3.21)$$

Finally we select the number k with the smallest corresponding $\widehat{\mathrm{AIC}}(k)$ or $\widehat{\mathrm{BIC}}(k)$ value.

Therefore, the same approach summarized in Section 2.3 can also be used here for spatial cluster detection.

• Denote $\mathcal{K}$ a pre-selected set of k's, for each fixed $k \in \mathcal{K}$, apply the Monte-Carlo EM algorithm in Section 3.2.2 to get the parameter estimates and use either the AIC or BIC rule to determine the number of clusters k.

• For the chosen k, use the results in Sections 3.2.3-3.2.4 to detect and determine the cluster regions.

3.4 Simulation studies and real data analysis

In Chapter 2, we have shown that our method performs well for the temporal case. In this section, we test our method performance for the spatial case by simulation studies and real data analysis.

3.4.1 Simulation studies

In this section, we perform simulation studies with fixed cluster regions similar to Subsection 2.4.1. We apply our method for single cluster and two clusters detection separately. We also compare our cluster detection performance with that by the stepwise

regression method [6]. For simplicity, the simulation studies are all done within the spatial region $I = (0, 2) \times (0, 1)$ with circular shaped clusters.

For the case of single-cluster detection, we fix the single cluster centered at $\mathbf{O} = o_1 = (.424, .426)'$ and with radius $\mathbf{r} = r_1 = .427$. Choosing $\alpha = 3.5$, we simulate $n = 100$ independently identically distributed spatial points $y_1, y_2, \ldots, y_{100}$ according to model (3.1). With these 100 y values and $k = 1$, applying the procedures developed in Section 3.2, we can get a set of parameter estimates, perform the hypotheses testing, and identify the potential cluster region. This simulation exercise is repeated 600 times.

For the case of two clusters ($k = 2$), we fix the first cluster centered at $\mathbf{o}_1 = (.099, .601)'$, the second cluster centered at $\mathbf{o}_2 = (1.379, .355)'$ and $\mathbf{r} = (r_1, r_2)' = (0.248, 0.194)'$. We then choose $\alpha_1 = 3.5$ and $\alpha_2 = 4.0$, and simulate $n = 150$ independently identically distributed time points $y_1, y_2, \ldots, y_{150}$ according to model (3.1). With these 150 y values and $k = 2$, we can get a set of parameter estimates, perform the hypotheses testing, and identify the potential cluster regions. Again, this simulation exercise is repeated 600 times.

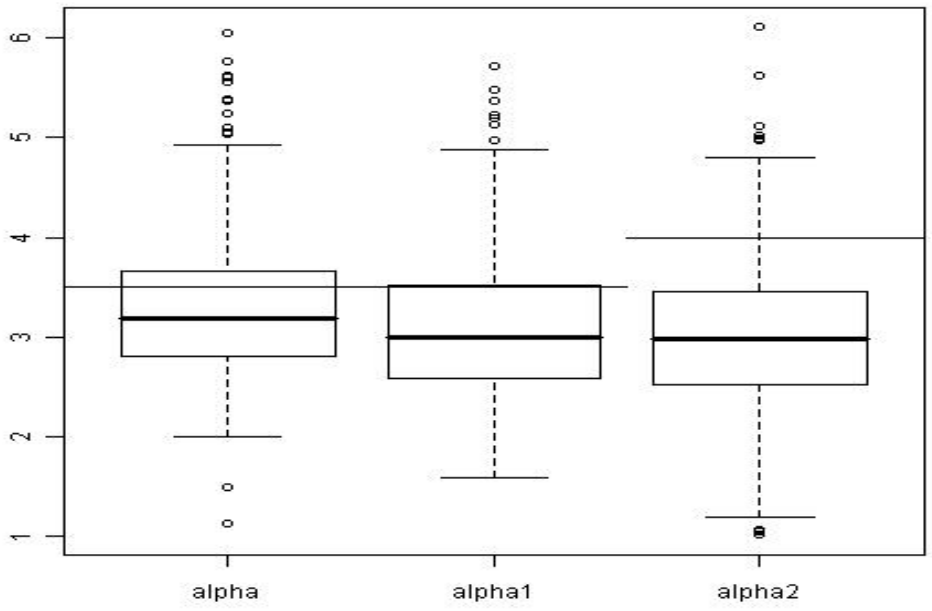

Figure 3.2: Box Plots for α_s estimates

Figure 3.2 shows the box plots of the estimates of α's in the two simulation studies. The first one is for the single cluster ($k = 1$) case and the other two are for α_1 and α_2 in the two clusters ($k = 2$) case. It seems the estimates underestimate the true parameters a little especially for the second smaller cluster in the two clusters case. The main reason may be because of the relatively smaller sample size ($n = 100$ for $k = 1$, $n = 150$ for $k = 2$) for spatial cluster detection. Another thing to mention is that we have compromised the convergence criterion a little to save the analysis time. This may affect the results' accuracy. Meanwhile, the estimates are not far from the true values.

To assess the performance of the likelihood based tests, including both Wald and likelihood ratio tests, we examine the powers and sizes of their level .05 tests. For power computation, we use the same $2 \times 600 = 1200$ simulated data sets described above and list the results in Table 3.1. To compute the size, we simulate another 600 data sets of 100 time points from the Uniform distribution on spatial region I (in the $k = 1$ case) and 600 data sets of 150 time points from the Uniform distribution on spatial region I (in the $k = 2$ case). We apply the same estimation and testing procedures to these two groups of 600 data sets with fixed $k = 1$ and $k = 2$ respectively. Then, we record the results in Table 3.1. In the single cluster case, both the Wald and LRT tests have over 99.5% power to detect the cluster. The Type I error of the Wald test is controlled at .05 and the LRT test is conservative at .012. In the two cluster case, the powers for Wald test are 85.5% and 66.7% with Type I errors .026 and .017 respectively. The LRT test has a 74% power with conservative Type I error .002.

Table 3.1: Power and Type I Error evaluation

$k = 1$						$k = 2$							
Power			Size			Power			Size				
Wald	LRT	SR	Wald	LRT	SR	Wald		LRT	SR	Wald		LRT	SR
						1	2			1	2		
99.7%	99.5%	90.5%	.05	.012	.26	85.5%	66.7%	74%	95%	.026	.017	.002	.235

Also included in Table 3.1 are the powers and type I errors of the stepwise regression

method [6]. They are calculated with the R package SPATCLUS [7] and on the same $2 \times 2 \times 600 = 2400$ simulated data sets. In order to apply this method, 3000 underlying population data points are simulated uniformly in the region I. This 3000 number is selected similarly as for the real data we will analyze in Section 3.4.2. In the single cluster case, the stepwise regression method has a power 90.5% with an inflated type I error .26. In the two cluster case, the power to detect one or more clusters is 95% with an inflated Type I error .235. For the particular type of data from model (3.1), it appears that our tests are a little conservative while the stepwise regression method has inflated type I errors.

Table 3.2: Cluster region estimates evaluation (%)

	Method	Statistics	Min	1^{st} QT	Median	3^{rd} QT	Max	Mean	sd
$k = 1$	Mean based	Sensitivity	7.50	96.83	100.00	100.00	100.00	97.19	7.814
		Specificity	52.08	79.11	87.50	92.31	100.00	85.24	10.064
		PPV	37.50	86.94	91.18	95.16	100.00	90.27	7.366
		NPV	59.78	94.97	100.00	100.00	100.00	96.80	5.867
	Median based	Sensitivity	5.00	95.29	100.00	100.00	100.00	96.58	8.302
		Specificity	52.08	81.08	88.60	94.29	100.00	86.57	10.010
		PPV	40.00	88.03	92.36	95.80	100.00	91.06	7.344
		NPV	60.00	93.68	100.00	100.00	100.00	96.11	6.265
	SR	Sensitivity	0.00	3.33	9.43	11.38	15.38	7.88	4.309
		Specificity	89.58	95.12	97.37	97.87	100.00	96.81	2.352
		PPV	0.00	62.50	75.00	87.50	100.00	73.01	24.446
		NPV	28.57	38.95	42.39	46.74	60.87	42.84	5.309
$k = 2$	Mean based	Sensitivity	0.00	80.91	92.81	98.21	100.00	85.63	18.123
		Specificity	0.00	88.17	93.88	97.00	100.00	90.71	10.508
		PPV	0.00	80.00	88.14	93.94	100.00	85.25	12.234
		NPV	0.00	89.88	95.65	98.81	100.00	93.25	7.723
	Median based	Sensitivity	0.00	85.71	93.33	98.25	100.00	89.26	13.695
		Specificity	0.00	89.90	95.63	97.96	100.00	92.01	10.352
		PPV	38.33	83.33	91.53	96.49	100.00	88.23	11.238
		NPV	0.00	92.47	96.02	98.85	100.00	94.75	6.390
	SR	Sensitivity	0.00	3.54	6.12	8.11	13.33	5.93	3.107
		Specificity	93.27	95.96	97.09	98.91	100.00	97.32	1.651
		PPV	0.00	50.00	50.00	75.00	100.00	57.01	22.220
		NPV	54.23	61.97	64.79	67.61	74.65	64.75	4.067

We also use the simulation to examine how accurately our proposed method can

identify the cluster regions. Table 3.2 lists the summary statistics of sensitivity, specificity, PPV and NPV values in the 600 repeated simulations for the two cluster region detection methods described in Subsection 3.2.4 as well as the stepwise regression method [7, 6]. Note, with the disc-based wrap method and a possible final union of the interacted proportions, it is not quite straightforward for the SR method to numerically identify the case points inside the final potential clusters. Therefore, only the case points between the selected cluster bounds ("breaks") are identified inside the clusters. This may underestimate the number of estimated inside case points and thus decrease sensitivity and increase specificity a little. The majority of these measurements by the mean and median based cluster region detection methods are around 90%. These indicate both methods can identify cluster regions accurately. The stepwise regression cluster detection method, however, gives poor sensitivity and NPV. It seems even though our estimates cover a little bigger area than the true clusters, and thus have lower Specificity and PPV, they are better than the stepwise regression method estimates for the particular type of data from model (3.1).

Table 3.3: Model selection evaluation (%)

	AIC			BIC		
	Estimated			Estimated		
	$k=1$	$k=2$	$k=3$	$k=1$	$k=2$	$k=3$
True $k=1$	76.0	21.0	3.0	93.5	6.0	.5
$k=2$	56.0	29.5	14.5	77.0	19.5	3.5

To study the performance of the proposed AIC and BIC criteria in determining the number of clusters, we use the procedures in Section 3.3 and define $\mathcal{K} = 1, 2, 3$ as the pre-selected set of ks. For each fixed $k \in \mathcal{K}$, we estimate the values of $\widehat{\text{AIC}}(k)$ and $\widehat{\text{BIC}}(k)$. We then pick the $\hat{k}$s which minimize $\widehat{\text{AIC}}(k)$ and $\widehat{\text{BIC}}(k)$ respectively to estimate the true number of clusters. Table 3.3 summarizes the model selection results using AIC and BIC criteria. In the single cluster case, 76% of the time AIC criterion, and 93.5% of the time BIC criterion choose the true cluster number $k = 1$. However, in the two cluster case, the selection percentage accuracy is only in the thirties with the AIC method, and in the twenties with the BIC method. Most of the misses are in the

estimated $k = 1$ category. This may be mainly because of the small size of the second cluster for the two clusters case.

Even though the simulation results here seem not as good as in Section 2.4, the method still works well. The α_s estimators are reasonable. Most of the time, the cluster region estimators are accurate. The Wald and LRT tests have high power to detect clusters at least in the cases of clusters within which an event is about 3.5 times likely to happen per unit area than that outside. Even though the tests are conservative, they are better than having inflated type I errors. The AIC and BIC criteria for two clusters detection do not perform well.

3.4.2 Pharmacy clusters in Montpellier

We implement our modeling procedure both with and without underlying population adjustment to analyze the Pharmacy clusters in Montpellier (a Frence city) data set previously analyzed by Demattei et. al [6]. This data set consists of 99 pharmacies in Montpellier located by the global positioning system (GPS). Previously, this data set was analyzed with the underlying inhomogeneous population taken into account. The population is defined by 30 IRIS Montpellier divisional system and obtained from the French 1999 population census. Since the population data is not at the same aggregation level as the case data (pharmacies), a new underlying population was built by simulating a uniform point process in each IRIS with size proportional to the IRIS population. Refer to Figure 3.3 for this simulated data. The stepwise regression method detected a significant cluster of pharmacies in the town centre with a p-value .0002. The cluster is shown in Figure 3.5 in dark grey. Also in this figure, the red circle represents the cluster located by the circular based spatial scan with Poisson model assumption. This cluster was also detected as significant with a p-value .002.

Here, we reanalyze this data set using our method both with circular and elliptical shaped clusters. We select the set of k's $\kappa = \{1, 2, 3\}$. First, the data is analyzed without underlying background population adjustment (i.e. with model (3.1)). Both AIC and

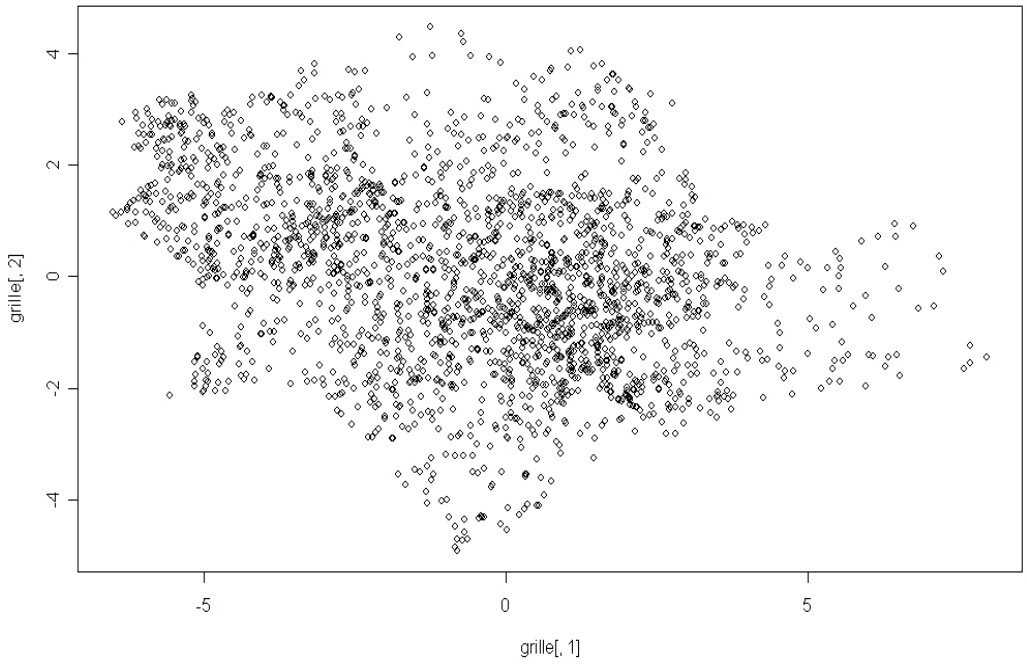

Figure 3.3: Simulated underlying population distribution of Montpellier in 1999

BIC criteria select the single cluster model. When analyzed with circular shaped clusters, the estimated $\alpha = 12.512$ via our Monte-Carlo EM algorithm in Subsection 3.2.2. This α is significantly different from 1 with both Wald and LRT test (P-value$\ll$.001) presented in Subsection 3.2.3. The estimated cluster is denoted in Figure 3.4 as the blue circle. Since it covers 25 events, we constrain the size of our elliptical cluster to be no bigger than that able to cover more than 25% of the total events when analyzed with elliptical-shaped cluster. The estimated α becomes 12.531, which is also significantly different from 1 with both Wald and LRT test (P-value$\ll$.001). The estimated cluster is denoted in Figure 3.4 as the red ellipsis. In order to compare our method with the SR and Scan methods, we build a second underlying population by simulating a uniform point process in the whole region and use it to reanalyze the case data with these two methods. The cluster detected with the stepwise regression method is shown in Figure 3.4 in dark grey and the spatial scan cluster is shown as the green circle. Both methods detect their clusters to be significant with a .001 p-value for the spatial scan test (the p-value for the SR method can not be directly achieved from the SPATCLUS package).

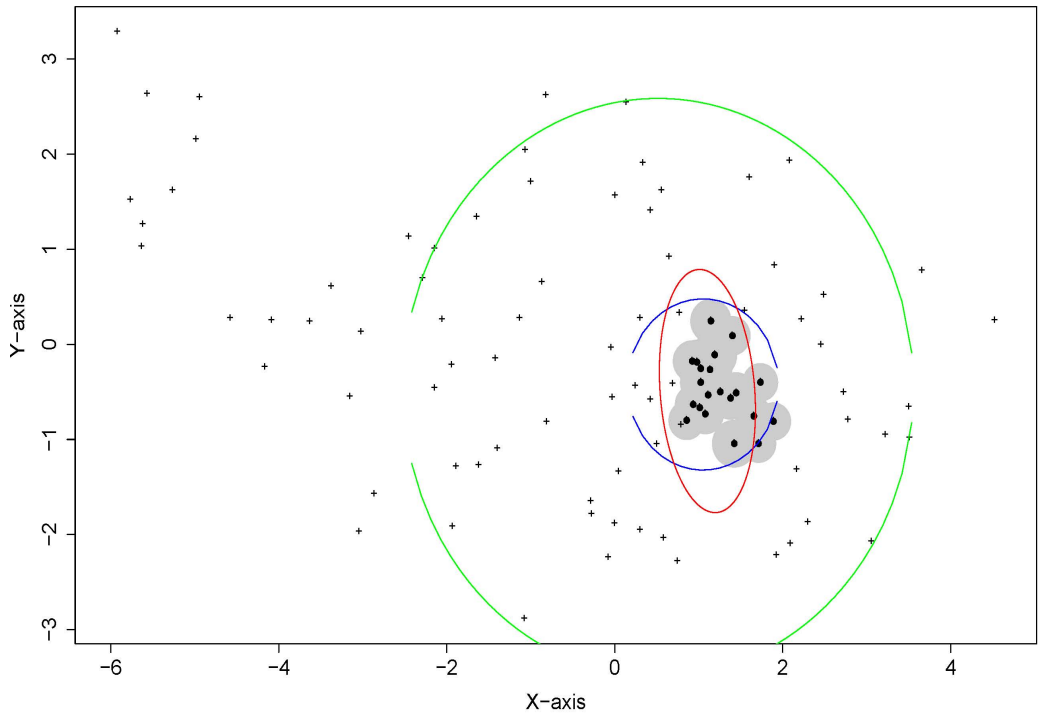

Figure 3.4: Cluster regions located using spatial scan statistic(green circle), stepwise regression(dark grey) and our method(blue circle & red ellipsis) without population adjustment

We can see that our estimates matches with the SR method result while the spatial scan greatly inflates the cluster region.

In order to adjust for the underlying population inhomogeneity, the underlying population density function can be naturally chosen as the baseline function $B(x^{(1)}, x^{(2)})$ in model (3.2), which together with the derived formulas in Appendix A, can then be used to analyze the data with population adjustment. When analyzed with circular shaped clusters, the single cluster model is selected again, with decreased estimated $\hat{\alpha} = 4.479$. This estimate is tested significantly different from 1 with both Wald (P-value= .02) and LRT tests (P-value$\ll$.001). The estimated cluster is shown also as the blue circle in Figure 3.5. Since it covers 16 events, we constrain the size of our elliptical cluster to be no bigger than that able to cover more than 20% of the events when analyzed with elliptical-shaped clusters. The single cluster model is selected a second time, with decreased estimated $\hat{\alpha} = 4.506$. This estimate is significantly different from 1 with both

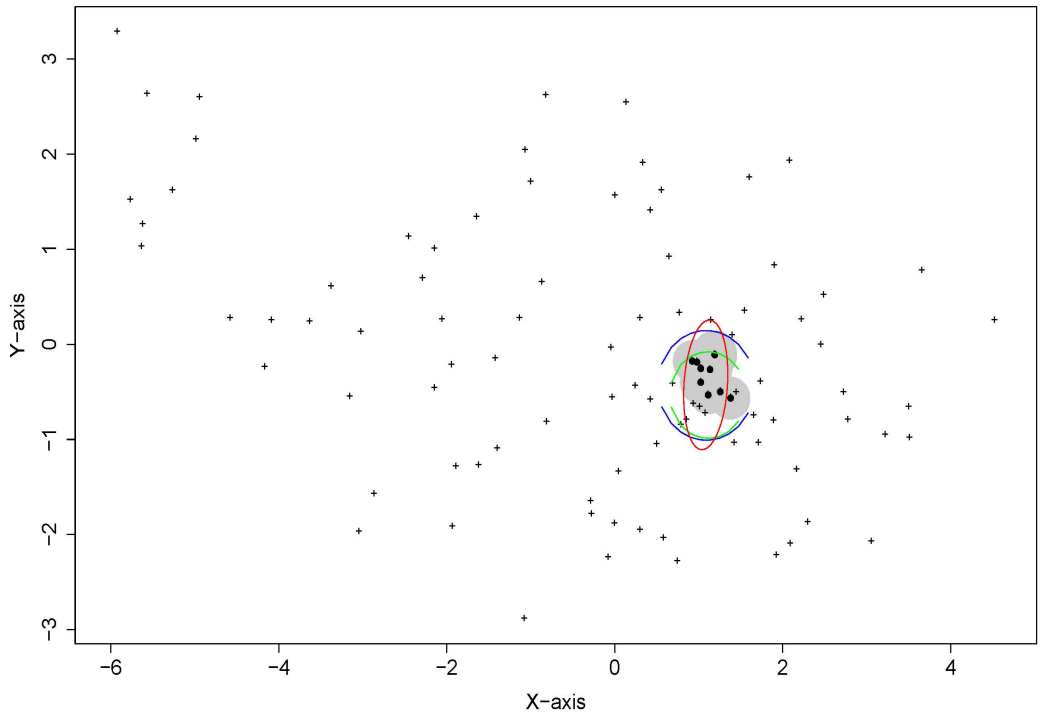

Figure 3.5: Cluster regions located using spatial scan statistic(green circle), stepwise regression(dark grey) and our method(blue circle & red ellipsis) with population adjustment

Wald (P-value= .02) and LRT tests (P-value$\ll$.001). The estimated cluster is shown as the red ellipse in Figure 3.5. Compared with the cluster detected with the stepwise regression method (the dark grey area in Figure 3.4) and the spatial scan cluster (the green circle in Figure 3.4), our estimates match with them with a little bigger sizes.

3.5 Discussion

We extend in this chapter the modeling and inference framework in Chapter 2 for spatial cluster detection. Because of the natural orderless property of the clusters in two-dimensional space, spatial cluster detection is always more difficult than for the temporal case. Therefore, it is not surprising that our method in this chapter does not work as well as in the previous chapter. In the temporal case, the waiting times $\mathbf{b} = (b_1, b_2, \ldots, b_{k+1})'$ and the cluster lengths $\mathbf{c} = (c_1, c_2, \ldots, c_k)'$ can be well

modeled with the help of each other because of the ordering property along the time axis. However, the spatial clusters have to be modeled differently. Taking this complex situation into consideration, our methodology performs well for the spatial case.

Here, we are able to use our approach to detect the elliptical clusters. Possible extensions can be to detect other shaped clusters. For example, a potential remedy is to have a collection of candidate shapes of various types, where each of the shapes has a certain probability to be selected as a candidate choice for a cluster shape. With enough computation, we can use training samples to obtain these candidate shapes as well as a probability model from which the shapes are selected. We can have latent modeling assumptions on these clusters and similar piecewise uniform distribution to model the observations ($\mathbf{y}_i$'s). We can then use a similar approach to detect the clusters. Another method for the irregular cluster detection can be realized in the following manner: after obtaining the estimated cluster regions by our approach, we can use the same disc-wrap method as the stepwise regression method to get irregular shaped clusters. Now we draw discs around all the points inside the estimated elliptical cluster regions.

Chapter 4

Conclusions and Discussions

Statistical modeling is one of the most widely used tools in modern applied statistics. A model that mimics the sample data generation process retrieves more information and provides greater insight into the problem. We present in this thesis latent models for temporal or spatial clusters and sample data generation processes. Based on this intuitive modeling framework, we develop likelihood inference based detection procedures and Monte-Carlo EM algorithms to identify individual clusters, estimate cluster locations and sizes in either temporal or spatial data. Simulation studies and real data analysis illustrate the efficiency of our proposed methodology.

The scan statistics procedures focus on the detection of the most unusual cluster instead of multiple ones. Our method can detect multiple clusters simultaneously. Unlike many existing approaches, our approach need not limit the potential clusters to finite sets. We can detect clusters of varying sizes. Compared with the stepwise regression method, our method has increased efficiency under the simulation study settings. Moreover, this latent modeling approach can flexibly adjust for non-uniform background variation. It can also be easily extended to three or higher dimensional cluster detection.

The likelihood inference presented in Subsections 2.2.3 and 3.2.3 are for two sided tests, which are appropriate to test either unusually dense ($\alpha_j > 1$) or sparse ($\alpha_j < 1$) clusters. In some applications, we may be interested in detecting only one type of cluster and one-sided tests are more suitable. Note that, if we limit the parameter space to $\{\alpha_j \geq 1, \text{ for } j = 1, 2, \ldots, k\}$ or $\{\alpha_j \leq 1, \text{ for } j = 1, 2, \ldots, k\}$, the point with $\alpha_j = 1$ is on the boundary of the parameter space. In this case, the constrained likelihood inference [43, 44] applies. If the one sided tests are for each single α_j or in the case of k=1, the

Wald type tests and the likelihood ratio tests described in Subsections 2.2.3 and 3.2.3 can be directly extended to one sided tests by dividing the p-values in half. For multiple α_j's, there are complications when using the likelihood ratio tests. This is inherited from the well known fact that the likelihood ratio tests (similar to the F test for equality of multiple population means or regression parameters) are not well suited for multiple parameter one sided tests. In this case, the likelihood ratio test statistic R^{**} is no longer asymptotically chi-square distributed. It instead follows a chi-bar-square distribution. This is the distribution of the weighted sum of several independent chi-squared variants, where the weights depend on the eigenvalues of the information matrices [44]. A useful algorithm to compute p-values for these types of testing problems has been provided in [44]. We can incorporate this algorithm to our problem, with few alterations to the Monte-Carlo EM parameter estimation and cluster location and size identification procedures. The theory for constrained likelihood inference is much more complex than for regular likelihood. The p-values from the constrained likelihood approach are usually smaller than those from the regular likelihood inference [44]. With some power loss, one can conservatively use the two-sided p-values for one-sided testing problems.

One limiting point of our method is the computation time. Mainly because of the model selection part, the analysis procedure costs more time as k increases, which makes the whole procedure less efficient. However, with the rapid development of current computational methods, the computation time may not be a problem in the near future.

Appendix A

Formulas with Baseline Function Incorporation for Chapter 3

With $B(x_1, x_2)$ the baseline function, from the generalized model (3.2),

$$f_\theta(y|\mathbf{O},\phi,\mathbf{a},\mathbf{b},k) = \begin{cases} \frac{\alpha_1 B(y^{(1)},y^{(2)})}{\tilde{E}+\sum_{j=1}^{k}(\alpha_j-1)\tilde{D}_j}, & \text{if } y \in A_1 \\ \qquad\qquad\cdots\cdots & \\ \frac{\alpha_k B(y^{(1)},y^{(2)})}{\tilde{E}+\sum_{j=1}^{k}(\alpha_j-1)\tilde{D}_j}, & \text{if } y \in A_k \\ \frac{B(y^{(1)},y^{(2)})}{\tilde{E}+\sum_{j=1}^{k}(\alpha_j-1)\tilde{D}_j}, & \text{if } y \notin \cup_{j=1}^{k} A_j \end{cases} \tag{A.1}$$

(3.5) becomes,

$$\begin{aligned} & f_\theta(\mathbf{y}|\mathbf{O},\mathbf{r},k) = \prod_{i=1}^{n} f(y_i|\mathbf{O},\phi,\mathbf{a},\mathbf{b},k) \\ = \; & \frac{\prod_{j=1}^{k}\{\prod_{y_i\in A_j}[\alpha_j B(y_i^{(1)},y_i^{(2)})]\}\{\prod_{y_i\notin\cup_{j=1}^{k}A_j}[B(y_i^{(1)},y_i^{(2)})]\}}{[\tilde{E}+\sum_{j=1}^{k}(\alpha_j-1)\tilde{D}_j]^n} = \frac{\prod_{j=1}^{k}\alpha_j^{Z_j}\prod_{i=1}^{n}B(y_i^{(1)},y_i^{(2)})}{[\tilde{E}+\sum_{j=1}^{k}(\alpha_j-1)\tilde{D}_j]^n} \\ = \; & e^{\sum_{j=1}^{k}[(\log\alpha_j)Z_j]-n\log[\tilde{E}+\sum_{j=1}^{k}(\alpha_j-1)\tilde{D}_j]+\sum_{i=1}^{n}\log[B(y_i^{(1)},y_i^{(2)})]}, \end{aligned} \tag{A.2}$$

where $Z_j = Z_j(\mathbf{y},\mathbf{O},\phi,\mathbf{a},\mathbf{b}) = \sum_{i=1}^{n}\mathbf{1}_{\{y_i\in A_j\}}$, number of incidences inside jth cluster

Detailed formula of 3.4 is,

$$\begin{aligned} & f_\theta(\mathbf{O},\phi,\mathbf{a},\mathbf{b}|\delta=k) = \frac{f_\theta(\mathbf{O},\phi,\mathbf{a},\mathbf{b},\delta=k)}{P_\lambda(\delta=k)} = \frac{f_\theta(\mathbf{O},\phi,\mathbf{a},\mathbf{b})f_\theta(\delta=k|\mathbf{O},\phi,\mathbf{a},\mathbf{b})}{P_\lambda(\delta=k)} \\ = \; & \frac{k\prod_{j=1}^{k}[\psi_{\mathbf{o}}(\mathbf{o}_j)\psi_\phi(\phi_j)2\psi_r(a_j)\psi_r(b_j)]\mathbf{1}_{\{\delta=k\}}}{P_\lambda(\delta=k)}. \end{aligned} \tag{A.3}$$

Equation (3.8) becomes,

$$\begin{aligned} & f_\theta(\mathbf{y},\mathbf{O},\phi,\mathbf{a},\mathbf{b},k) = f_\theta(\mathbf{y}|\mathbf{O},\phi,\mathbf{a},\mathbf{b},k) f_\theta(\mathbf{O},\phi,\mathbf{a},\mathbf{b}|k) P_\lambda(\delta=k) \\ = & \frac{\prod_{j=1}^k \alpha_j^{Z_j} \prod_{i=1}^n B(y_i^{(1)},y_i^{(2)})}{[\tilde{E}+\sum_{j=1}^k(\alpha_j-1)\tilde{D}_j]^n} k \prod_{j=1}^k [\psi_{\mathbf{o}}(\mathbf{o}_j)\psi_\phi(\phi_j)2\psi_r(a_j)\psi_r(b_j)]\mathbf{1}_{(\delta=k)} \end{aligned} \quad (A.4)$$

From (A.4), the complete log-likelihood function

$$\begin{aligned} & \ell(\theta) = \ell(\theta|\mathbf{O},\phi,\mathbf{a},\mathbf{b},\mathbf{y},\delta=k) = \log f_\theta(\mathbf{y},\mathbf{O},\phi,\mathbf{a},\mathbf{b},k) \\ = & \sum_{j=1}^k [(\log\alpha_j)Z_j] + \sum_{i=1}^n \log[B(y_i^{(1)},y_i^{(2)})] - n\log[\tilde{E}+\sum_{j=1}^k(\alpha_j-1)\tilde{D}_j] \\ & + \sum_{j=1}^k [\log\psi_{\mathbf{o}}(\mathbf{o}_j)+\log\psi_\phi(\phi_j)+\log\psi_r(a_j)+\log\psi_r(b_j)] + \log[\mathbf{1}_{(\delta=k)}] \end{aligned} \quad (A.5)$$

Thus,

$$Q(\theta|\theta^{(s)}) = E[\ell(\theta)|\mathbf{y},k,\theta^{(s)}] = Q_1(\alpha|\theta^{(s)}) + Q_2(\lambda|\theta^{(s)}) + Q_3 \quad (A.6)$$

where

$$\begin{aligned} Q_1(\alpha|\theta^{(s)}) &= \sum_{j=1}^k \mathrm{E}(Z_j|\mathbf{y},k,\theta^{(s)})\log\alpha_j - n\mathrm{E}\{\log[\tilde{E}+\sum_{j=1}^k(\alpha_j-1)\tilde{D}_j]|\mathbf{y},k,\theta^{(s)}\}, \\ Q_2(\lambda|\theta^{(s)}) &= \sum_{j=1}^k \mathrm{E}[\log\psi_{\mathbf{o}}(\mathbf{o}_j)+\log\psi_\phi(\phi_j)+\log\psi_r(a_j)+\log\psi_r(b_j)|\mathbf{y},k,\theta^{(s)}] \\ Q_3 &= \sum_{i=1}^n \log[B(y_i^{(1)},y_i^{(2)})]. \end{aligned} \quad (A.7)$$

Equation (3.12) becomes

$$\begin{aligned} & f(A_j|A_l,l=1,2,\ldots,k,l\neq j,\mathbf{y},k) = \frac{f(\mathbf{O},\phi,\mathbf{a},\mathbf{b},\mathbf{y},k)}{\int f(\mathbf{O},\phi,\mathbf{a},\mathbf{b},\mathbf{y},k)d\mathbf{o}_j d\phi_j da_j db_j} \\ = & \frac{\frac{\prod_{l=1}^k \alpha_l^{Z_l}\prod_{i=1}^n B(y_i^{(1)},y_i^{(2)})}{[\tilde{E}+\sum_{l=1}^k(\alpha_l-1)\tilde{D}_l]^n}\psi_{\mathbf{o}}(\mathbf{o_j})\psi_\phi(\phi_j)\psi_r(a_j)\psi_r(b_j)\mathbf{1}_{(\delta=k)}}{\int \frac{\prod_{l=1}^k \alpha_l^{Z_l}\prod_{i=1}^n B(y_i^{(1)},y_i^{(2)})}{[\tilde{E}+\sum_{l=1}^k(\alpha_l-1)\tilde{D}_l]^n}\psi_{\mathbf{o}}(\mathbf{o_j})\psi_\phi(\phi_j)\psi_r(a_j)\psi_r(b_j)\mathbf{1}_{(\delta=k)}d\mathbf{o}_j d\phi_j da_j db_j} \\ \propto & \frac{\alpha_j^{Z_j}}{[\tilde{E}+\sum_{l=1}^k(\alpha_l-1)\tilde{D}_l]^n}\psi_{\mathbf{o}}(\mathbf{o_j})\psi_\phi(\phi_j)\psi_r(a_j)\psi_r(b_j)\mathbf{1}_{(\delta=k)}, \end{aligned} \quad (A.8)$$

From (A.5), detailed procedure of the information matrix,

$$\frac{\partial}{\partial\theta}\ell(\theta) = (\frac{\partial}{\partial\alpha_1}\ell(\theta),\cdots,\frac{\partial}{\partial\alpha_k}\ell(\theta),\frac{\partial}{\partial\lambda_o}\ell(\theta),\frac{\partial}{\partial\lambda_\phi}\ell(\theta),\frac{\partial}{\partial\lambda_r}\ell(\theta))^T \quad (A.9)$$

where,

$$\frac{\partial}{\partial\alpha_l}\ell(\theta)=\frac{Z_l}{\alpha_l}-\frac{n\tilde{D}_l}{[\tilde{E}+\sum_{j=1}^{k}(\alpha_j-1)\tilde{D}_j]},l=1,2,\ldots,k$$

$$\frac{\partial}{\partial\lambda_{\mathbf{o}}}\ell(\theta)=\sum_{j=1}^{k}\frac{\psi_{\mathbf{o}}^{'}(\mathbf{o}_j)}{\psi_{\mathbf{o}}(\mathbf{o}_j)},\quad \frac{\partial}{\partial\lambda_{\phi}}\ell(\theta)=\sum_{j=1}^{k}\frac{\psi_{\phi}^{'}(\phi_j)}{\psi_{\phi}(\phi_j)},\quad \frac{\partial}{\partial\lambda_{r}}\ell(\theta)=\sum_{j=1}^{k}[\frac{\psi_{r}^{'}(a_j)}{\psi_{r}(a_j)}+\frac{\psi_{r}^{'}(b_j)}{\psi_{r}(b_j)}]$$

and

$$\frac{\partial^2}{\partial\theta^2}\ell(\theta)=\begin{pmatrix}\frac{\partial^2}{\partial\alpha_1^2}\ell(\theta) & \cdots & \frac{\partial}{\partial\alpha_1\alpha_k}\ell(\theta) & 0 & 0 & 0\\ \vdots & \cdots & \vdots & \vdots & \vdots & \vdots\\ \frac{\partial^2}{\partial\alpha_k\alpha_1}\ell(\theta) & \cdots & \frac{\partial^2}{\partial\alpha_k^2}\ell(\theta) & 0 & 0 & 0\\ 0 & \cdots & 0 & \frac{\partial^2}{\partial\lambda_{\mathbf{o}}^2}\ell(\theta) & 0 & 0\\ 0 & \cdots & 0 & 0 & \frac{\partial^2}{\partial\lambda_{\phi}^2}\ell(\theta) & 0\\ 0 & \cdots & 0 & 0 & 0 & \frac{\partial^2}{\partial\lambda_{r}^2}\ell(\theta)\end{pmatrix}_{(k+3)\times(k+3)} \tag{A.10}$$

where,

$$\frac{\partial^2}{\partial\alpha_l\partial\alpha_i}\ell_k(\theta)=\frac{n\tilde{D}_l\tilde{D}_i}{[\tilde{E}+\sum_{j=1}^{k}(\alpha_j-1)\tilde{D}_j]^2},\quad \frac{\partial^2}{\partial\alpha_l^2}\ell_k(\theta)=-\frac{Z_l}{\alpha_l^2}+\frac{n\tilde{D}_l^2}{[\tilde{E}+\sum_{j=1}^{k}(\alpha_j-1)\tilde{D}_j]^2}$$

$$\frac{\partial^2}{\partial\lambda_{\mathbf{o}}^2}\ell_k(\theta)=\sum_{j=1}^{k}\{\frac{\psi_{\mathbf{o}}^{''}(\mathbf{o}_j)}{\psi_{\mathbf{o}}(\mathbf{o}_j)}-[\frac{\psi_{o}^{'}(o_j)}{\psi_{\mathbf{o}}(\mathbf{o}_j)}]^2\},$$

$$\frac{\partial^2}{\partial\lambda_{\phi}^2}\ell_k(\theta)=\sum_{j=1}^{k}\{\frac{\psi_{\phi}^{''}(\phi_j)}{\psi_{\phi}(\phi_j)}-[\frac{\psi_{\phi}^{'}(\phi_j)}{\psi_{\phi}(\phi_j)}]^2\},$$

$$\frac{\partial^2}{\partial\lambda_{r}^2}\ell_k(\theta)=\sum_{j=1}^{k}\{\frac{\psi_{r}^{''}(a_j)}{\psi_{r}(a_j)}-[\frac{\psi_{r}^{'}(a_j)}{\psi_{r}(a_j)}]^2+\frac{\psi_{r}^{''}(a_j)}{\psi_{r}(b_j)}-[\frac{\psi_{r}^{'}(b_j)}{\psi_{r}(b_j)}]^2\}$$

Equation (3.16) becomes,

$$\begin{aligned}R &= 2\log\{\frac{\max_{H_1}f_{\theta}(\mathbf{y},k)}{\max_{H_0}f_{\theta}(\mathbf{y},k)}\}\\ &= 2\log\Big\{\frac{\int\int f_{\hat{\theta}}(\mathbf{y}|\mathbf{O},\phi,\mathbf{a},\mathbf{b},k)f_{\hat{\theta}}(\mathbf{O},\phi,\mathbf{a},\mathbf{b}|k)d\mathbf{O}d\phi d\mathbf{a}d\mathbf{b}P_{\hat{\lambda}}(\delta=k)}{\max_{H_0}e^{\sum_{j=1}^{k}\{\sum_{y_i\in A_j}\log[B(y_i^{(1)},y_i^{(2)})]\}-n\log(\tilde{E})}P_{\lambda}(\delta=k)}\Big\}\\ &= 2\Big\{\log\int\int f_{\hat{\theta}}(\mathbf{y}|\mathbf{O},\phi,\mathbf{a},\mathbf{b},k)f_{\hat{\theta}}(\mathbf{O},\phi,\mathbf{a},\mathbf{b}|k)d\mathbf{O}d\phi,\mathbf{a},\mathbf{b}+\log P_{\hat{\lambda}}(\delta=k)\\ &\quad -\sum_{i=1}^{n}\log[B(y_i^{(1)},y_i^{(2)})]+n\log(\tilde{E})-\max_{\lambda}[\log P_{\lambda}(\delta=k)]\Big\}\end{aligned} \tag{A.11}$$

When $B(x_1,x_2)$ is population density function, $\tilde{E}=\int\int_I B(x_1,x_2)dx_1dx_2=N$,

the total population inside region I, $\tilde{D}_j = \int\int_{A_j \bigcap I} B(x_1, x_2)dx_1dx_2 = N_j$, number of population inside jth cluster

When $B(x_1, x_2) \equiv 1$, $\tilde{E} = \int\int_I B(x_1, x_2)dx_1dx_2 = E$, $\tilde{D}_j = \int\int_{A_j \bigcap I} B(x_1, x_2)dx_1dx_2 = D_j$, it becomes no background situation.

Appendix B

Real Data Sets Analyzed in Chapter 2

Hospital Hemoptysis Admission Data Set

Days of hemoptysis admission at Nice University Hospital from January 1to December 31, 1995, are as follows:

2 8 23 29 43 48 58 60 61 63 69 71 74 74 78 80 85 86 86 87 93 105 106 108 115 117 121 126 135 140 141 156 159 179 187 188 188 191 191 198 201 214 225 225 235 235 239 249 262 271 279 279 282 292 296 302 317 323 337 342 352 354.

Revised brucellosis counts per week and predicted values

1997-2003 Data Average:

0.86 1.00 1.29 0.72 1.15 1.43 0.86 1.29 1.86 1.29 2.00 1.58 1.29 1.29 1.15 2.00 0.72 1.29 2.43 2.15 3.58 1.86 1.00 2.00 3.00 2.58 3.29 2.15 2.29 3.29 3.43 2.72 2.43 1.58 2.58 2.86 3.29 3.00 2.72 1.86 1.72 2.43 3.58 2.00 1.29 2.00 1.29 2.43 3.15 2.15 3.43 7.15

2004 Data: 0 0 0 0 2 3 1 1 0 5 4 1 1 0 3 2 0 1 1 4 3 2 4 6 0 9 3 2 5 4 0 10 0 0 3 1 8 5 5 4 4 5 0 27 7 12 0 5 6 6 4 2

Appendix C

R and C Codes for Chapter 2

C.1 Main R codes stored in "RunAnalyzeBack.txt"

```
source("AnalyzeBack.txt")    # Call R subroutines stored in "AnalyzeBack.txt"
    library(MASS)    # Upload needed R package
    # Population at risk function (2.27) in Section 2.5.1
    Wy.back <- function(x) ifelse((x>182/365)&(x<=244/365), 1+72*x/10000+55/355, 1+72*x/10000)
    min.NlPdk <- scan("minimum.txt")    #
    # Main function to analyze the data
    ABIC.fun(
      kk.vec=1:3,    # Pre-specified set of ks κ
      hospital.data,    # Observed data: vector of length nn
      min.NlPdk,    tt=1,    tt.scale=365,    # Re-scale the observed data
      n.gib1=100, m.gib1=50, iter1=5, tol1=.1,    # Parameters for model estimation
      n.gib=1000, m.gib=501, iter=50, tol=0.001,    # Parameters for model estimation
      sig=0.05, Ln.gib=50000, Lm.gib=30001    # Parameters for likelihood inference
      )
```

C.2 R subroutines stored in "AnalyzeBack.txt"

C.2.1 Model estimation in Section 2.2.2

```
# EM Iteration and Information Matrix
    EM.gib.k <- function(
      kk=1,    # Given number of clusters: integer ∈ κ
      ys=hospital.data,    # Transformed observed data: vector of length nn
      nn=length(ys),    # # of observed data
      tt=1,    # Given time window [0,T]
```

```
  betas.0=rep(log(kk+1),kk),    # Initial values of log(α_s)
  lam1.0=2*kk,    # Initial value of λ_b
  lam2.0=2*kk,    # Initial value of λ_c
  bs.0,    # Initial waiting times: vector of length kk+1
  cs.0,    # Initial clusters lengths: vector of length kk
  iterUp=100,    # Maximum EM iteration steps
  tol=.005,    # Convergence criterion
  nn.gibbs=10000,    # Gibbs sampling
  mm.gibbs=5000,    # Gibbs sampling "Burn-in"
  tt.tu    # Integration of background function over [0,T]: T̃
  )
{
# EM Parameter estimate
  betas=betas.0; lam1=lam1.0; lam2=lam2.0; MM=nn.gibbs-mm.gibbs
  converge <- F
  for (iter in 1:iterUp) {
    bc.gibbs <- rep(0, nn.gibbs*(2*kk+1))
    # call EM Gibbs sampling subroutine NN.gibbs
    bc.gibbs <- NN.gibbs(bc.gibbs,nn.gibbs,bs.0,cs.0,betas,lam1,lam2,ys,nn,kk,tt,tt.tu)[[1]]
    bc.gibbs <- matrix(bc.gibbs, nrow=nn.gibbs, ncol=2*kk+1, byrow=T)
    bc.use <- bc.gibbs[(mm.gibbs+1):nn.gibbs,]
    bs.use <- bc.use[,1:(kk+1)]; cs.use <- as.matrix( bc.use[,(kk+2):(2*kk+1)] )
    # call subroutine ZL.fun to calculate z_j and c̃_j
    tmpout <- apply(bc.use, 1, ZL.fun, kk, ys )
    Zs <- as.matrix(tmpout[1:kk,]); LL.tu <- as.matrix(tmpout[(kk+1):(2*kk),])
    # call subroutine Nll.betas to update parameter estimate logα
    betas.new<-optim(betas,Nll.betas,d1Nll.betas,Zs=Zs,LL=LL.tu, tt=tt.tu,nn=nn,kk=kk)§par
    # update parameter estimate λ_b and λ_c
    lam1.new <- (kk+1)*MM/sum(bs.use); lam2.new <- kk*MM/sum(cs.use)
    # update waiting times and cluster lengths
    bs.0 <- apply( bs.use, 2 ,median ); cs.0 <- apply( cs.use, 2, median )
```

```
    if ( (sum(abs(betas-betas.new)<tol*abs(betas)+tol)==kk)*(abs(lam1-lam1.new)<tol*abs(lam1)+tol)
*(abs(lam2-lam2.new)<tol*abs(lam2)+tol)==1 ) { converge <- T; break }
    # update parameter estimates
    betas <- betas.new; lam1 <- lam1.new; lam2 <- lam2.new
    }
  # Final parameter estimates θ̂
  betas <- betas.new; lam1 <- lam1.new; lam2 <- lam2.new

  # cluster intervals identification in Section 2.2.4
  Ijs.med <- cumsum(c(rbind(bs.0[1:kk],cs.0)))    # Median method
  bs.mean <- colMeans( bs.use ); cs.mean <- colMeans( cs.use )
  Ijs.mean<- cumsum(c(rbind(bs.mean[1:kk],cs.mean)))    # Mean method

  # variance-covariance matrix estimate
  const1 <- (kk+1)/lam1; const2 <- kk/lam2
  abb.3 <- rowSums( bs.use ); abb.4 <- rowSums( cs.use )
  if (kk==1) { abb <- nn*LL.tu*exp(betas)/(tt.tu+(exp(betas)-1)*LL.tu)
    } else { abb<-t(nn*LL.tu*exp(betas))/(tt.tu+colSums((exp(betas)-1)*LL.tu)) }
  abb <- t(abb)
  InfMa.comp <- InfMa.miss.1 <- matrix( 0, kk+2, kk+2 )
  InfMa.comp[1:kk, 1:kk] <- abb %*% t(abb)/nn/MM
  diag(InfMa.comp)[1:kk] <- - rowMeans( as.matrix(abb) ) + diag(InfMa.comp)[1:kk]
  diag(InfMa.comp)[kk+1] <- - const1/lam1; diag(InfMa.comp)[kk+2] <- - const2/lam2
  InfMa.comp <- - InfMa.comp
  InfMa.miss.vecMa <- rbind( Zs-abb, const1-abb.3, const2-abb.4 )
  InfMa.miss.1 <- InfMa.miss.vecMa%*%t(InfMa.miss.vecMa)/MM
  InfMa.miss.2.vec <- rowMeans(InfMa.miss.vecMa)
  InfMa.miss.2 <- InfMa.miss.2.vec %*% t( InfMa.miss.2.vec )
  InfMa.miss <- InfMa.miss.1 - InfMa.miss.2
  InfMa.obs <- InfMa.comp - InfMa.miss; VarMa <- solve( InfMa.obs )
  return( betas, lam1, lam2, iter, converge, Ijs.med, bs.0, cs.0, Ijs.mean, VarMa )
```

```
}

# Negative Q_1 function 2.12
Nll.betas <- function(betas, Zs, LL, tt, nn, kk)
{
  if (kk==1) { -( betas*mean(Zs) - nn*mean(log(tt+(exp(betas)-1)*LL)) )
    } else { -(sum(betas*rowMeans(Zs))-nn*mean(log(tt+colSums((exp(betas)-1)*LL)))) }
}

# 1st derivative of negative Q_1 function
d1Nll.betas <- function(betas, Zs, LL, tt, nn, kk)
{
  if (kk==1) { - mean(Zs) + nn*mean( exp(betas)*LL / ( tt+(exp(betas)-1)*LL ) )
    } else {-rowMeans(Zs)+nn*rowMeans(exp(betas)*LL/(tt+colSums((exp(betas)-1)*LL)))}
}

# Subroutine to calculate z_j and c̃_j
ZL.fun <- function(bs.gibbs.vector, kk, ys )
{
  bs <- bs.gibbs.vector[1:kk]; cs <- bs.gibbs.vector[(kk+2):(2*kk+1)]
  Ijs<-cumsum( c(rbind(bs,cs)) ); Zs <- rep(0,kk); LL.tu <- cs
  for(j in 1:kk){
    Zs[j] <- sum( (ys>Ijs[2*j-1])&(ys<Ijs[2*j]) ) # # of events within j_th cluster
    LL.tu[j] <- integrate(Wy.back, lower=Ijs[2*j-1], upper=Ijs[2*j])§value }
  return( cbind(Zs,LL.tu) )
}

# R and C interface to call C subroutine for EM Gibbs Sampling: (b,c|y,k,θ̂)
NN.gibbs <- function( bc.gibbs,nn.gibbs,bs.0,cs.0,betas,lam1,lam2,ys,nn,kk,tt,tt.tu )
{
  if ( !is.loaded(symbol.C("NNgibbsBack")) ) dyn.load("NgibbsBack.so")
  .C( "NNgibbsBack",
```

```
    as.double(bc.gibbs), as.integer(nn.gibbs), as.double(bs.0), as.double(cs.0),
    as.double(betas), as.double(lam1), as.double(lam2),
    as.double(ys), as.integer(nn), as.integer(kk), as.double(tt), as.double(tt.tu) )
}
```

C.2.2 Likelihood inference in Section 2.2.3

```
# Wald test for H_0 : α_j = 1 vs H_1 : α_j ≠ 1
  TestWald <- function( sig, VarMa, betas, kk )
  {
    betas.var <- diag(VarMa)[1:kk]; Z.stat <- betas/betas.var^0.5
    Pvalue <- pnorm( abs(Z.stat), lower.tail=F )*2; Walds <- (Pvalue<sig)
    return(Walds, Z.stat, Pvalue)
  }

  # Likelihood ratio test for H_0 : α_1 = α_2 = ... = α_k = 1 vs H_1 : at least one α_j ≠ 1
  TestLRT<-function(sig,betas,lam1,lam2,tt,kk,ys,min.NlPdk,nn.gibbs,mm.gibbs,tt.tu)
  {
    bs.0 <- rep( tt/(kk/lam1+kk/lam2)/lam1, times=kk+1 )
    cs.0 <- rep( tt/(kk/lam1+kk/lam2)/lam2, times=kk )
    bc.gibbs <- rep( 0, nn.gibbs*(2*kk+1) )
    # call testing Gibbs sampling subroutine LRT.gibbs
    bc.gibbs <- LRT.gibbs( bc.gibbs, nn.gibbs, bs.0, cs.0, lam1, lam2, kk, tt )[[1]]
    bc.gibbs <- matrix( bc.gibbs, nrow=nn.gibbs, ncol=2*kk+1, byrow=T )
    bc.use <- bc.gibbs[(mm.gibbs+1):nn.gibbs,]
    bs.use <- bc.use[,1:(kk+1)]; cs.use <- as.matrix( bc.use[,(kk+2):(2*kk+1)] )
    tmpout <- apply( bc.use, 1, ZL.fun, kk, ys )
    Zs <- as.matrix( tmpout[1:kk,] ); LL.tu <- as.matrix( tmpout[(kk+1):(2*kk),] )
    if ( kk==1 ) { fy.bck <- exp(betas*Zs)/(tt.tu+(exp(betas)-1)*LL.tu)^nn
      } else {fy.bck<-exp(colSums(betas*Zs)-nn*log(tt.tu+colSums((exp(betas)-1)*LL.tu)))}
    lams <- c(lam1, lam2); logfyk <- log(mean(fy.bck))-NlPdk(lams,tt,kk)
    LRT.stat <- 2*( logfyk + nn*log(tt.tu) + min.NlPdk )
    Pvalue <- pchisq( LRT.stat, df=kk, lower.tail=F ); LRT <- (Pvalue<sig)
```

```
    return(LRT, logfyk, LRT.stat, Pvalue)
  }

  # Negative log-likelihood function of P_λ(δ = k) (2.5)
  NlPdk <- function( lams, tt, kk )
  {
    - ( kk*(log(lams[1]*lams[2]))-lams[1]*tt+2*kk*log(tt)+log(tG.fun(tt*(lams[1]-lams[2]),kk))-
log(kk)-2*log(gamma(kk)) )
  }
  tG.fun<-function( u, kk )
  {
    if ( (kk<=0) || (kk!=as.integer(kk)) ) { cat("Wrong: k should be a positive integer")
      } else { ff <- function(ss) (1-ss)^kk*ss^(kk-1)*exp(u*ss)
        integrate(ff, lower=0, upper=1)§value }
  }

  # R and C interface to call C subroutine for testing Gibbs Sampling: (b,c|k,θ̂)
  LRT.gibbs <- function( bc.gibbs, nn.gibbs, bs.0, cs.0, lam1, lam2, kk, tt )
  {
    if ( !is.loaded(symbol.C("LRTgibbs")) ) dyn.load("LRTgibbs.so")
    .C( "LRTgibbs",
      as.double(bc.gibbs), as.integer(nn.gibbs),
      as.double(bs.0), as.double(cs.0), as.double(lam1), as.double(lam2),
      as.integer(kk), as.double(tt) )
  }
```

C.2.3 Model selection in Section 2.3

```
ABIC.fun <- function( kk.vec, ys, min.NlPdk, tt, tt.scale, n.gib1, m.gib1, iter1, tol1, n.gib,
m.gib, iter, tol, sig, Ln.gib, Lm.gib, nn=length(ys) )
  {
    mm <- length(kk.vec); kk.max <- max(kk.vec)
```

```
iter.vec = converge.vec = logfyk.vec = lam1.vec = lam2.vec = LRT.vec = StatLRT.vec
= PvalueLRT.vec = rep( NA, mm )
betas.Ma = Wald.Ma = PvalueWald.Ma = StatWald.Ma = matrix( , mm, kk.max )
IjsMean.Ma = IjsMed.Ma = MeanCov.Ma = MedCov.Ma = matrix( , mm, 2*kk.max )
VarMa.Ma <- matrix( , sum(kk.vec+2), kk.max+2 )
indi <- 0; Data.rank <- 1:nn; tt.tu <- integrate( Vectorize(Wy.back), 0, tt )§value
for (kk in kk.vec) {
  indi <- indi+1; betas.0 = rep(log(kk+1),kk); lam1.0 = lam2.0 = 2*kk/tt
  bs.0 <- rep( tt/(kk/lam1.0+kk/lam2.0)/lam1.0, times=kk+1 )
  cs.0 <- rep( tt/(kk/lam1.0+kk/lam2.0)/lam2.0, times=kk )
  res1 <- EM.gib.k(kk,ys,nn,tt,betas.0,lam1.0,lam2.0,bs.0,cs.0, iter1,tol1,n.gib1,m.gib1,tt.tu)
  res <- EM.gib.k(kk,ys,nn,tt,res1§betas,res1§lam1,res1§lam2,res1§bs.0,res1§cs.0, iter,tol,n.gib,m.gib,tt.tu)
  iter.vec[indi] <- res§iter; converge.vec[indi] = res§converge
  betas.Ma[indi,1:kk] <- res§betas; lam1.vec[indi] <- res§lam1; lam2.vec[indi] <- res§lam2
  IjsMean.Ma[indi,1:(2*kk)] <- res§Ijs.mean; IjsMed.Ma[indi,1:(2*kk)] <- res§Ijs.med
  VarMa.Ma[( sum(kk.vec[1:indi]+2)-kk-1 ):( sum(kk.vec[1:indi]+2) ), 1:(kk+2)] <- res§VarMa
  for ( ii in 1:kk ){
    Data.logi <- (ys>=IjsMean.Ma[indi,2*ii-1])&(ys<=IjsMean.Ma[indi,2*ii])
    Data.dica <- Data.rank[Data.logi]
    MeanCov.Ma[indi,(2*ii-1):(2*ii)] <- tt.scale*c(ys[Data.dica[1]],ys[Data.dica[length(Data.dica)]])
    Data.logi <- (ys>=IjsMed.Ma[indi,2*ii-1])&(ys<=IjsMed.Ma[indi,ii*2])
    Data.dica <- Data.rank[Data.logi]
    MedCov.Ma[indi,(2*ii-1):(2*ii)] <- tt.scale*c(ys[Data.dica[1]],ys[Data.dica[length(Data.dica)]])
    }
  betasWaldtest <- TestWald(sig, res§VarMa, res§betas, kk)
  Wald.Ma[indi,1:kk] = betasWaldtest§Walds; StatWald.Ma[indi,1:kk] = betasWaldtest§Z.stat
  PvalueWald.Ma[indi,1:kk] = betasWaldtest§Pvalue
  betasLRTest <- TestLRT(sig, res§betas, res§lam1, res§lam2, tt, kk, ys, min.NlPdk[kk],
Ln.gib, Lm.gib, tt.tu)
  LRT.vec[indi] = betasLRTest§LRT; StatLRT.vec[indi] = betasLRTest§LRT.stat
  PvalueLRT.vec[indi] = betasLRTest§Pvalue; logfyk.vec[indi] = betasLRTest§logfyk
  }
```

```
AIC.vec <- 2*(logfyk.vec - kk.vec); BIC.vec <- 2*logfyk.vec - log(nn)*kk.vec
pdf("result/AIC-BIC.pdf")
  matplot( cbind(1:indi,1:indi), cbind(AIC.vec,BIC.vec), type="b", lty=1:2 )
  text( 1:indi, AIC.vec, kk.vec )
dev.off()
indi.AIC <- which.max(AIC.vec); kkopt.AIC <- kk.vec[indi.AIC]
indi.BIC <- which.max(BIC.vec); kkopt.BIC <- kk.vec[indi.BIC]
file <- paste("kk-AIC=", kkopt.AIC, ".pdf", sep="")
pdf(file)
  plot( c(0,tt), c(0,.4), type="n" )
  for (ii in 1:kk) {
    lines( IjsMean.Ma[indi.AIC, (2*ii-1):(2*ii)], c(.1, .1), lty=1 )
    lines( IjsMed.Ma[indi.AIC, (2*ii-1):(2*ii)], c(.2, .2), lty=2 ) }
  points( ys, rep(0, nn) )
dev.off()
if ( kkopt.BIC!=kkopt.AIC ) {
  file <- paste( "result/kk-BIC=", kkopt.BIC, ".pdf", sep="" )
  pdf(file)
    plot( c(0,tt), c(0,.4), type="n" )
    for (ii in 1:kk) {
      lines( IjsMean.Ma[indi.BIC, (2*ii-1):(2*ii)], c(.1, .1), lty=2 )
      lines( IjsMed.Ma[indi.BIC, (2*ii-1):(2*ii)], c(.2, .2), lty=3 ) }
    points( ys, rep(0, nn) )
  dev.off()
  }
sink( "result/result.txt" )
  cat("\n iter, converge: \n\n"); print( cbind(iter.vec, converge.vec) )
  cat("\n kk, Alphas, lam1, lam2:\n\n");
  print( cbind(kk.vec, exp(betas.Ma), lam1.vec, lam2.vec) )
  cat("\n Mean Intervals, Data Coverage:\n\n");
  print( cbind(IjsMean.Ma*tt.scale, MeanCov.Ma) )
  cat("\n Median Intervals, Data Coverage:\n\n")
```

```
        print( cbind(IjsMed.Ma*tt.scale, MedCov.Ma) )
        cat("\n Wald Test, LRT Test:\n\n"); print( cbind(Wald.Ma, LRT.vec) )
        cat("\n Wald.Stat, Wald Pvalue, LRT.Stat, LRT Pvalue:\n\n")
        print( cbind(StatWald.Ma, PvalueWald.Ma, StatLRT.vec, PvalueLRT.vec) )
        cat("\n VarMa:\n\n"); print( VarMa.Ma )
        cat("\n AIC, BIC :\n\n"); print( cbind(AIC.vec, BIC.vec) )
        cat("\n\n The Final Optimal Cluster # by AIC =", kkopt.AIC, "\n\n")
        cat("\n\n The Final Optimal Cluster # by BIC =", kkopt.BIC, "\n\n")
    sink()
}
```

C.3 C Subroutines stored in "NgibbsBack.c"

```
# include <stdlib.h>
  # include <stdio.h>
  # include <math.h>
  # include <time.h>
  // Driver for subroutine qsimp
  # define EPS 5.0e-6
  # define JMAX 20
  # define FUNC(x) ((*func)(x))
  static double SimuB( double,int,double*,double*,double*,double*,int,int,double,double );
  static double SimuC( double,int,double*,double*,double*,double*,int,int,double,double );
  static void ZLFun( double*, double*, int, double*, int, double* );
  static double sum( double*, int );
  static void cumsum( double*, int );
  static double sample( double*, double*, int );
  static double trapzd( double (*func)(double), double , double, int );
  static double qsimp( double (*func)(double), double, double );
  static double WyBack( double );

  /* MCMC EM Gibbs sampling */
```

```
void NNgibbsBack( double *bcGibbs, int *nnGibbs, double *bs0, double *cs0, double *betas, double *lam1, double *lam2, double *ys, int *nn, int *kk, double *tt, double *ttTu )
{
  int ss, k=*kk, ii;
  srand( (unsigned)time( NULL ) ); // get random seed according to system time
  // (Main Loop) Generate nnGibbs gibbs samples
  for( ss=0; ss<*nnGibbs; ss++ ) {
    for( ii=0; ii<k; ii++ ) {
      SimuB( *lam1, ii, betas, bs0, cs0, ys, *nn, k, *tt, *ttTu );
      bcGibbs[ss*(2*k+1)+ii] = bs0[ii];
      SimuC( *lam2, ii, betas, bs0, cs0, ys, *nn, k, *tt, *ttTu );
      bcGibbs[ss*(2*k+1)+(k+1)+ii] = cs0[ii]; }
    SimuB( *lam1, k, betas, bs0, cs0, ys, *nn, k, *tt, *ttTu );
    bcGibbs[ss*(2*k+1)+k] = bs0[k]; }
}

/* Importance sampling: simulate a (b_j|b_l, l ≠ j, c, y, k) */
static double SimuB( double lam1, int ii, double *betasCur, double *bsCur, double *csCur, double *ys, int nn, int kk, double tt, double ttTu )
{
  int kImp=500, i, j; double rrVec[kImp], ws[kImp], tmpout[2*kk], tmp, nom, upBs, lowBs;
  lowBs = tt-sum(bsCur,kk)-sum(csCur,kk)-bsCur[kk]+bsCur[ii]; if ( lowBs<0 ) lowBs=0;
  if ( ii==kk ) { tmp = (double) rand()/RAND_MAX; // random # ~ Uniform(0,1)
    bsCur[ii] = lowBs-log(tmp)/lam1; // random number from truncated Exp(λ_b)
    } else { upBs = tt-sum(bsCur,kk)-sum(csCur,kk)+bsCur[ii];
      for( i=0; i<kImp; i++ ) {
        tmp = (double) rand()/RAND_MAX;
        rrVec[i] = lowBs-log(1-(1-exp(-lam1*(upBs-lowBs)))*tmp)/lam1;
        bsCur[ii] = rrVec[i]; ZLFun(bsCur, csCur, kk, ys, nn, tmpout);
        ws[i] = 0; for( j=0; j<kk; j++) ws[i] = ws[i]+betasCur[j]*tmpout[j];
        nom = 0; for( j=0; j<kk; j++) nom = nom+(exp(betasCur[j])-1)*tmpout[j+kk];
        ws[i] = exp(ws[i]-nn*log(ttTu+nom)); }
```

```
        bsCur[ii] = sample( rrVec, ws, kImp ); }
    }

    /* Importance sampling: simulate a (c_j|c_l, l ≠ j, b, y, k) */
    static double SimuC( double lam2, int ii, double *betasCur, double *bsCur, double *csCur,
double *ys, int nn, int kk, double tt, double ttTu )
    {
      int kImp=500, i, j; double rrVec[kImp], ws[kImp], tmpout[2*kk], tmp, nom, upCs, lowCs;
      upCs = tt-sum(bsCur,kk)-sum(csCur,kk)+csCur[ii];
      lowCs = upCs-bsCur[kk]; if ( lowCs<0 ) lowCs = 0;
      for ( i=0; i<kImp; i++ ) { tmp = (double) rand()/RAND_MAX;
        rrVec[i] = lowCs-log(1-(1-exp(-lam2*(upCs-lowCs)))*tmp)/lam2;
        csCur[ii] = rrVec[i]; ZLFun( bsCur, csCur, kk, ys, nn, tmpout );
        ws[i] = 0; for( j=0; j<kk; j++ ) ws[i] = ws[i]+betasCur[j]*tmpout[j];
        nom = 0; for( j=0; j<kk; j++ ) nom = nom+(exp(betasCur[j])-1)*tmpout[j+kk];
        ws[i] = exp(ws[i]-nn*log(ttTu+nom)); }
      csCur[ii] = sample(rrVec, ws, kImp);
    }

    /* C Subroutine to calculate z_j and c̃_j */
    static void ZLFun( double *bsk, double *csk, int kk, double *ys, int nn, double *zl )
    {
      double low, up; int ii, jj; low = 0.0; up = 0.0;
      for ( jj=0; jj<kk; jj++ ) { zl[jj] = 0.0; low = up+bsk[jj]; up = low+csk[jj];
        for ( ii=0; ii<nn; ii++ ) { if ( (ys[ii]>low)&(ys[ii]<up) ) zl[jj] = zl[jj]+1; }
        zl[jj+kk] = qsimp(WyBack,low,up); }
    }

    /* Sum of first n elements of an array */
    static double sum( double *data, int n )
    {
      int j; double sum=0.0; for (j=0; j<n; j++) sum=sum+data[j]; return(sum);
```

```
}

/* Cumulative sum of first n elements of an array */
static void cumsum( double *data, int n )
{
   int j; for ( j=1; j<n; j++ ) data[j] = data[j-1]+data[j];
}

/* Get a random sample from a size n array "data" with a size n weight array "prob" */
static double sample( double *data, double *prob, int n )
{
   int index = 0; double uni;
   cumsum( prob, n ); uni = (double) rand()/RAND_MAX;
   while ( (prob[index]<uni*prob[n-1])&(index<n-1) ) index = index+1;
   return data[index];
}

/* Integration Functions! Obtained from Numerical Recipes! */
static double trapzd( double (*func)(double), double a, double b, int n )
{
   double x, tnm, sum, del; static double s; int it, j;
   if ( n==1 ) { return ( s=0.5*(b-a)*(FUNC(a)+FUNC(b)) );
      } else { for ( it=1,j=1; j<n-1; j++ ) it <<= 1;
         tnm = it; del = (b-a)/tnm; x = a+0.5*del;
         for ( sum=0.0,j=1; j<=it; j++,x+=del ) sum += FUNC(x);
         s = 0.5*(s+(b-a)*sum/tnm);
         return s; }
}

static double qsimp( double (*func)(double), double a, double b )
{
   int j; double s, st, ost=0.0, os=0.0;
```

```
    for ( j=1; j<=JMAX; j++ ) { st = trapzd(func,a,b,j); s = (4.0*st-ost)/3.0;
      if ( j>5 )
        if ( fabs(s-os)<EPS*fabs(os) || (s==0.0 & os==0.0) ) return s;
      os = s; ost = st; }
    printf( "Too many steps in routine qsimp" ); return 0.0;
  }

  /* Background function */
  static double WyBack( double x )
  {
    if ( (x>=182.0/365.0)&(x<=244.0/365.0) ) { return ( 1+72.0*x/10000.0+55.0/355.0 );
      } else { return ( 1+72.0*x/10000.0 ); }
  }
  # undef EPS
  # undef JMAX
  # undef FUNC
```

C.4 C Subroutines stored in "LRTgibbs.c"

```
# include <stdlib.h>
  # include <stdio.h>
  # include <math.h>
  # include <time.h>
  static double SimuB ( double, int, double*, double*, int, double );
  static double SimuC ( double, int, double*, double*, int, double );
  static double sum ( double*, int );

  /* MCMC LRT Gibbs sampling */
  void LRTgibbs( double *bcGibbs, int *nnGibbs, double *bs0, double *cs0, double *lam1,
double *lam2, int *kk, double *tt )
  {
    int ss, k = *kk, ii; srand( (unsigned)time( NULL ) );
    for ( ss=0; ss<*nnGibbs; ss++ ) {
```

```
    for ( ii=0; ii<k; ii++ ) {
      SimuB( *lam1, ii, bs0, cs0, k, *tt ); bcGibbs[ss*(2*k+1)+ii] = bs0[ii];
      SimuC( *lam2, ii, bs0, cs0, k, *tt ); bcGibbs[ss*(2*k+1)+(k+1)+ii] = cs0[ii]; }
    SimuB( *lam1, k, bs0, cs0, k, *tt ); bcGibbs[ss*(2*k+1)+k] = bs0[k]; }
}

/* Importance sampling: simulate a (b_j|b_l, l ≠ j, c, k) */
static double SimuB( double lam1, int ii, double *bsCur, double *csCur, int kk, double tt )
{
  double upBs, lowBs, tmp;
  lowBs = tt-sum(bsCur,kk)-sum(csCur,kk)-bsCur[kk]+bsCur[ii]; if ( lowBs<0 ) lowBs=0;
  if ( ii==kk ) { tmp = (double) rand()/RAND_MAX;
    bsCur[ii] = lowBs-log(tmp)/lam1;
    } else { upBs = tt-sum(bsCur,kk)-sum(csCur,kk)+bsCur[ii];
      tmp = (double) rand()/RAND_MAX;
      bsCur[ii] = lowBs-log(1-(1-exp(-lam1*(upBs-lowBs)))*tmp)/lam1; }
}

/* Importance sampling: simulate a (c_j|c_l, l ≠ j, b, k) */
static double SimuC( double lam2, int ii, double *bsCur, double *csCur, int kk, double tt )
{
  double upCs, lowCs, tmp;
  upCs = tt-sum(bsCur,kk)-sum(csCur,kk)+csCur[ii];
  lowCs = upCs-bsCur[kk]; if ( lowCs<0 ) lowCs = 0;
  tmp = (double) rand()/RAND_MAX;
  csCur[ii] = lowCs-log(1-(1-exp(-lam2*(upCs-lowCs)))*tmp)/lam2;
}
```

References

[1] Chang M, Glynn MK, Groseclose SL (2003). Endemic, notifiable bioterrorism-related diseases, United States, 1992C1999. Emerg Infect Dis 9, 556C64.

[2] Akaike H (1974). A new look at the statistical model identification. IEEE Transactions on Automatic Control 19, 716C723.

[3] Balakrishnan N, Koutras, MV (2001). Runs and Scans with Applications. New York: John Wiley and Sons.

[4] Demattei C, Molinari N (2006). Multiple temporal cluster detection test using exponential inequalities. *Far East Journal of Theoretical Statistics,* 19(2): 231-244 - Preprint.

[5] Demattei C, Molinari N (2007). *p*-value calculations for multiple temporal cluster detection. *C.R.Acad. Sci. Paris*, Ser.I 344. pp697-6701.

[6] Demattei C, Molinari N, Daures JP (2007). Arbitrarily shaped multiple spatial cluster detection for case event data. *Computational Statistics and Data Analysis*. 51, 3931-3945.

[7] Demattei C, Molinari N, Daures JP (2006). SPATCLUS: An R package for arbitrarily shaped multiple spatial cluster detection for case event data. *Computer Methods and Programs in Biomedicine*. 84, 42-49.

[8] Dembo A, Karlin S (1992). Poisson Approximations for r-Scan Processes. The Annals of Applied Probability 2, 329-357

[9] Denison D, Holmes C (2001). Bayesian partitioning for estimating disease risk. Biometrics 57, 143C149.

[10] Diggle P, Rowlingson B, Su T-L (2005). Point process methodology for on-line spatio-temporal disease surveillance. Environmetrics 16, 423-434.

[11] Dwass M (1957). Modified randomization tests for nonparametric hypotheses. Annals of Mathematical Statistics 28, 181-187.

[12] Fu JC, Lou WY (2003). Distribution Theory of Runs and Patterns and its Applications: A Finite Markov Chain Imbedding Approach. World Scientific.

[13] Gangnon RE, Clayton MK (2000). Bayesian detection and modeling of spatial disease clustering. Biometrics 56, 922-935.

[14] Gangnon RE, Clayton MK (2001). A weighted average likelihood ratio test for spatial clustering of disease. Statistics in Medicine 20, 2977-2987.

[15] Gangnon RE, Clayton MK (2003). A hierarchical model for spatially clustered disease rates. Statistics in Medicine 22, 3213-3228.

[16] Gangnon RE, Clayton MK (2004). Likelihood-based tests for localized spatial clustering of disease. Environmetrics 15, 797-810.

[17] Ghosh M, Natarajan K, Waller LA, Kim D (1999). Hierarchical Bayes GLMs for the analysis of spatial data: an application to disease mapping. Journal of Statistical Planning and Inference 75, 305C318.

[18] Glaz J, Naus J (1983). Multiple clusters on the line. Communications in Statistics-Theory and Methods 12, 1961-1986.

[19] Glaz J, Balakrishnan N, eds. (1999) Scan Statistics and Applications. Boston: Birkhauser.

[20] Glaz J, Naus J, Wallenstein S (2001). Scan Statistics. New York: Springer.

[21] Knorr-Held L, RaBer G (2000). Bayesian detection of clusters and discontinuities in disease maps. Biometrics 56, 13C21.

[22] Kulldorff M and Nagarwalla N (1995). Spatial disease clusters: detection and inference. Statistics in Medicine 14, 799-810.

[23] Kulldorff M (1997). A spatial scan statistic. Communication in Statistics 26(6), 1481-1496.

[24] Kulldorff M (1999). Spatial scan statistics: models, calculations, and applications.

[25] Kulldorff M, Huang L, Pickle L, Duczmal L (2006). An elliptic spatial scan statistic. Statistics in Medicine 25, 3929-3943.

[26] Lawson AB (1995). Markov chain Monte Carlo methods for putative pollution source problems in environmental epidemiology. Statistics in Medicine 14, 2473C2486.

[27] Lawson AB, Clark A (1999). Markov chain Monte Carlo methods for putative sources of hazard and general clustering. In Disease Mapping and Risk Assessment for Public Health, Lawson AB, Bohning D, Biggeri A, Viel J-F, Bertollini R (eds). Wiley WHO. , chapter 9.

[28] Leung M-Y, Choi K-P, Xai A, Chen LHY (2005). Non-random clusters of palindromes in Herpesvirus genomes. J. Computational Biology 12, 331-354.

[29] Mandl KD, Overhage JM, Wagner MM et al (2004). Implementing syndromic surveillance: a practical guide informed by the early experience. J Am Med Inform Assoc 11, 141C50.

[30] Molinari N, Bonaldi C, Daures JP (2001). Multiple temporal cluster detection. Biometrics 57, 577-583.

[31] Nagarwalla N (1996). A scan statistic with a variable window. Statistics in Medicine 15, 845-850.

[32] Naus JI (1965). The distribution of the size of the maximum cluster of points on a line. Journal of the American Statistical Association 60, 532-538.

[33] Naus JI (1965). Clustering of random points in two dimensions. Biometrika 52, 263-267.

[34] Naus JI (1966). Some probabilities, expectations and variances for the size of largest clusters and smallest intervals. Journal of the American Statistical Association 61, 1191-1199.

[35] Naus JI (1966). A power comparison of two tests of non-random clustering. Technometrics 8, 493-517.

[36] Naus JI (1968). An extension of the birthday problem. The American Statistician 22, 27-29.

[37] Naus JI (1982). Approximations for distributions of scan statistics. Journal of the American Statistical Association 77, 177-183.

[38] Naus JI, Wartenberg D (1997). A double-scan statistic for clusters of two types of events. Journal of the American Statistical Association 92, 1105-1113.

[39] Naus JI, Wallenstein S (2004). Multiple window and cluster size scan procedures. Methodology and Computing in Applied Probability 6, 389-400.

[40] Neill DB, Moore AW, Cooper GF (2006). A Bayesian spatial scan statistic. In Advances in Neural Information Processing Systems 18, 1003C1010.

[41] Openshaw S, Charlton M, Wymer C, Craft AW (1987). A mark 1 analysis machine for the automated analysis of point data sets. International Journal of Geographical Information Systems 1, 335-358.

[42] Pan W (2001). Akaike's Information Criterion in generalized estimating equations. Biometrics 57, 120-125

[43] Self SG, Liang KY (1987). Asymptotic Properties of Maximum Likelihood Estimators and Likelihood Ratio Tests Under Nonstandard. Journal of the American Statistical Association, 605-610.

[44] Silvapulle MJ, Sen PK (2005). Constrained statistical inference: inequality, order, and shape restrictions.

[45] Song C, Kulldorff M (2005). Tango's maximized excess events test with different weights. International Journal of Health Geographics 4, .

[46] Schwarz G 1978. Estimating the dimension of a model. Annals of Statistics 6, 461-464.

[47] Su X, Wallenstein S, Bishop D (2001). Non-overlapping clusters: Approximate distribution and application to molecular biology. Biometrics 57, 420-426.

[48] Tanner MA (1993). Tools for statistical inference: methods for the exploration of posterior distributions and likelihood functions.

[49] Turnbull BW, Iwana EJ, Burnett WS, Howe HL, Clark LC (1990). Monitoring for clusters of disease: application to leukemia incidence in upstate New York. American Journal of Epidemiology 132, S136-S143.

[50] Wallenstein SR, Naus JI (1973). Probabilities for a k^{th} nearest neighbor problem on the Line. The Annals of Probability 1, 188-190.

[51] Wallenstein SR, Naus JI (1974). Probabilities for the size of largest clusters and smallest intervals. Journal of the American Statistical Association 69, 690-697.

[52] Wallenstein S, Naus J, Glaz J (1994). Power of the scan statistic in detecting a changed segment in a bernoulli sequence. Biometrika 81, 595-601.

[53] Wallenstein S, Naus J (2004). Scan Statistics for Temporal Surveillance for Biologic Terrorism, Morbidity & Mortality Weekly Report 53, 74-78.

[54] Waller LA, Carlin BP, Xia H, Gelfand AE (1997). Hierarchical spatio-temporal mapping of disease rates. Journal of the American Statistical Association 92, 607C617.

[55] Zheng X, Palffy-Muhoray P (2007). Distance of closest approach of two arbitrary hard ellipses in two dimensions. Physical Review E 75, 061709.

Vita

Qiankun Sun

1998-2002 Undergraduate Study at the Department of Statistics and Finance, University of Science and Technology of China, Hefei, Anhui, China.

2002 B.S. in Mathematical Statistics, University of Science and Technology of China.

2002-2008 Graduate Study at the Department of Statistics and Biostatistics, Rutgers University, New Brunswick, New Jersey, United States of America.

2008 M.S. in Statistics and Biostatistics, Rutgers University.

2008 (Expect) Ph.D. in Statistics and Biostatistics, Rutgers University.

2002-2003 Fellowship, Department of Statistics and Biostatistics, Rutgers University.

2003-2006 Teaching Assistant, Department of Statistics and Biostatistics, Rutgers University.

2005 Statistical Consultant, Summer Intern, Office of Statistical Consulting, Rutgers University

2006 Statistician, Summer Intern, Oncology Department, Novartis, Florham Park, New Jersey.

2006-2007 Statistical Consultant, Part-time, Methods and Technology Group, Sanofi-Aventis, Bridgewater, New Jersey.

2008 Statistician, Summer Intern, Central Nerve System Group, Janssen, Titusville, New Jersey.

2008 Minge Xie, Qiankun Sun and Joseph Naus. "*A Latent Model to Detect Multiple Clusters of Varying Sizes*", Biometrics.

2008 Hui Quan, Qiankun Sun, Ji Zhang and Joe Shih. "*Comparisons between ITT and Treatment Emergent Adverse Event Analyses*", Statistics in Medicine.

2008 Venkat Sethuraman and Qiankun Sun. "*Impact of Baseline ECG Collection on the Planning, Analysis and Interpretation of 'thorough' QT Trials*", Pharmaceuticals Statistics.