Distribution Agreement

Signature:

____________________________ _______________

Ming Wang Date

Spatial and spatial-temporal point process analysis

By

Ming Wang

Doctor of Philosophy

Biostatistics

____
Lance A. Waller, Ph.D.
Advisor

____
Kang, Jian, Ph.D.
Committee Member

____
Long, Qi, Ph.D.
Committee Member

____
Johnson, Brent A., Ph.D.
Committee Member

____
Kramer, Michael R., Ph.D.
Committee Member

Accepted:

____
Lisa A. Tedesco, Ph.D.
Dean of the James T. Laney School of Graduate Studies

____
Date

Spatial and spatial-temporal point process analysis

By

Ming Wang
M.S., University of Louisville, 2008
B.S., Peking University, 2006

Advisor: Lance A. Waller, Ph.D.

An abstract of
A dissertation submitted to the Faculty of the
James T. Laney School of Graduate Studies of Emory University
in partial fulfillment of the requirements for the degree of
Doctor of Philosophy
in Biostatistics
2013

Abstract

Spatial and spatial-temporal point process analysis

By

Ming Wang

Spatial-temporal point pattern data are increasingly available including one-dimens -ional, directionless observations observed along a line. Our motivating example involves sea turtle nesting data with space and time-specific emergence locations along Juno Beach, Palm Beach County, Florida for the years 1998-2000. Our objectives are to assess spatial and temporal heterogeneity in sea turtle nesting patterns, and detect possible effects due to a 990-foot fishing pier constructed in year 1998-1999. We mainly focus on one-dimensional spatial and spatial-temporal point processes to conduct statistical inference through non-parametric and parametric methods, and yield insights about the first-order and second-order properties of point processes as well as space-time interaction.

A Log-Gaussian Cox Process (LGCP), a Cox point process with the logarithm of intensity function following a Gaussian Process, provides a flexible framework for modeling heterogeneous spatial point processes. The pair correlation function (PCF) plays a vital role in characterizing second-order spatial dependency in LGCPs and delivers key inputs on spatial association structures, yet empirical estimation of the PCF remains challenging, even more so for spatial point processes in one dimension (points along a line). We consider two common edge-correction approaches during nonparametric estimation of the PCF, and evaluate their performance via simulation for one-dimensional spatial data. We also provide a novel algorithm to estimate the PCF based on theoretical derivation combined with finite sample simulation of the k^{th} ($k <= 4$) moment, revealing useful information for optimal spatial designs.

To assess local variations in sea turtle nesting density, we develop a novel hierarchical Bayesian non-parametric model based on Dirichlet processes. Autoregressive temporal dependencies are incorporated in a three-level hierarchical structure. This model allows the potential for time-evolving mixed components/weights across groups. We compare our model with the existing models, e.g., Dirichlet process mixture models, hierarchical Dirichlet process models, and dynamic hierarchical Dirichlet process models, to show its advantage via simulation and real data application to our motivating example.

Spatial and spatial-temporal point process analysis

By

Ming Wang
M.S., University of Louisville, 2008
B.S., Peking University, 2006

Advisor: Lance A. Waller, Ph.D.

A dissertation submitted to the Faculty of the
James T. Laney School of Graduate Studies of Emory University
in partial fulfillment of the requirements for the degree of
Doctor of Philosophy
in Biostatistics
2013

Acknowledgement

I first would like to express my utmost gratitude to my advisor Dr. Lance A. Waller for his continual encouragement, insightful guidance, and kindness during my Ph.D. studies. Without his inspired thoughts, leading expertise in spatial statistics, and constant support, my dissertation cannot be done. I learned a lot from him, not only academic knowledge but also professional skills. As the Chair of Department of Biostatistics and Bioinformatics, Rollins School of Public Health, Emory University and one of the key committee members in various statistical institutions, he has a busy schedule due to administrative business, but he still insists meeting with me weekly, stimulating his suggestions, and helping me solving the challenges for my dissertation research. His academic rigour, positive life attitude as well as enthusiasm of spatial statistics research influenced me a lot, enabling me to pursue a promising academic career as a faculty. In particular, his reference and boundless recommendation without hesitation are appreciated during the period of my job hunting. I feel very fortunate to be his student, and this learning experience under his supervision will be never forgotten.

On the other hand, I would like to thank my committee members, Dr. Jian Kang, Dr. Qi Long, Dr. Brent A. Johnson, and Dr. Michael R. Kramer for their helpful comments, detailed review, and active discussion on my dissertation topics and structures. Their professional background on spatial point processes, Bayesian modeling, non-parametric methodology, and epidemiology provide me constructive advice, making the work integrated and comprehensive. Especially, I want to express thank you to Dr. Kang for his generosity in helping me overcome technical difficulties I encountered during the research; I am also grateful for Dr. Long's mentor in equipping

myself methodological research and consulting skills through collaborative projects.

In addition, I extremely thanks to all those who helped me during my five-year Ph.D. studies in the Department of Biostatistics and Bioinformatics. Thanks to our excellent staffs, Mary Abosi, Melissa Sherrer, Bob Waggoner, and Angela D. Thomas, for assisting me solving the questions whenever I have difficulties in class registration or conference traveling; Thanks to those wonderful faculties, i.e., Dr. Amita K. Manatunga, Dr. Limin Peng, Dr. Ying Guo, Dr. Hao Wu and so on, for providing me insightful research projects and useful comments on my career development; Thanks to my great classmates for their friendship to let me experience the graduate study's life beauty. Although I cannot thank all of them individually, my heart will remember them forever, and thank you very much to you all.

Last but not least, I would like to give my deepest thanks to my parents and parents-in law, my husband Lijun Zhang, and my little baby son Louis Zhang for their love and support on my work and life. I cannot image how my life turns out to be without them, and I hope this work could be the best reward to them. Thanks for all your unconditional dedication, belief and encouragement.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background

Spatial statistics has gained increasing attention, and has been widely applied into public health and biomedical research. A key feature of spatial analysis is that the use of location information. There are three main types of spatial data: point data consisting event locations; geostatistical data containing the measurements taken at multiple locations; and region data describing event counts within a finite set of spatial regions, each of which addresses different specific questions of interest. In this dissertation, we focus on realizations of spatial point process to quantitatively conduct spatial point pattern analysis. A spatial point process is a stochastic process with each realization including a finite or countably infinite set of points observed within a bounded window. One goal is to detect clusters or clustering based on point process theory and apply stochastic models describing occurrences of events in space. Methods build on several important concepts listed below

- A point process X is *stationary* if its distribution is translation invariant, i.e., $X + s = \{\xi + s : \xi \in X\}$ has the same distribution as X for any $s \in \mathbb{R}^d$.

- A point process X is *isotropic* if its distribution is invariant under rotations around the origin, i.e., $\psi X = \{\psi\xi : \xi \in X\}$ has the same distribution as X for any rotation ψ around the origin.

- A point process X is *homogeneous* or *first order stationary* if its intensity function $\rho(s)$ is constant, otherwise X is said to be *inhomogeneous*.

- A homogeneous point process X is *second-order stationary* if its second-order properties (i.e., $K-$function or pair correlation function) only depend on spatial distance between points but not the locations of the points, while for a heterogeneous point process X, it is called *second-order intensity-reweighted stationary* given the intensities are bounded away from 0.

There exist various parametric point process models, and the simplest and most fundamental one is the Poisson process. Due to its property of complete spatial randomness, Poisson process is often treated as a baseline for detecting clustering or inhibition patterns. In the following sections and chapters, we will introduce technical details as needed. The dissertation is organized as follows: the remainder of Chapter 1 defines another standard and flexible point process model, the Cox process, with the log-Gaussian Cox process as a special case, and outlines a brief introduction of concepts for first and second-order properties in spatial point processes which provide the elements for our extensions to existing spatial-temporal point processes. Chapter 2 presents on evaluation of non-parametric pair correlation functions for spatial log-Gaussian Cox processes to better understand its performance in one-dimensional spatial data analysis. Chapter 3 applies our spatial-temporal point pattern analysis to sea turtle nesting and emergence locations along Juno Beach, Florida to seek refined estimates of the spatial and temporal heterogeneity. Chapter 4 develops Bayesian non-parametric modeling for density estimation of sea turtle emergence locations to efficiently estimate the evolution of the density over time.

1.2 First and Second-order properties

1.2.1 Intensity

The intensity, ρ, is a first-order property of a point process defining the expected number of events occurring per unit area. For an arbitrary Borel set $B \subset W \subset \mathbb{R}^d$, let $|B|$ and $N(B)$ denote the area of B and the number of events in B. The intensity function of X is defined by Diggle (2003) as

$$\rho(s) = \lim_{|ds| \to 0} \{ \frac{E[N(ds)]}{|ds|} \} \tag{1.1}$$

As mentioned above, $\rho(s)ds$ is the probability of observing exactly one point in the infinitesimally small region ds. If a spatial point process is homogeneous, then $\rho(s) = \rho > 0$, a constant; otherwise, the process is heterogeneous. When modeling a spatial point process, our first goal is estimation of the intensity function from data.

For a homogeneous process, $\hat{\rho} = N(W)/|W|$. While for an inhomogeneous process, we first consider parametric estimation. If we assume the intensity function belongs to a parametric family $\{\rho(s;\theta) : \theta \in \Theta\}$. One typical example is the family of modulated Poisson processes defined by Cox (1972), where $\rho(s;\theta) = \exp\{\theta' z(s)\}$ with $z(s)$ as a specified vector of covariates observed at location s. Therefore, for n events in the observed window W, the associated likelihood function can be specified as

$$L(\theta; s_1, s_2, \ldots, s_n) \propto \exp\left\{ - \int_W \rho(s;\theta) ds \right\} \left\{ \prod_{i=1}^{n} \rho(s_i;\theta) \right\} \tag{1.2}$$

The maximum likelihood estimate of θ is obtained by maximizing L; however, this approach requires an assumption that we observe locations without error. Parametric models have also been discussed by others (Diggle, 2003; Waagepetersen, 2007). Although the parametric approach is useful, estimates may not be reliable if the assumed parametric model deviates greatly from the true intensity function, i.e., if the covariates do not describe the intensity well. More often, non-parametric methods for estimating the intensity function are widely applied. Berman and Diggle (1989) proposed the following intensity estimator using kernel smoothing:

$$\hat{\rho}(s;h) = \frac{1}{nh}\sum_{i=1}^{n}\kappa\left(\frac{s-s_i}{h}\right) \tag{1.3}$$

where s_i denotes the location of i^{th} event in X, n the total observed number of events in W, κ a kernel function satisfying $\int_W \kappa(s)ds = 1$, and h a smoothing parameter called the bandwidth. Most commonly, κ is a Gaussian kernel following a standard normal density, $\frac{1}{\sqrt{2\pi}}\exp(-s^2/2)$.

The choice of the bandwidth, h, has particular influence on the accuracy of intensity estimate. According to Diggle (1985) and Berman and Diggle (1989), the bandwidth, h, can be chosen as the value which minimizes the mean square error, $MSE(h) = E[(\hat{\rho}(s;h) - \rho(s))^2]$. Another method for selecting h is the direct plug-in approach by Sheather and Jones (1991), incorporating kernel estimates to replace unknown quantities in formula for the asymptotically optimal bandwidth. This approach can be simply applied by using the **dpik** function in R package **Spatstat** or **KernSmooth**. We discuss bandwidth selection in more detail for our application in the chapters below.

1.2.2 $K-$function and $L-$function

One important second-order property for characterizing spatial correlation or dependency is $K-$function (Ripley, 1976). This function evaluates the expected number of events within a certain distance given an arbitrary event in the study region. Thus, for a stationary point process $X \in \mathbb{R}^d$ in a window W, it is defined as

$$K(r) = \frac{E(\psi(r))}{\rho} \tag{1.4}$$

where $\psi(r) = \frac{1}{\rho|W|} \sum_{s_1 \neq s_2 \in W} I(s_1 \in W, \|s_1 - s_2\| \leq r)$ represents the count of the neighbors located at a distance less than or equal to a given distance r away from a randomly chosen point in the window W with $\|.\|$ denoting the Lebesgue measure. If X is stationary, $\psi(r)$ is translation invariant, and the intensity function ρ is constant. To ease inference, Besag (1977) proposed a variance-stabilized version of the $K-$function known as the $L-$function. The $L-$function is defined as $L(r) = (K(r)/\delta_d)^{1/d}$ with $\delta_d = \frac{\pi^{d/2}}{\Gamma(1+d/2)}$ representing the volume of a $d-$dimensional unit ball. Furthermore, for a stationary Poisson point processes, the $K-$function and $L-$function are written by

$$K(r)_{\mathbb{R}^d} = \delta_d r^d, \qquad L(r) = r \tag{1.5}$$

For our application, $K(r)_{R^1} = 2r$ for an one-dimensional stationary Poisson point process; while for a two-dimensional Poisson process, $K(r)_{\mathbb{R}^2} = \pi r^2$. The theoretical K and $L-$functions for the Poisson point process serve as reference functions to identify clustering or regularity patterns. $K(r) > \delta_d r^d$ or $L(r) > r$ suggests clustering or aggregate patterns; $K(r) = \delta_d r^d$ or $L(r) = r$ suggests complete spatial randomness (CSR); and, if $K(r) < \delta_d r^d$ or $L(r) < r$ indicates segregated patterns with gaps. As noted by Waller and Gotway (2004), a given realization can include clustered, CSR, and regular behavior at different distances.

The $K-$function for stationary and isotropic Poisson processes is defined by Ripley (1977) as:

$$K(r) = \int_{s=0}^{r} \tau_d t^{d-1} g(s) ds \tag{1.6}$$

where $\tau_d = \frac{\partial \delta_d}{\partial d} = \frac{2\pi^{d/2}}{\Gamma(d/2)}$ represents the surface area of an unit sphere in $\mathbb{R}^d$, and $g(\cdot)$ is the pair correlation function (defined in Chapter 1.2.3). If the events are distributed independently from each other, $g(\cdot) = 1$, therefore, $K(r) = \delta_d r^d$ as above. A non-parametric estimate of K function with edge-correction (Ripley, 1976) may be written as:

$$\hat{K}(r) = \frac{1}{|W|} \sum_{i=1}^{n} \sum_{j \neq i} \frac{\omega_{ij} I(\|s_i - s_j\| \leq r)}{\hat{\rho}(s_i)\hat{\rho}(s_j)} \tag{1.7}$$

If the point process is stationary, then $\hat{\rho}(s_i)\hat{\rho}(s_j)$ can be replaced by its unbiased estimator, thus

$$\hat{K}(r) = \frac{|W|}{n(n-1)} \sum_{i=1}^{n} \sum_{j \neq i} \omega_{ij} I(\|s_i - s_j\| \leq r) \tag{1.8}$$

If the point process is not stationary, then $\hat{\rho}(s_i)$ and $\hat{\rho}(s_j)$ can be estimated using a leave-one-out kernel smoother as described by Baddeley *et al.* (2000). In the formulas above, n is the total number of observed events in the window W, $I(\|s_i - s_j\| \leq r)$ is an indicator function taking value 1 when the distance of points s_i and s_j is less than or equal to r and value 0 otherwise. Here, $|W|$ denotes the area of the window W. ω_{ij} is an edge-correction factor to adjust for the bias due to edge effects, and Ripley's (1976) edge correction is commonly used, where ω_{ij} is the inverse of the proportion of the circumference of the circle centered at event i with radius $d(i,j)$ within the study area. For one-dimensional spatial data analysis, an analogue of Ripley's correction is specified by the reciprocal of the ratio of the interval within the length of $2 \times |s_i - s_j|$ for each point i within the study window W serving as the center point in turn. Also, Diggle (1989) proposed setting ω_{ij} to 2 when the distance $|s_i - s_j|$ is larger than the distance of point i to the nearest edge (end) of the line segment defining the study area W, otherwise ω_{ij} is set to 1. In Chapter 2, we will compare the performance of these two edge-correction methods for one-dimensional data.

1.2.3 Pair correlation function (PCF)

Another important second-order property of spatial point process is the PCF, which provides a summary of the joint expectation of events and first-order intensity. The PCF is highly related to the $K-$function because both of them take into account spatial correlation. Without assuming stationary, $\forall \xi \in \mathbb{R}^d, \eta \in \mathbb{R}^d$, the theoretical definition of PCF provided by Baddeley *et al.* (2000) is

$$g(\xi, \eta) = \frac{\rho^{(2)}(\xi, \eta)}{\rho(\xi)\rho(\eta)} = 1 + \frac{Cov(N(d\xi), N(d\eta))}{\rho(\xi)\rho(\eta)d\xi d\eta} \tag{1.9}$$

where the second-order intensity $\rho^{(2)}(\xi, \eta) = \lim_{|d\xi| \to 0; \|d\eta\| \to 0} \{\frac{E[N(d\xi)N(d\eta)]}{\|d\xi\| \|d\eta\|}\}$ and $\rho(\cdot)$ is defined as above. The PCF equals one under complete spatial randomness, i.e., homogeneous Poisson processes. A value of PCF greater than one indicates an aggregated (clustered) spatial point pattern. A PCF value less than one suggests an inhibitive spatial point pattern.

For first and second-order stationary spatial point processes, the PCF reduces to $g(r) = \frac{\rho^{(2)}(r)}{\rho^2}$ with r as a distance lag. In practice, the empirical PCF (defined below) is often plotted at varying distance lags in order to detect the scale of clustering or regularity compared with complete spatial randomness suggesting appropriate parametric models to fit to the data. Furthermore, many statisticians prefer the PCF over the K-function defined above, because the PCF may yield more stable estimates of second-order property than the K-function in some cases, e.g., the Cox process models (Brix, 1999).

A non-parametric estimate of PCF can be obtained via $\hat{g}(r) = \frac{\hat{\rho}^{(2)}(r)}{\hat{\rho}^2}$, where $\hat{\rho}^{(2)}$ is the empirical estimate of the second-order intensity using kernel smoothing techniques (Møller *et al.*, 1998).

$$\hat{\rho}^{(2)}(r;h) = \sum_{i=1}^{n} \sum_{j \neq i} \frac{\kappa_h(r - \|s_i - s_j\|)\omega_{ij}}{\nu |W|} \tag{1.10}$$

where $\nu = \sigma_d r^{d-1}$ with $\sigma_d = \frac{2\pi^{d/2}}{\Gamma(d/2)}$ as the surface area of unit sphere in $\mathbb{R}^d$. Therefore, for one-dimensional and two-dimensional isotropic spatial point processes, PCFs are estimated via

$$\hat{g}(r)_{\mathbb{R}^1} = \frac{1}{2|W|} \sum_{i=1}^{n} \sum_{j \neq i} \frac{\kappa_h(r - \|s_i - s_j\|)\omega_{ij}}{\hat{\rho}(s_i)\hat{\rho}(s_j)}$$

$$\hat{g}(r)_{\mathbb{R}^2} = \frac{1}{2\pi r|W|} \sum_{i=1}^{n} \sum_{j \neq i} \frac{\kappa_h(r - \|s_i - s_j\|)\omega_{ij}}{\hat{\rho}(s_i)\hat{\rho}(s_j)}$$

Specially, for homogeneous point processes, $\hat{\rho}(s_i)\hat{\rho}(s_j)$ can be replaced by the unbiased estimator $\frac{n(n-1)}{|W|^2}$. $\kappa_h(\cdot)$ denotes the scaled kernel specified as $\kappa_h(s) = \frac{1}{h}\kappa(\frac{s}{h})$ with κ as a kernel function such as the Epanecnikov kernel by $\kappa(t) = \frac{3}{4}(1 - \frac{t^2}{h^2})I[-h \leq a \leq h]$. $|W|$ and Ripley's isotropic edge correction factor ω_{ij} are defined as the same as above. The PCF can also be derived from the $K-$function, shown as $g(r) = \frac{K'(r)}{\tau_d r^{d-1}}$. Intuitively, an estimate of $g(r)$ can be obtained by $\hat{K}(r)$, however, this is not easy because $\hat{K}(r)$ is usually a step-function. In such cases, numerical methods (e.g., approximation by splines) provide a better approach. On the other hand, the issues of how to choose the bandwidth of second-order product intensity also need exploration due to their impact on the accuracy of the estimator. Recent work related to bandwidth selection for the PCF estimation are shown by Guan (2008), who proposed a composite likelihood cross-validation approach. Other PCF estimators as well as bandwidth selection strategies are summarized by Stoyan and Stoyan (1994).

To link non-parametric estimates to parametric PCFs involves the selection of a

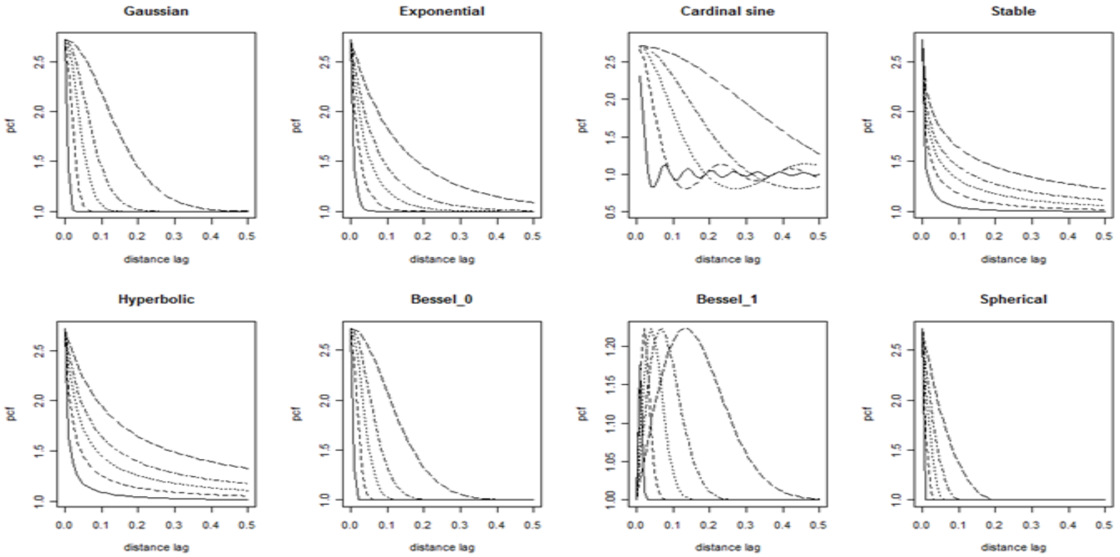


Figure 1.1: Parametric PCFs with $\sigma^2 = 1$ and $\beta = c(0.01, 0.03, 0.06, 0.1, 0.2)$, where the solid line is for the smallest β.

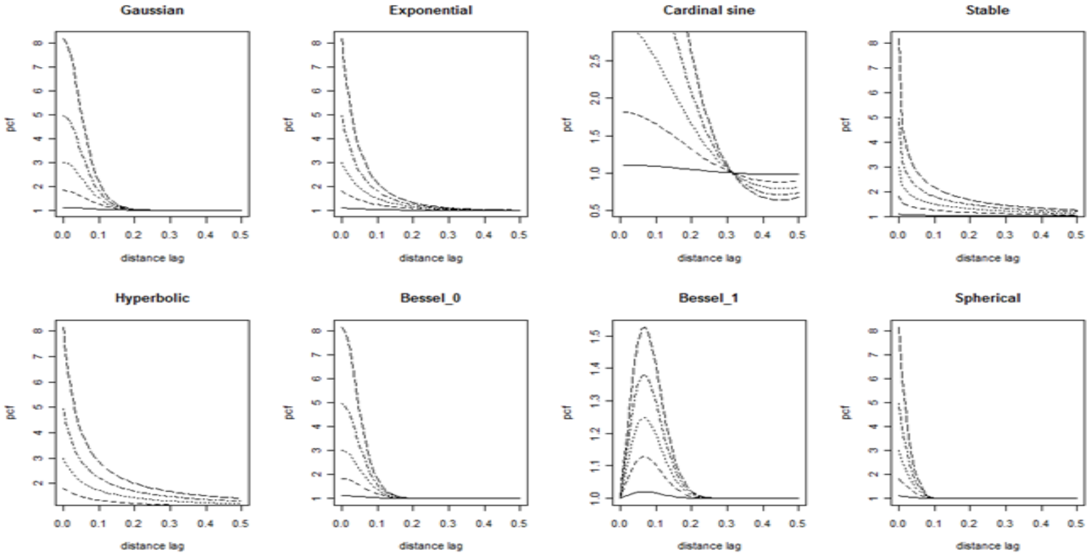


Figure 1.2: Parametric PCFs with $\sigma^2 = c(0.1, 0.6, 1.1, 1.6, 2.1)$ and $\beta = 0.1$ where the solid line is for the smallest σ^2.

parametric family and parameter estimation. Examples of commonly used parametric PCF families for stationary and isotropic spatial point processes are shown in Figure 1.1 and 1.2, where σ is the variance and β is a scale parameter (Møller *et al.*, 1998). To choose an appropriate parametric correlation model for a specific observed spatial point pattern, non-parametric estimates of PCFs versus different distance lags are always preliminarily analyzed. Afterwards, the parameter estimates for the parametric PCF models can be achieved by a minimum contrast method or maximum likelihood as detailed below.

1.3 Cox process: Log-Gaussian Cox process

A Cox process called a double stochastic process arises as an inhomogeneous Poisson process with a random intensity measure which itself is a realization of a stochastic process (Diggle, 2003; Christakos, 1992). A Cox process is characterized by a random intensity process $\Lambda = \{\rho(s) : s \in \mathbb{R}^d\}$. Conditional on $\Lambda = \rho(s)$, X is a Poisson Process with intensity function $\rho(s)$ which means there is no interaction between points given $\rho(s)$. Further, for any bounded Borel set $B \subset \mathbb{R}^d$, if $\Lambda = \rho(s)$ is known, $X \cap B$ is Poisson distributed with mean $\int_B \rho(s)ds$ which must be non-negative and finite. For simplicity, we consider cases where Λ as well as X is stationary and isotropic.

A log Gaussian Cox Process (LGCP) is a member of a family of Cox processes, where heterogeneity is modeled by $\Lambda = \exp(Y)$ where $Y = \{Y(s) : s \in \mathbb{R}^d\}$ is a real-valued Gaussian Process, i.e., the joint distribution of any finite vector $(Y(s_1), Y(s_2), \ldots, Y(s_n))$ is Gaussian (Møller *et al.*, 1998). By assuming stationarity and isotropy, the Gaussian distribution of Y is specified by the mean $\mu = E(Y(s))$ and the variance $\sigma^2 = Var(Y(s))$. The correlation function for Y is $\zeta(r) =$

$Cov(Y(s_1), Y(s_2))/\sigma^2$ where $r = \|s_1 - s_2\|$ is the Euclidean distance between points s_1 and s_2.

In particular, we discuss one of the most useful characteristics for a univariate log Gaussian Cox process, the n^{th} order product densities $\rho^{(n)}$, $n = 1, 2, \ldots$. The moments of the intensity process are given by

$$\rho^{(n)}(s_1, s_2, \ldots, s_n) = E\prod_{i=1}^{n} \rho(s_i) \tag{1.11}$$

for different $s_1, s_2, \ldots, s_n \in \mathbb{R}^d$. $\rho^{(n)}(s_1, s_2, \ldots, s_n)ds_1ds_2\cdots ds_n$ represents the probability that X has a point in each of the n infinitesimally small disjoint regions, ds_1, ds_2, $\cdots$, ds_n. For X a stationary log Gaussian Cox process,

$$\begin{aligned}\rho^{(n)}(s_1, s_2, \ldots, s_n) &= \exp\left\{n\mu + \sigma^2\left[\frac{n}{2} + \sum_{1\le i<j\le n}\zeta(r_{ij})\right]\right\}\\ &= \rho^n \prod_{1\le i<j\le n} g(r_{ij})\end{aligned}$$

where $r_{ij} = \|s_i - s_j\|$. Hence, the intensity and PCF can be respectively written as:

$$\rho = \rho^{(1)}(s) = \exp(\mu + \sigma^2/2) \tag{1.12}$$

$$g(r) = g(\|s_i - s_j\|) = \rho^{(2)}(s_1, s_2)/\rho^2 = \exp(\sigma^2\zeta(r)) \tag{1.13}$$

where $\zeta(r) = 1$ when $r = 0$. As $\sigma^2 \to 0$, a log Gaussian Cox Process tends to a homogenous Poisson process; If $\zeta(\cdot) = 1$, a mixed Poisson process can be obtained with a randomized density Λ which is log Gaussian distributed.

In addition, if X is stationary, then $\rho^{(1)}(s) = \rho$ and $g(s_1, s_2) = g(\|s_1 - s_2\|)$. Thus, $\mu(s) = \mu = \log(\rho) - \sigma^2/2$ and $\zeta(s_1, s_2) = \zeta(\|s_1 - s_2\|) = \log(g(\|s_1 - s_2\|))/\sigma^2$, indicating the Gaussian process Y is stationary. Inversely, stationarity of Y also implies stationarity of X based on the above information. Note that the distribution of X is completely determined by $(\rho, g(.))$ or $(\mu, \sigma^2, \zeta(.))$, which makes parametric models accessible. LGCPs possesses appealing theoretical properties, which can be flexibly extended to spatial-temporal modeling.

There exists an extensive literature on log-Gaussian Cox processes for spatial-temporal point data. Diggle *et al.* (2005) focused on the development of on-line spatial-temporal disease surveillance system for non-specific gastroenteric disease in the country of Hampshire, UK, using a non-stationary log-Cox Poisson process to model spatial-temporal intensity including a deterministic component for spatial and temporal variation in the normal disease pattern, and an unobserved stochastic component for localized departure from normal pattern. Under the assumption of separability in the PCF between space and time, they used moment-based methods to estimate the spatial and temporal correlation structures. They provide methods of minimizing the empirical and theoretical descriptors for estimation and prediction of potential anomalies, thus alerting a public health department to take action. Møller *et al.* (1997) applied a planar Cox process with a log Gaussian intensity process to analyze stand structural heterogeneity in forestry using an exponential covariance function as a parametric model for the covariance. By using a minimum contrast estimation procedure, the estimates of the parameters were achieved; therefore, completely determining the distribution of the log Gaussian Cox process. During model checking, they found the model provided a better fit than a Matern cluster process by comparing the $L-$, $F-$ and $G-$functions. Finally, using an empirical Bayesian approach, they predicted the unknown environmental heterogeneity in the soil through the envelopes from the posterior distribution of the log intensity given the observed pattern of trees. Brix and Diggle (2001) explored the properties of Cox process models for space-time point process data to describe and predict the space-time variation in intensity and corresponding spatial-temporal correlation structures. Specifically, they assume that the time lag between successive observations is sufficiently small that a continuous space-time model is approximated, using a spatial Ornstein-Uhlenbeck process. Afterwards, they derived moment-based methods of parameter estimation and developed a Metropolis-Hastings of Markov chain Monte Carlo algorithm to simulate from the posterior distributions of the observed process data.

However, the assumption of a separable space-time covariance structure is pragmatic, and the robustness of predictions to departure from this assumption needs exploration. Toward this end, Møller and Ghorbani (2010) conducted second-order analysis on the PCF and $K-$function for inhomogeneous spatial-temporal point process including Cox process. Assuming the second-order spatial-temporal separability, they considered a log-Gaussian Cox process with an additive covariate structure. More important, they propose the diagnosis procedures for checking the hypothesis of spatial-temporal separability using simulated and real data. Also, Møller *et al.* (1998) investigated an univariate and multivariate log-Gaussian Cox process, and summarized first, second and third-order statistical properties for clustered point patterns analysis. We will explore the performance of a log-Gaussian Cox processes with non-separable space-time covariance function due to our specific motivating example described next.

1.4 Motivating Example

Our motivating data set involves nesting and emergence locations of sea turtles along Juno Beach, Palm Beach County, Florida from the 1998-2000 nesting seasons. The primary interest aims to employ statistical methodology for spatial-temporal point processes to access spatial and temporal heterogeneity of sea turtle emergences and nesting, and also to assess possible effects due to a 990-foot fishing pier constructed in 1998-1999.

Palm Beach County contracted the Marine life Center of Juno Beach to gather data containing sea turtle emergence locations at sub meter resolution each nesting season over 6 miles (9.7 km) along the beach. Data inculde Universal Transverse Mercator (UTM) coordinates of each emergence, nest fate, sea turtle species, human activities and so on. Our data were collected on three species of sea turtles, loggerhead (*Caretta caretta, CC*), green (*Chelonia mydas, CM*) and leatherback (*Dermochelys coriacea, DC*) daily from year 1998 to 2000. The three species exhibit varied sample sizes with CC providing 8,000-10,000 emergences per year. According to the Florida Fish and Wildlife Conservation Commission, Juno Beach is divided into 11 Index Nesting Beach Survey Zones ("INBS zones") of approximately 0.5 mile shown in Figure 1.3. Of note is a major construction project with potential impacts on sea turtle nesting, namely the construction of the 990 foot-long Juno Beach Fishing Pier in northern Palm Beach County in year 1998-1999. Here, we focus on detecting changes in nesting and emergence patterns near the pier.

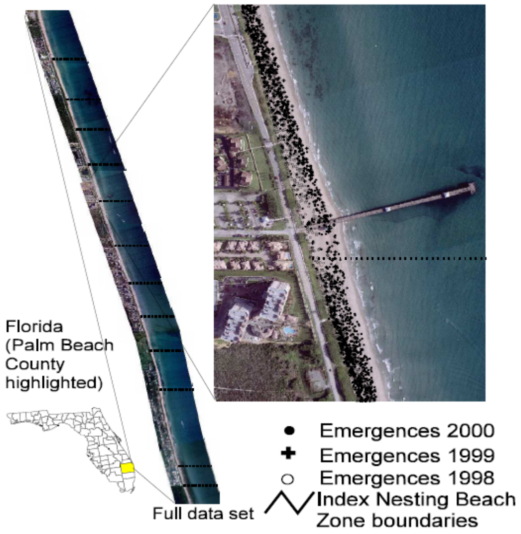


Figure 1.3: Juno Beach Sea Turtle Nesting Data 1998-2000

In this dissertation, we consider statistical inference and modeling for one-dimensional spatial-temporal point processes using the PCF to provide insight into the underlying pattern in Chapter 2. These data were analyzed through non-parametric analysis as well as a qualitative model via a Log Gaussian Cox Process (LGCP) in Chapter 3. Also, in Chapter 4, the intensity estimates were investigated by Bayesian non-parametric modeling and novel statistical methodology proposed to incorporate temporal autocorrelation within the spatial-temporal processes.

Chapter 2

Evaluation of non-parametric pair correlation functions for spatial Log-Gaussian Cox Processes

2.1 Introduction

The pair correlation function (PCF) plays a vital role in characterizing interpoint spatial dependency in spatial point processes. It is recommended by many authors, because it is more interpretable as a function of distance than the often-used second-order $K-$function which is a cumulative function of distance (Penttinen and Stoyan, 2000). That is, the estimated PCF provides inference regarding clustering or regularity in the process at any particular distance, while the estimated $K-$function provides inference regarding clustering or regularity *up to* a given distance. Nonparametric estimates of the $K-$function or PCF often aid exploratory analyses assessing the degree of clustering or regularity in spatial point patterns (Ripley, 1977). In our motivating example, these estimates deliver key insights regarding association structure within a log-Gaussian Cox Process (LGCP), allowing investigators to propose parametric correlation families and estimate parameters through, for example, the minimum contrast method (Diggle *et al.*, 2005; Stoyan *et al.*, 1993).

In this section, we evaluate the performance of different approaches in estimating the PCF for one-dimensional spatial point patterns. This type of data arises when one observes events occurring along a line segment, without directionality. However, estimating the PCF remains challenging, especially when we consider a spatial point process in one dimension (points along a line). We are particularly interested in two elements of the PCF estimation for one-dimensional data namely, edge-effect adjustments and evaluation of bias and variance under infill asymptotics.

Adjustment for edge-effects is necessary because of bias arising due to events occurring outside the observation window (Doguwa, 1990). Most work on edge-correction methods has been done for two-dimensional spatial data, with little work in the one-dimensional case. For instance, Diggle (1985) demonstrated a kernel method for estimating the local intensity with an end-effect correction, which was illustrated using simulated data from an one-dimensional stationary Cox process; however, this paper did not consider the second-order properties. Also, Ripley (1988) proposed an edge-correction approach which can be computed for a window of arbitrary shape, and we consider the one-dimensional case. We provide details of these two edge-correction methods and compare them in estimation of the PCF for LGCPs via simulation.

Our second focus relies on evaluation of bias and variance of the non-parametric PCF estimates for one-dimensional spatial data under infill asymptotics. Two types of asymptotics exist in the spatial statistics literature, namely increasing domain asymptotics and infill asymptotics (Lahiri, 1996; Zhu and Zhang, 2006). The former assumes a stochastic process observed in an increasing region which eventually becomes unbounded as the sample size goes to infinity, thus the (expected) distance between any pair of locations is bounded away from zero. In contrast, infill asymptotics represents observing an increasing number of events from a spatial process in a fixed bounded domain where events become increasingly closer together, thus the minimum distance between events goes to zero as sample size goes to infinity. The consistency of the nonparametric PCF estimate using case-control data under increasing domain asymptotics has been discussed (Guan *et al.*, 2010). Due to the wide application of LGCPs, we would like to explore infill asymptotics for one-dimensional LGCPs via theory and simulation.

In addition, we develop an efficient algorithm for PCF estimation via Monte Carlo simulations. We also develop an edge-correction method for PCF estimation, and measure the resulting reduction in the edge-effect bias, especially for moderately large distance lags, compared with the methods by Diggle and Ripley mentioned above. It is worth mentioning that various software packages provide simulation and analysis of LGCPs for two-dimensional spatial data, e.g., the "spatstat" package in *R*)Baddeley and Turner, 2005; Baddeley, 2010). However, those packages are not directly applicable to one-dimensional spatial data. In the end, we provide a R package for one-dimensional spatial point pattern analysis.

The rest of the contents is organized as follows: In Section 2.2, we provide an overview of LGCPs and existing edge-corrected kernel-based estimators of the PCF. We derive infill asymptotic bias and variance for nonparametric PCF estimators in Section 2.3. In Section 2.4, we conduct simulation studies to identify consistency properties of two classic edge-correction methods for one or two dimensional LGCPs, and compare these to a new, heuristic edge-correction method. Section 2.5 provides discussion and directions for further research.

2.2 Methods

A non-parametric estimate of the PCF is given by $\hat{g}(r) = \hat{\rho}^{(2)}(r)/[\hat{\rho}(s_i)\hat{\rho}(s_j)]$, where $\hat{\rho}(s_i)\hat{\rho}(s_j)$ can be replaced by an unbiased estimator $N(N-1)/|W|^2$ under an assumption of stationary. Also, a kernel estimate of $\rho^{(2)}$ is provided by Baddeley *et al.* (2000).

$$\hat{\rho}^{(2)}(r;h) = \sum_{i=1}^{N}\sum_{j\neq i} \frac{\kappa_h(r - \|s_i - s_j\|)\omega_{ij}}{\nu|W|},$$

where $\nu = \tau_d r^{d-1}$ with $\tau_d = 2\pi^{d/2}/\Gamma(d/2)$ as the surface area of a unit sphere in in $d^{th}-$dimensional space $\mathbb{R}^d$ and ω_{ij} is the edge-correction factor. This leads to a non-parametric estimate of the PCF:

$$\hat{g}(r) = \frac{|W|}{\nu N(N-1)} \sum_{i=1}^{N}\sum_{j\neq i} \kappa_h(r - \|s_i - s_j\|)\omega_{ij}$$

In particular, for one-dimensional spatial data, $\nu = 2$, while for two-dimensional spatial data, $\nu = 2\pi r$. $\kappa(\cdot)$ is a a kernel function with $\kappa_h(t) = \kappa(t/h)/h$. Throughout this article, we adopt the Epanechnikov kernel, i.e.,

$$\kappa(s) = \begin{cases} \frac{3}{4}(1-s^2) & \text{if } |s| \leq 1 \\ 0 & \text{otherwise.} \end{cases}$$

The parameter h denotes the bandwidth, a smoothing parameter. Here, we select $h = 0.1\sqrt{5}/\sqrt{\rho}$ as suggested by Fiksel (1988).

For edge-correction, we consider the use of an edge-corrected weight ω_{ij}. Events near the boundary of the window W may have other potential neighboring events outside the window which are not included above, thus an uncorrected PCF estimator may be biased in this region. For two-dimensional data, Ripley (1988) suggest the use of inverse probability sampling to obtain an unbiased estimate of the PCF, where ω_{ij}

is the reciprocal of the ratio of the interval within the length of $2 \times \|s_i - s_j\|$ taking point i as the center point that lies within the study area. For one-dimensional spatial data, Diggle (1985) proposes setting ω_{ij} to 2 when the distance $\|s_i - s_j\|$ is larger than the distance of point i to the nearest edge of the line segment defining the study area, otherwise ω_{ij} is set to 1. However, whether the performance of these two methods is satisfactory for one-dimensional LGCPs under infill asymptotics is unclear. Performance for any edge-correction approach deteriorates for large values of r, leading to recommendations to focus attention on values of r less than half the length of the study segment. We propose a heuristic edge-corrected method to moderately reduce the bias for such situations, where ω_{ij} is the reciprocal of the proportion of the interval of an annulus with s_i as the center and radii $r_1 = r - h'$ and $r_2 = r + h'$ that lies within the study area. For $r < h'$, we use Diggle's correction method. We illustrate and assess performance of his method below.

2.3 Statistical inference

In this section, we derive the bias and variance of the non-parametric PCF estimate. We assume the following regularity conditions:

1. The PCF $g(\cdot)$ is bounded and continuous.
2. The k^{th} order product intensity is bounded for $k = 1, 2, 3, 4$, i.e., there exists $C > 0$ such that $\rho^{(k)}(s_1, s_2, \ldots, s_k) = E\left(\prod_{l=1}^{k} \rho(s_l)\right) < C$.
3. The kernel density κ satisfies with $\int x\kappa(x)dx = 0$ and $\int x^2\kappa(x)dx < \infty$.

Theorem 2.3.1. *Under regularity conditions 1–3, the infill asymptotic bias and vari-*

ance of $\hat{g}(r)$ under the framework of LGCPs are given by

$$\begin{aligned}
Bias(\hat{g}(r)) &= \frac{\lambda^2|W|\varpi(r)}{\nu\Delta_2}\Big[1+\frac{\Delta_4}{\Delta_2^2}+4\frac{\Delta_3}{\Delta_2^2}+\frac{2}{\Delta_2}\Big]-\frac{\lambda^2|W|\left(2\varpi(r)+4\lambda\xi(r)+\lambda^2\varrho(r)\right)}{\nu\Delta_2^2}-\exp(\sigma^2\zeta(r)),\\
Var(\hat{g}(r)) &= \frac{\lambda^4|W|^2\varpi^2(r)}{\nu^2\Delta_2^2}\Big[\frac{M_2(r)|W|^2}{\lambda^4\varpi^2(r)}-\frac{2\left(2\varpi(r)+4\lambda\xi(r)+\lambda^2\varrho(r)\right)}{\varpi(r)\Delta_2}+\frac{\Delta_4}{\Delta_2^2}+\frac{4\Delta_3}{\Delta_2^2}+\frac{2}{\Delta_2}\Big].
\end{aligned}$$

$$\begin{aligned}
\Delta_k &= E\left(\int_W \rho(s)ds\right)^k,\\
\varpi(r) &= \int_W\int_W \kappa_h(r-\|u-v\|)\omega(u,v)g(\|u-v\|)dudv,\\
\xi(r) &= \int_W\int_W\int_W \kappa_h(r-\|u-v\|)\omega(u,v)\prod_{\substack{m,n\in\{u,v,h\}\\ m\neq n}} g(\|m-n\|)dudvdh,\\
\varrho(r) &= \int_W\int_W\int_W\int_W \kappa_h(r-\|u-v\|)\omega(u,v)\prod_{\substack{m,n\in\{u,v,h,g\}\\ m\neq n}} g(\|m-n\|)dudvdhdg,\\
M_k(r) &= E[\vartheta^k(r)], \quad where \quad \vartheta(r)=\frac{1}{|W|}\sum_{i=1}^{N}\sum_{j\neq i}\kappa_h(r-\|s_i-s_j\|)\omega_{ij}.
\end{aligned}$$

In particular, after rearrangement, the bias of $\hat{g}(r)$ for one-dimensional LGCPs is

$$\begin{aligned}
\text{Bias}(\hat{g}(r)_{\mathbb{R}^1}) &= \frac{\lambda^2|W|}{2\Delta_2}\Big[\varpi(r)\Big(1+\frac{\Delta_4}{\Delta_2^2}+4\frac{\Delta_3}{\Delta_2^2}\Big)-\frac{4\lambda\xi(r)}{\Delta_2}-\frac{\lambda^2\varrho(r)}{\Delta_2}\Big]-\exp(\sigma^2\zeta(r)),\\
\text{Var}(\hat{g}(r)_{\mathbb{R}^1}) &= \frac{M_2(r)|W|^4}{4\Delta_2^2}-\frac{\lambda^4|W|^2\varpi^2(r)}{4\Delta_2^2}\Big[\frac{2\left(2\varpi(r)+4\lambda\xi(r)+\lambda^2\varrho(r)\right)}{\varpi(r)\Delta_2}-\frac{\Delta_4}{\Delta_2^2}-\frac{4\Delta_3}{\Delta_2^2}-\frac{2}{\Delta_2}\Big].
\end{aligned}$$

Evaluation require calculation of Δ_k as follows. To begin, note that $\Delta_1 = E\left(\int_W \rho(s)ds\right) = \lambda|W|$, but for higher orders, calculation of Δ_k is not trivial, and we consider two potential methods. First, we calculate the average of $(\sum_{s\in W}\rho(s)ds)^k$ to obtain an approximation based on N' simulated events. This approximation improves as N' increases, so we assess the accuracy. The variability of the estimates under different choices of N', where N'=100, 500, 1000, 5000, and 10000. From Figure 2.1, we see that as N' increases, the variability of the estimates decreases, thus leading to narrower confidence intervals. Also, we find that the empirical standard errors associated with N'=100, 500, 1000, 5000, and 10000 for the cases with $\mu = 3$, $\sigma^2 = 0.2$ and scale parameter $\beta = 0.1$ are 20.278, 9.031, 6.216, 2.882 and 2.020. To assess performance, we fit a linear relationship between the ratios of the error and the square root of N', yielding estimates for the intercept and slope of 0.009 and 1.003 respectively, providing some information about the convergence rate. Similar findings exist for Δ_k with $k = 3, 4$, therefore, the desired estimate can be obtained according to different initial values and preliminary simulations. For Δ_2, we need to run at least 20,000 simulations to get standard error less than 1.5, and the estimate is $\Delta_2 \approx 511.3$. Table 2.1 shows that the point estimates do not change much for larger values of N'. Therefore, based on a similar philosophy using 250,000 simulations, the third-order and fourth-order estimates are $\Delta_3 \approx 12223.32$ ($sd = 15.86$) and $\Delta_4 \approx 303307.8$ ($sd = 561.00$).

We can also refine the simulation to calculate Δ_k based on further mathematical derivation. Taking Δ_2 as an example, we see that

$$\Delta_2 = \exp(2\mu + \sigma^2) \lim_{N\to\infty} \sum_{n\neq m}^{N} \exp(\sigma^2 \zeta(\frac{|n-m|}{N})) \frac{|W|^2}{N^2}$$

where $\lim_{N\to\infty} \sum_{n\neq m}^{N} \exp\left(\sigma^2\zeta(\frac{|n-m|}{N})\right) \frac{|W|^2}{N^2} \approx |W|^2 + 2|W|^2 \Big[\beta^2 \exp\left(\sigma^2 \exp(-\frac{1}{\beta})\right) E(\frac{1}{\Psi^2} |\Psi > 0) - \beta \exp\left(\sigma^2 \exp(-\frac{2}{\beta})\right) \left(E(\frac{1}{\Omega}|\Omega > 0) + \beta E(\frac{1}{\Omega^2}|\Omega > 0)\right)\Big]$ with Ψ and Ω following Poisson distributions, i.e., $\Psi \sim Pois\left(\sigma^2 \exp(-\frac{1}{\beta})\right)$; $\Omega \sim Pois\left(\sigma^2 \exp(-\frac{2}{\beta})\right)$. Although this method simplifies the simulation, requiring only draws from the two Poisson distribution, it still does not have a closed form and has limitations especially when σ^2 or β are small. As a result, we prefer the first method to estimate Δ_k.

To calculate the multiple integrations in $\varpi(r)$, $\xi(r)$ and $\varrho(r)$, some numerical techniques will be necessary, i.e., the Gaussian-Hermite Quadrature method. The basic idea is to replace integrals by summations of the value of the integrand at a sequence of points within the range of integration. In general,

$$\int_R \cdots \int_R f(x_1, \ldots, x_m) dx_1 \cdots dx_m \approx \sum_{i=1}^{M} \omega_i f(y_{i1}, \ldots, y_{im})$$

where ω_i are the weights and $(y_{i1}, \ldots, y_{im})$ are called the nodes. The most commonly used rule is the Gauss-Legendre rule. Here, we use MATLAB Quadrature function to compute approximate values of the multiple integrals. For instance, given $r = 0.2$, we get the following results $\varpi(r) \approx 2.069$, $\xi(r) \approx 2.233$ and $\varrho(r) \approx 2.5025$ when we use Diggle's edge correction method, yielding an estimated bias of -0.036, which is similar to our simulation result -0.032. Figure 2.2 shows the results for the estimated bias based on the theoretical formula and simulation for scenarios 1 and 4. The similarity between estimates provides some verification of our theoretical derivation; however, due to the approximation of the Taylor expansion, there still exists a gap between the estimate and the true value. This combined algorithm based on theoretical formulation and finite sample simulation of the k^{th} $(k <= 4)$ moment has comparable results and fast calculation compared with the non-parametric approach.

2.4 Simulation

W first assess the performance of Ripley's correction method in two-dimensional spatial data through simulation. We investigate three scenarios with $\mu = 2, 4, 6$, $\sigma^2 = 3$ and an exponential correlation structure with a scale parameter $\beta = 0.01$ under infill asymptotics, and 1000 Monte Carlo data sets generated from the unit square $[0, 1] \times [0, 1]$ for each scenario. Figure 2.3 shows the non-parametric estimates versus the underlying true values of PCF and the boxplots of estimates at ten distinct distance lags in each set-up. Figure 2.4 provides the bias of the PCF and $K-$function estimates. Note that $\hat{g}(0)$ is undefined and in practice, estimation of $\hat{g}(r)$ are unstable

Table 2.1: The estimates (standard error) of Δ_2, Δ_3 and Δ_4 for different N'

	$\mu=3,\sigma^2=0.2$			$\mu=3,\sigma^2=1$		
	Δ_2 (SD)	Δ_3 (SD)	Δ_4 (SD)	Δ_2 (SD)	Δ_3 (SD)	$\Delta_4(SD)$ (SD)
$N'=100$	510.75 (20.28)	12197.84 (763.80)	302265.08 (27109.45)	1350.23 (165.16)	69043.19 (20756.86)	4517442 (4068465)
$N'=500$	511.03 (9.03)	12211.46 (337.50)	302829.81 (11918.64)	1355.92 (71.82)	69672.89 (9193.27)	4591710 (1976767)
$N'=1000$	511.15 (6.22)	12217.14 (235.66)	303059.06 (8456.84)	1357.54 (51.62)	69822.23 (6410.37)	4593132 (1212178)
$N'=5000$	511.16 (2.88)	12216.77 (108.39)	303042.17 (3840.12)	1357.25 (23.19)	69758.43 (2774.77)	4573329 (489640)
$N'=10000$	511.17 (2.02)	12217.60 (76.08)	303084.79 (2701.53)	1357.58 (16.49)	69864.93 (2067.03)	4605783 (435673)

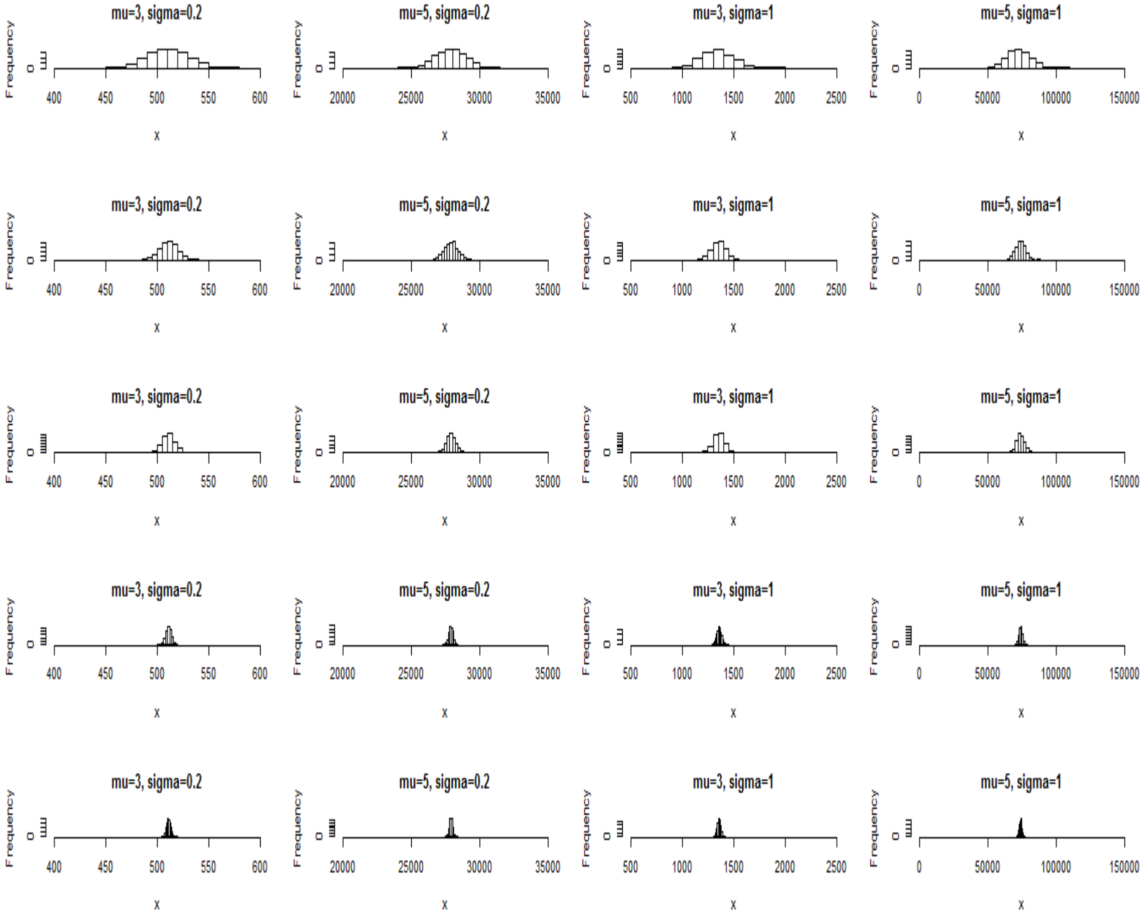


Figure 2.1: The histograms of Δ_2 for four scenarios. From top to bottom, $N'=100, 500, 1000, 5000, 10000$.

at very small values of r. Moreover, we can see that the PCF estimates adjusted by Ripley's correction perform well for moderate distance values, i.e., $0.02 < r < 0.25$. As we expect, as μ increases, the bias of PCF estimates decreases, and the variability of the estimates also decreases. In particular, the degree of bias reduction of the PCF at distance lags $r = 0.05, 0.10, 0.15$ is shown in Figure 2.5, which shows the bias decreasing to 0 with some oscillations as μ increases to 6. The results indicate that non-parametric estimates of the PCF are consistent under infill asymptotics and that Ripley's edge-correction performs well in our simulation studies for the two-dimensional spatial data.

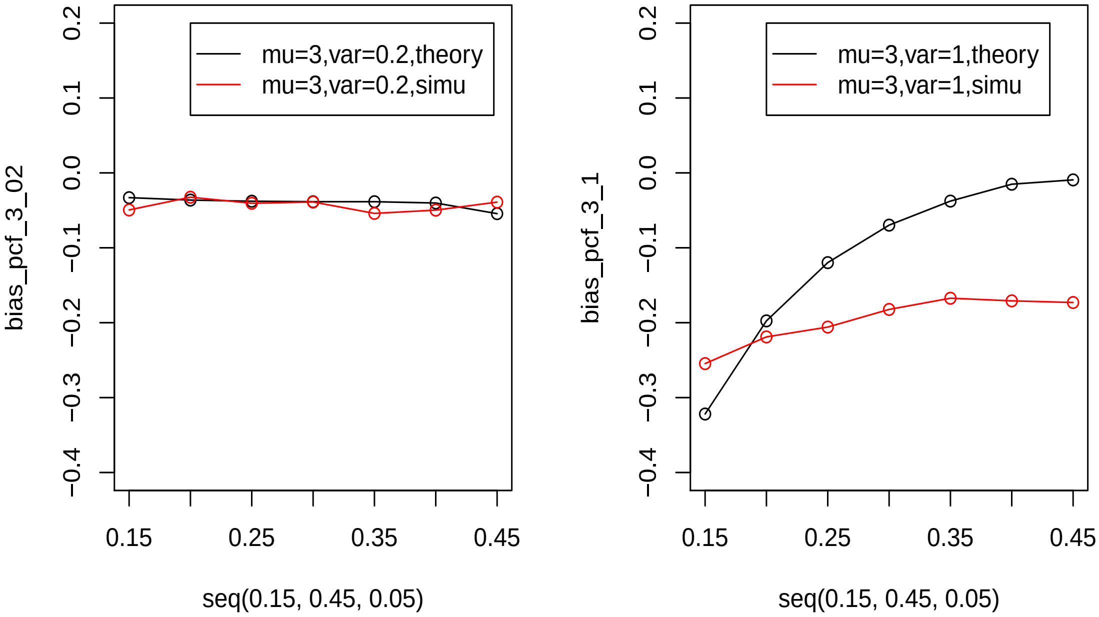


Figure 2.2: The estimated bias based on the theoretical derivation and simulation

Next we conduct extensive simulations to assess the performance of the two edge-correction approaches on the bias of non-parametric PCF estimates for one-dimensional LGCPs. Nine scenarios with different set-ups of parameters are listed as follows: 1)

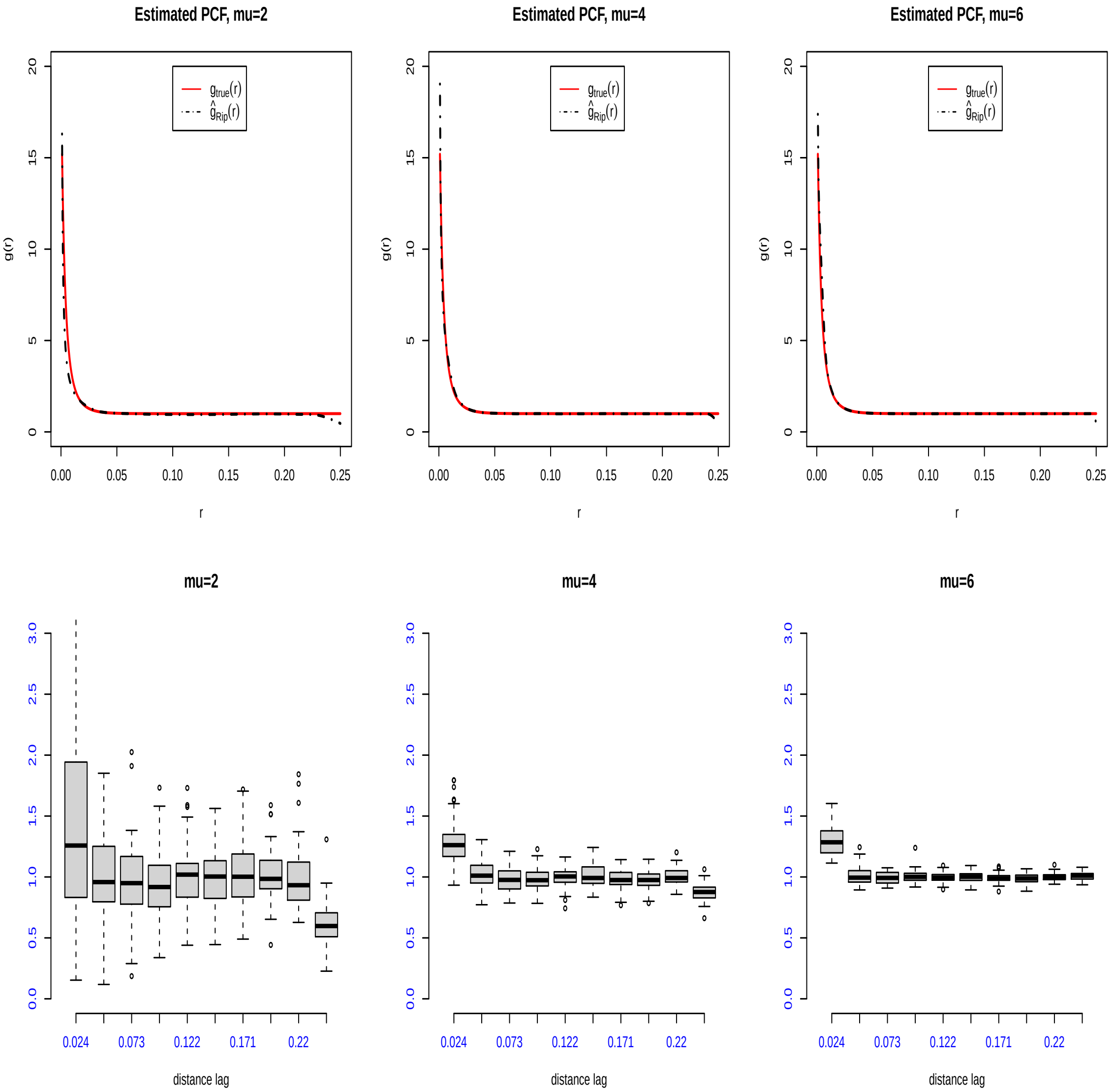


Figure 2.3: Non-parametric PCF estimates and the corresponding boxplots at ten distinct distance lags for two-dimensional LGCPs

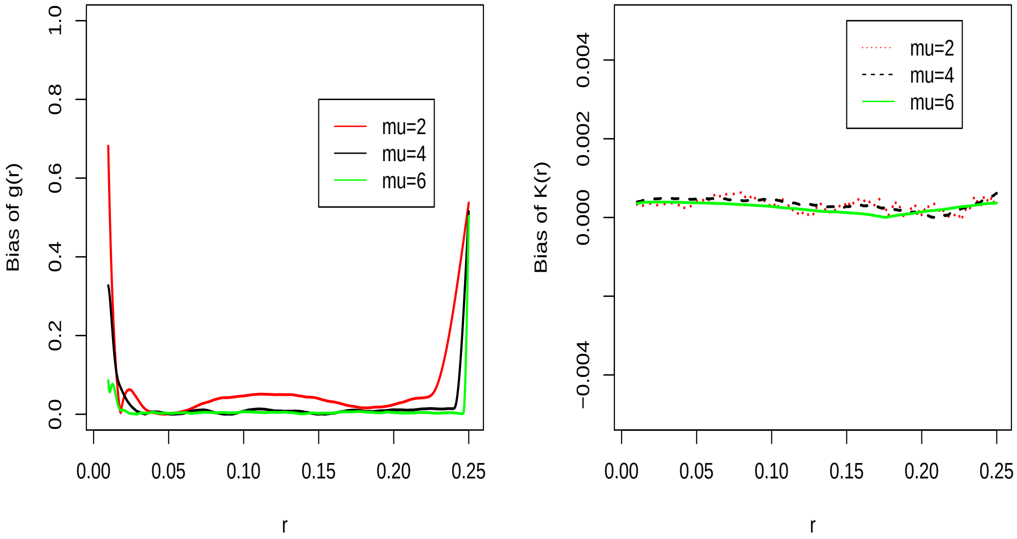


Figure 2.4: Bias of non-parametric PCF estimates for two-dimensional LGCPs

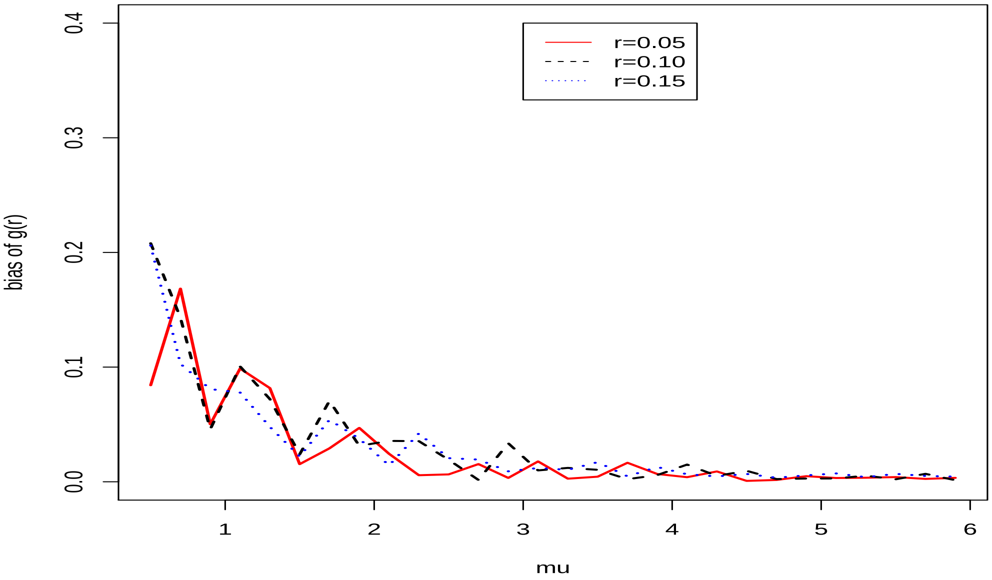


Figure 2.5: Bias of non-parametric PCF estimates over different means for three distance lags from two-dimensional LGCPs

$\mu = 3, \sigma^2 = 0.2$; 2) $\mu = 5, \sigma^2 = 0.2$; 3) $\mu = 8, \sigma^2 = 0.2$; 4) $\mu = 3, \sigma^2 = 1$; 5) $\mu = 5, \sigma^2 = 1$; 6) $\mu = 8, \sigma^2 = 1$; 7) $\mu = 3, \sigma^2 = 3$; 8) $\mu = 5, \sigma^2 = 3$; 9) $\mu = 8, \sigma^2 = 3$. The same exponential correlation form with $\beta = 0.1$ is used for all scenarios. Without loss of generality, suppose that each realization of an one-dimensional log-Gaussian Cox process is simulated within the line region [0, 1]. For each scenario, we generate 1000 realizations, and calculate the mean and standard deviation of the nonparametric edge-corrected PCF estimates. Let $g_{true}(r)$ denote the true underlying PCF, $\hat{g}_{rip}(r)$ as the PCF estimate with Ripley's edge-correction, and $\hat{g}_{dig}(r)$ denotes the estimate with Diggle's edge-correction. We note that the theoretical formula of the PCF is $g(r) = \exp\{\sigma^2 \exp(-\frac{r}{\beta})\}$ where r is the length of the interval for any pair of points.

According to the definition of LGCPs, the intensity measure $\rho(s)$ follows a Gaussian Process and may be used to generate events $X(s)$ by acting as its intensity. Our procedures to simulate one realization of one-dimension LGCPs are as follows:

- Generate a realization $\{Y(s) : s \in I\}$ from a stationary Gaussian process with mean μ and covariance function

  $$Cov(s_i, s_j) = \sigma^2 \exp\left(-\frac{\|s_i - s_j\|}{\beta}\right)$$

  where $I = \{0, \frac{1}{K}, \frac{2}{K}, \ldots, \frac{K-1}{K}, 1\}$ with $K = 100$ and $\|s_i - s_j\|$ is the length of the interval between s_i and s_j;

- Given $Y(s)$, generate one realization $\rho(s) = \exp(Y(s))$ of the intensity process;

- For each grid, generate $N(s)$ from a Poisson distribution with mean $\rho(s)/K$;

- Generate $\{X(s), i = 1, 2, \ldots, N(s)\}$ which are uniformly distributed in the s^{th} grid, and remove the points outside the interval [0,1];

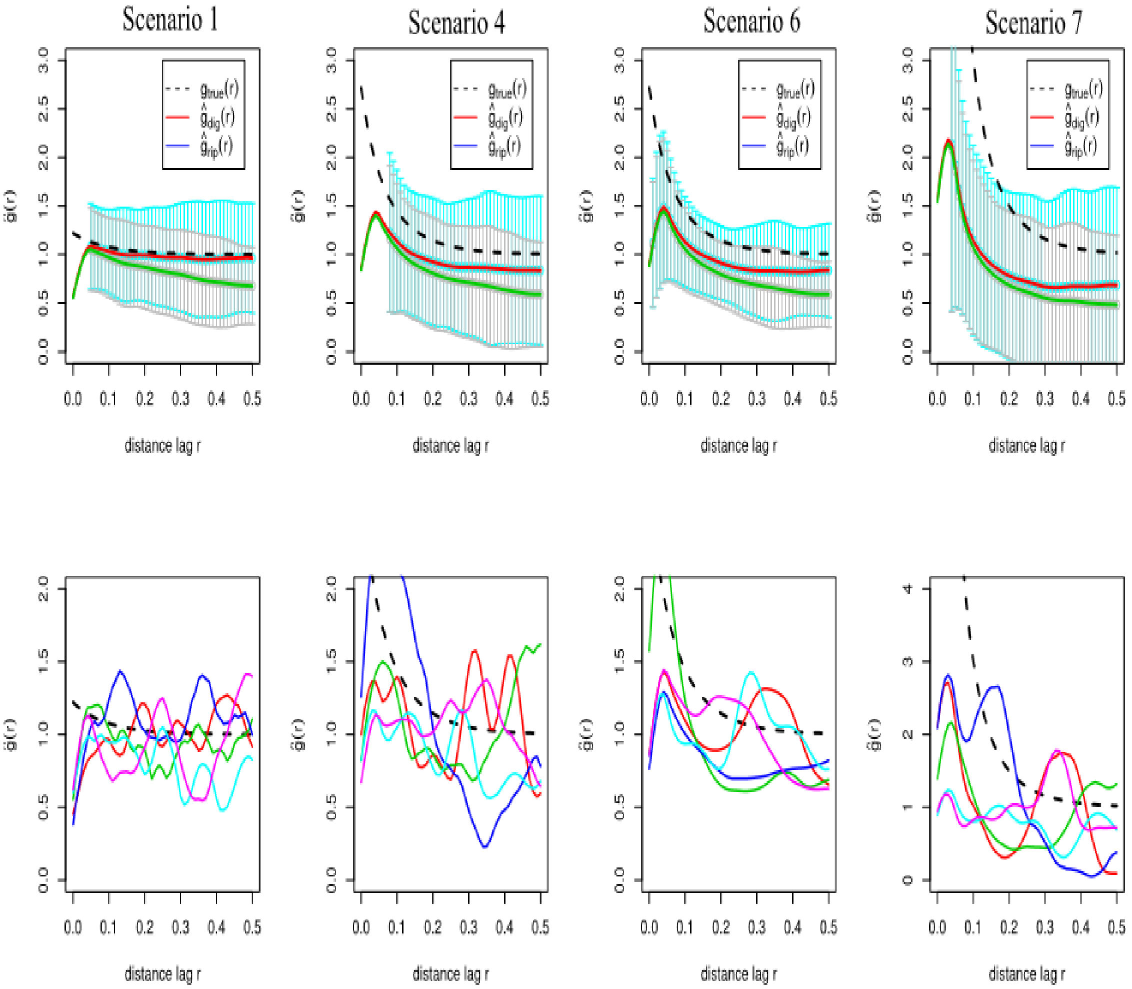


Figure 2.6: Mean of PCF estimates adjusted by Ripley's edge-correction method and Diggle's edge-correction method with 95% empirical confidence interval from 1000 realizations of one-dimensional LGCPs and the bottom plots give five plots of PCF estimates with Diggle's edge-correction method randomly sampled from the full set of simulation. The plots are for scenarios 1, 4, 6, 7 respectively from left to right. In the top row of plots, the black dashed line is the true underlying PCF; the red line represents the PCF estimates with Diggle's edge-correction method; the blue dashed line means the PCF estimates with Ripley's edge-correction method.

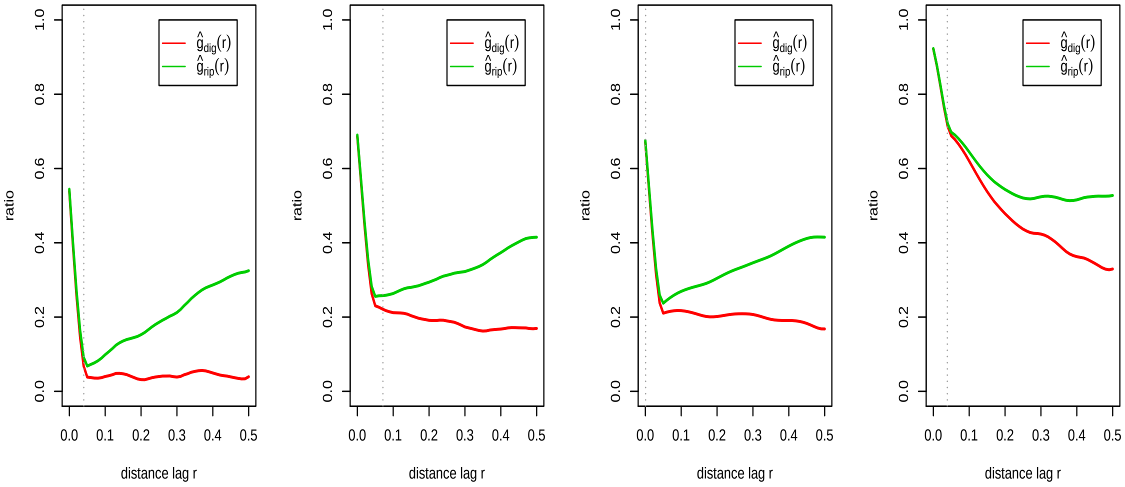


Figure 2.7: Relative ratio of the mean PCF estimates adjusted by Ripley's edge-correction method and Diggle's edge-correction method versus the true value for scenarios 1, 4, 6, 7 respectively from left to right. The dashed gray line represents the maximum of 1000 minimum pairwise distances from 1000 Monte Carlo data.

The plots for the mean of PCF estimates in scenarios 1, 4, 6, 7 are shown in Figure 2.6, where $\hat{g}_{dig}$ performs better than $\hat{g}_{rip}$ in terms of bias reduction; however, the estimates at small distance lags r are not reliable for all scenarios because in some realizations with small μ or σ^2, the minimum pairwise distance may be greater than the smallest value of r in the plot, thus few pairs will be incorporated in the formula to calculate the estimate of PCF. Therefore, we do not provide the 95% confidence interval for the estimates at distance lag r less than the maximum of 1000 minimum pairwise distances. We also find out that as σ^2 increases, the bias will increase. In the middle two plots, the confidence intervals tend to be narrower as μ increases, which is similar as two-dimensional cases and can be explained and confirmed by theoretical inference. Five random samples of the PCF estimates with Diggle's edge-correction method are plotted at the bottom to show the variability of estimated PCFs from the same underlying process. Similar findings are obtained from Figure 2.7 showing the

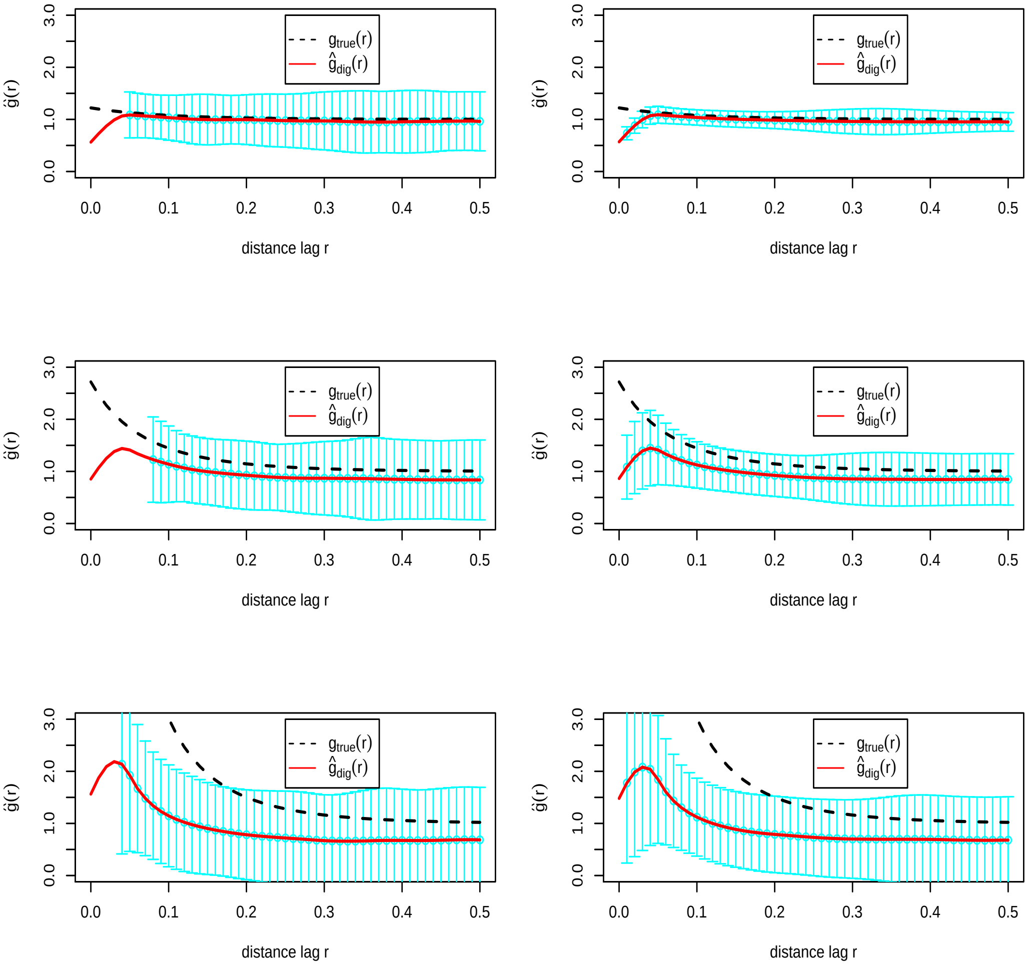


Figure 2.8: Mean of PCF estimates adjusted by Diggle's edge-correction method with 95% empirical confidence intervals from 1000 realizations of one-dimensional LGCPs on $[0, 1]$. The top two plots are based on simulations under Scenarios 1 and 2; the middle two plots on simulations under Scenarios 4 and 5; and the bottom two plots on simulations under Scenarios 7 and 8. In each plot, the black dashed line is the true underlying PCF; the red line represents the mean of PCF estimates with Diggle's edge-correction method.

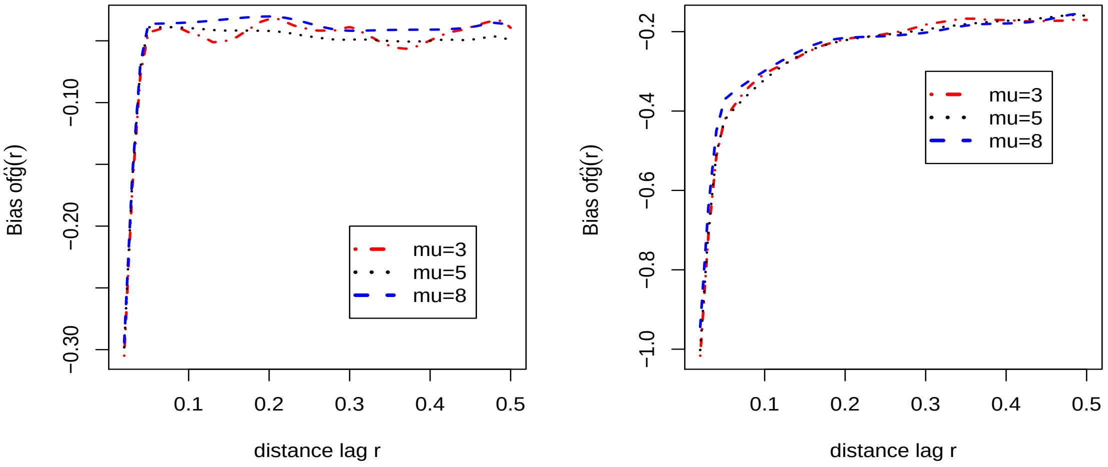


Figure 2.9: Bias of PCF estimates adjusted by Diggle's edge-correction method from 1000 realization of one-dimensional LGCPs on $[0, 1]$. The left panel is for scenarios 1-3, and the right panel is for scenarios 4-6.

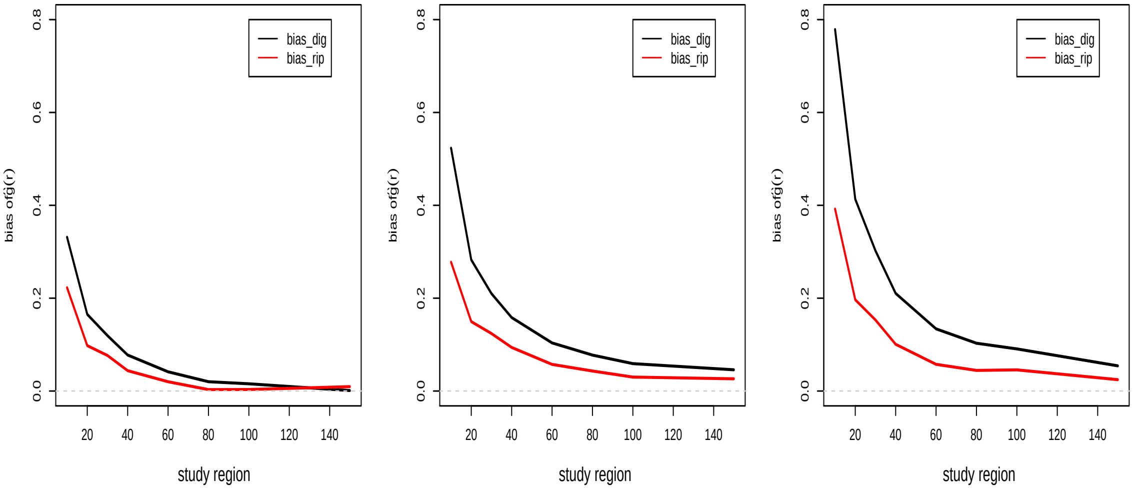


Figure 2.10: Bias of PCF estimates under increasing domain asymptotics based on 1000 realizations of one-dimensional LGCPs on $[0, 1]$. The left plot is for distance lag $r = 1$; the middle plot is for distance lag $r = 3$ and the right plot is for distance lag $r = 5$.

relative ratio of the mean PCF estimates with both edge-corrections, illustrating that Diggle's method versus the true value is preferable for one-dimensional spatial data analysis for the scenarios under consideration. More information about the effect of μ and σ^2 on PCF estimation appears in Figures 2.8-2.9. As above, we find that for the scenarios with the same μ, greater σ^2 leads to larger range of variation and increased bias. Also, for distance lags $0.2 < r < 0.5$, estimation becomes more reliable as the number of point pairs increases, however this does not hold for very small distance lags, consistent with previous studies in literature (Diggle, 2003). Another interesting note is that for the scenarios with equal σ^2, as μ increases under infill asymptotics, the expected number of points increases, and the bias of PCF estimates at small distance lags decreases due to the emergence of more pairs of points with short inter-point distances, but there exists little difference for very small distance lags $r < 0.1$. In conclusion, the bias of non-parametric PCF estimation under infill asymptotics is sensitive to both μ and σ^2, but particularly to σ^2. Also, Diggle's edge-correction method is recommended for one-dimensional spatial LGCPs with relatively small value of σ^2 to reduce bias at moderate distance lags, but additional analysis is necessary to fully determine this edge-correction's feasible usage.

Additional results on the bias of PCF estimates under increasing domain asymptotics are shown in Figure 2.10. We first simulate spatial points from one realization of LGCPs with $\mu = 1$, $\sigma^2 = 1$ and $\beta = 0.5$ in a fixed interval length [0, 300]. Afterwards, we can choose subset windows with lengths of 10, 20, 30, 40, 60, 80, 100 and 150 to generate the increasing domains. In this case, Ripley's edge-correction method performs better than Diggle's under increasing domain asymptotics, and, as the interval length increases, the bias decreases and a wider interval window would be needed to reduce bias at larger distance lags. This gives some evidence that the choice of edge-correction method for one-dimensional spatial data depends on the sample design the asymptotics of interest and the research questions.

In addition, we discuss the issues about nonparametric PCF estimates for large distance lags. Simulation results show that the means of the non-parametric PCF estimates decrease for distance lags $r > 0.5$, where partial results for scenarios 1 and 3 shown in Figure 2.11. To investigate the reason for this, we check the number of unique pairs of points for calculating the non-parametric estimate of the PCF, and note that after $r > 0.5$, all of the observed pairs would be edge-corrected, and that weights in Diggle's correction methods only depend on the pointwise distances, which is not large enough to fully correct the edge effect for large r. To adjust the substantial bias for large $r >= 0.5$, we propose another edge-corrected method, where ω_{ij} is the reciprocal of the proportion of the interval of an annulus with s_i as the center and radius $r_1 = r - \sqrt{5}h$ and $r_2 = r + \sqrt{5}h$ that lies within the study area, while for $r < 0.5$, we use the same ω_{ij} as Diggle's correction method. From Figure 2.12, we can see the proposal performs better than either of the Ripley's or Diggle's correction for relatively larger distance lags.

2.5 Conclusion and Discussion

In this article, we investigate the bias and variance of non-parametric estimators of the PCF from log-Gaussian Cox processes, especially one-dimensional spatial data via simulation and theoretical inference. The non-parametric PCF estimates are important, because those values could be used, say in the minimum contrast method, to estimate parametric forms of log-Gaussian Cox processes. In most cases, empirical estimation remains challenging for a number of reasons, and more so if we consider a spatial point process in one dimension (points along a line). Our results provide informative to build optimal sampling designs and obtain efficient parametric modeling.

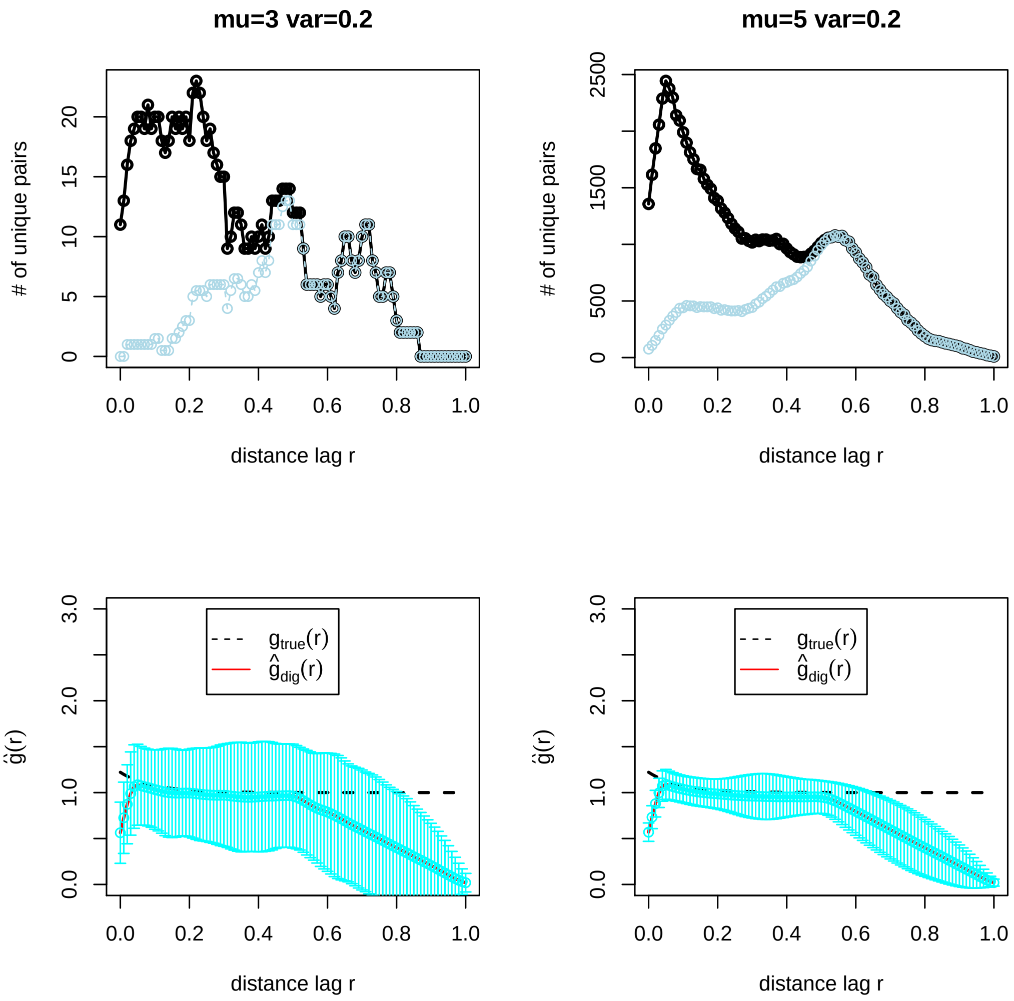


Figure 2.11: The left panel is for scenario 1 and the right panel is simulated from scenario 2. The top two plots indicate the number of unique pairs used for calculating non-parametric estimate of PCF and the number of pairs having edge-correction adjustment, and the bottom two plots show the mean of the PCF estimates adjusted by Diggle's edge-correction method with 95% empirical confidence interval from 1000 realizations of one-dimensional LGCP on $[0, 1]$.

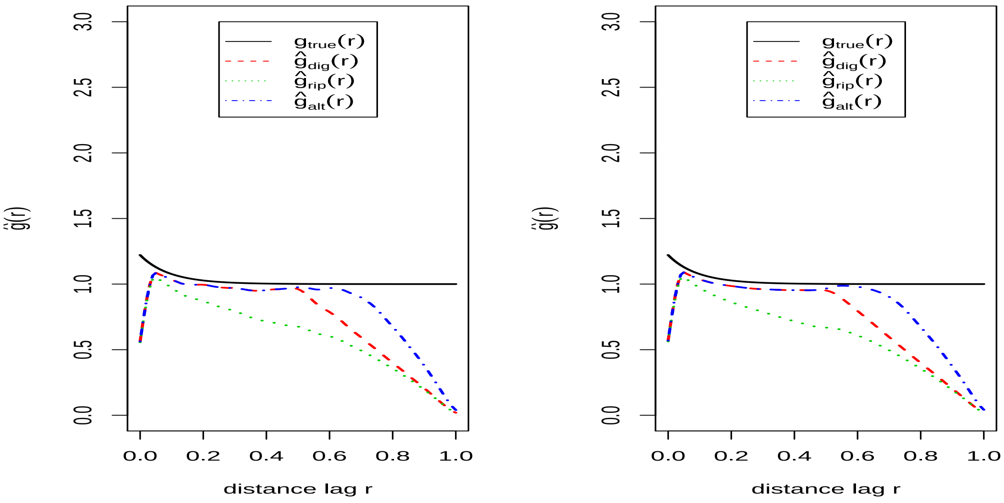


Figure 2.12: The PCF estimates with different edge-correction methods for scenarios 1 and 2

We consider two asymptotic settings, infill asymptotics and increasing domain asymptotics, but our focus is primarily on the former. We evaluated two edge-correction approaches to assess their impact on the performance of non-parametric estimators of PCF. Our simulations suggest that Diggle's edge-correction is easy to implement and has less bias than Ripley's approach at moderate distance lags, and performs well under infill asymptotics. We provided an algorithm based on theoretical formulation combined with finite sample simulation of the k^{th} ($k <= 4$) moment with respect to the integration of the intensity function to estimate the PCF. Compared with the non-parametric approach, this algorithm has comparable results and fast calculation. Also, we propose a new edge-correction method for larger distance lags to (slightly) improve existing methods by further reducing bias.

The derivation of the bias and the variance for the estimate of the PCF is not trivial. Accordingly, we found out that bias decreases as the mean of the Gaussian process defining intensity, μ, increases, and also that the bias increases as the variance of the intensity, σ^2, increases given fixed values of other parameters. This trend can be tracked for both infill and increasing domain asymptotics. For the increasing domain asymptotics, we found the bias decreases as the study region increases, which agrees with our general expectation. Future work will explore the impact of additional elements of the estimator on performance. The bandwidth is of particular interest to build on earlier work of Fiksel (1998).

2.6 Appendix

2.6.1 The bias of $\hat{g}(r)$

Using the law of iterated expectation, we have $E(\hat{g}(r)) = E\Big[E(\hat{g}(r)|\rho(\cdot))\Big]$. Let

$$\hat{\psi}(r) = \sum_{i=1}^{N} \sum_{j \neq i} \kappa_h(r - \|s_i - s_j\|)\omega_{ij}$$

$$\vartheta = \frac{\hat{\psi}(r)}{|W|}$$

$$M = \frac{\nu N(N-1)}{|W|^2}$$

Therefore, $\hat{g}(r) = \frac{\vartheta}{M}$. Based on the delta method, we have

$$E(\hat{g}(r)) = E(\frac{\vartheta}{M}) \approx \frac{E(\vartheta)}{E(M)}\Big[1 + \frac{Var(M)}{(E(M))^2} - \frac{Cov(\vartheta, M)}{E(\vartheta)E(M)}\Big]. \tag{2.1}$$

First, note that $E(N(N-1)) = E[E(N(N-1))|\rho(\cdot)]$. Also, under the framework of LGCP, we have $Var(N|\rho(\cdot)) = E(N|\rho(\cdot)) = \int_W \rho(s)ds$, and $E(N^2|\rho(\cdot)) = \int_W \rho(s)ds+$

$(\int_W \rho(s)ds)^2$. Hence, we find

$$E(N(N-1)) = E\left(\int_W \rho(s)ds\right)^2. \tag{2.2}$$

For log-Gaussian Cox processes, we know that $\lambda = E(\rho(s)) = \exp(\mu + \frac{1}{2}\sigma^2)$, and $Var(\rho(s)) = [\exp(\sigma^2) - 1]\exp(2\mu + \sigma^2)$. Thus, we have $E(\rho(s)^2) = Var(\rho(s)) + \lambda^2 = \exp(2\mu + 2\sigma^2)$. Therefore,

$$\begin{aligned}
E(\int_W \rho(s)ds)^2 &= E\Big[\lim_{N\to\infty}\Big(\sum_{n=1}^{N}\rho(\frac{n}{N})\frac{|W|}{N}\Big)\Big(\sum_{m=1}^{N}\rho(\frac{m}{N})\frac{|W|}{N}\Big)\Big] \\
&= E\Big[\lim_{N\to\infty}\Big(\sum_{n=m}^{N}\rho^2(\frac{n}{N})\frac{|W|^2}{N^2} + \sum_{n\neq m}^{N}\rho(\frac{n}{N})\rho(\frac{m}{N})\frac{|W|^2}{N^2}\Big)\Big] \\
&= \lim_{N\to\infty}\Big[\sum_{n=m}^{N}E(\rho^2(\frac{n}{N}))\frac{|W|^2}{N^2} + \sum_{n\neq m}^{N}E(\rho(\frac{n}{N})\rho(\frac{m}{N}))\frac{|W|^2}{N^2}\Big] \\
&= \lim_{N\to\infty}\Big[\exp(2\mu+2\sigma^2)\sum_{n=m}^{N}\frac{|W|^2}{N^2} + \sum_{n\neq m}^{N}E(\exp(Y(\frac{n}{N}))\exp(Y(\frac{m}{N})))\frac{|W|^2}{N^2}\Big] \\
&= \lim_{N\to\infty}\Big[\exp(2\mu+2\sigma^2)\sum_{n=m}^{N}\frac{|W|^2}{N^2} + \sum_{n\neq m}^{N}E(\exp(Y(\frac{n}{N}) + Y(\frac{m}{N})))\frac{|W|^2}{N^2}\Big] \\
&= \lim_{N\to\infty}\Big[\exp(2\mu+2\sigma^2)\sum_{n=m}^{N}\frac{|W|^2}{N^2} + \sum_{n\neq m}^{N}\exp\Big(2\mu + \sigma^2(1+\zeta(\frac{|n-m|}{N}))\Big)\frac{|W|^2}{N^2}\Big] \\
&= \exp(2\mu+2\sigma^2)\lim_{N\to\infty}(\frac{|W|^2}{N}) + \exp(2\mu+\sigma^2)\lim_{N\to\infty}\sum_{n\neq m}^{N}\exp(\sigma^2\zeta(\frac{|n-m|}{N}))\frac{|W|^2}{N^2}.
\end{aligned}$$

Unfortunately, there is no closed form for this result, thus we calculate the average of $(\sum_{s\in W}\rho(s)\Delta s)^2$ to obtain the approximation based on N simulated data sets, which has discussed in Section 2.3. Also, we know that $E(M) = \frac{\nu}{|W|^2}E(N(N-1)) = \frac{\nu}{|W|^2}E(\int_W \rho(s)ds)^2$. Next,

$$\begin{aligned}
Var(N(N-1)) &= Var\Big[E(N(N-1))|\rho(\cdot)\Big] + E\Big[Var(N(N-1))|\rho(\cdot)\Big] \\
&= Var\Big[(\int_W \rho(s)ds)^2\Big] + E\Big[4(\int_W \rho(s)ds)^3 + 2(\int_W \rho(s)ds)^2\Big] \\
&= E(\int_W \rho(s)ds)^4 - (E(\int_W \rho(s)ds)^2)^2 + 4E(\int_W \rho(s)ds)^3 + 2E(\int_W \rho(s)ds)^2.
\end{aligned}$$

We can derive that

$$
\begin{aligned}
E(\int_W \rho(s)ds)^3 &= E\Big[\lim_{N\to\infty}\Big(\sum_{n=1}^{N}\rho(\frac{n}{N})\frac{|W|}{N}\Big)\Big(\sum_{m=1}^{N}\rho(\frac{m}{N})\frac{|W|}{N}\Big)\Big(\sum_{j=1}^{N}\rho(\frac{j}{N})\frac{|W|}{N}\Big)\Big] \\
&= E\Big[\lim_{N\to\infty}\Big(\sum_{n=m=j}^{N}\rho^3(\frac{n}{N})\frac{|W|^3}{N^3} + \sum_{\substack{n\neq m\\ m=j}}^{N}\rho(\frac{n}{N})\rho^2(\frac{m}{N})\frac{|W|^3}{N^3} + \sum_{\substack{n\neq m\\ n=j}}^{N}\rho^2(\frac{n}{N})\rho(\frac{m}{N})\frac{|W|^3}{N^3} \\
&\quad + \sum_{\substack{n=m\\ m\neq j}}^{N}\rho^2(\frac{n}{N})\rho(\frac{j}{N})\frac{|W|^3}{N^3} + \sum_{n\neq m\neq j}^{N}\rho(\frac{n}{N})\rho(\frac{m}{N})\rho(\frac{j}{N})\frac{|W|^3}{N^3}\Big)\Big] \\
&= \lim_{N\to\infty}\Big(\sum_{n=m=j}^{N}\exp(3\mu+\frac{9}{2}\sigma^2)\frac{|W|^3}{N^3} + 2\sum_{\substack{n\neq m\\ m=j}}^{N}\exp(3\mu+\frac{5}{2}\sigma^2+2\sigma^2\zeta(\frac{|n-m|}{N}))\frac{|W|^3}{N^3} \\
&\quad + \sum_{\substack{n\neq m\\ m=j}}^{N}\exp(3\mu+\frac{5}{2}\sigma^2+2\sigma^2\zeta(\frac{|n-j|}{N}))\frac{|W|^3}{N^3} + \sum_{n\neq m\neq j}^{N}E(\rho(\frac{n}{N})\rho(\frac{m}{N})\rho(\frac{j}{N}))\frac{|W|^3}{N^3}\Big) \\
&= \exp(3\mu+\frac{9}{2}\sigma^2)\lim_{N\to\infty}(\frac{|W|^3}{N^2}) + 2\lim_{N\to\infty}\sum_{\substack{n\neq m\\ m=j}}^{N}\exp(3\mu+\frac{5}{2}\sigma^2+2\sigma^2\zeta(\frac{|n-m|}{N}))\frac{|W|^3}{N^3} \\
&\quad + \lim_{N\to\infty}\sum_{\substack{n\neq m\\ m=j}}^{N}\exp(3\mu+\frac{5}{2}\sigma^2+2\sigma^2\zeta(\frac{|n-j|}{N}))\frac{|W|^3}{N^3} \\
&\quad + \lim_{N\to\infty}\sum_{n\neq m\neq j}^{N}E(\rho(\frac{n}{N})\rho(\frac{m}{N})\rho(\frac{j}{N}))\frac{|W|^3}{N^3}
\end{aligned}
$$

where the first term above approaches to 0 as $\lim_{N\to\infty}\frac{|W|^3}{N^2}\to 0$. The second and third terms also approximately to 0 because $\exp(3\mu+\frac{5}{2}\sigma^2+2\sigma^2\zeta(\frac{|n-m|}{N}))$ and $\exp(3\mu+\frac{5}{2}\sigma^2+2\sigma^2\zeta(\frac{|n-j|}{N}))$ are bounded and $\lim_{N\to\infty}N(N-1)\times\frac{|W|^3}{N^3}\to 0$. For the fourth term, $E(\rho(\frac{n}{N})\rho(\frac{m}{N})\rho(\frac{j}{N})) = \exp(3\mu+\frac{3}{2}\sigma^2+\sigma^2(\zeta(\frac{|n-m|}{N})+\zeta(\frac{|m-j|}{N})+\zeta(\frac{|n-j|}{N}))$. Similarly, $E(\int_W \rho(s)ds)^3$ does not have a closed form, thus it is also approximated by simulated data. A similar argument holds for $E(\int_W \rho(s)ds)^4$. Afterwards, we find

$$
\begin{aligned}
\frac{Var(M)}{(E(M))^2} &= \frac{Var(N(N-1))}{(E(N(N-1)))^2} \\
&= \frac{E(\int_W \rho(s)ds)^4}{(E(\int_W \rho(s)ds)^2)^2} + \frac{4E(\int_W \rho(s)ds)^3}{(E(\int_W \rho(s)ds)^2)^2} + \frac{2}{E(\int_W \rho(s)ds)^2} - 1.
\end{aligned}
$$

Second, by a change of variables (Federer, 1969), we can derive that

$$\begin{aligned}
E(\vartheta) &= E(\frac{\hat{\psi}(r)}{|W|}) \\
&= \frac{1}{|W|}E(E(\hat{\psi}(r)|\rho(\cdot))) \\
&= \frac{1}{|W|}E\Big[\int_W\int_W \kappa_h(r-\|u-v\|)\omega(u,v)\rho^{(2)}(u,v)dudv\Big] \\
&= \frac{\lambda^2}{|W|}\int_W\int_W \kappa_h(r-\|u-v\|)\omega(u,v)g(\|u-v\|)dudv \\
&= \frac{\lambda^2}{|W|}\int_W\Big[\int_0^L \sum_{v\in W:\|u-v\|=t} \kappa_h(r-\|u-v\|)\omega(u,v)g(\|u-v\|)dt\Big]du \\
&= \frac{\lambda^2}{|W|}\int_W\Big[\int_0^L \sum_{v\in W:\|u-v\|=t} \kappa_h(r-t)\omega(u,v)g(t)dt\Big]du.
\end{aligned}$$

where L is the maximum pairwise distance in the fixed window under infill asymptotics, otherwise, $L\to\infty$ for increasing domain. Under one-dimensional spatial point processes, the summation on the right hand side of Equation (11) is simply the number of points in $v \in W$ that are situated at the same distance of t from u. Thus, $\sum_{v\in W:\|u-v\|=t}\omega(u,v)$ should be the number of points on the surface of the boundary. Given $\omega(u,v)$ as Diggle's edge correction factor, $E(\vartheta) = 2\lambda^2\int_0^L \kappa_h(r-t)g(t)dt$.

Third, for $Cov(\vartheta, M)$, we need to find $E(N(N-1)\hat{\psi}(r))$. Motivated by the expression, we can rewrite

$$N(N-1)\hat{\psi}(r) = \sum_{i=1}^{N}\sum_{j\neq i}\sum_{k=1}^{N}\sum_{l\neq k}\varrho(s_i,s_j)\varrho(s_k,s_l)$$

where $\varrho(s_i,s_j)=\kappa_h(r-\|s_i-s_j\|)\omega_{ij}$ and $\varrho(s_k,s_l)=1$. After regrouping the summation above, we have seven types of combinations, $\{(s_i,s_j),(s_i,s_j)\}$, $\{(s_i,s_j),(s_j,s_i)\}$, $\{(s_i,s_j),(s_k,s_j)\}$, $\{(s_i,s_j),(s_k,s_i)\}$, $\{(s_i,s_j),(s_i,s_l)\}$, $\{(s_i,s_j),(s_j,s_l)\}$, $\{(s_i,s_j),(s_k,s_l)\}$ with summation denoted by S_A, S_B, S_C, S_D, S_E, S_F, S_G respectively. Obviously,

$S_A = S_B = \sum_{i=1}^{N}\sum_{j\neq i}\varrho(s_i, s_j)$, and we find

$$\begin{aligned} E(S_A) &= \lambda^2\int_W\int_W \kappa_h(r-\|u-v\|)\omega(u,v)g(\|u-v\|)dudv \\ &= \lambda^2\varpi(r). \end{aligned}$$

S_C, S_D, S_E and S_F have the same expectation. We can derive the expectations for S_C and S_G as follows

$$\begin{aligned} E(S_C) &= E(\int_W\int_W\int_W \kappa_h(r-\|u-v\|)\omega(u,v)\rho^{(3)}(u,v,h)dudvdh) \\ &= \lambda^3\int_W\int_W\int_W \kappa_h(r-\|u-v\|)\omega(u,v)\prod_{\substack{m,n\in\{u,v,h\}\\ m\neq n}} g(\|m-n\|)dudvdh \\ &= \lambda^3\xi(r), \end{aligned}$$

$$\begin{aligned} E(S_G) &= E(\int_W\int_W\int_W\int_W \kappa_h(r-\|u-v\|)\omega(u,v)\rho^{(4)}(u,v,h,g)dudvdhdg) \\ &= \lambda^4\int_W\int_W\int_W\int_W \kappa_h(r-\|u-v\|)\omega(u,v)\prod_{\substack{m,n\in\{u,v,g,h\}\\ m\neq n}} g(\|m-n\|)dudvdhdg \\ &= \lambda^4\varrho(r). \end{aligned}$$

Finally, we sum all the above results to find $E(N(N-1)\hat{\psi}(r)) = 2\lambda^2\varpi(r) + 4\lambda^3\xi(r) + \lambda^4\varrho(r) = \lambda^2\eta$ where $\eta = 2\varpi + 4\lambda\xi(r) + \lambda^2\varrho(r)$. Accordingly, we obtain $\frac{Cov(\vartheta,M)}{E(\vartheta)E(M)} = \frac{\eta}{\varpi(r)E(\int_W\rho(s)ds)^2} - 1$. In the end, we have

$$\begin{aligned} E(\hat{g}(r)) &= \frac{\lambda^2|W|\varpi(r)}{\nu E(\int_W\rho(s)ds)^2}\Big[1 + \frac{E(\int_W\rho(s)ds)^4}{(E(\int_W\rho(s)ds)^2)^2} + 4\frac{E(\int_W\rho(s)ds)^3}{(E(\int_W\rho(s)ds)^2)^2} \\ &\quad + \frac{2}{E(\int_W\rho(s)ds)^2}\Big] - \frac{\lambda^2|W|\eta}{\nu E(\int_W\rho(s)ds)^2}. \end{aligned}$$

In particular, for one-dimensional data, the above estimator can be written as

$$
\begin{aligned}
E(\hat{g}(r)_{\mathbb{R}^1}) &= \frac{\lambda^2|W|^2\int_0^L \kappa_h(r-t)g(t)dt}{E(\int_W \rho(s)ds)^2}\Big[1+\frac{E(\int_W \rho(s)ds)^4}{(E(\int_W \rho(s)ds)^2)^2}+4\frac{E(\int_W \rho(s)ds)^3}{(E(\int_W \rho(s)ds)^2)^2} \\
&\quad +\frac{2}{E(\int_W \rho(s)ds)^2}\Big]-\frac{\lambda^2|W|\eta}{2E(\int_W \rho(s)ds)^2}.
\end{aligned}
$$

Also, using Taylor expansion, we can get that

$$
\begin{aligned}
\int_0^L \kappa_h(r-t)g(t)dt &= \int_{r-L}^r \kappa_h(m)g(r-m)dm \\
&\approx \int_{r-L}^r \kappa_h(m)(g(r)-\dot{g}(r)m+\frac{\ddot{g}(r)}{2}m^2)dm \\
&\approx g(r)\int_{r-L}^r \kappa_h(m)dm-\dot{g}(r)\int_{r-L}^r \kappa_h(m)mdm \\
&\quad +\frac{\ddot{g}(r)}{2}\int_{r-L}^r \kappa_h(m)m^2dm
\end{aligned}
$$

Then, the bias of the estimated PCF can be obtained by $E(\hat{g}(r))-g(r)$. After plugging in the exponential correlation, i.e., $g(r)=exp(\sigma^2 exp(-\frac{r}{\beta}))$, we know that

$$
\dot{g}(r)=-\frac{\sigma^2}{\beta}\exp(-\frac{r}{\beta})g(r) \;\; and \;\; \ddot{g}(r)=\frac{\sigma^2}{\beta^2}\exp(-\frac{r}{\beta})g(r)\Big[1+log(g(r))\Big]
$$

Hence, the bias of $\hat{g}(r)$ is approximated by

$$
\begin{aligned}
E(\hat{g}(r))-g(r) &= \frac{\lambda^2|W|^2}{2E(\int_W \rho(s)ds)^2}\Big[2\int_0^L \kappa_h(r-t)g(t)dt\Big(1+\frac{E(\int_W \rho(s)ds)^4}{(E(\int_W \rho(s)ds)^2)^2} \\
&\quad +4\frac{E(\int_W \rho(s)ds)^3}{(E(\int_W \rho(s)ds)^2)^2}\Big)-\frac{4\lambda\xi(r)}{|W|E(\int_W \rho(s)ds)^2} \\
&\quad -\frac{\lambda^2\varrho(r)}{|W|E(\int_W \rho(s)ds)^2}\Big]-\exp(\sigma^2\exp(-\frac{r}{\beta}))
\end{aligned}
$$

Based on estimates from simulation, $E(\int_W \rho(s)ds)^4$ is greater than $(E(\int_W \rho(s)ds)^2)^2$, therefore, the bias from the above formula should be negative. Given fixed μ, as the variance σ^2 increases, the absolute value in the square brackets will increase, suggesting an increase in the absolute value of overall bias; however, it is not easy to detect consistent trends of bias when μ increases for a given value of σ^2.

2.6.2 The variance of $\hat{g}(r)$

We can also get the variance estimate of $\hat{g}(r)$ by the delta method as follows:

$$Var(\hat{g}(r)) = Var(\frac{\vartheta}{M}) \approx \frac{(E(\vartheta))^2}{(E(M))^2}\Big[\frac{Var(\vartheta)}{(E(\vartheta))^2} - 2\frac{Cov(\vartheta, M)}{E(\vartheta)E(M)} + \frac{Var(M)}{(E(M))^2}\Big]$$

The key part here is to calculate $Var(\vartheta) = E(\vartheta^2) - E(\vartheta)^2$. Similarly as above, we know that

$$\hat{\psi}(r)^2 = \sum_{i=1}^{N}\sum_{j\neq i}\sum_{k=1}^{N}\sum_{l\neq k}\varrho(s_i, s_j)\varrho(s_k, s_l)$$

where $\varrho(s_i, s_j) = \kappa_h(r - \|s_i - s_j\|)\omega_{ij}$ and $\varrho(s_k, s_l) = \kappa_h(r - \|s_k - s_l\|)\omega_{kl}$. We denote S^*_A, S^*_B, S^*_C, S^*_D, S^*_E, S^*_F, S^*_G which have the same types of summations as defined above.

$$\begin{aligned} E(S^*_A) &= E(\int_W\int_W \kappa_h^2(r - \|u - v\|)\omega^2(u, v)\rho^{(2)}(u, v)dudv) \\ &= \lambda^2\int_W\int_W \kappa_h^2(r - \|u - v\|)\omega^2(u, v)g(\|u - v\|)dudv \\ &= \mu^*_A \end{aligned}$$

Similarly,

$$\begin{aligned} E(S^*_E) &= \lambda^3\int_W\int_W\int_W \kappa_h(r - \|u - v\|)\kappa_h(r - \|u - h\|)\omega(u, v)\omega(u, h)\Xi dudvdh \\ &= \mu^*_E, \qquad \textit{where} \quad \Xi = \prod_{\substack{m,n\in\{u,v,h\}\\ m\neq n}} g(\|m - n\|) \end{aligned}$$

$$\begin{aligned} E(S^*_G) &= \lambda^4\int_W\int_W\int_W\int_W \kappa_h(r - \|u - v\|)\kappa_h(r - \|h - g\|)\omega(u, v)\omega(h, g)\Omega dudvdhdg \\ &= \mu^*_G, \qquad \textit{where} \quad \Omega = \prod_{\substack{m,n\in\{u,v,g,h\}\\ m\neq n}} g(\|m - n\|) \end{aligned}$$

However, S_C^*, S_D^* and S_F^* cannot be simplified, thus we use μ_C^*, μ_D^* and μ_F^* denoting their expected value.

$$\begin{aligned}
E(S_B^*) &= E(\int_W \int_W \kappa_h^2(r - \|u-v\|)\omega(u,v)\omega(v,u)\rho^{(2)}(u,v)dudv) \\
&= \lambda^2 \int_W \int_W \kappa_h^2(r - \|u-v\|)\omega(u,v)\omega(v,u)g(\|u-v\|)dudv \\
&= \mu_B^*
\end{aligned}$$

Therefore, the variance estimate can be written as

$$\begin{aligned}
Var(\hat{g}(r)) &\approx \frac{(E(\vartheta))^2}{(E(M))^2}\Big[\frac{Var(\vartheta)}{(E(\vartheta))^2} - 2\frac{Cov(\vartheta,M)}{E(\vartheta)E(M)} + \frac{Var(M)}{(E(M))^2}\Big] \\
&= \frac{\lambda^4|W|^2\varpi(r)^2}{\nu^2(E(\int_W \rho(s)ds)^2)^2}\Big[\frac{E(\vartheta^2)|W|^2}{\lambda^4\varpi^2} - \frac{2\eta}{\varpi E(\int_W \rho(s)ds)^2} \\
&\quad + \frac{E(\int_W \rho(s)ds)^4}{(E(\int_W \rho(s)ds)^2)^2} + \frac{4E(\int_W \rho(s)ds)^3}{(E(\int_W \rho(s)ds)^2)^2} + \frac{2}{E(\int_W \rho(s)ds)^2}\Big] \\
&= \frac{\lambda^4|W|^4(\int_0^L \kappa_h(r-t)g(t)dt)^2}{(E(\int_W \rho(s)ds)^2)^2}\Big[\frac{E(\vartheta^2)}{|W|^2\lambda^4(\int_0^L \kappa_h(r-t)g(t)dt)^2} \\
&\quad - \frac{\eta}{|W|\int_0^L \kappa_h(r-t)g(t)dtE(\int_W \rho(s)ds)^2} + \frac{E(\int_W \rho(s)ds)^4}{(E(\int_W \rho(s)ds)^2)^2} \\
&\quad + \frac{4E(\int_W \rho(s)ds)^3}{(E(\int_W \rho(s)ds)^2)^2} + \frac{2}{E(\int_W \rho(s)ds)^2} - 2\Big]
\end{aligned}$$

where $E(\vartheta^2)$ can be obtained by the summation of $\mu_i^*, i \in \{A, B, C, D, E, F, G\}$.From the above derived formula, the variance decreases as μ increases or σ^2 decreases.

Chapter 3

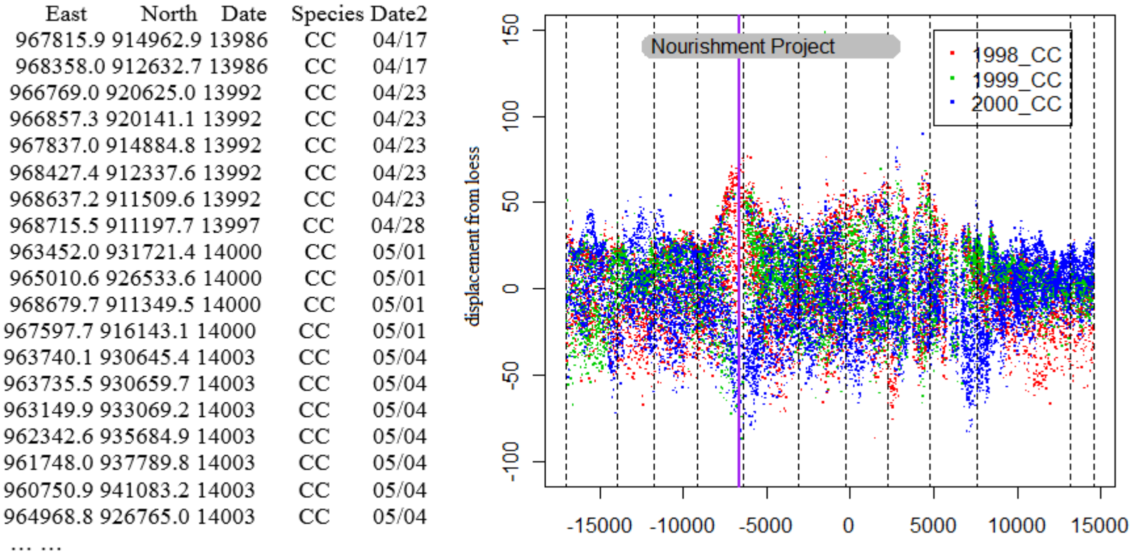
Spatial-temporal point pattern analysis of sea turtle nesting and emergence locations

3.1 Introduction

Research on sea turtles often focuses on emergence (nesting and non-nesting) locations or hatchling success on nesting beaches because those studies provide vital information on the reproductive cycles and success of these endangered threatened species. Such ecological and environmental investigations can provide information regarding varieties of potential factors impacting nesting behavior, and thus more effective protection strategies can be applied to prevent declines of sea turtle population. According to *Florida's sea turtles* (1992), "Concern for the plight of sea turtles is growing and around the world. Conservationists, governmental agencies, public and private organizations, corporations and individuals are working to protect sea turtles on nesting beaches and at sea." There exist a lot of literatures on point pattern analysis of emergences, i.e., Weishampel *et al.* (2003) evaluated the spatial-temporal patterns of sea turtle nesting behaviors along an east central Florida beach by autocorrelation analysis and indicated emergence patterns of loggerhead and green turtles were non-random; Antworth *et al.* (2006) monitored spatial and temporal nesting patterns and hatching success on Canaveral National Seashore, and pointed out that higher density of nests were deposited on the southern end than the northern end and growing populations were detected over the nineteen-year study period. These works of statistical estimation and comparison based on descriptive statistics offer valuable insights into the assessment of management policies on preserving nesting beaches for sea turtles as well as possible influential factors for particular species.

Our motivating data involves emergence (nesting and non-nesting) locations of sea turtles along Juno Beach, Palm Beach County, Florida from the nesting seasons of year 1998-2000. Juno Beach has been recognized as one of the most densely nested beaches in southeast Florida. Based on the resting reports along Florida's east coast from 2001 to 2010, Juno Beach has the highest nesting concentrations of leatherback turtles and the second highest nesting concentrations of loggerhead and green turtles. Obviously, Juno Beach plays a pivotal and representative role in evaluating how point pattern of nesting locations change over space and time. We provide detailed description of our data in Chapter 1.

Waller and Leong (2007) quantified local impacts of a beach nourishment project on emergence and nesting patterns by calculating the relative ratio of pre- and post-nourishment kernel density estimates at each point along Juno Beach, and found significant reduction of loggerhead and green turtle emergences in the northern portion of the nourishment zone and a significant increase just to the south of the nourishment zone. Later, by utilizing Waller and Leong's approach, Welch (2007) provides comprehensive analysis and summary of the effects of pier and nourishment construction projects on sea turtle nesting to help design "turtle-friendly" future beach nourishment profiles. Annual Juno Beach nesting reports since 2000 monitor spatial-temporal sea turtle nesting patterns and reproductive success, and also identify the impacts of construction projects. Most of these focus on frequency approaches (histograms, scatter plots and tables). However, few studies have investigated the spatial and temporal distribution patterns of sea turtles along Juno Beach by using stochastic techniques. To fill this gap, we propose point processes methodology to investigate spatial and temporal heterogeneity of loggerhead turtle emergences and nesting, and further assess potential effects due to local fishing pier construction. Our data includes 8,832 loggerhead emergences comprised of 4,357 nesting and 4,475 non-nesting emergences (false crawling) for the year 1998; 8,672 loggerhead emergences with 4,326 nesting and 4,346 non-nesting emergences (false crawling) for the year 1999; 10,003 loggerhead emergences including 5,139 nesting and 4,864 non-nesting emergences (false crawling) for the year 2000.



East	North	Date	Species	Date2
967815.9	914962.9	13986	CC	04/17
968358.0	912632.7	13986	CC	04/17
966769.0	920625.0	13992	CC	04/23
966857.3	920141.1	13992	CC	04/23
967837.0	914884.8	13992	CC	04/23
968427.4	912337.6	13992	CC	04/23
968637.2	911509.6	13992	CC	04/23
968715.5	911197.7	13997	CC	04/28
963452.0	931721.4	14000	CC	05/01
965010.6	926533.6	14000	CC	05/01
968679.7	911349.5	14000	CC	05/01
967597.7	916143.1	14000	CC	05/01
963740.1	930645.4	14003	CC	05/04
963735.5	930659.7	14003	CC	05/04
963149.9	933069.2	14003	CC	05/04
962342.6	935684.9	14003	CC	05/04
961748.0	937789.8	14003	CC	05/04
960750.9	941083.2	14003	CC	05/04
964968.8	926765.0	14003	CC	05/04
… …				

Figure 3.1: The data structure and Displacement of CC nesting sites to loess beach

Many tools for the statistical analysis of spatial point process can detect clusters and/or clustering. Based on Besag and Newell (1991)'s terminology, a cluster defines local collection of events occurring at a higher rate than elsewhere. Tests to detect clusters typically evaluate a p-value for each potential cluster, creating a challenging multiple test situation. In comparison, the term clustering indicates a summary aspect of an observed pattern tests for clustering pattern evaluate the entire data set with a single p-vlaue. The first and second-order properties of spatial point process can also be applied for both of these via Monte Carlo inference. For instance, comparison of estimated intensities can find out where patterns differ, and thus detect unusual collection of events (clusters); second-order properties such as $K-$ function or pair correlation function (PCF) typically compare spatial-temporal dependency across varied distance lags and over whole pattern, thus assessing clustering. In this article, we apply both approaches into spatial and temporal pattern analysis. We also evaluate the hypothesis of space-time separability to assess space-time interaction. More importantly, we develop corresponding R programming for one-dimensional spatial-temporal point processes analysis, implementing results from Chapters 1 and 2.

Building from our analytic framework, we use a Log-Gaussian Cox Process to capture environmental variation in the intensity function. Here, we apply spatial-temporal log-Gaussian Cox processes and corresponding results for loggerhead nests in observed in 1998. We fit our parametric model using minimum contrast method (detailed below) to further understand the potential clustering/inhabitative point patterns across space and time. The details of spatial-temporal point process techniques including non-parametric estimation and the testing of space-time separability as well as log-Gaussian Cox processes are provided in Section 3.2. In Section 3.3, we summarize results for our spatial-temporal point pattern analysis as well as parametric model fitting. Finally, in Section 3.4, we discuss the impact of our results on protection of loggerhead turtles and outline future studies.

3.2 Methods

3.2.1 Data preprocessing

Our GPS data require some preprocessing before we can apply the methods in Chapter 1 and 2. We first rotate the UTM coordinates by $\theta = argtan(\beta)$ with β as the slope estimate from linear regression in order to minimize transformation of points from the beach to a line. Next, we perform local regression to define a curve that describes the deterministic part of the variation in the data. The span parameter is chosen as 0.1 to smooth local noise without losing general beach features.

The displacement of all emergences of loggerhead turtles onto a horizontal line after rotation and loess defines data around a linear beach for year 1998-2000, shown in Figure 3.1. INBS zones as well as the area of the nourishment project and pier are also provided. Accordingly, we calculate the distance of sequential pairs of locations ordered by rotated east coordinate along the linear beach to form an one-dimensional spatial point process. Table 3.1 shows the emergence count per unit distance by INBS zones for year 1998-2000 to ensure their comparability among zones due to varied zone length (range: 0.27-1.1 miles or 0.43-1.77 km). After comparison, we can see there exists a reduction of nests count in INBS zone 4, especially in year 1999, and an increase in INBS zone 5, indicating potential spatial heterogeneity and environmental impact near the pier. The next section will further evaluate the spatial and temporal variation through estimated first-order and second-order properties of point processes.

Table 3.1: Emergence count per unit distance by INBS zones for years 1998-2000

INBS zone	Nests count 1998	Nests count 1999	Nests count 2000
1	218	213	241
2	244	264	259
3	290	245	295
4	272	195	269
5	308	344	343
6	353	290	385
7	269	285	310
8	277	265	319
9	241	234	281
10	300	334	384
11	258	274	304

3.2.2 Notation

For one-dimensional spatial-temporal point process in $\mathbb{R}^1 \times \mathbb{R}$, we consider the point pattern projected onto the X-axis in Figure 3.1 as a realization of a locally finite random subset X in the study region $A = W \times T$ consisting a sequence of events $\{(s_i, t_i) : i = 1, 2, \ldots, n\}$ with $s \in \mathbb{R}^1$ denoting location, $t \in \mathbb{R}$ representing time within a bounded spatial region W and a time interval $[0, T]$. Let $N(B)$ denote the number of events within study region B, and N denote the total number of events within region A. The locally finite observed spatial component process X_s includes the events with times in the time interval $[0, T]$. Similarly, X_t denotes the locally finite observed temporal component process consisting the events in the area W. first define important terminology for spatial-temporal point point analysis. Immediately after, we introduce spatial-temporal properties through definitions and estimation in details.

- A point process X has *space-time clustering* if a statistically significant excess of events, occurs within a limited space-time subset.
- A point process X has *space-time interaction* if pairs of events which are close to each other in space are also close to each other in time.
- *First-order spatial-temporal separability* is defined by $\rho(s, t) = \rho(s)\rho(t)/N$.

- *Second-order spatial-temporal separability* id defined by $K(u,v) = K_s(u)K_t(v)$.

3.2.3 Spatial-temporal Intensity

The first-order property of a space-time process is the intensity function defining the expected number of events occurring per unit area and time. For a spatial-temporal point process X, the intensity $\rho(s,t)$ is defined as follows

$$\rho(s,t) = lim_{|ds \times dt| \to 0} \frac{E(N(ds \times dt))}{|ds \times dt|}$$

where $ds \times dt$ is a small region around the point (s,t) with ds denoting an infinitesimally small line segment and dt an infinitesimally time interval. Hence, $\rho(s,t)|ds \times dt|$ defines the probability that one event occurs in $ds \times dt$, which satisfies $\int_W \int_T \rho(s,t)d(s,t) = N$ assuming that events cannot occur at exactly the same point in space-time. If we have the intensity for a spatial point process X_s, $\rho(s)$, and the intensity for a temporal point process X_t, $\rho(t)$, and we assume a separable spatial-temporal intensity function, $\rho(s,t)$ may be estimated as a product form given by $\hat{\rho}(s,t) = \hat{\rho}(s)\hat{\rho}(t)/N$; otherwise, a two-dimensional space-time kernel estimator may be used.

In addition, if the process X is homogeneous, the intensity will be a constant, and can be estimated by $\frac{N}{|W| \times T}$, whilst for heterogeneous processes, non-parametric techniques are widely applied. For instance, for separable processes, the kernel intensity estimator can be written as (Berman and Diggle, 1989):

$$\hat{\rho}(s,t;h_s,h_t) = \frac{1}{Nh_sh_t} \sum_{i=1}^{N} \kappa_s \left(\frac{s - s_i}{h_s} \right) \kappa_t \left(\frac{t - t_i}{h_t} \right)$$

where s_i and t_i are the location and time index of i^{th} event, κ is the kernel function which is a Gaussian kernel in common, and h is a smoothing parameter or bandwidth often based on the direct plug-in approach by Sheather and Jones (1991) which has been described in Chapter 1.

In our application, to find out where nesting patterns differ between months (May-August), we use spatial intensities to detect the clusters under an assumption of heterogeneous Poisson point processes. In particular, for each pairwise comparison of intensities from month i and j ($i \neq j$, $i, j \in \{5, 6, 7, 8\}$), we calculate the natural logarithm of the relative ratios of monthly intensities, i.e., $\hat{r}(s) = log\{\hat{\rho}_i(s)/\hat{\rho}_j(s)\}$, along the the beach surface following the general approach of Kelsall and Diggle (1995). We apply a random labelling Monte Carlo method to get the 95% tolerance envelopes based on pointwise quartiles at each grid point based on 1000 simulations. The locations along the beach where the observed value of $\hat{r}(s)$ stays outside the tolerance regions reveal significance difference between two month groups. This approach is useful in discerning locally significant changes in nesting locations that may be difficult to detect using the traditional, less sensitive method of analysis by zones (Waller and Leong, 2005).

3.2.4 Spatial-temporal $K-$function and PCF

In this section, we extend spatial $K-$function and pair correlation function to spatial-temporal point processes, thus exploring second-order spatial-temporal dependency. Under the assumption of second-order (intensity-reweighted) stationarity, the non-parametric estimate of K-function is given by

$$\hat{K}(u, v) = \frac{1}{|W||T|} \sum_{i=1}^{N} \sum_{j \neq i} \frac{I(|s_i - s_j| \leq u, |t_i - t_j| \leq v)\omega(s_i, s_j)\omega(t_i, t_j)}{\hat{\rho}(s_i, t_i)\hat{\rho}(s_j, t_j)} \tag{3.1}$$

where $I(\|s_i - s_j\| \leq u, |t_i - t_j| \leq v)$ is an indicator function with value 1 when the distance of points s_i and s_j is no larger than u and associated time lag between t_i

and t_j is less than or equal to v simultaneously and 0 otherwise. $\omega(s_i, s_j)$ and $\omega(t_i, t_j)$ denote edge correction factors to adjust bias, where Diggle's edge-correction (1985) is applied. $K(u, v)$ is used to evaluate the expected number of events within a certain distance u and time interval v given an arbitrary event in the study region. When X follows complete spatial-temporal randomness, $K(u, v) = \int_{-v}^{v} \int_{-u}^{u} du'dv' = 4uv$, while $K(u, v) > 4uv$ indicates aggregate spatial-temporal patterns and $K(u, v) < 4uv$ shows regular spatial-temporal patterns.

PCF is another informative second-order statistic related to the joint expectation of events and first-order intensity. Without assuming stationary, the theoretical definition of PCF provided by Baddeley *et al.* (2000) is

$$\begin{aligned} g((s_i, t_i), (s_j, t_j)) &= \frac{\rho^{(2)}((s_i, t_i), (s_j, t_j))}{\rho(s_i, t_i)\rho(s_j, t_j)} \\ &= 1 + \frac{Cov(N(D_i), N(D_j))}{\rho(s_i, t_i)\rho(s_j, t_j)|D_i||D_j|}, \quad D = ds \times dt \end{aligned}$$

where $\rho^{(2)}(\cdot)$ is the second-order intensity. Note that the PCF equals one for completely random spatial-temporal point patterns, i.e., a Poisson process; A value of PCF greater than one indicates aggregate spatial-temporal point patterns; at that space-time lags, the PCF less than one suggests inhibitive spatial-temporal point patterns. There exists a close relationship between the PCF and $K-$function, where $K(u, v) = \int_{-v}^{v} \int_{-u}^{u} g(u', v')du'dv'$ (past and future events included). A non-parametric estimate of the PCF for X can be obtained by

$$\hat{g}(u, v) = \frac{1}{|W||T|} \sum_{i=1}^{N} \sum_{j \neq i} \frac{\kappa_{h_s}(u - \|s_i - s_j\|)\kappa_{h_t}(v - |t_i - t_j|)\omega(s_i, s_j)\omega(t_i, t_j)}{\hat{\rho}(s_i, t_i)\hat{\rho}(s_j, t_j)}$$

where the edge correction factor $\omega(s_i, s_j)$, $\omega(t_i, t_j)$ and $\kappa(\cdot)$ are defined the same as before. In our study, we use both $K-$functions and PCFs to detect clustering. In next section, $K-$functions are utilized to test space-time separability by simulation.

3.2.5 Space-time clustering and interaction

Hypothesis testing of space-time separability is important in investigation of spatial-temporal structure, providing suggestions for subsequent model fitting. There exist two types of separability, first-order separability and second-order separability, which have been briefly introduced above.

As we know, a non-parametric estimate of spatial-temporal intensity function is given by kernel smoothing which could induce space-time dependency into the data. To assess separable space-time intensity functions, Schoenberg (2004) proposed several non-parametric test statistics, and found a Cramer-von Mises type statistic had the greatest power to detect gradual departures from separability. This statistic is defined by

$$S_1 = \int_W \int_T \left(\hat{\rho}(s,t) - \hat{\rho}(s)\hat{\rho}(t)/N\right)^2 dtds$$

Given discrete time and a relatively small grid size in space partition, the above statistic can be approximated by $\sum_{s \in W} \sum_{t \in T} \left(\hat{\rho}(s,t) - \hat{\rho}(s)\hat{\rho}(t)/N\right)^2$. However, in Schoenberg's work, he only considered hypothesis testing on one realization (quite variable sometimes), and did not provide any information on the sensitivity of this statistic. Here, we will evaluate this approach regarding its performance in preserving Type I error and we will testing power for one-dimensional spatial-temporal point processes by simulation for better understanding. Under a null hypothesis of space-time separability, $\hat{\rho}(s,t)$ and $\hat{\rho}(s)\hat{\rho}(t)/N$ should be similar, thus large values of the above statistic indicate a departure from separability hypothesis. The p-value can be achieved by independent simulated realizations of inhomogeneous Poisson processes with a separable intensity function of $\hat{\rho}(s)\hat{\rho}(t)/N$. In our data analysis, if the null hypothesis is rejected, the joint spatial-temporal (two-dimensional) kernel estimate will be utilized in our situation.

For second-order separability, we consider hypothesis tests of space-time clustering and interaction under an assumption of reweighted second-order stationary. Note that the $K-$function is commonly used as measure of spatial-temporal clustering or interaction. The null hypothesis of no spatial-temporal clustering means the data are realizations from inhomogeneous Poisson process with the estimated intensity function which we simulate using a permutation method, and we propose

$$S_2 = \sum_u \sum_v \left(\hat{K}(u,v) - 2u * 2v \right)^2$$

where u and v are distance and temporal lags which are less than half the spatial segment length for u and less than half the temporal period for v. As Diggle (1995) mentioned, absence of spatial-temporal clustering is a special case of absence of spatial-temporal interaction. Therefore, for space-time interaction, we consider two common functional summary statistics

$$\hat{\psi}_1 = \sum_u \sum_v \left(\hat{K}(u,v) - \hat{K}_s(u)\hat{K}_t(v) \right)^2$$

$$\hat{\psi}_2 = \sum_u \sum_v \frac{\hat{K}(u,v)}{\hat{K}_s(u)\hat{K}_t(v)}$$

Under the null hypothesis of no space-time interaction, $K(u,v)$ should be the product of the estimates from separate space and time $K-$functions under the null hypothesis (Diggle et al. 1995; Gatrell et al. 1996), i.e., $K(u,v) = K_s(u) \times K_t(v)$, thus $\hat{\psi}_1$ is expected to be 0 and $\hat{\psi}_2$ is expected to be 1. ψ_1 proposed by Diggle (1995) and ψ_2 suggested by Møller et al (2012) may yield different results and we will be compared them in different settings. Tolerance envelopes will be constructed by random label-

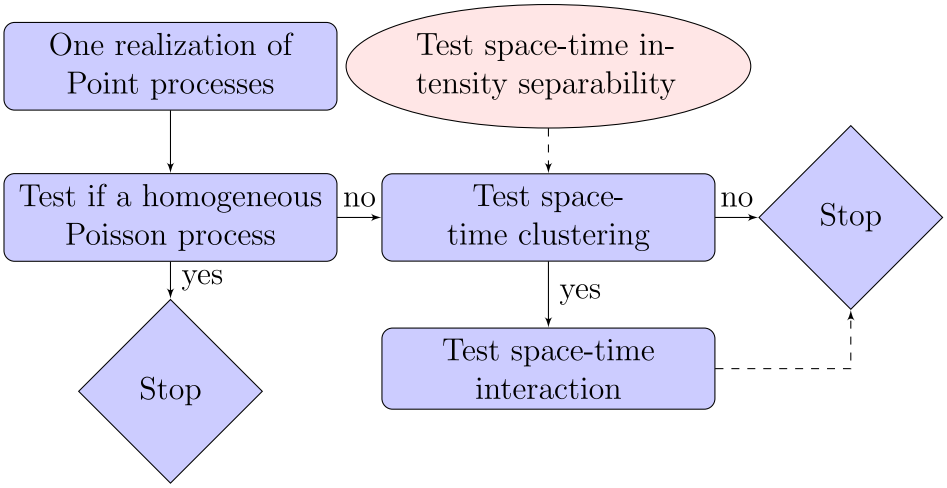

One realization of Point processes
Test space-time intensity separability
Test if a homogeneous Poisson process
no
Test space-time clustering
no
Stop
yes
yes
Stop
Test space-time interaction

Figure 3.2: Testing procedures for spatial-temporal point process

ing of temporal components holding the event locations fixed, where the marginal spatial and temporal structures are preserved. We consider two permutations, one is random labeling for each point and the other one is random labeling for daily points. In Figure 3.2, we summarize the testing procedures for an arbitrary realization of spatial-temporal point processes, which will be followed up in simulation studies of performance and applied in our case study.

Simulation Studies

We investigate four scenarios via simulation: a homogeneous Poisson process (the null hypothesis), an inhomogeneous Poisson process (separable intensity function and non-separable intensity function), a log-Gaussian Cox process with separable covariance function, and a log-Gaussian Cox process with non-separable covariance function. Throughout our simulations for one-dimensional spatial-temporal point process, we generate points in the line interval $[0, 1]$ and the time interval $[1, 30]$, where discrete times are assumed at steps of 1 (day). For each scenario, 500 realizations with 100 permutations are performed.

- ***Homogenous Poisson Process***: The intensity is a constant, i.e., $\lambda(s,t) = \lambda = 10$. Note that space and time are separable.
    - **Simulation Algorithm**: Generate the total number of points N from Poisson distribution with mean $\lambda|W||T|$; Uniformly distribute these points within the spatial region, and randomly select the time with replacement from the set of discrete times.
    - **Results**: Three sample realizations from homogeneous Poisson processes with $\lambda = 10$ are shown in Figure 3.3. 1) For each realization, we first test if the process is homogeneous, and the 95% tolerance envelope is based on 100 simulated realizations from a homogenous Poisson process with $\hat{\lambda} = N/|W||T|$. Type I error is calculated as 0.06, and the testing result for one realization is shown in Figure 3.4. 2) We test spatial-temporal interaction based on random labeling of temporal components. By random labeling. We find a Type I errors of 0.052 ($\hat{\psi}_1$) and 0.06 ($\hat{\psi}_2$); By random labeling daily points, Type I errors are 0.056 ($\hat{\psi}_1$) and 0.05 ($\hat{\psi}_2$). The results for one-realization are shown in Figure 3.5. Overall, for homogeneous Poisson processes, these test statistics perform well. We find random labeling each point and daily points have comparable performance in detecting space-time interaction under the null hypothesis.

- ***Inhomogeneous Poisson process***: We consider two types of intensity functions, one is separable, i.e., $\lambda(s,t) = \exp(3s)\exp(-0.1t)$ and the other one is non-separable, i.e., $\lambda(s,t) = \exp(\frac{10s}{5-0.1t})$. Note that there exists spatial-temporal clustering compared to realizations from a homogenous Poisson process, but there is no residual space-time clustering or interaction based on realizations from the inhomogeneous Poisson processes (ie., the observed clustering is defined entirely by the intensity function; in other words, observations are condi-

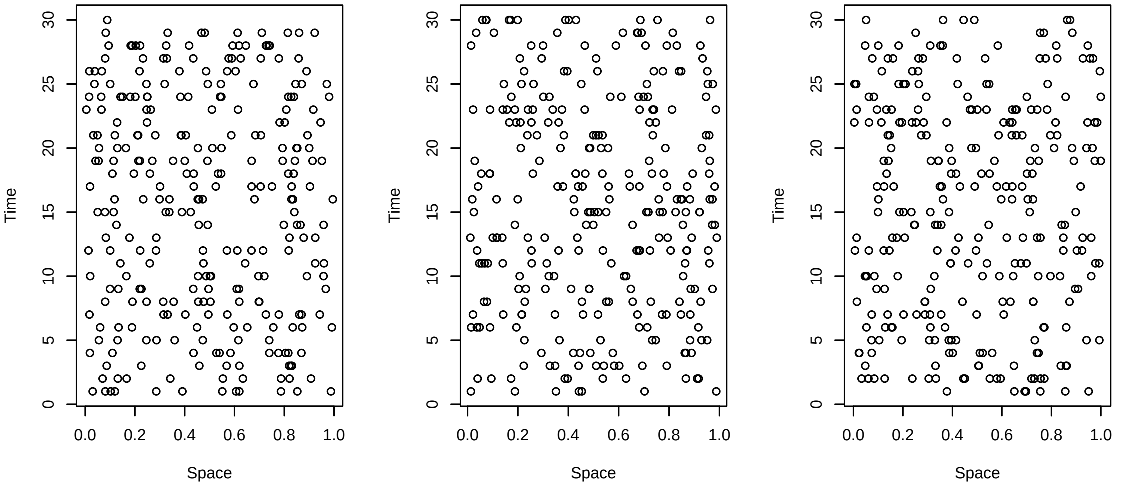


Figure 3.3: Three realizations for a spatial-temporal homogeneous Poisson process

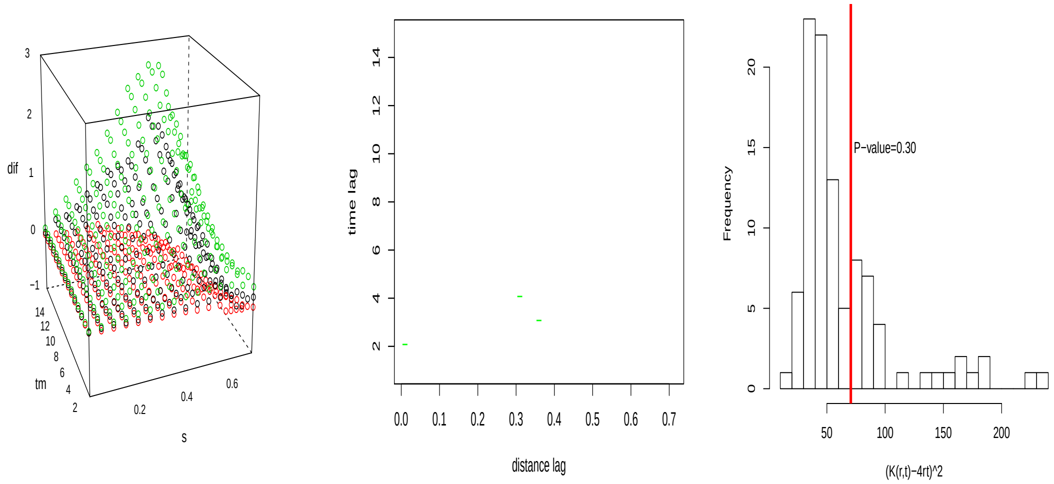


Figure 3.4: Testing for space-time clustering based on a spatial-temporal homogeneous Poisson process; For the left panel, the points in green are the upper bound of the 95% tolerance envelope, the points in red are the lower bound of the 95% tolerance envelope, and the points in black are the estimates based on the data realization; For the middle panel, the positive sign means the estimate stays above the upper 95% bound, and the negative sign means the estimate stays below the lower 95% bound; The right panel shows the observed test statistic (in red) and the histogram of the statistic values across simulations.

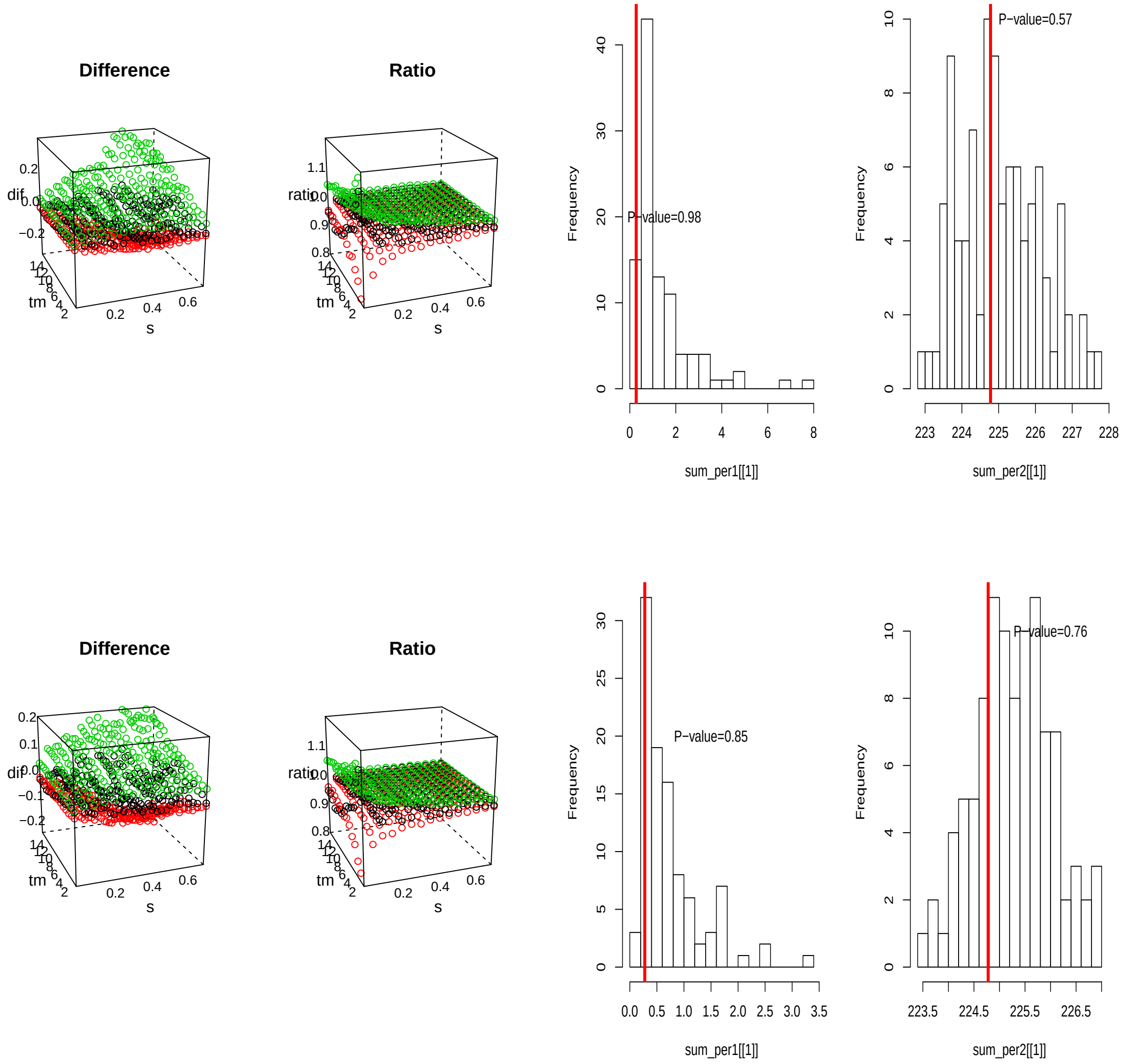


Figure 3.5: Testing for space-time interaction based on a spatial-temporal homogeneous Poisson process; The top panel is based on random labeling on each point, and the bottom panel is based on random labeling on daily points; For the left two plots, the points in green are the upper bound of the 95% tolerance envelope, the points in red are the lower bound of the 95% tolerance envelope, and the points in black are the estimates based on the data realization; For the right two plots, the line (in red) shows the observed test statistic and the histogram of the statistic values are based on simulations.

tionally independent given the intensity function.).

- **Simulation Algorithm**: Find an upper bound λ_{max}, and simulate a homogeneous Poisson process with λ_{max} as above; calculate the intensity at each simulated point using the defined intensity function; generate random samples from a uniform distribution $U(0, 1)$ to thin the simulated Poisson process yielding a set of remaining events consistent with the heterogeneous intensity via an acceptance-rejection algorithm.

- **Results**: Six sample realizations from the same spatial-temporal inhomogeneous Poisson process are shown in Figure 3.6. 1) For each realization, we first test if the intensity function is separable. For the case with separable intensity function, Type I error is 0.023 ; for the case with non-separable intensity function, the power is 0.58. Testing results for one realization in each case are shown in Figure 3.7. 2) For the inhomogeneous Poisson process with non-separable intensity function, we further test space-time clustering based on the realizations from an inhomogeneous Poisson process with $\hat{\lambda}(s, t) = \hat{\lambda}_s \hat{\lambda}_t$. Type 1 error is 0.04, and the results for one-realization are shown in Figure 3.8. 3) In addition, we test spatial-temporal interaction based on random labeling of temporal components. By random labeling of each point, Type I errors are 0.01 ($\hat{\psi}_1$) and 0.02 ($\hat{\psi}_2$); By random labeling of daily points, Type I errors are 0.08 ($\hat{\psi}_1$) and 0.04 ($\hat{\psi}_2$). The results for one realization are shown in Figure 3.9. Overall, for inhomogeneous Poisson processes, the statistic S_1 perform moderately well (underestimating Type I error a little) in testing first-order separability; also, random labeling of daily points performs better than random labeling of each point in terms of Type I error, and in particular, the statistic $\hat{\psi}_2$ has better control of Type I error than $\hat{\psi}_1$.

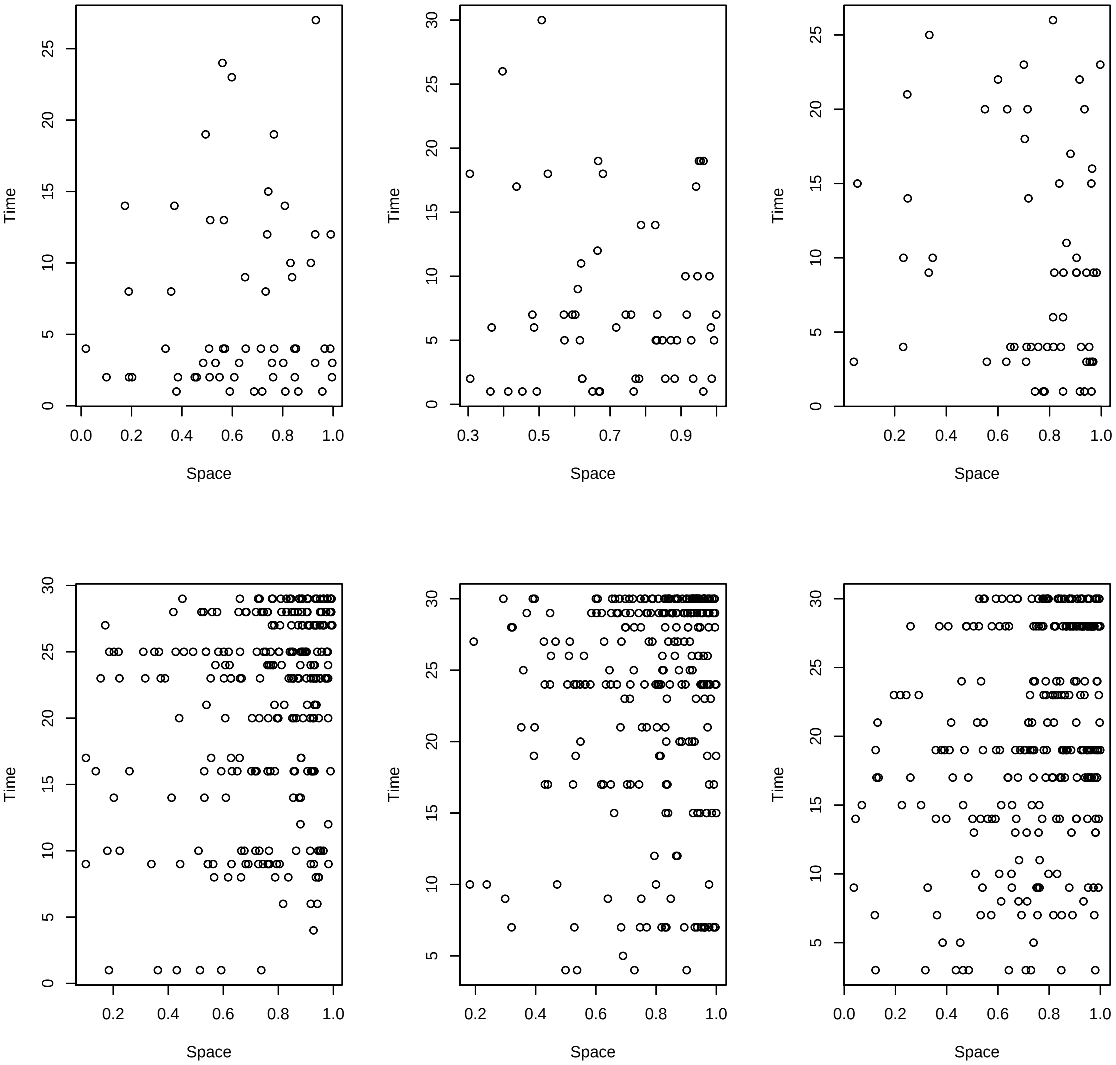


Figure 3.6: Three realizations for a spatial-temporal inhomogeneous Poisson process with separable intensity function (above) and non-separable intensity function (below)

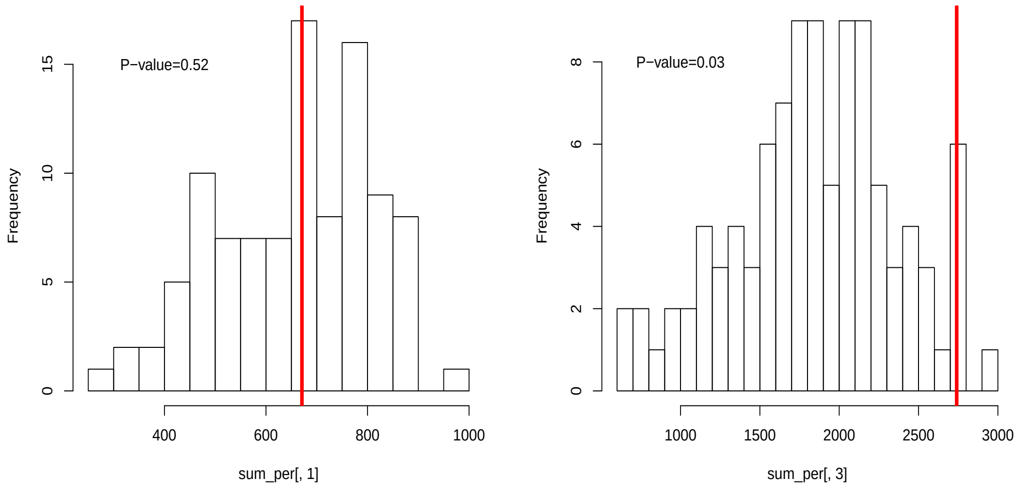


Figure 3.7: Testing separable space-time intensity functions for two realizations from spatial-temporal inhomogeneous Poisson process with a separable intensity function (left) and a non-separable intensity function (right)

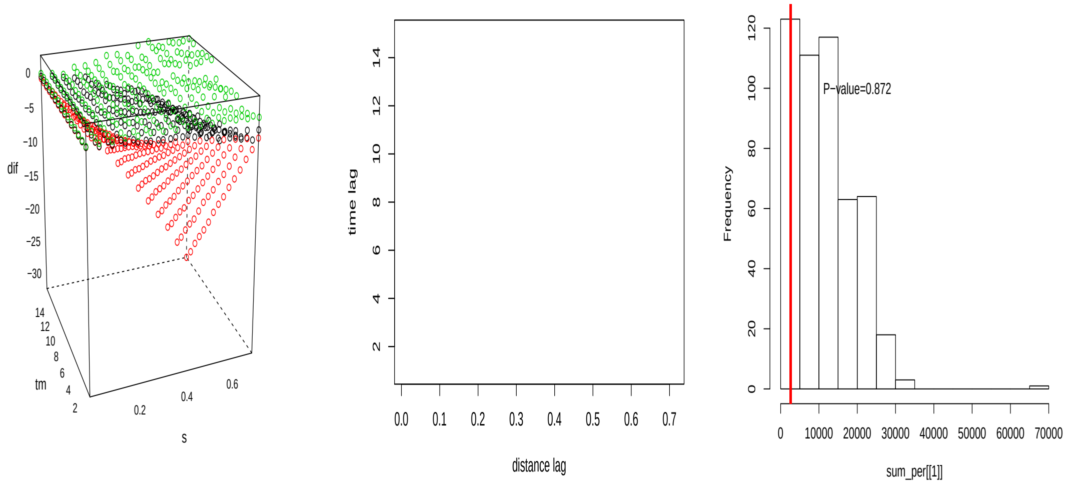


Figure 3.8: Testing space-time clustering based on inhomogeneous Poisson process; For the left panel, the points in green are the upper bound of the 95% tolerance envelope, the points in red are the lower bound of the 95% tolerance envelope, and the points in black are the estimates based on the data realization; For the middle panel, the positive sign means the estimate stays above the upper 95% bound, and the negative sign means the estimate stays below the lower 95% bound; The right panel shows the observed test statistic (in red) and the histogram of the statistic values across simulations.

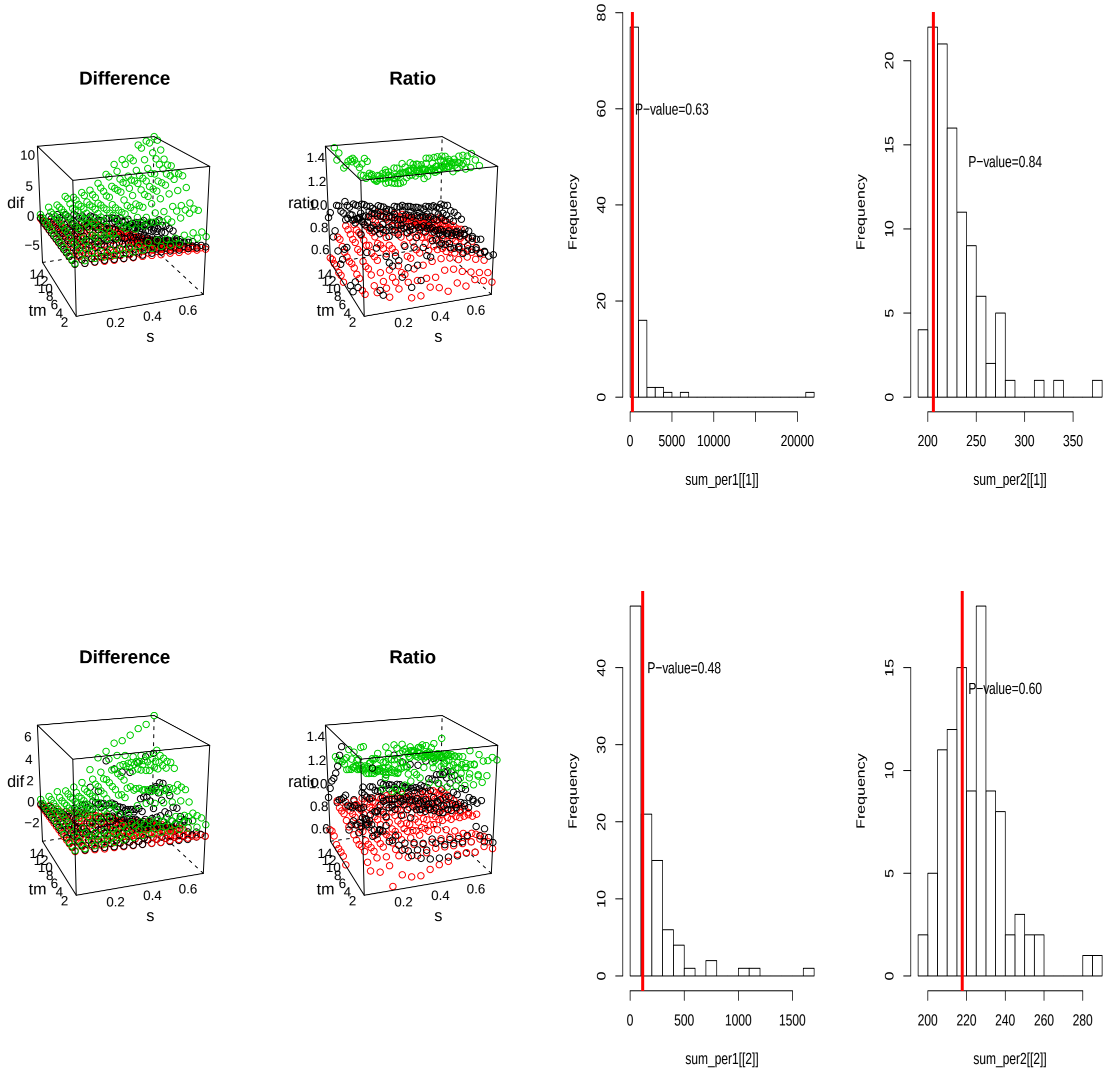


Figure 3.9: Testing space-time interaction in a spatial-temporal inhomogeneous Poisson process; The top panel is based on random labeling on each point, and the bottom panel is based on random labeling on daily points; For the left two plots, the points in green are the upper bound of the 95% tolerance envelope, the points in red are the lower bound of the 95% tolerance envelope, and the points in black are the estimates based on the data realization; For the right two plots, the line (in red) shows the observed test statistic and the histogram of the statistic values are based on simulations.

- ***Log-Gaussian Cox process with separable covariance function***: The intensity function $\Lambda = \exp(Y)$, where Y follows a Gaussian process with mean 2 and covariance function $Cov(u, v) = \exp(-5u) + \exp(-0.01v)$. Note that there is no space-time interaction.

  - **Simulation Algorithm**: Simulate one realization of Gaussian random field $Y(s, t)$ with mean 2 and covariance function $Cov(u, v) = \exp(-5u) + \exp(-0.01v)$ over a defined set of spatial-temporal grids; Get the intensity function $\lambda(s, t) = \exp(Y(s, t))$, and find an upper bound λ_{max}; Simulate a homogeneous Poisson process with λ_{max} as above; Generate random samples from uniform distribution $U(0, 1)$ to thin the simulated Poisson process leaving a realization following the desired intensity function via an acceptance-rejection algorithm.

  - **Results**: Three realizations from spatial-temporal Log-Gaussian Cox processes are shown in Figure 3.10. 1) We first test space-time clustering based on the realizations from inhomogeneous Poisson process with $\hat{\lambda}(s, t) = \hat{\lambda}_s \hat{\lambda}_t$. Type I error is 0.075, and the results for one realization are shown in Figure 3.11. 2) We further test spatial-temporal interaction based on random labeling of temporal components for daily points. Type I errors are 0.068 ($\hat{\psi}_1$) and 0.054 ($\hat{\psi}_2$). The results for one-realization are shown in Figure 3.12. Overall, for Log-Gaussian Cox processes with separable covariance function, the statistic S_2 perform a slightly inflated Type I error in testing space-time clustering; We also find that the statistic $\hat{\psi}_2$ with random labeling of daily points performs better than $\hat{\psi}_1$ in term of preserving Type I error.

- ***Log-Gaussian Cox process with non-separable covariance function***: The intensity function $\Lambda = \exp(Y)$, where Y follows up a Gaussian process with

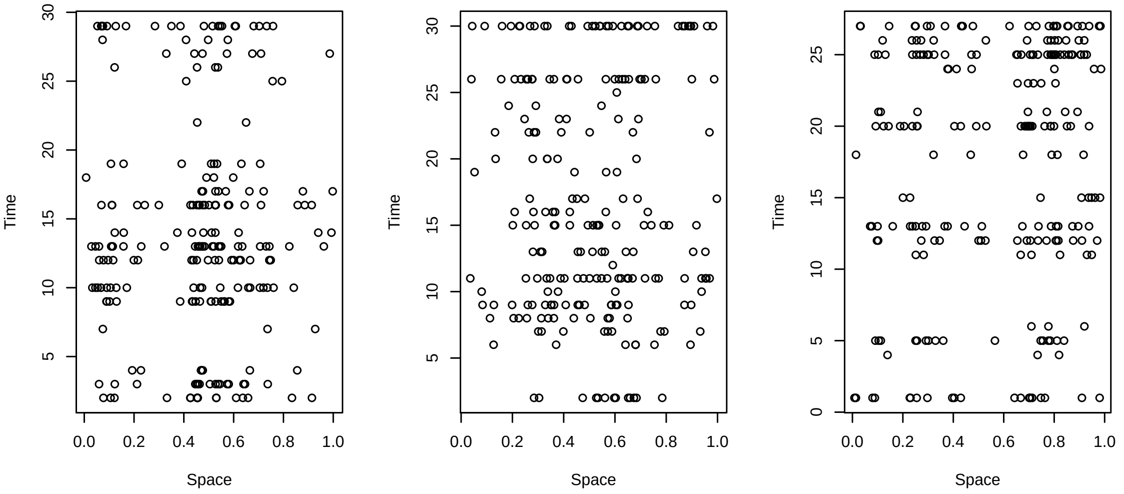


Figure 3.10: Three realizations for a spatial-temporal Log-Gaussian Cox process with separable covariance function

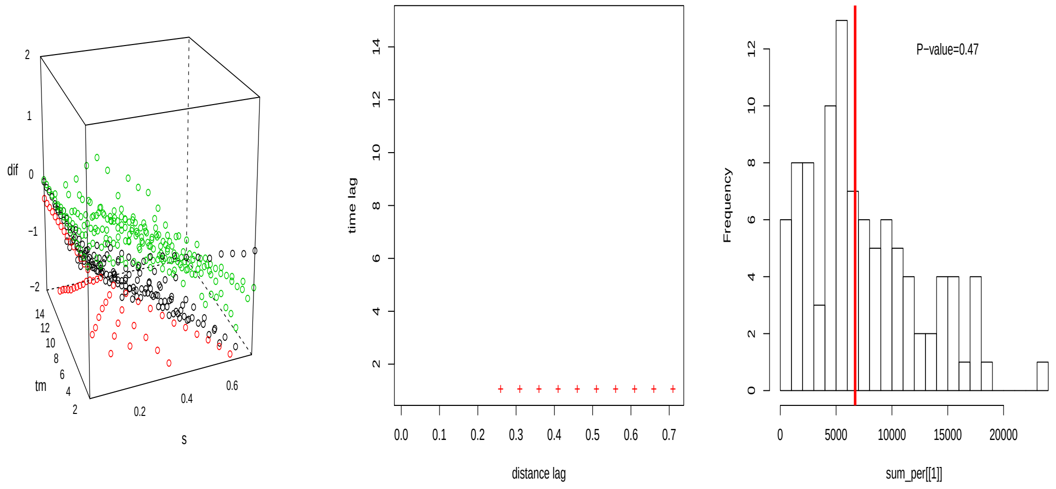


Figure 3.11: Testing space-time clustering in a Log-Gaussian Cox process with a separable covariance function; For the left plot, the points in green are the upper bound of the 95% tolerance envelope, the points in red are the lower bound of the 95% tolerance envelope, and the points in black are the estimates based on the data realization; For the middle panel, the positive sign means the estimate stays above the upper 95% bound, and the negative sign means the estimate stays below the lower 95% bound; The right panel shows the observed test statistic (in red) and the histogram of the statistic values across simulations.

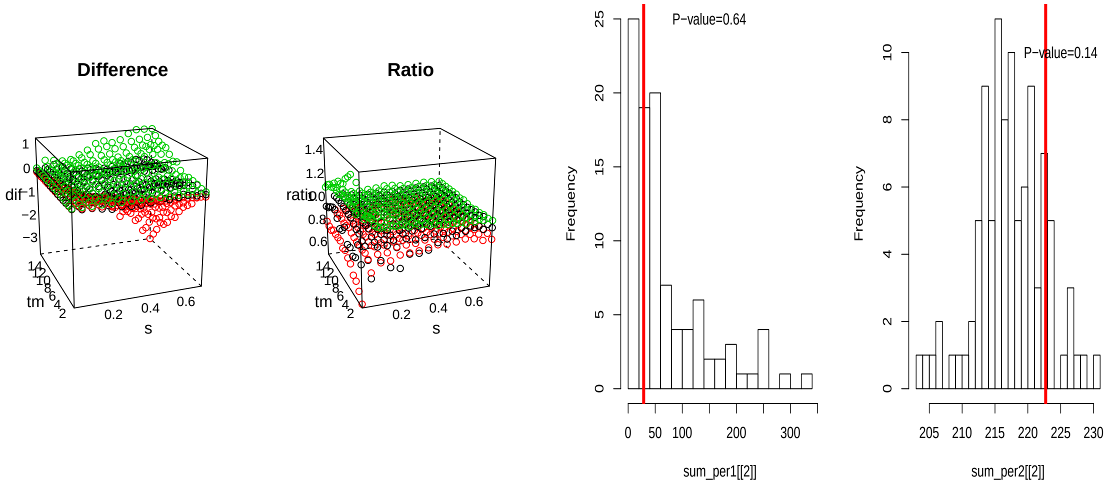


Figure 3.12: Testing space-time interaction for a Log-Gaussian Cox process with a separable covariance function by random labeling of daily points; For the left panel, the points in green are the upper bound of the 95% tolerance envelope, the points in red are the lower bound of the 95% tolerance envelope, and the points in black are the estimates based on the data realization; For the right panel, the line (in red) shows the observed test statistic and the histogram of the statistic values are based on simulations.

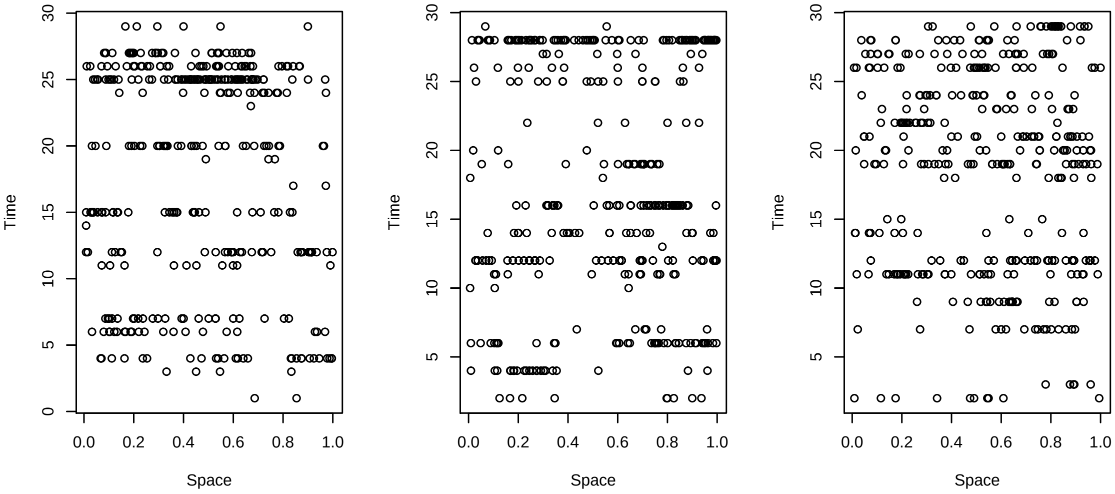


Figure 3.13: Three realizations for a spatial-temporal Log-Gaussian Cox process with non-separable covariance function

mean 2 and covariance function $Cov(u, v) = \exp(-\sqrt{0.1u^2 + 10uv + v^2})$. Note that there exists space-time interaction.

- **Simulation Algorithm**: Simulate one realization of Gaussian random field $Y(s, t)$ with mean 2 and covariance function $Cov(u, v)$ defined by $\exp(-\sqrt{0.1u^2 + 10uv + v^2})$ over a defined set of spatial-temporal grids; The intensity function $\lambda(s, t) = \exp(Y(s, t))$, and we find an upper bound λ_{max}; Next, we simulate a homogeneous Poisson process with λ_{max} as above; We generate random samples from a uniform distribution $U(0, 1)$ and thin the simulated Poisson process leaving a realization following the desired intensity function via an acceptance-rejection algorithm.
- **Results**: Three realizations from spatial-temporal Log-Gaussian Cox processes are shown in Figure 3.13. We are interested in testing space-time interaction based on random labeling of temporal components for daily points. We find power of detecting space-time interaction for $\hat{\psi}_1$ is 0.65 and 0.74 for $\hat{\psi}_2$. The results for one realization are shown in Figure 3.14. Overall, for Log-Gaussian Cox processes with a non-separable covariance function, the statistic $\hat{\psi}_2$ performs better than $\hat{\psi}_1$ in term of higher power in detecting space-time interaction.

3.3 Application to the sea turtle nesting data

3.3.1 Spatial-temporal point pattern analysis

We plot the the kernel density and intensity estimates of loggerhead turtle nestings along the beach for year 1998-2000, which can be seen from Figures 3.15, 3.16, and 3.17 respectively. Note that the bandwidth h for kernel smoothing is 758.57 (ft)

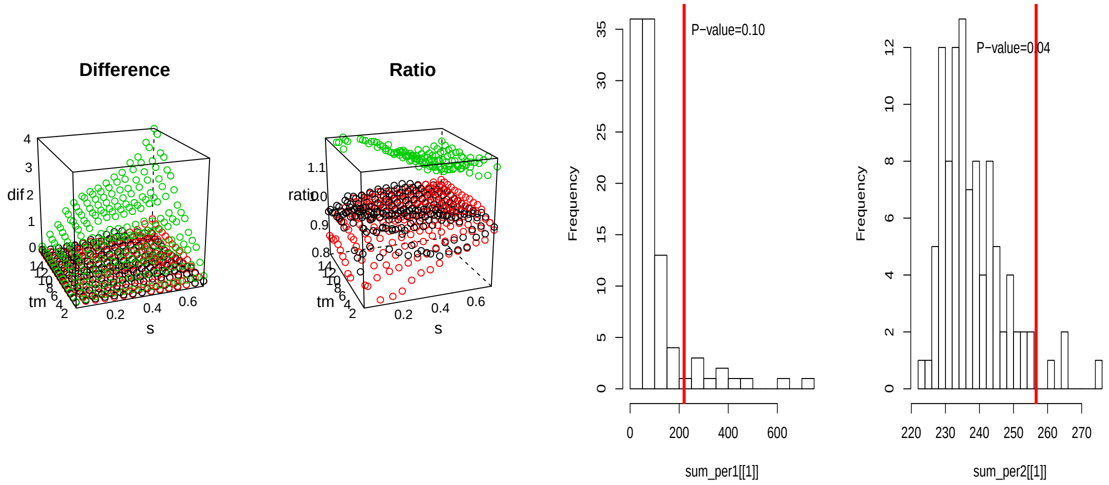


Figure 3.14: Testing space-time interaction in a Log-Gaussian Cox process with non-separable covariance function

based on the whole data using the direct plug-in method mentioned in the Method section, and for pairwise density comparison, log risk ratios and their 95% tolerance envelopes based on 1000 random labeling simulation (green dotted line: the upper envelope bound; red dotted line: the lower envelope bound).

We can see that nesting densities change over space and time within each year. For instance, for the year 1998, the number of loggerhead nests varies across months (May:889; June:1,523; July: 1,621; August:319) with the highest occurrences in June and July and the lowest ones in August. Overall point patterns show significant relative nesting reduction in the area north of the pier accompanied by an increase to the south of the pier except in that month of August. This exception may be due to the small sample size recorded in that month, which can be seen from the kernel intensity plot. In addition, there is significant nesting reduction in the area north of the pier accompanied by a noticeable increase to the south in year 1999 compared with year 1998, and the effect of the pier alleviates in year 2000 which is still in doubt.

Zones 5-9 have the highest nesting intensity in July whilst the others reach their

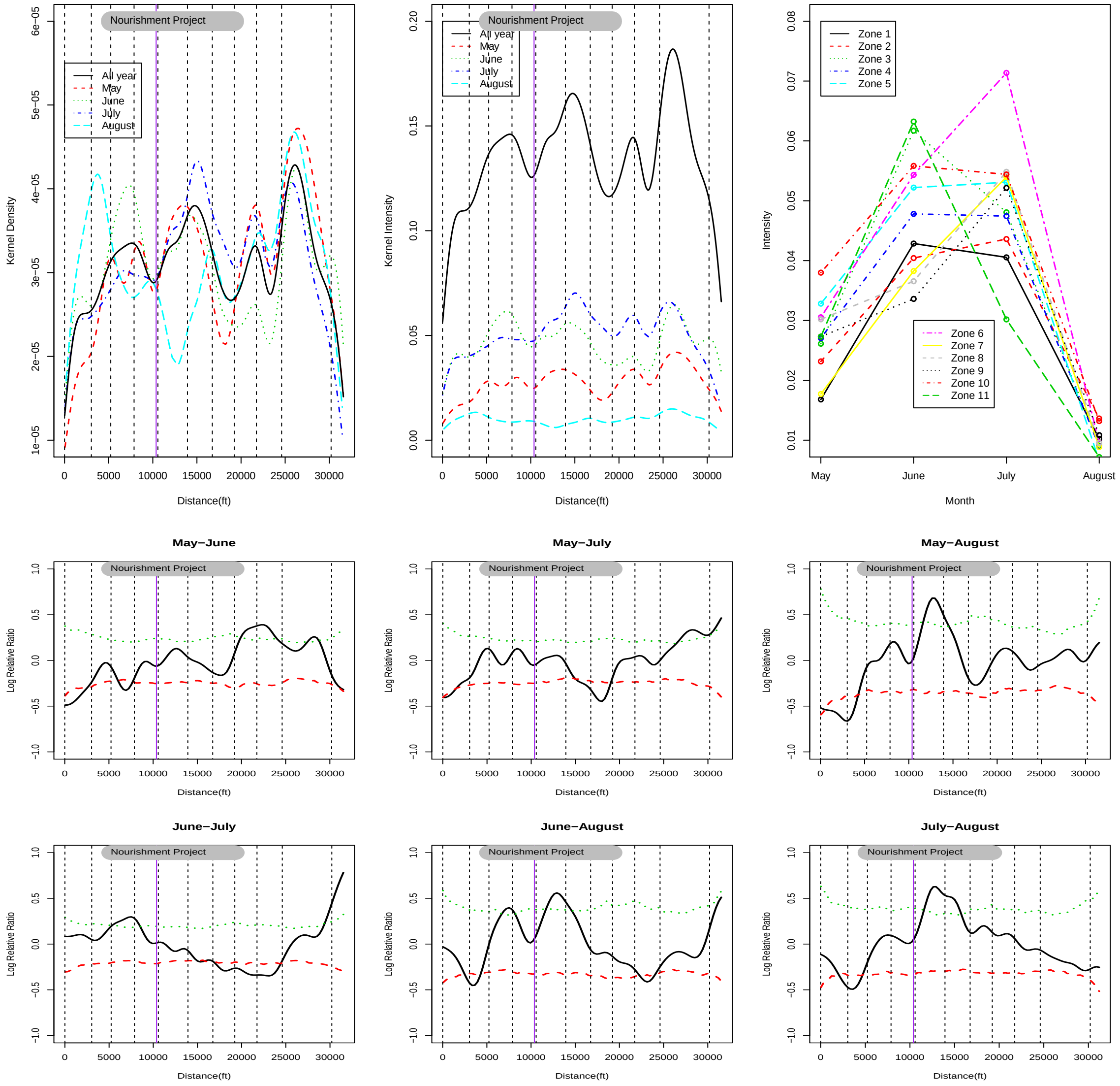


Figure 3.15: The top two: Kernel estimates of loggerhead nesting density and intensity (year 1998) by month and zone; The bottom plots: pairwise comparison of monthly kernel density estimates via Monte-Carlo random labeling

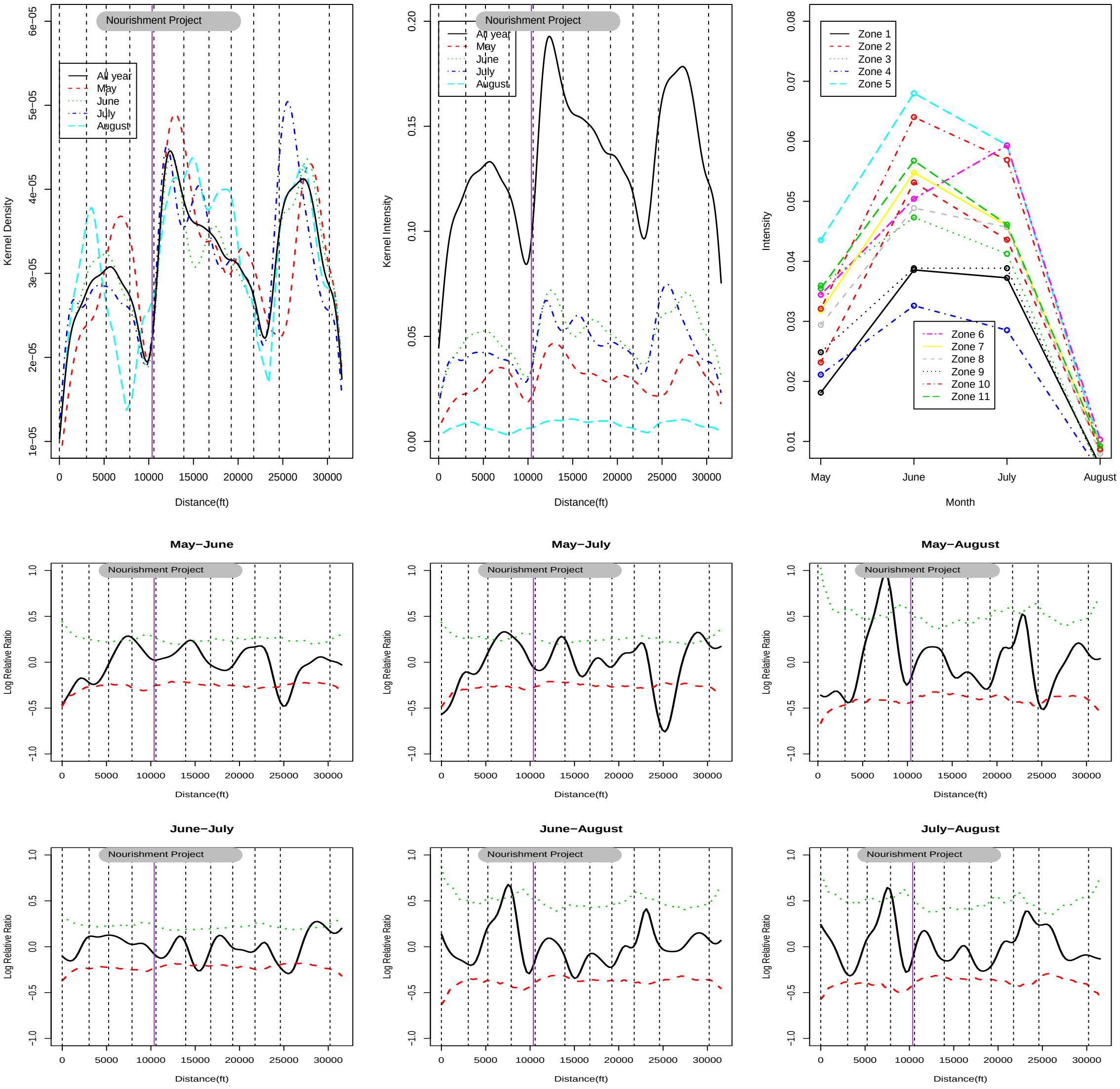


Figure 3.16: The top two: Kernel estimates of loggerhead nesting density and intensity (year 1999) by month and zone; The bottom plots: pairwise comparison of monthly kernel density estimates via Monte-Carlo random labeling

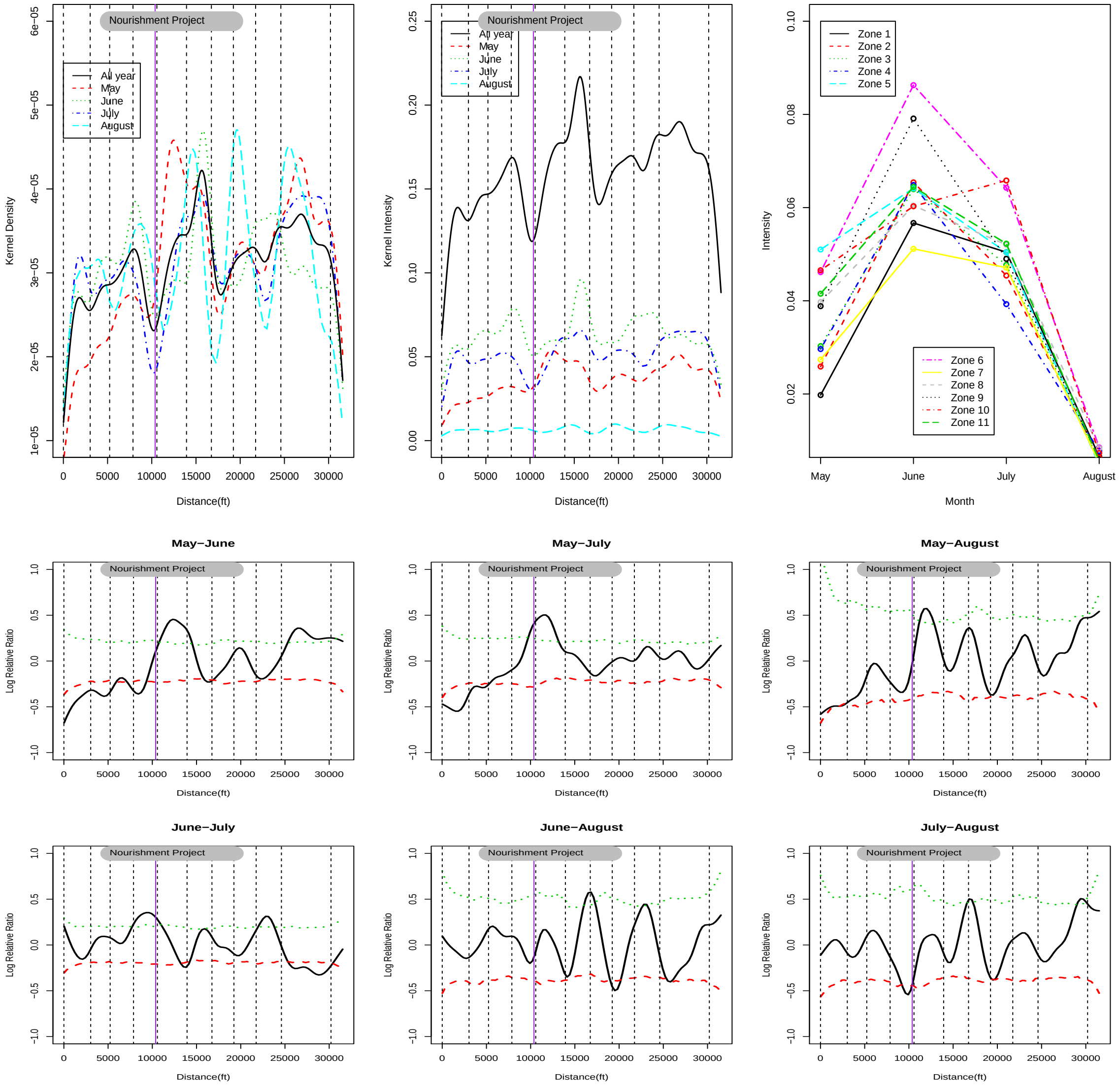


Figure 3.17: The top two: Kernel estimates of loggerhead nesting density and intensity (year 2000) by month and zone; The bottom plots: pairwise comparison of monthly kernel density estimates via Monte-Carlo random labeling

maxima in June, following by May and August in year 1998; All zones except zone 6 have their highest nesting intensity in June, followed by June, May and August sequentially in year 1999; For year 2000, we find similar trends to year 1999 except that zone 10 has a slight higher intensity in July than June. For year 1998, there exists a significant increase of nests to the north of the pier and a significant decrease just south of the pier in June compared with July. In comparisons to August, there is a significant increase on the south side of the pier in May, June and July. In year 1999, we observe significant nesting increases around zone 10 in July compared with May and June, and also significantly lower nesting intensity between zone 3 and 4 in August compared with May, June and July. For year 2000, we observe significant increases in nesting intensity on the south of the pier in May compared with June, July and August.

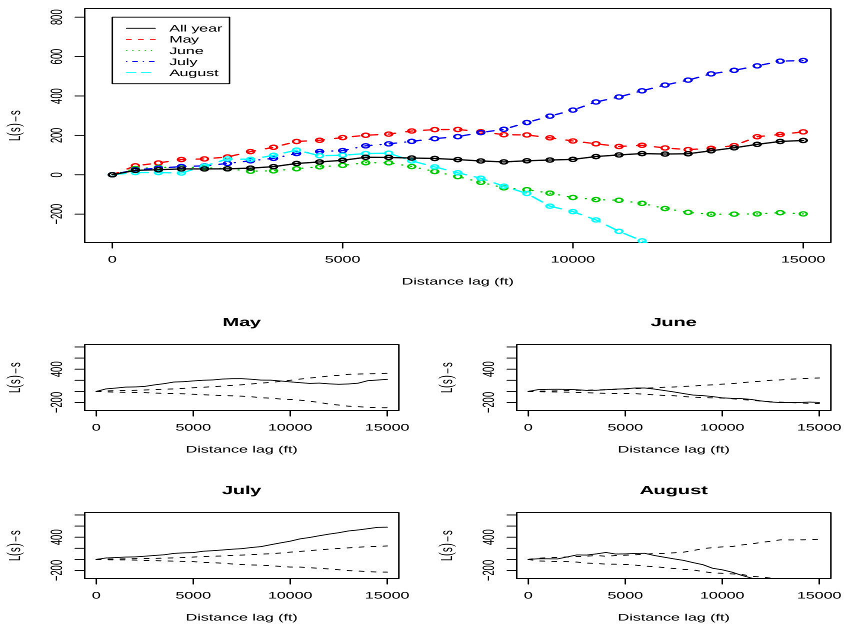


Figure 3.18: The non-parametric estimates of $K-$function for loggerhead nesting locations (year 1998)

In our second-order non-parametric analysis, we estimate the spatial $L-$function

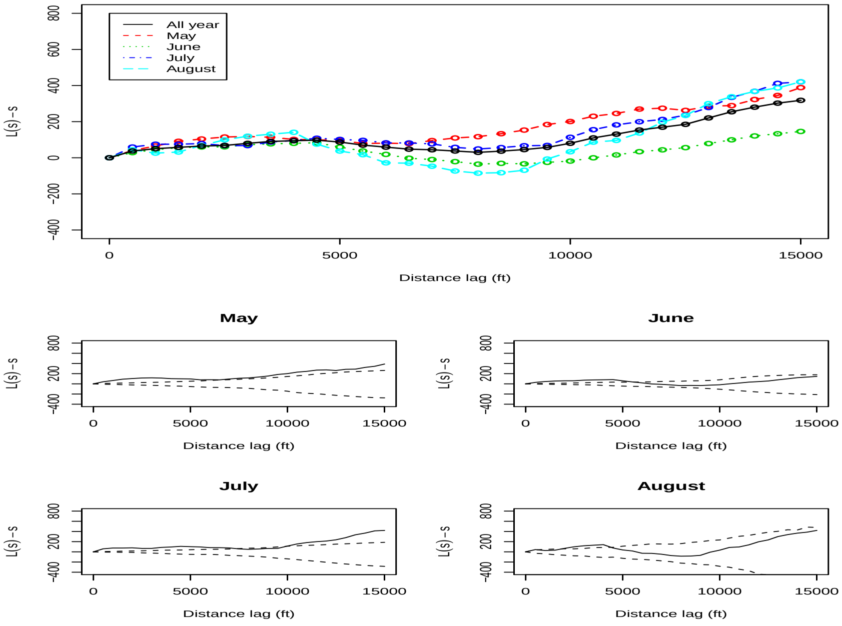


Figure 3.19: The non-parametric estimates of $K-$function for loggerhead nesting locations (year 1999)

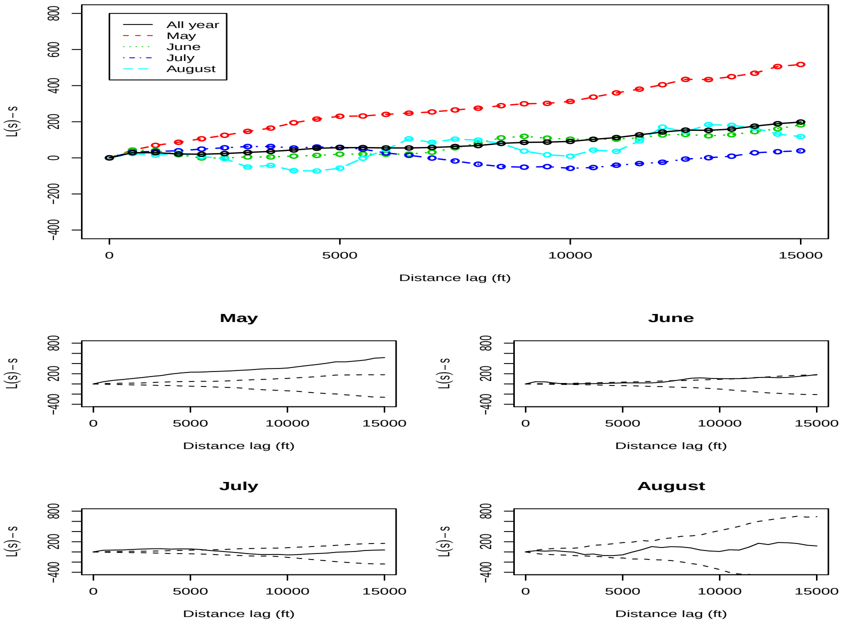


Figure 3.20: The non-parametric estimates of $K-$function for loggerhead nesting locations (year 2000)

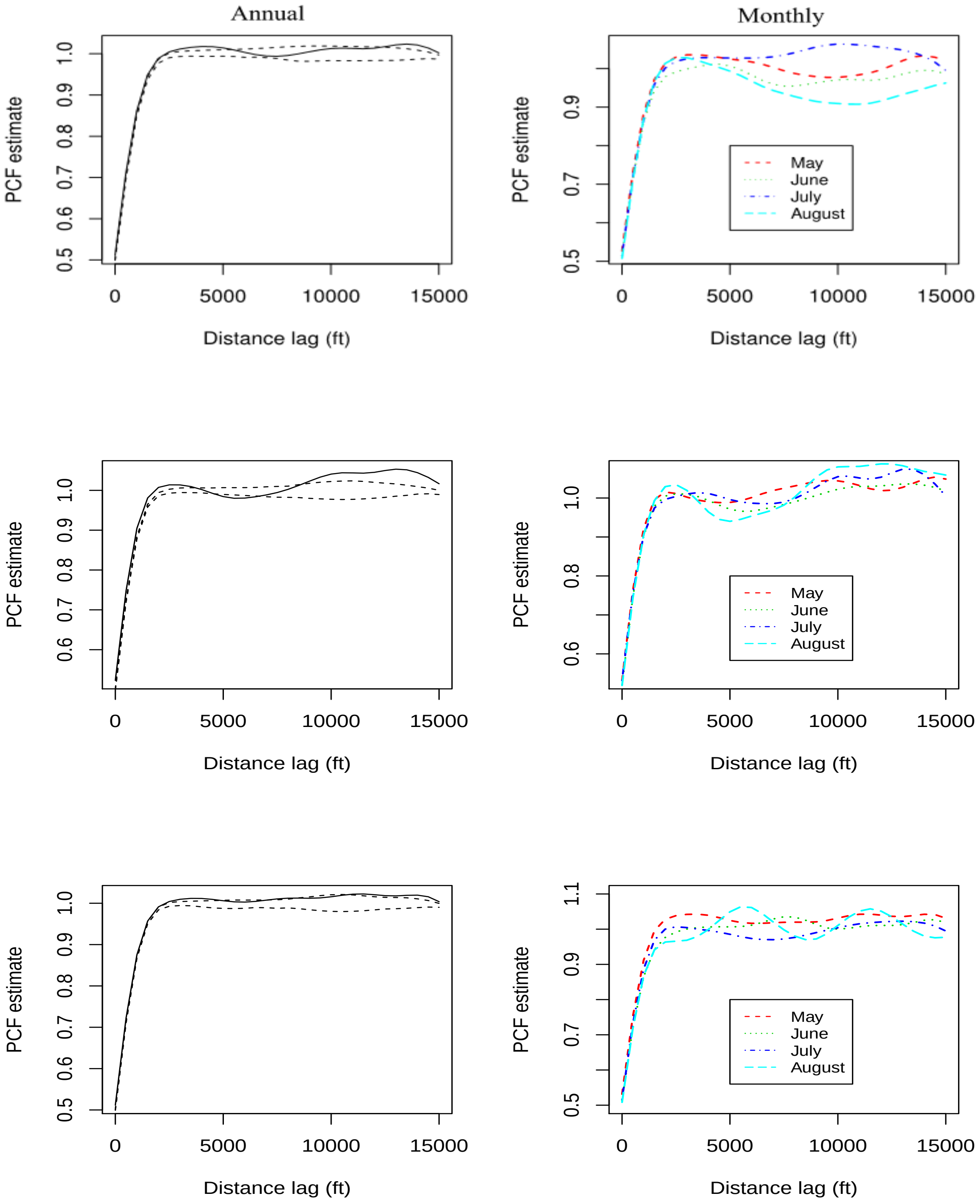


Figure 3.21: The non-parametric estimates of pair correlation function for loggerhead nesting locations (From top to bottom: year 1998-2000)

(solid line) for May-August with a 95% tolerance envelope (dotted lines) calculated from 99 simulations of a Poisson process for years 1998-2000, plotted in Figures 3.18, 3.19, and 3.20 respectively. The cumulative spatial point patterns change over time: for 1998, suggesting significant nesting aggregation for $0 < r < 10,000(ft)$ in May, $0 < r < 15,000(ft)$ in July and $0 < r < 7,000(ft)$ in June and August. Also, we observe a significant inhibitive pattern for $r > 11,000$ in August with values staying below the lower tolerance limits; in 1999, we observe more clustering in nesting patterns for larger distance lags r up to $15,000(ft)$ in May. In June and August 2000, only smaller distance lags $0 < r < 5,000(ft)$ exhibits significant clustering patterns. For 2000, no strong evidence of clustering are detected for $0 < r < 15,000(ft)$ in June and August, but there exists significant clustering pattern for May.

We can also investigate the clustering/regular patterns at particular distance lags, as can be seen from the plots of the estimated PCF for years 1998-2000 in Figure 3.21. We observe the patterns (at a particular r) change over time; for 1998, significantly large PCF estimates appear at a smaller distance lag, i.e., $r < 5,000(ft)$ and a larger one, i.e., $12,000(ft) < r < 15,000(ft)$; for year 1999, significance due to exceedance over the upper envelope at a relatively smaller distance lag, i.e., $r < 3,000(ft)$) and at distances as large as $10,000(ft) < r < 15,000(ft)$; for year 2000, trends are similar to year 1999. These second-order analyses illustrate and quantity how and at what scales spatial clustering in loggerhead nesting pattern change over the course of nesting seasons.

In all, we have quantified spatial-temporal patterns of loggerhead nesting from years 1998-2000 by month and zone. The results showed that loggerhead nesting patterns are not completely random, favoring the southern side of beach area and the seasons of June and July; loggerhead nesting exhibits clustering at distance lags less than $5,000(ft)$ and at relatively large distance lags between $12,000(ft)$ and $15,000(ft)$, and that this clustering pattern changes over time; The fishing pier significantly decreases loggerhead nesting during the first season following the construction; We also find a significant decrease in loggerhead nesting near Zone 9 in tear 1999 which merits further investigation. In addition, the results of spatial-temporal point pattern analysis for loggerhead emergence locations for years 1998-2000 are provided in Figures 3.26-3.32, and the pattern seems quite similar, but there still exist slight differences compared with nesting patterns.

3.3.2 Log-Gaussian Cox Processes

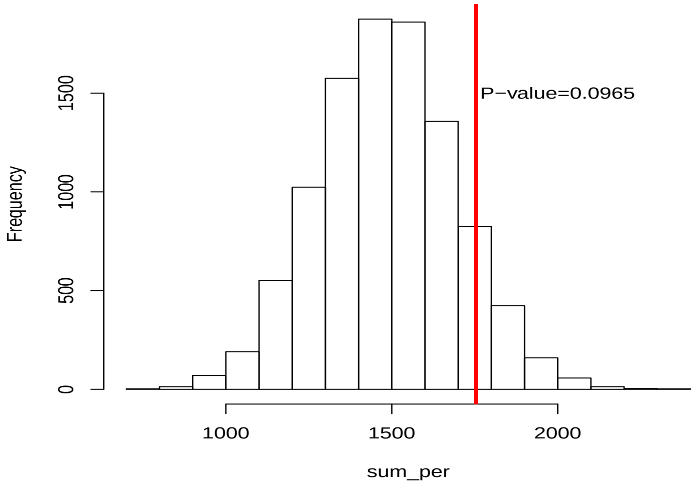


Figure 3.22: Testing of separable intensity function for year 1998

In this section, we take only year 1998 loggerhead nesting data for an example to fit a parametric model of space-time point processes. To begin, we test space-time separability using 10,000 simulated data from an inhomogeneous Poisson process with separable space-time intensity estimate $\hat{\lambda}_s\hat{\lambda}_t$. From Figure 3.22, we see the test statistic is 1753.813 as well as the 95% confidence interval under the null hypothesis is (1090.557, 1894.797). Based on the one-sided p-value (the probability that the statis-

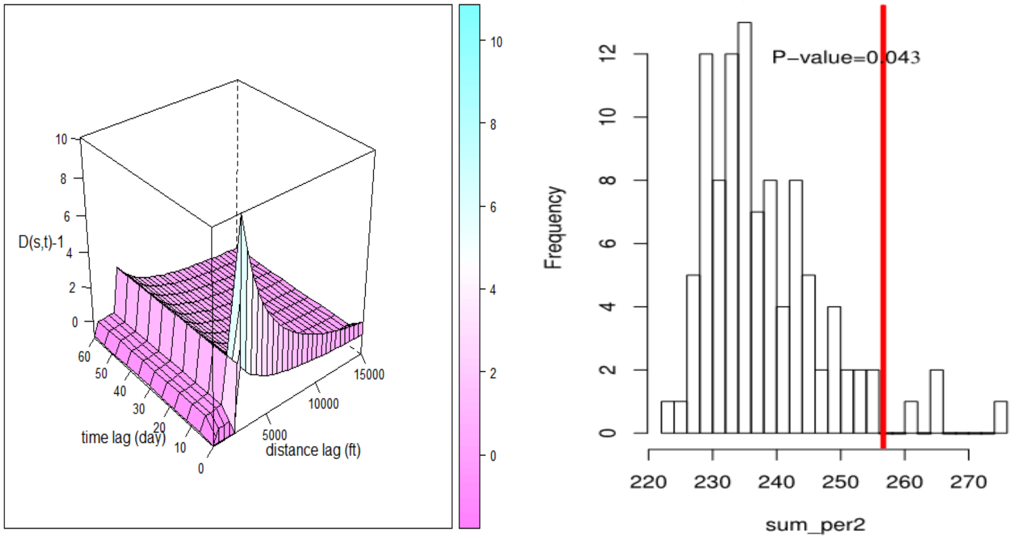


Figure 3.23: Testing of space-time clustering and interaction for year 1998

tic S_1 is greater than 1753.813) is $0.096 > 0.05$, thus the (first-order) spatial-temporal intensity function can be treated as separable. Also, we test space-time interaction based on 100 simulated realizations by random labeling of temporal components for daily points, the p-value is 0.043 which is less than 0.05 shown in Figure 3.23. Therefore, we can indicate there exists significant space-time interaction, and we will fit a parametric model with a non-separable covariance function.

Here, we consider log-Gaussian Cox process for parametric mode fitting, where the non-negative intensity follows a stochastic process defined as $\lambda(s,t) = \exp[Y(s,t)]$ where $Y(s,t)$ is a Gaussian process. Given assumptions of stationary and isotropy, we have $\rho = E(\lambda(s,t)) = \exp(\mu + \frac{1}{2}\sigma^2)$ and $g(u,v) = \exp(C(u,v))$. Note that $C(u,v)$ is not space-time separable, which could be chosen from Gneiting's Family of non-separable covariance functions shown below.

$$C(u,v) = \frac{\sigma^2}{\psi(|v|^2)^{d/2}}\varphi(\frac{\|u\|^2}{\psi(|v|^2)})$$

where $\varphi(u)$ is a completely monotone function, $\psi(v)$ is a positive function with a completely monotone derivative, and d is the spatial dimension. Gneiting (2001) provided various choices of functions for $\phi(u)$ and $\psi(v)$ (Goeiting, 2002).

To estimate the parameters and determine the functions $\varphi(\cdot)$ and $\psi(\cdot)$, we first conduct separable estimation of spatial and temporal covariance functions via minimum contrast method, where the parametric forms of $\varphi(\cdot)$ and $\psi(\cdot)$ can be recommended.

Based on the moment-based methods of Brix and Diggle (2001) and Diggle et.al (2004), we get the time-averaged kernel-based estimator of pair correlation function by

$$\hat{g}(u) = \frac{1}{2|T||W|} \sum_{t=1}^{T} \sum_{i=1}^{n_t} \sum_{j \neq i} \frac{\kappa_h(u - \|s_i - s_j\|)\omega_{ij}}{\hat{\lambda}_0(s_i)\hat{\lambda}_0(s_j)\hat{\rho}^2(t)}$$

Thus, for the marginal spatial correlation structure, we try $\varphi(u) = \exp(-cu^{\gamma})$, and the estimates are $\hat{c} = 1/27500$ and $\hat{\gamma} = 1.2$, and the results are shown in Figure 3.24. Based on our previous simulation studies in Chapter 2, it is reasonable for us to consider an exponential form here due to invalid estimates of pair correlation function for small distance lags. Also, $\hat{\sigma}^2 = 3.7$ and $\hat{\mu} = -3.8$. In addition, for temporal covariance function, the non-parametric estimate is

$$\widehat{Cov}(N_{t'}, N_{t'-v}) = N_{t'}N_{t'-v} - \hat{\lambda}_t(t')\hat{\lambda}_t(t'-v).$$

Thus, using the data information from 14020 to 14040, we select $\psi(v) = (1+v^{\alpha})^{a}$, and the estimates are $\hat{\alpha} = 0.9$ and $\hat{a} = -0.98$, and the results are shown in Figure 3.25. Next, considering space-time non-separability, we assume the following covariance function

$$C(u, v) = \frac{\hat{\sigma}^2}{(1 + |v|^{2\hat{\alpha}})^{\beta/2}} \exp\left(\frac{\hat{c}\|u\|^{2\hat{\gamma}}}{(1 + |v|^{2\hat{\alpha}})^{\beta\hat{\gamma}}}\right)$$

Note that β quantifies the degree of space-time dependency, and $\beta \in [0, 1]$. A weighted least square approach using the following equation can be applied to estimate β, and $\hat{\beta} = 0.56$.

$$W = \sum_{i,j} \sum_{v=1}^{T} \left(\frac{\hat{C}(u_{ij}; v) - C(u_{ij}; v|\beta)}{1 - C(u_{ij}; v|\beta)} \right)$$

where $\hat{C}(u_{ij}; v)$ is the empirical correlation between the points i and j at temporal lag t. Note that larger values of $\hat{\beta}$ means stronger correlation between space and time, and here, it indicates moderate spatial-temporal correlation. One potential future work is hypothesis testing for checking its significance (away from 0), which needs further investigation by considering the boundary issues. Furthermore, the goodness of fit can be assessed by the permutation method based on the estimated parametric model, but the complexity of the covariance function raise challenge for simulating Gaussian random fields, which will be the direction of another future work. This section shows that how to conduct parametric modeling fitting with non-separable spatial-temporal covariance function, and provide more mechanistic information about the point process.

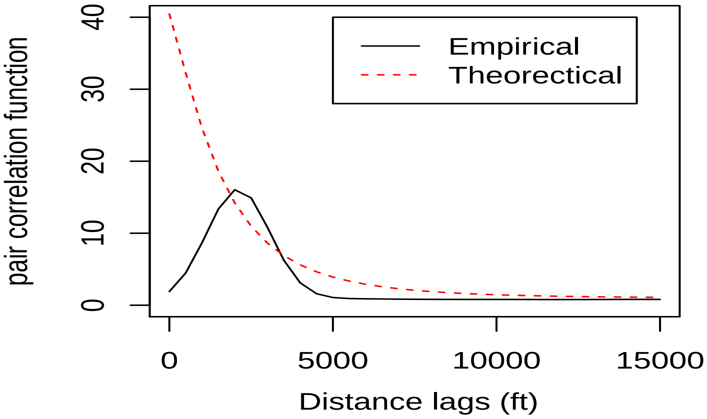


Figure 3.24: Estimation of spatial pair correlation function for loggerhead nesting locations, 1998

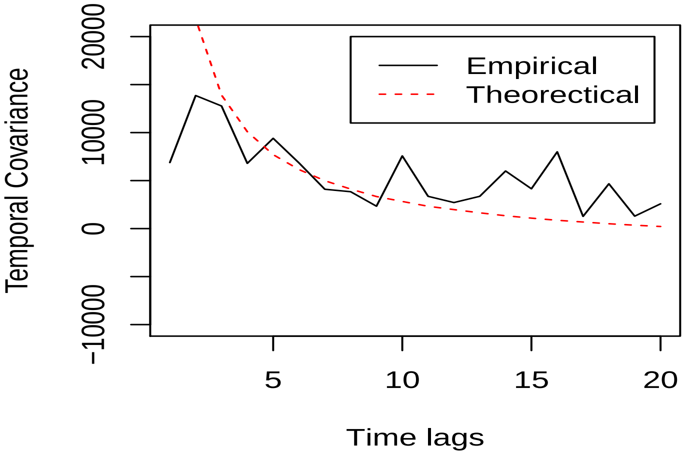


Figure 3.25: Estimation of temporal covariance function for loggerhead nesting locations, 1998

3.4 Conclusion and Discussion

In this chapter, we have conducted spatial-temporal point pattern analysis for loggerhead nesting locations using point process techniques, and quantified patterns of clustering within the date, and determined changes over space and time. Also, extensive simulation studies have evaluated the performance of testing statistics in hypothesis tests on space-time separability under different scenarios. Thus, we have a better understanding of spatial-temporal separability regarding the first-order and second-order properties of log-Gaussian Cox Processes. However, in practice the underlying true processes are not observed can be difficult to assess. We compared separable and non-separable spatial-temporal covariance models, and evaluated statistics for hypothesis testing. Finally, we illustrate how to fit a non-separable covariance model via Log-Gaussian Cox Processes in practice, illustrating how this technique could be applied to real data applications.

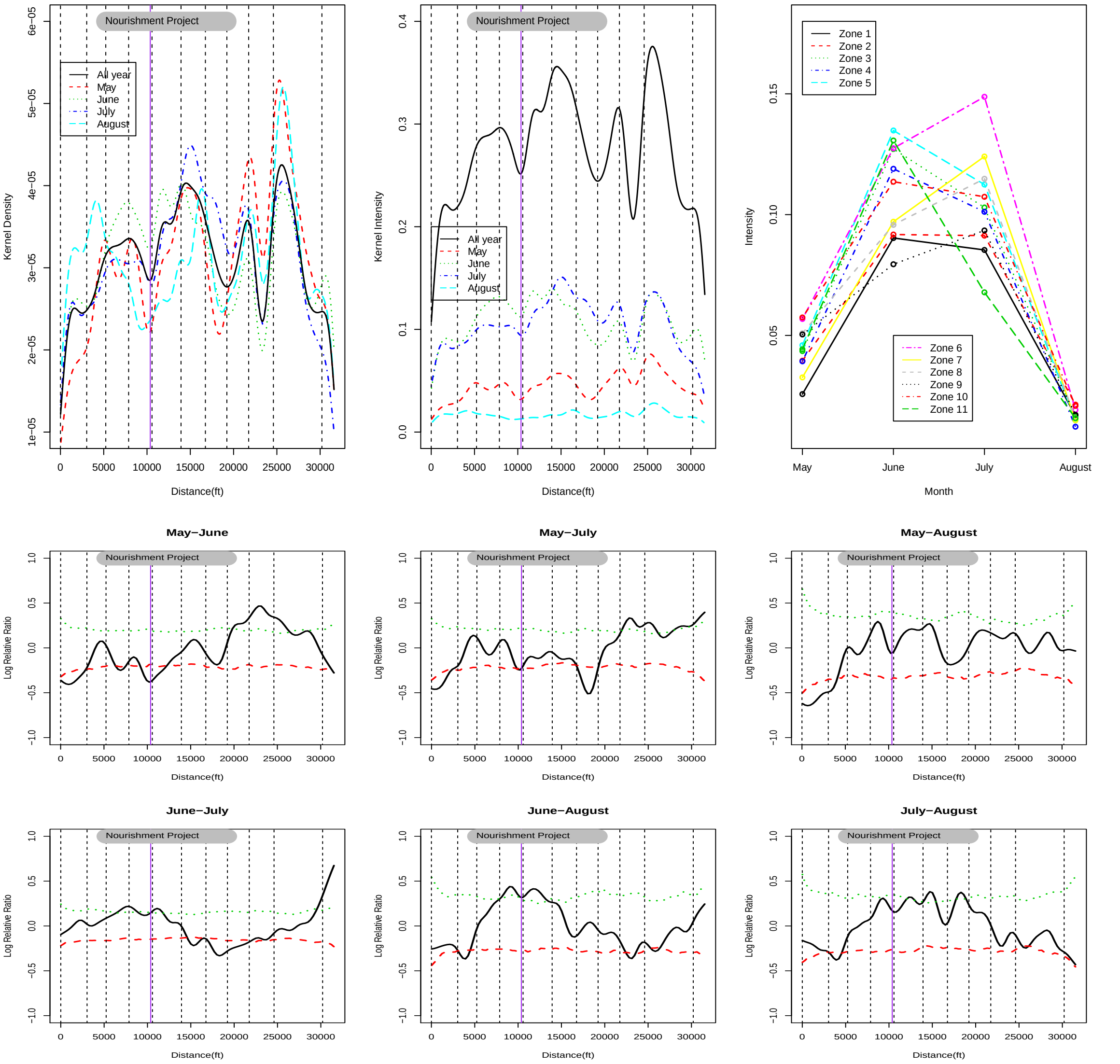


Figure 3.26: The above: Kernel estimates of loggerhead emergence locations (year 1998) by month and zone; The bottom: pairwise comparison of kernel density estimates via Monte-Carlo inference

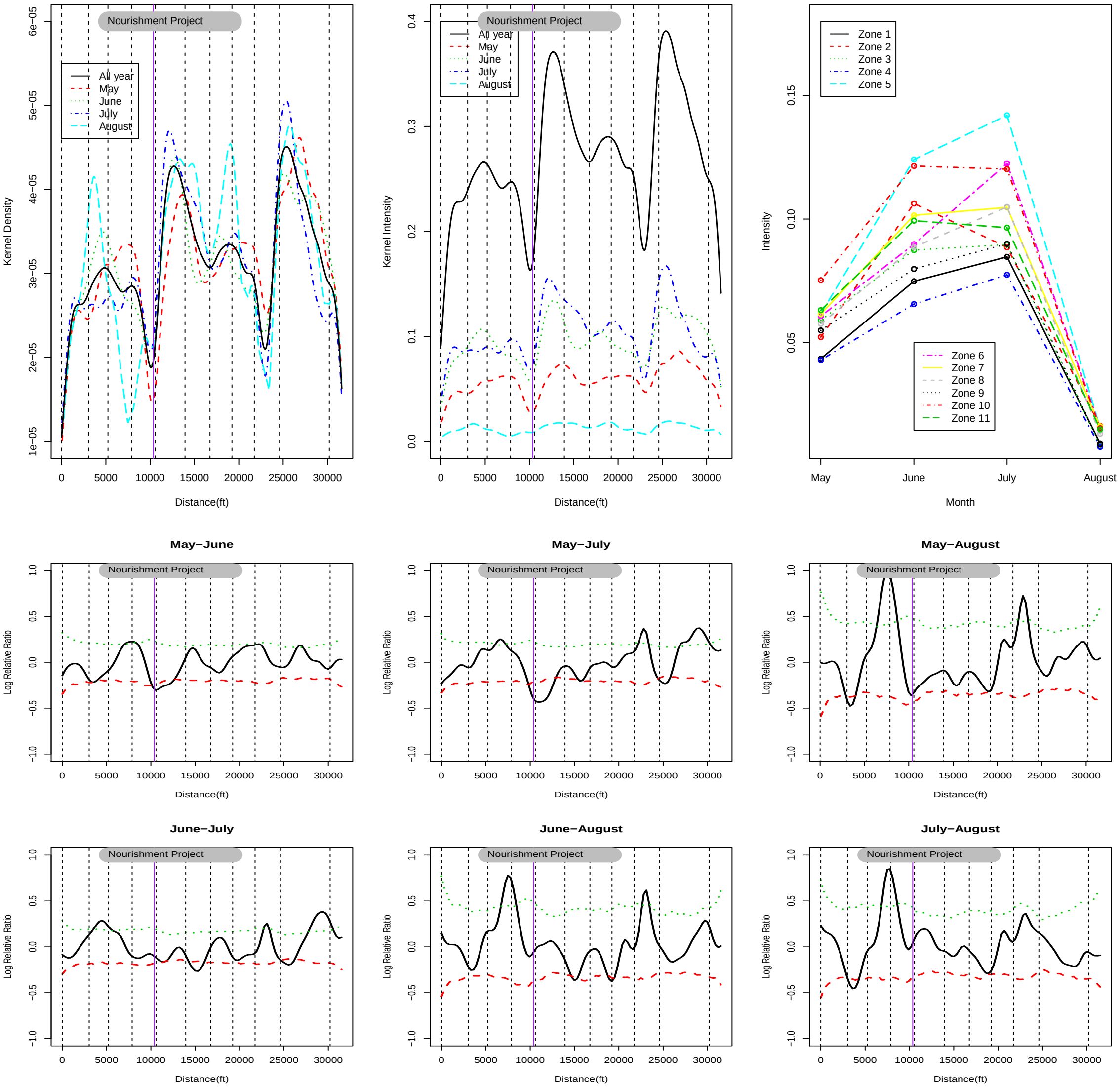


Figure 3.27: The above: Kernel estimates of loggerhead emergence locations (year 1999) by month and zone; The bottom: pairwise comparison of kernel density estimates via Monte-Carlo infe rence

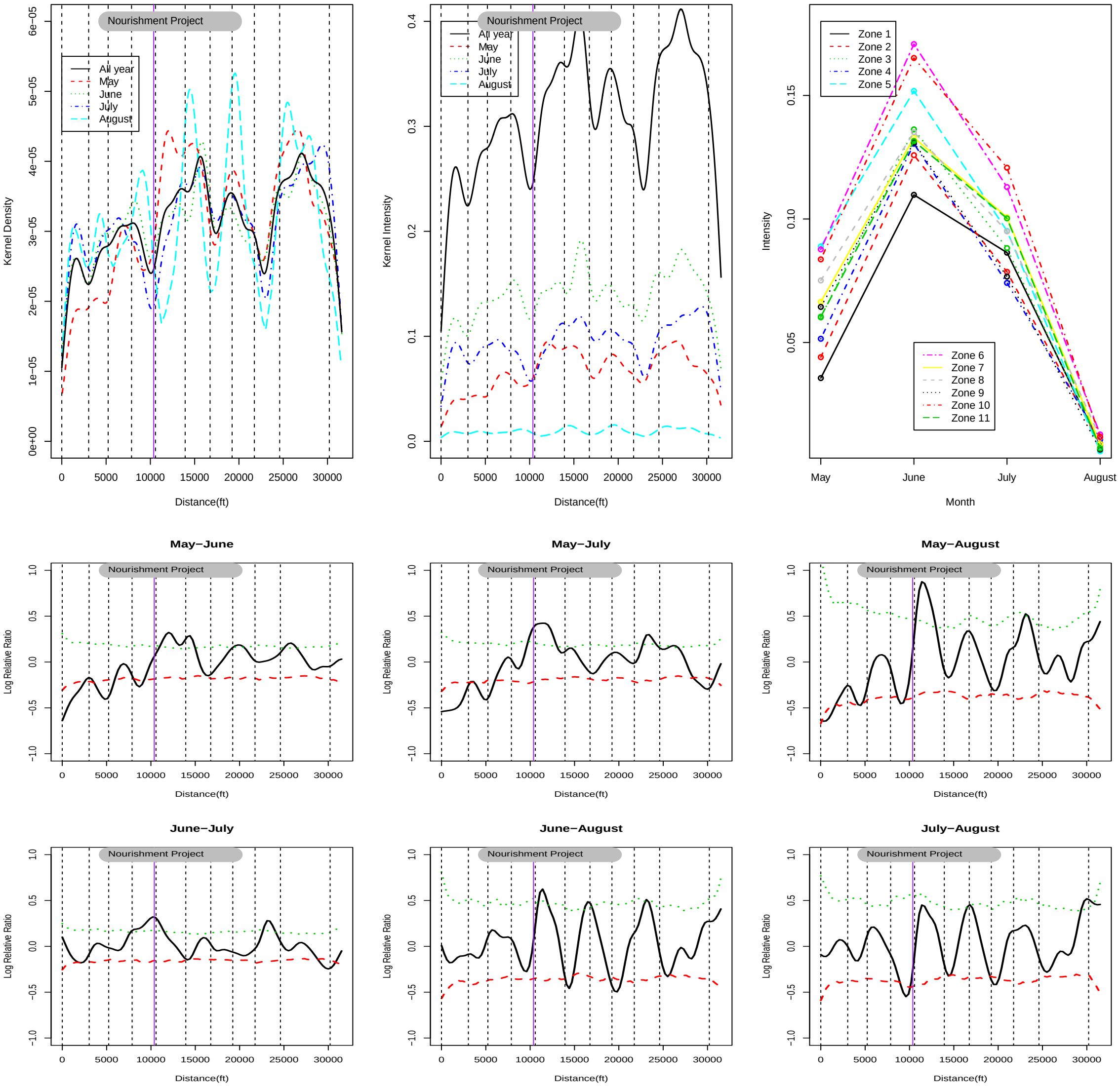


Figure 3.28: The above: Kernel estimates of loggerhead emergence locations (year 2000) by month and zone; The bottom: pairwise comparison of kernel density estimates via Monte-Carlo infe rence

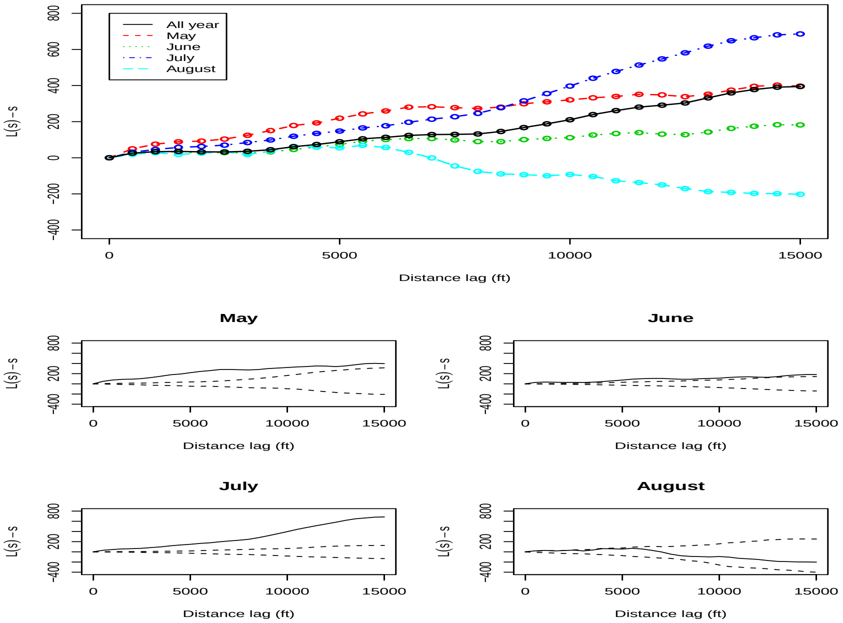


Figure 3.29: The non-parametric estimates of $K-$function for loggerhead emergence locations (year 1998)

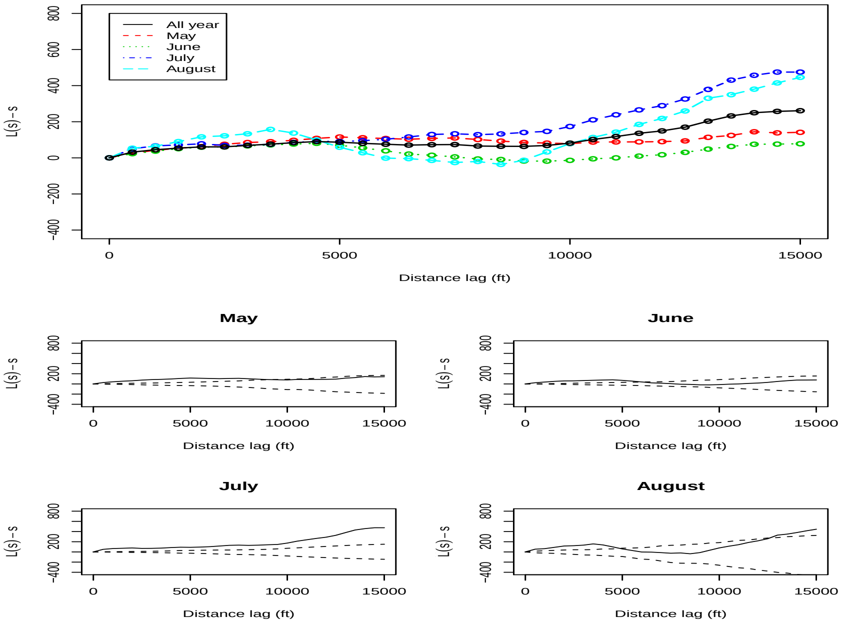


Figure 3.30: The non-parametric estimates of $K-$function for loggerhead emergence locations (year 1999)

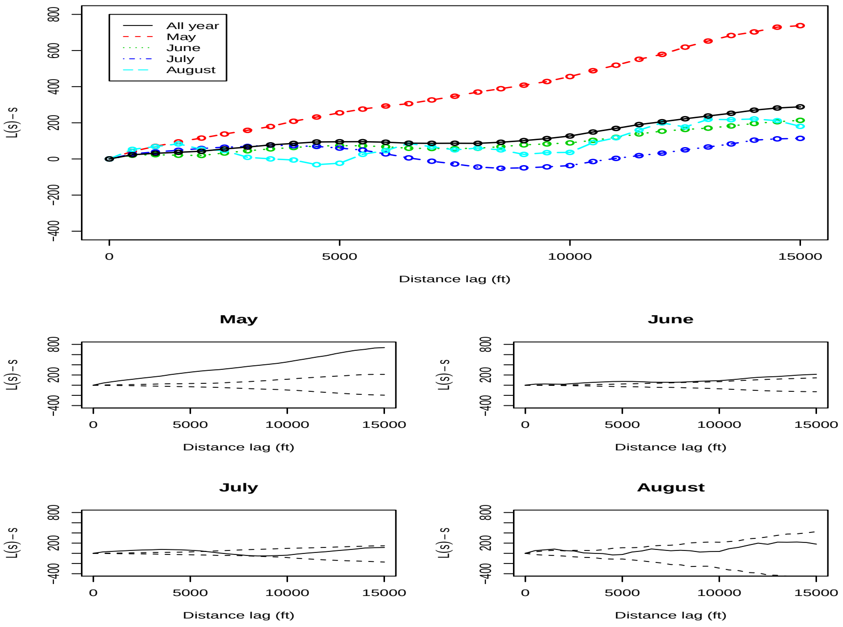


Figure 3.31: The non-parametric estimates of $K-$function for loggerhead emergence locations (year 2000)

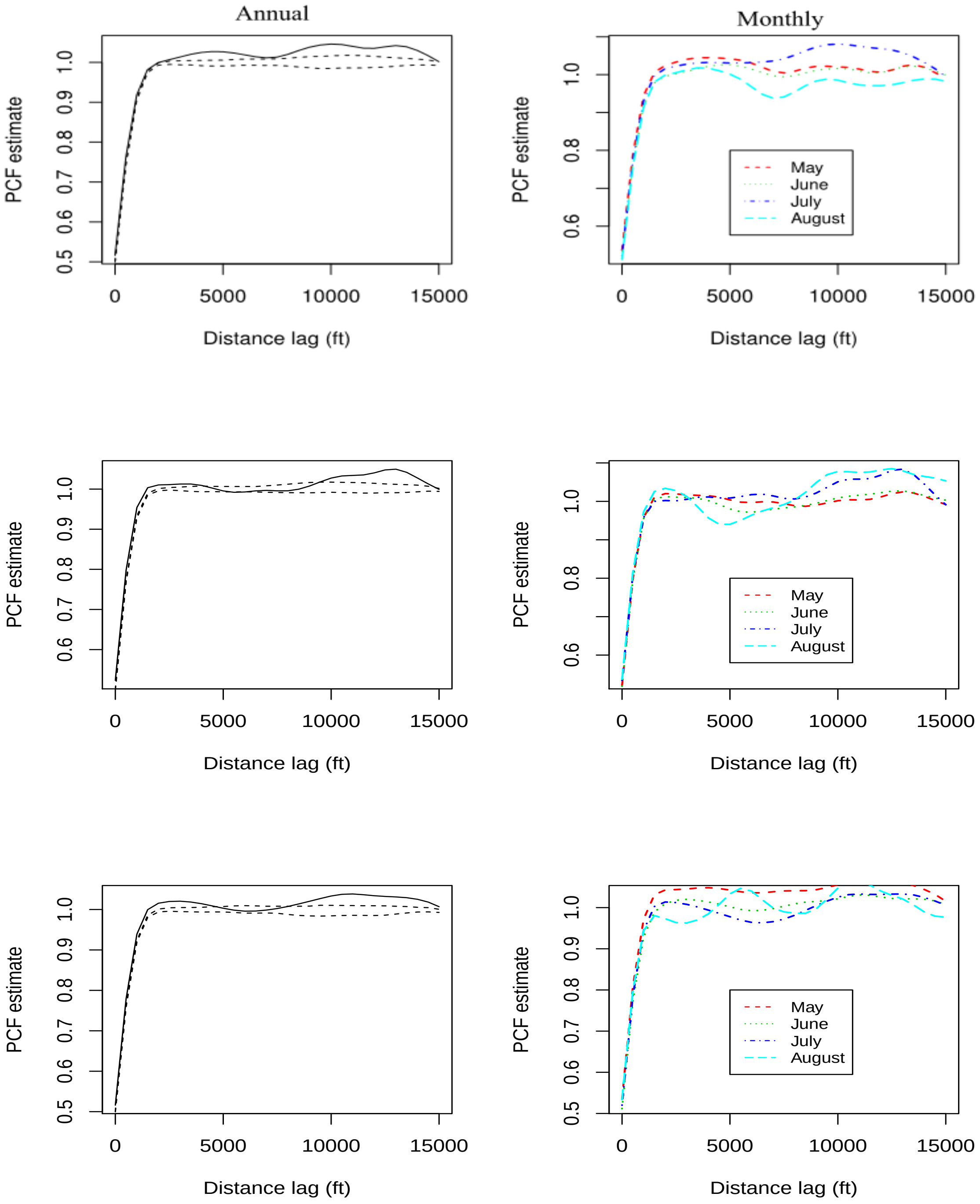


Figure 3.32: The non-parametric estimates of pair correlation function for loggerhead emergence locations (From top to bottom: year 1998-2000)

Chapter 4

Non-parametric Bayesian modeling for density estimation of sea turtle nesting locations along Juno Beach in Florida

4.1 Introduction and motivation data

In this chapter, we focus on spatial-temporal point pattern analysis, in particular, on density estimation to detect the clusters and their trends evolving over time. Our data application involves sea turtle nesting locations along Juno Beach, Palm Beach County, Florida for the years 1998-2000. Palm Beach County contracted the Marine Life Center of Juno Beach to gather daily data regarding sea turtle emergence locations along the beach. We provided a detailed data description in Chapter 1, and conducted kernel density estimation by month and year in Chapter 3. Our earlier comparisons of density estimates between months, we detected clusters and identified unusual collections of events by a random labeling Monte Carlo method. Here, we develop non-parametric Bayesian models for density estimation to track the evolution of these clusters in discrete time units (week/month).

Earlier estimates of the density of turtle nesting locations rely on non-parametric techniques, such as kernel-based estimation (Rosenblatt, 1969; Fan *et al.*, 1996), spline smoothing (Gu and Wang, 2003), and wavelets (Clemencon, 2000). In our previous analysis, we use a kernel method, where the basic idea is to place a kernel of density at each observed location, then sum the kernels together to provide the estimate of the overall density function at that particular location, where the kernel (shape) and bandwidth impact density estimation. Figures 4.1 and 4.2 show kernel density and intensity estimates of sea turtle nesting locations by week for the year 1998. Note that we select a Gaussian kernel function and the bandwidth for each week is chosen based on the direct plug-in approach (Sheather and Jones, 1991). We note that there are 22 weeks in total for year 1998, and the date ranges by week are shown in Table 4.1. The number of events per week varies significantly, from 1 (weeks 1, 2, and 21) to 438 (week 13). Clearly, the distribution is multi-modal, and when the sample size is small, the estimation curve can be quite bumpy. Also, the week-specific estimates do not reveal clear evidence of temporal evolution in the shape of the distributions.

There is an extensive literature on Bayesian density estimation using Gaussian distributions (Escobar and West, 1994; Richardson and Green, 1997). For instance, Fraley et al. (2012) produced a density estimate using a Gaussian finite mixture model and model-based clustering where the optimal model is selected according to the Bayesian Information Criterion (BIC). To avoid the need to pre-specify the number of mixture components for finite mixture models, a Dirichlet processes (DP) prior allows flexibly infinite (non-parametric) mixture modeling. Note that this non-parametric Bayesian approaches has gained wide popularity, and has been extended

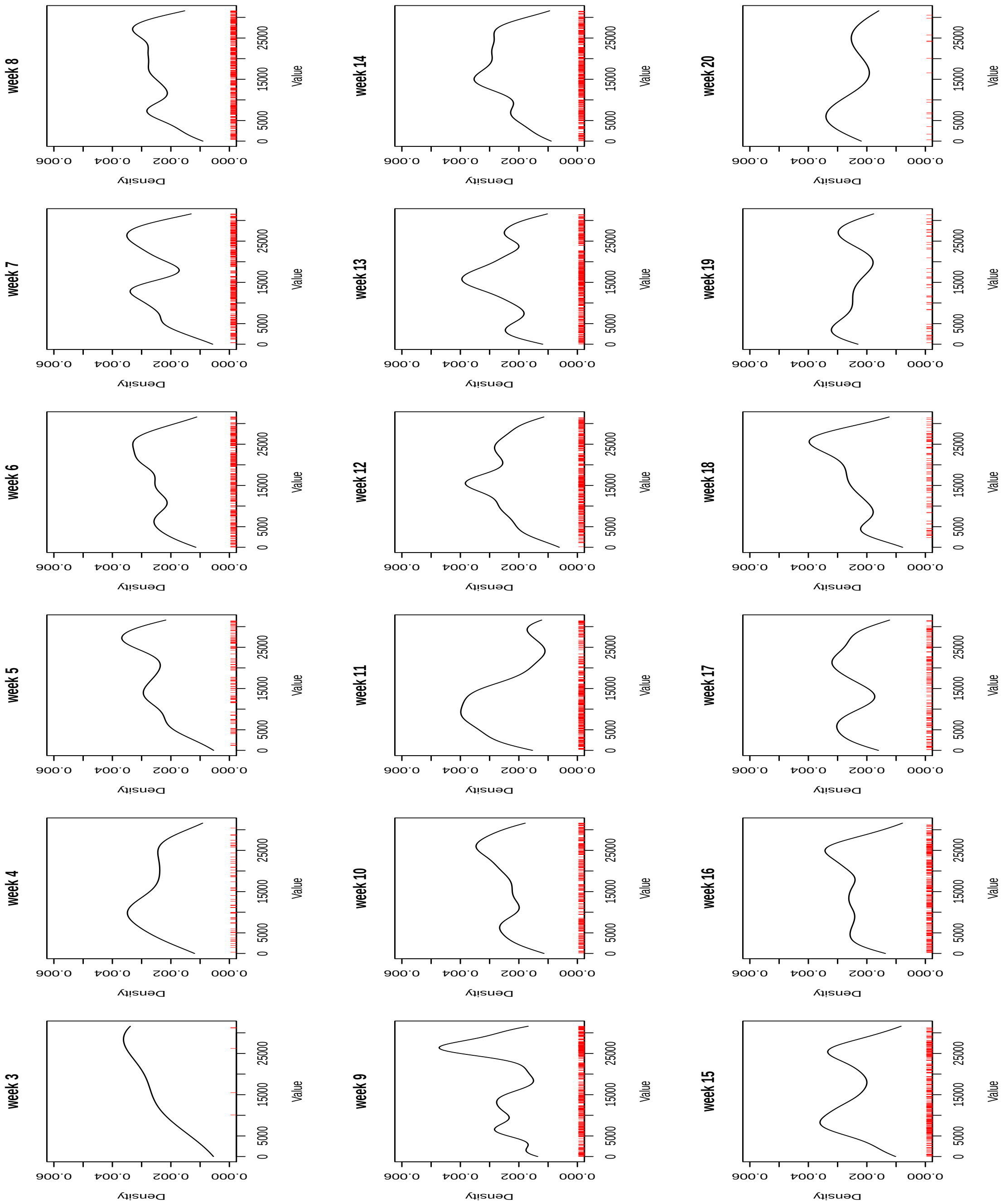


Figure 4.1: The Kernel density estimates of CC nesting along beach by week

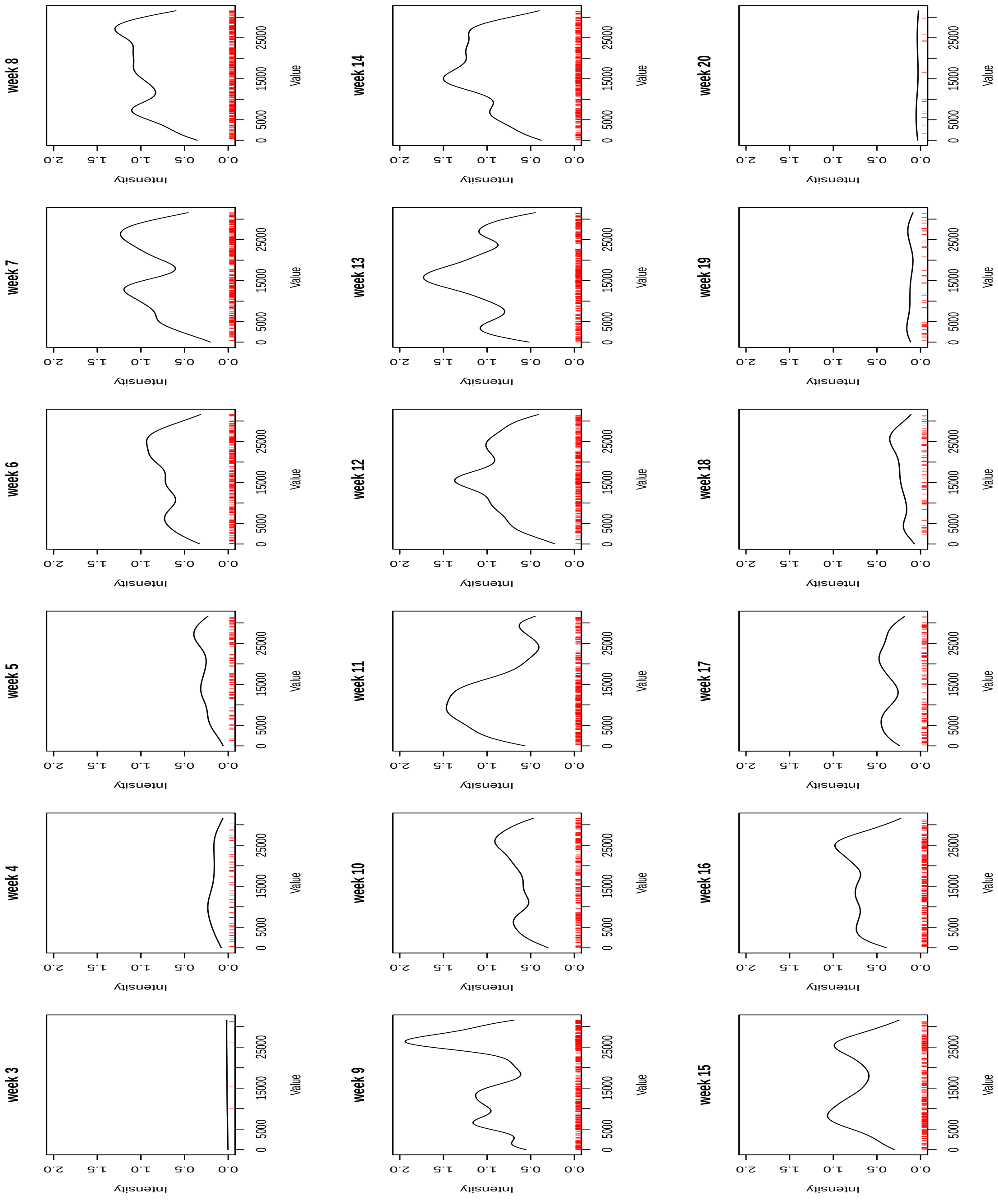


Figure 4.2: The Kernel intensity estimates of CC nesting along beach by week

Table 4.1: The week information in sea turtle nesting data for year 1998

Week	Date	Week	Date
1	Apr 17	2	Apr 23
3	Apr 28- May 1	4	May 4-May 9
5	May 10-May 16	6	May 17-May 23
7	May 24-May 30	8	May 31- Jun 6
9	Jun 7-Jun 13	10	Jun 14-Jun 20
11	Jun 21-Jun 27	12	Jun 28- Jul 4
13	Jul 5-Jul 11	14	Jul 12-Jul 18
15	Jul 19-Jul 25	16	Jul 26- Aug 1
17	Jul 26- Aug 1	18	Aug 9-Aug 15
19	Aug 16-Aug 22	20	Aug 23-Aug 29
21	Aug 30-Sep 4	22	Sep 23

to hierarchical data structure (Teh *et al.*, 2006); however, most previous work assumes exchangeability in sampling. More recent approaches include dependency, particularly temporal dependency (Griffin, 2009; Tang and Ghosal, 2007; Dunson, 2006; Rodriguez and Horst, 2008), which we describe in detail. Here, we consider countable infinitely mixtures of Gaussian distributions at each time period (week), and incorporate temporal dependency into our density estimation to provide insight into sea turtle nesting density evolution over time. We present our model in the framework of dependent DP priors and provide an analytic framework for wider application. For comparison, we also compare our approach to a Dirichlet process mixture model and a hierarchical Dirichlet process model.

The rest of the chapter is organized as follows: In Section 4.2, we describe the Dirichlet Process Mixture model as well as associated computational algorithms and inference. Similarly, we provide an overview of hierarchical Dirichlet Process models including computational and statistical inference in Section 4.3. In Section 4.4, we show the details of our proposed algorithm and associated inference. Section 4.6 provides simulation results based on these alternative approaches, and compares their performance. Finally, short discussion and directions for further research will be seen in Section 4.7 .

4.2 Dirichlet Process Mixture model

4.2.1 Model

We first fit a Dirichlet process mixture (DPM) model for each week separately. Let $\mathbf{X} = \{x_1, x_2, \ldots, x_n\}$ be a set of observed events in $\mathbb{R}^1$. The DPM model can be written as follow

$$G \sim DP(\alpha, H),$$

$$\boldsymbol{\theta}_i | G \sim G, \quad and$$

$$x_i | \boldsymbol{\theta}_i \sim f(\cdot | \boldsymbol{\theta}_i),$$

where $\boldsymbol{\theta}_i$ is denoted as the latent class parameter, which could be a scalar or a vector. Each data point x_i is assigned to a class of distributions with probability density function $f(\cdot|\boldsymbol{\theta}_i)$. We note that G is a prior distribution on $\boldsymbol{\theta}_i$, and $DP(\alpha, H)$ is a DP prior on G with the concentration parameter α and a base distribution H. Notice that $G = \sum_{k=1}^{\infty} \pi_k \delta_{\boldsymbol{\theta}_k^*}$, based on a stick-breaking construction, where $\boldsymbol{\theta}_i^* \sim H$, $\pi_k = \beta_k \prod_{i=1}^{k-1} (1 - \beta_i)$ with $\beta_k \sim Beta(1, \alpha)$. This mixture model can assign the points into classes, where the points in the same class are sampled from the same distribution with the same class parameter due to the DP prior. Also, this model enables infinite classes where the number of clusters is unknown or unbounded as data sample size increases. In another words, the shape of the distribution is driven by the data, allowing flexible inference. The equivalent model can be written as

$$z_i \sim Discrete(\pi),$$

$$\boldsymbol{\theta}_i = \boldsymbol{\theta}^*_{z_i}, \quad and$$

$$x_i|z_i \sim f(\cdot|\boldsymbol{\theta}^*_{z_i}),$$

where z_i is the class label for the i^{th} point.

4.2.2 Algorithm and Inference

A conditional prior distribution for the class parameter $\boldsymbol{\theta}_i$ can be obtained by integrating out G, i.e.,

$$\boldsymbol{\theta}_i|\boldsymbol{\theta}_{-i}, \alpha, H \sim \frac{1}{N-1+\alpha}\Sigma_{j\neq i}\delta(\boldsymbol{\theta}_j) + \frac{\alpha}{N-1+\alpha}H$$

where $\delta(\boldsymbol{\theta}_j)$ is a distribution concentrated at the point $\boldsymbol{\theta}_j$, and $\boldsymbol{\theta}_{-i}$ denotes class assignments for all points except x_i. Combining this prior with the likelihood, we derive the conditional posterior for $\boldsymbol{\theta}_i$ as

$$\boldsymbol{\theta}_i|\boldsymbol{\theta}_{-i}, X, \alpha, H \sim \Sigma_{j\neq i} f(x_i|\boldsymbol{\theta}_j)\delta(\boldsymbol{\theta}_j) + \alpha\left(\int f(x_i|\boldsymbol{\theta})H(\boldsymbol{\theta})d\boldsymbol{\theta}\right)H(\boldsymbol{\theta}_i).$$

If there are K unique values of the class parameters, i.e., $\boldsymbol{\theta}^*_1, \ldots, \boldsymbol{\theta}^*_K$, then using class labels, we can also write the conditional posterior, given z_{-i} as

$$\boldsymbol{\theta}_i|z_{-i}, \mathbf{X}, \alpha, H \sim \Sigma_{k=1}^{K} n_k^{-i} f(x_i|\boldsymbol{\theta}^*_k) + \alpha\left(\int f(x_i|\boldsymbol{\theta})H(\boldsymbol{\theta})d\boldsymbol{\theta}\right)H(\boldsymbol{\theta}_i),$$

where n_k^{-i} shows the number of points in class k after removing the i^{th} point. If we choose a conjugate base distribution H, the predictive probabilities can be computed using sufficient statistics. Thus, efficient Gibbs sampling can be applied to update the class parameter in each iteration (Escobar and West, 1994).

The algorithm for DPM model has been extensively studied by literature (Escobar and West, 1994; Richardson and Green, 1997). The R function “DPdensity” allows density estimation. In our application, the generic function fits a DPM with mixture Gaussian distribution for density estimation is used.

$$G \sim DP(\alpha, H),$$

$$(\mu_i, \sigma_i^2)|G \sim G, \quad and$$

$$x_i|(\mu_i, \sigma_i^2) \sim N(\mu_i, \sigma_i^2),$$

where H is specified as normal-inverse-Wishart distribution, i.e., a conjugate prior

$$H = N(\mu|\kappa, (1/\tau)\sigma^2)IW(\sigma^2|\nu, \psi),$$

where the hyperparameters are defined as $\alpha = 1$, $\kappa = mean(\mathbf{X})$, $\tau \sim Gamma(0.1, 1)$, $\nu = 4$, and $\psi \sim IW(4, 1/var(\mathbf{X}))$.

The basic algorithm is shown as follows:

- Initialization: randomly assign data points into an arbitrary number of K clusters $1 <= K <= N$ (N: the total number of points)
- For each data point x_i, perform the following procedures:
  - Remove the point x_i from its current cluster. If that cluster becomes empty, then this component will be deleted. Given the current assignment of clusters for all the other points, calculate the predictive probability of this point joining each of the existing clusters as well as being a new class (ω is normalizing constant). More specifically,
    - the probability for joining the existing cluster is $\omega n_k^{-i} f(x_i|\boldsymbol{\theta}_k)$, and
    - the probability for joining the new cluster is $\omega\alpha \int f(x_i|\boldsymbol{\theta})H(\boldsymbol{\theta})d\boldsymbol{\theta}$.

- Assign the point x_i to the $K'+1$ possible clusters according to the predictive probabilities, where K' is the current total number of clusters. If a new cluster is created, then θ_i is sampled from the base distribution H. Update z_i and the number of clusters K' based on the assignment.
- For the current assignment, draw a new value from the conditional posterior of cluster parameters $\boldsymbol{\theta}_z | x_i$ such that $z_i = z$.

4.3 Hierarchical Dirichlet Process Model

4.3.1 Model

Notice the separate DPM models above do not share cluster parameters across different groups (week in our application) because the base distribution H is continuous. In this section, we consider an extension of the above DPM model, i.e., a hierarchical Dirichlet Process (HDP) model by placing a DP prior on the base distribution and incorporating a hierarchical structure. Let $\mathbf{X}_j = \{x_{j1}, x_{j2}, \ldots, x_{jn_j}\}$ denote the observed points in $\mathbb{R}^1$ from the j^{th} group (week), $j \in J$ with J as the index set. This model allows different groups of points (by week) to share the same prior on the cluster parameters, i.e., the atoms of mixture distribution. In our application, given that sea turtle nesting locations may have similar point patterns within each week due to limited changes in environmental conditions within a week, it is reasonable to assign the points to clusters that share group-specific parameters. An advantage of this model is the borrowing of information across groups, allowing estimation, even with small sample sizes. The HDP model can be written as follows

$$G_0 \sim DP(\gamma, H),$$

$$G_j \sim DP(\alpha_j, G_0), j \in J,$$

$$\boldsymbol{\theta}_{ji}|G_j \sim G_j, i = 1, 2, \ldots, n_j, \quad and$$

$$x_{ji}|\boldsymbol{\theta}_{ji} \sim f(\cdot|\boldsymbol{\theta}_{ji}),$$

where x_{ji} is the i^{th} point in the j^{th} group and $\boldsymbol{\theta}_{ji}$ denotes a latent class parameter. As above, the $G_j' s$ denote the group prior distributions for the class parameters as a DP with concentration parameter α_j and baseline distribution G_0, and G_0 itself follows another DP having concentration parameter γ and baseline distribution H. Similar to DPM models, this HDP model can be represented by a stick-breaking construction. We write down $G_0 = \sum_{k=1}^{\infty} \beta_k \delta_{\boldsymbol{\theta}_k^*}$, where $\boldsymbol{\beta}|\gamma \sim Stick(\gamma)$ and $\boldsymbol{\theta}_k^*|H \sim H$. Furthermore, $G_j = \sum_{k=1}^{\infty} \pi_{jk} \delta_{\boldsymbol{\theta}_k^*}$, where $\pi_j|\alpha_j, \boldsymbol{\beta} \sim DP(\alpha_j, \boldsymbol{\beta})$. The calculation details involve $\pi_{jk} = \nu_{jk} \Pi_{l=1}^{k-1}(1 - \nu_{jl})$ with $\nu_{jk}|\alpha_j, \beta_1, \ldots, \beta_k \sim Beta(\alpha_j \beta_k, \alpha_j(1 - \Sigma_{l=1}^{k-1} \beta_l))$. An analogous representation is called the Chinese restaurant franchise (CRF) (each customer represents a point), which is popularly used in Markov chain Monte Carlo (MCMC) sampling schemes for the HDP model (Teh *et al.*, 2006), as discussed next.

Let $\boldsymbol{\theta}_{jt}^*$ denote the random variable for the t^{th} table in the j^{th} restaurant, which is sampled from G_0; $\boldsymbol{\theta}_k^{**}$ denotes iid variables from the base distribution H. The class parameter $\boldsymbol{\theta}_{ji}$ is associated with table t_{ji} in the j^{th} restaurant, $\boldsymbol{\theta}_{ji} = \boldsymbol{\theta}_{jt_{ji}}^*$; the variable $\boldsymbol{\theta}_{jt}^*$ is related to the mixture component k_{jt}, $\boldsymbol{\theta}_{jt}^* = \boldsymbol{\theta}_{k_{jt}}^{**}$. Several important features include: n_{jtk} denotes the number of customers who are served dish k at table t in restaurant j; m_{jk} is the total number of tables serving dish k in restaurant j; K is denoted as the total number of unique dishes served in restaurants; $x^{-ji} = \mathbf{X} \setminus x_{ji}$, $k^{-jt} = \mathbf{k} \setminus k_{jt}$, and n_{jt}^{-ji} denotes the number of customers at table t except x_{ji} in restaurant j. The marginal probabilities of HDP can be captured by CRF. The conditional prior distribution of the class parameter $\boldsymbol{\theta}_{ji}$ are obtained by integrating out G_j, i.e.,

$$\boldsymbol{\theta}_{ji}|\boldsymbol{\theta}_{j(-i)}, \alpha, G_0 \sim \Sigma_{t=1}^{m_{j.}} \frac{n_{jt.}}{\alpha + n_{j..}} \delta(\boldsymbol{\theta}^*_{jt}) + \frac{\alpha}{\alpha + n_{j..}} G_0.$$

Similarly, after integrating out G_0, the conditional distribution for $\boldsymbol{\theta}^*_{jt}$ is

$$\boldsymbol{\theta}^*_{jt}|\boldsymbol{\theta}^*_{j(-t)}, \gamma, H \sim \Sigma_{k=1}^{K} \frac{m_{.k}}{\gamma + m_{..}} \delta(\boldsymbol{\theta}^{**}_k) + \frac{\gamma}{\gamma + m_{..}} H$$

Also, since H is a conjugate prior in our situation, the conditional density of x_{ji} with $k_{jt_{ji}} = k$ given all the other data can be obtained by integrating out $\boldsymbol{\theta}^{**}_k$ as follows:

$$f^{-x_{ji}}(x_{ji}; k) = \frac{\int f(x_{ji}|\boldsymbol{\theta}^{**}_k) \prod_{j'i' \neq ji; k_{jt_{ji}}=k} f(x_{j'i'}|\boldsymbol{\theta}^{**}_k) h(\boldsymbol{\theta}^{**}_k) d\boldsymbol{\theta}^{**}_k}{\int \prod_{j'i' \neq ji; k_{jt_{ji}}=k} f(x_{j'i'}|\boldsymbol{\theta}^{**}_k) h(\boldsymbol{\theta}^{**}_k) d\boldsymbol{\theta}^{**}_k},$$

which is the conditional density of x_{jt} given all the other data.

4.3.2 Algorithm and Inference

There are three sampling schemes mentioned by Teh et al. (2006) and Neal (2000), allowing MCMC sampling from the desired posterior distribution. Here, we focus on the CRF algorithm. The generic function fits a HDP model with mixture of normal distributions for the density estimation as follows:

$$G_0 \sim DP(\gamma, H),$$

$$G_j \sim DP(\alpha, G_0),$$

$$(\mu_{ji}, \sigma^2_{ji})|G_j \sim G_j, \quad and$$

$$x_{ji}|(\mu_{ji}, \sigma^2_{ji}) \sim N(\mu_{ji}, \sigma^2_{ji}),$$

where H is specified as normal-inverse-Wishart distribution as a conjugate prior

$$H = N(\mu|\kappa, (1/\tau)\sigma^2)IW(\sigma^2|\nu, \psi),$$

with hyperparameters defined as $\gamma = 1$, $\alpha = 1$, $\kappa = mean(\mathbf{X})$, $\tau \sim Gamma(500, 500)$, $\nu = 1$, and $\psi \sim IW(1, 1/var(\mathbf{X}))$.

According to the sampling scheme introduced by Neal (2000), we update index variables t_{ji} and k_{jt} by taking advantage of exchangeability, and afterwards, the class parameters $\boldsymbol{\theta}_{ji}$ and $\boldsymbol{\theta}^*_{jt}$ are reconstructed from the current index variables and $\boldsymbol{\theta}^{**}_k$. The conditional posterior distribution of t_{ji} given the other data items can be obtained by combining the conditional prior of t_{ji} and the likelihood of x_{ji}, which is

$$p(t_{ji} = t|\mathbf{t}^{-ji}, \mathbf{k}) \propto \begin{cases} n^{-ji}_{jt.} f^{-x_{ji}}(x_{ji}; k) & \text{if table } t \text{ has been occupied,} \\ \alpha p(x_{ji}|\mathbf{t}^{-ji}, t_{ji} = t^{new}; \mathbf{k}) & \text{if } t = t^{new}, \end{cases}$$

where $p(x_{ji}|\mathbf{t}^{-ji}, t_{ji} = t^{new}; \mathbf{k}) = \sum_{k=1}^{K} \frac{m_{.k}}{m_{..}+\gamma} f^{-x_{ji}}(x_{ji}; k) + \frac{\gamma}{m_{..}+\gamma} f^{-x_{ji}}(x_{ji}; k^{new})$, and $k_{jt^{new}}$ can be sampled from the following distribution

$$p(k_{jt^{new}} = k|\mathbf{t}, \mathbf{k}) \propto \begin{cases} m_{.k} f^{-x_{ji}}(x_{ji}; k) & \text{if } k \text{ has already exsited,} \\ \gamma f^{-x_{ji}}(x_{ji}; k^{new}) & \text{if } k = k^{new}. \end{cases}$$

Based on the calculated predictive conditional probabilities from above, we assign x_{ji} to the table previously occupied or to a new table, and if the table is new, then it will be assigned to the existing k or to k^{new} which is sampled from the base distribution H. Next, to update the index k_{jt}, the conditional probability follows

$$p(k_{jt} = k|\mathbf{t}, \mathbf{k}^{-jt}) \propto \begin{cases} m^{-jt}_{.k} f^{-x_{jt}}(\mathbf{x}_{jt}; k) & \text{if } k \text{ has already exsited,} \\ \gamma f^{-x_{jt}}(\mathbf{x}_{jt}; k^{new}) & \text{if } k = k^{new}. \end{cases}$$

The basic algorithm used here is detailed below:

- **Initialization**: For each group, randomly assign customers to an arbitrary table t ($1 <= t <= n_j$), and also randomly assign each table to a cluster k, $1 <= k <= N$ (N: the total number of customers).

- **Sampling** t_{ji}: For each customer x_{ji}, perform the following procedure:

  - Remove the customer x_{ji} from its current table. If that table becomes empty, then the probability that this table is reoccupied is 0. Also, the corresponding k_{jt} may be deleted, and if after deleting k_{jt}, the cluster label k is unallocated in this value, thus this mixture component should also be deleted. Given the current assignment of tables for all the other customers, assign this customer to join any of the existing tables as well as sitting at a new table based on the conditional probabilities above.

- **Sampling** k_{jt}: For each table t, perform the following procedures:

  - Remove the table t from its current cluster. If that cluster becomes empty, then the mixture component will be deleted. Given the current assignment of clusters for all the tables, calculate the predictive probability of this table joining any of the existing clusters as well as joining a new class based on the above derivation.

4.4 Hierarchical Dirichlet Process Autoregressive Model

4.4.1 Model

As we know, HDP models allow different groups to share the same atoms and exchangeable collections of distributions, which does not mirror application with its sequential daily data. In our case, nesting locations are collected by day, and the event locations also have week/month assignments. We expect patterns in adjacent weeks to have closer relationships than pattern far apart, which violates the typical assumptions in HDP models. To incorporate temporal dependency among the infinite mixture distributions across weeks, Rodriguez and Horst (2008) applied dependent Dirichlet processes (DDP) to time series data. This model has the same weights for all the mixture components across different groups (week) and induces dependency through random permutations of the atoms for the mixture distributions, i.e., $\boldsymbol{\theta}^*_{jk}|\boldsymbol{\theta}^*_{j-1,k} \sim \zeta(\boldsymbol{\theta}^*_{j-1,k})$ with $\zeta(\cdot)$ is a known pre-specified function. To consider varying weights of the atoms, there exist several approaches, i.e., the order-based DDP (Griffin and Steel, 2006), the local DP (Chung and Dunson, 2011), dynamic linear models with DP components (Caron *et al.*, 2008) and so on, but all of them have limitation for applications with long-range dependency. Ren and Dunson (2008) proposed an alternative model called a dynamic hierarchical Dirichlet process (dHDP) model (Ren *et al.*, 2008). This model defines $G_j = (1 - \tilde{\omega}_{j-1})G_{j-1} + \tilde{\omega}_{j-1}H_{j-1}$ to allow G_j share features from G_{j-1} and may also include the atoms from an innovation distribution H_{j-1}, where H_{j-1} is from a DP and the probability $\tilde{\omega}_{j-1} \sim Beta(a_{j-1}, b_{j-1})$. We extend this model to a hierarchical Dirichlet process autoregressive (HDPA) model by incorporating hierarchical data structure with a specific form of temporal dependency.

We use the same notation as the HDP models in Section 4.2. We also consider a nested structure of weeks within months by having weeks in the same month to share the same innovation probability $\tilde{\omega}$, perhaps reflecting weeks within similar lunar cycles. The model becomes:

$$G_0 \sim DP(\gamma, H),$$

$$G^*_{m_{j-1}} \sim DP(\varphi_{m_{j-1}}, G_0), \quad G^*_1 \sim DP(\varphi_1, G_0),$$

$$H_{j-1} \sim DP(\alpha_j, G^*_{m_{j-1}}), \quad G_1 \sim DP(\alpha_1, G^*_1),$$

$$G_j = (1 - \tilde{\omega}_{m_{j-1}})G_{j-1} + \tilde{\omega}_{m_{j-1}} H_{j-1},$$

$$\boldsymbol{\theta}_{ji} | G_j \sim G_j, i = 1, 2, \ldots, n_j, \quad and$$

$$x_{ji} | \boldsymbol{\theta}_{ji} \sim f(\cdot | \boldsymbol{\theta}_{ji}),$$

where m_j indexes month for the j^{th} week taking value $\{1, 2, \ldots, M\}$, and the weeks within the m^{th} month share the same probability generated from $Beta(a_{\tilde{m}}, b_{\tilde{m}})$ with $\mathbf{a}$ and $\mathbf{b}$ as hyperparameters.; $\tilde{\omega}_{m(j)} \sim Beta(a_{m(j)}, b_{m(j)})$ with $\mathbf{a}$ and $\mathbf{b}$ as hyperparameters. This application of a periodic density process has many applications since annual or monthly patterns exist in economic or financial data. Taking these patterns into account in developing models is crucial for accurate inference. We also note that if all $\tilde{\omega}_m$ are ones, the model is a HDP model; if all $\tilde{\omega}_m$ are zeros, the model is a DPM model, illustrating that both HDP and DPM models are special cases of the more general formulation. Their graphic representations are shown in Figure 4.3.

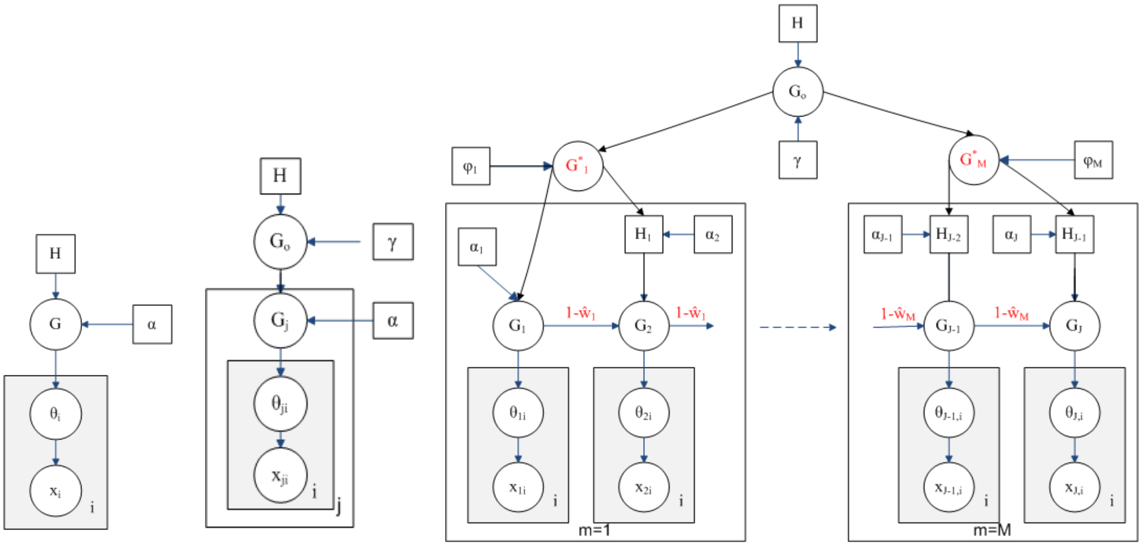


Figure 4.3: Graphic representations of DPM, HDP and HDPA models

4.4.2 Algorithm and Inference

Similar to dHDP models, we have

$$G_j = t_{j1} \sum_{k=1}^{\infty} \pi_{1k} \delta_{\boldsymbol{\theta}_k^*} + t_{j2} \sum_{k=1}^{\infty} \pi_{2k} \delta_{\boldsymbol{\theta}_k^*} + \ldots + t_{jj} \sum_{k=1}^{\infty} \pi_{jk} \delta_{\boldsymbol{\theta}_k^*},$$

where $t_{jl} = \tilde{\omega}_{m_{l-1}} \prod_{n=l}^{j-1} (1 - \tilde{\omega}_{m_n})$ with $\tilde{\omega}_{m_0} = 1$, and $\sum_{l=1}^{j} t_{jl} = 1$. Here, we denote η_{ji} as the index indicating which mixture distribution is taken based on a multinomial distribution, and z_{ji} denotes which parameter component is chosen. Therefore, we get

$$\{\boldsymbol{\theta}_k^*\}_{k=1}^{\infty} | H \sim H, \quad \tilde{\omega}_{m_j} | \boldsymbol{a}, \boldsymbol{b} \sim Beta(a_{m_j}, b_{m_j}),$$

$$\boldsymbol{\beta} | \gamma \sim Stick(\gamma), \quad \boldsymbol{\beta}_{m_j} | \boldsymbol{\varphi}_{m_j}, \boldsymbol{\beta} \sim Dir(\boldsymbol{\varphi}_{m_j}, \boldsymbol{\beta}),$$

$$\boldsymbol{\pi}_j | \alpha_j, \boldsymbol{\beta}_{m_j} \sim DP(\alpha_j, \boldsymbol{\beta}_{m_j}),$$

$$\eta_{ji} | \tilde{\omega} \sim \boldsymbol{t}_j, \quad z_{ji} | \{\boldsymbol{\pi}_1, \ldots, \boldsymbol{\pi}_j\}, \eta_{ji} \sim \boldsymbol{\pi}_{\eta_{ji}}, \quad and$$

$$x_{ji}|z_{ji}, \{\boldsymbol{\theta}^*_k\}_{k=1}^{\infty} \sim f(\boldsymbol{\theta}^*_{z_{ji}}).$$

The Gibbs sampling algorithm for this model is similar to for dHDP models using a truncated stick-breaking process (Ren *et al.*, 2008), and the procedures is as follows:

- Update $\tilde{\boldsymbol{\omega}}$ from

$$(\tilde{\omega}_m|\cdots) \sim Beta\left(a_m + \sum_{j:m_j>m} n_{jm}, b_m + \sum_{j:m_j>m}\sum_{h=0}^{m-1} n_{jh}\right),$$

  where $n_{jm} = \sum_{i=1}^{n_j}\sum_{k:m_{k-1}=m}\delta(\eta_{ji}=k)$.

- Update $\{\boldsymbol{\beta}_{m_j}\}_{j=1}^J$ and $\{\pi_j\}_{j=1}^J$;

- Update the indicator η_{ji} from

$$Pr(\eta_{ji}=l|\cdots) \propto \tilde{\omega}_{m_{l-1}}\prod_{n=l}^{j-1}(1-\tilde{\omega}_{m_n})\pi_{lz_{ji}}\prod_{\tau=l}^{z_{ji}-1}(1-\pi_{l\tau})Pr(x_{ji}|\boldsymbol{\theta}^*_{z_{ji}}).$$

- Update the indicator z_{ji} from

$$Pr(z_{ji}=k|\cdots) \propto \pi_{\eta_{ji},k}\prod_{\tau=l}^{k-1}(1-\pi_{\eta_{ji}\tau})Pr(x_{ji}|\boldsymbol{\theta}^*_k).$$

- Update $\{\boldsymbol{\theta}^*_k\}_{k=1}^K$ and hyperparameters γ, $\boldsymbol{\varphi}$ and $\boldsymbol{\alpha}$.

Note that $n_{jm} = \sum_{i=1}^{n_j}\sum_{k:m(k-1)=m}\delta(\eta_{ji}=k)$ and $n_{j0} = \sum_{i=1}^{n_j}\delta(\eta_{ji}=1)$. Similar to the previous two models, H is specified as normal-inverse-Wishart distribution, a conjugate prior $H = N(\mu|\kappa,(1/\tau)\sigma^2)IW(\sigma^2|\nu,\psi)$ where the hyperparameters are defined as $\kappa = mean(\mathbf{X})$, $\tau \sim Gamma(0.1,1)$, $\nu = 4$, and $\psi \sim IW(4, 1/var(\mathbf{X}))$; $a_m \sim Gamma(1,1)$ and $b_m \sim Gamma(0.1,0.1)$; $\alpha \sim Gamma(1,1)$, $\varphi \sim Gamma(0.5,0.5)$, and $\gamma \sim Gamma(0.5,0.5)$.

4.5 Simulation

To explore the difference between our proposal and the other modeling alternatives, we conduct a simple simulation to compare results. To mimic our data, we use the settings shown in Table 4.2 to simulate the data. There are four combinations of Gaussian mixture distributions with different values for μ, σ^2, and p. We consider four months $D1$, $D2$, $D3$ and $D4$, each of which has four weeks indexed by the subscript in the table. We consider similar patterns for the weeks within each month by sharing the same mixture components and slightly different weights; while between months, the patterns are quite different due to different components and weights though all models share a common component $N(0,1)$. Note that there exists time dependency between weeks. For each mixture distribution, we generate 100 points, and there are 1600 points within our 16 weeks in total.

We apply the kernel method, DPM models, HDP models, dHDP models, and HDPA models to estimate the density for each week, where posterior inference is based on 1000-sample burn in and 5000 samples saving every 10^{th} observation. The results are shown graphically in Figure 4.4. We find that HDPA models perform the best in terms of localizing the mixture components more accurately and they also exhibit the smallest Kullback-Leibler (KL) divergence measures for weeks 3, 7, 11 and 14, as shown in Figure 4.5. Although HDPA models have KL divergence measure slightly higher than dHDP models in week 3, the difference is quite small. We see that HDPA models not only incorporate time-dependency but also incorporate the hierarchical structure (month), therefore, the mixture atoms are captured well, allowing superiority over the other alternatives. HDP models perform satisfactory in some weeks (smaller bias), but do not retain the mixture components as well as HDPA models although HDP estimates borrow information benefit across weeks. Finally, kernel methods perform the worst showing the highest KL divergence measures.

Table 4.2: Parameter settings for simulation studies

	Comp 1			Comp 2			Comp 3			Comp 4		
	p	μ	σ^2	p	μ	σ^2	p	μ	σ^2	p	μ	σ^2
$D1_1$	0.75	0	1	0.25	3	2						
$D1_2$	0.73	0	1	0.27	3	2						
$D1_3$	0.71	0	1	0.29	3	2						
$D1_4$	0.69	0	1	0.31	3	2						
$D2_1$	0.65	0	1	0.35	$a_1 : N(3, 2)$	2						
$D2_2$	0.63	0	1	0.37	$a_1 : N(3, 2)$	2						
$D2_3$	0.61	0	1	0.39	$a_1 : N(3, 2)$	2						
$D2_4$	0.59	0	1	0.41	$a_1 : N(3, 2)$	2						
$D3_1$	0.55	0	1	0.30	$a_2 : N(a_1, 2)$	2	0.15	2	2			
$D3_2$	0.53	0	1	0.30	$a_2 : N(a_1, 2)$	2	0.17	2	2			
$D3_3$	0.51	0	1	0.30	$a_2 : N(a_1, 2)$	2	0.19	2	2			
$D3_4$	0.49	0	1	0.30	$a_2 : N(a_1, 2)$	2	0.21	2	2			
$D4_1$	0.45	0	1	0.20	$a_3 : N(a_2, 2)$	2	0.20	2	2	0.15	10	1
$D4_2$	0.43	0	1	0.20	$a_3 : N(a_2, 2)$	2	0.20	2	2	0.17	10	1
$D4_3$	0.41	0	1	0.20	$a_3 : N(a_2, 2)$	2	0.20	2	2	0.19	10	1
$D4_4$	0.39	0	1	0.20	$a_3 : N(a_2, 2)$	2	0.20	2	2	0.21	10	1

DPM models outperform kernel methods by considering infinity clusters of Gaussian components, thus providing more adaptive results.

4.6 Results

We summarize the kernel density and intensity estimates by month for the sea turtle nesting data in Figure 4.6 and 4.7. Similarly, the results based on DPM models, HDP models and HDPA models appear in Figure 4.8, 4.9 and 4.10 respectively. Note that the sample sizes are extremely small in weeks 1-3 (late April) and weeks 20-22 (late August), containing only one or two points, thus the estimation line is quite flat and close to 0 in Figure 4.2. Therefore, we do not consider these weeks here.

To understand the temporal trend, we aggregate the estimation plots by month in Figures 4.6-4.10, where weeks 4-7 represent May, weeks 8-11 June, weeks 12-15 July,

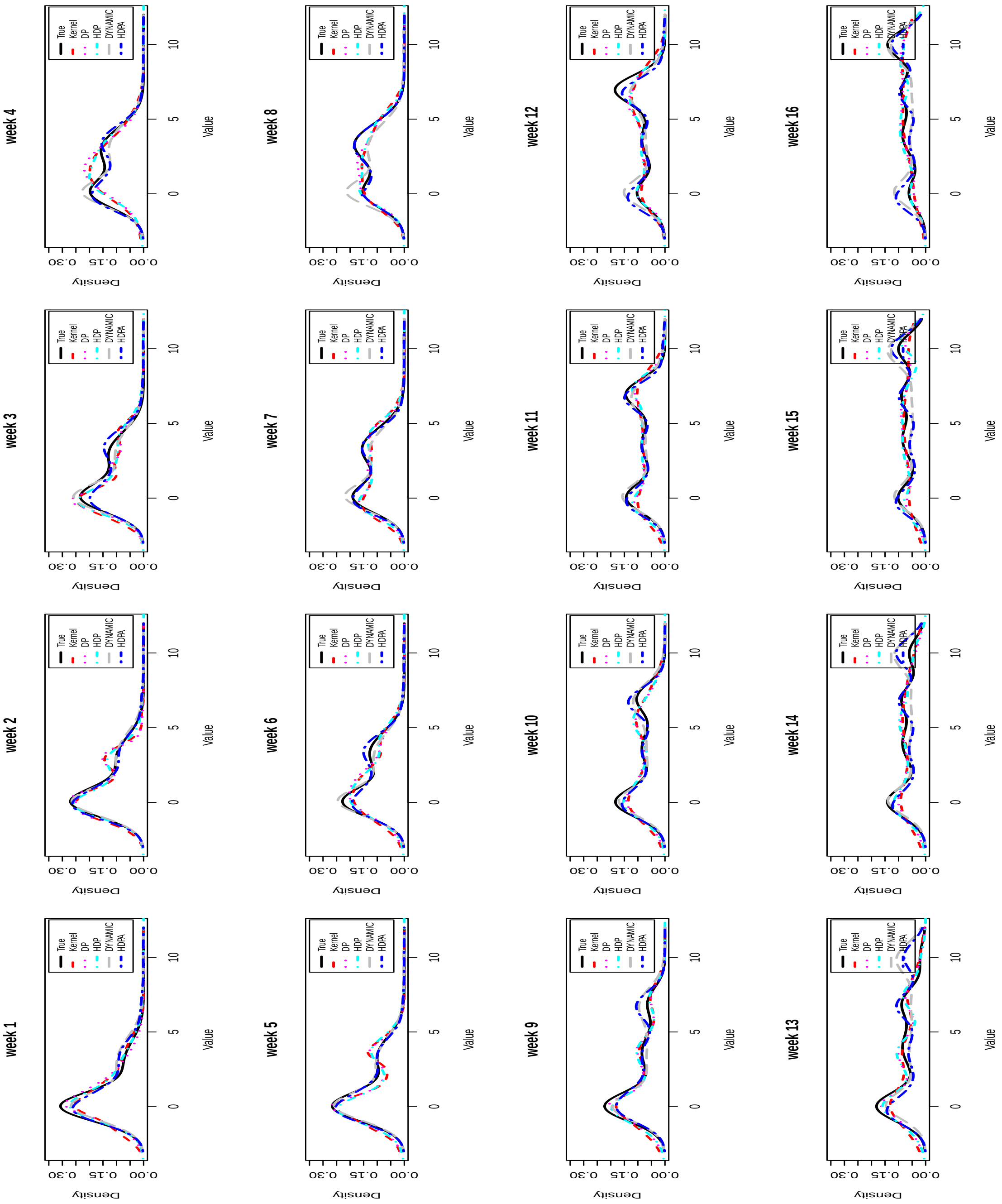


Figure 4.4: Summary of the density estimates for each week by month

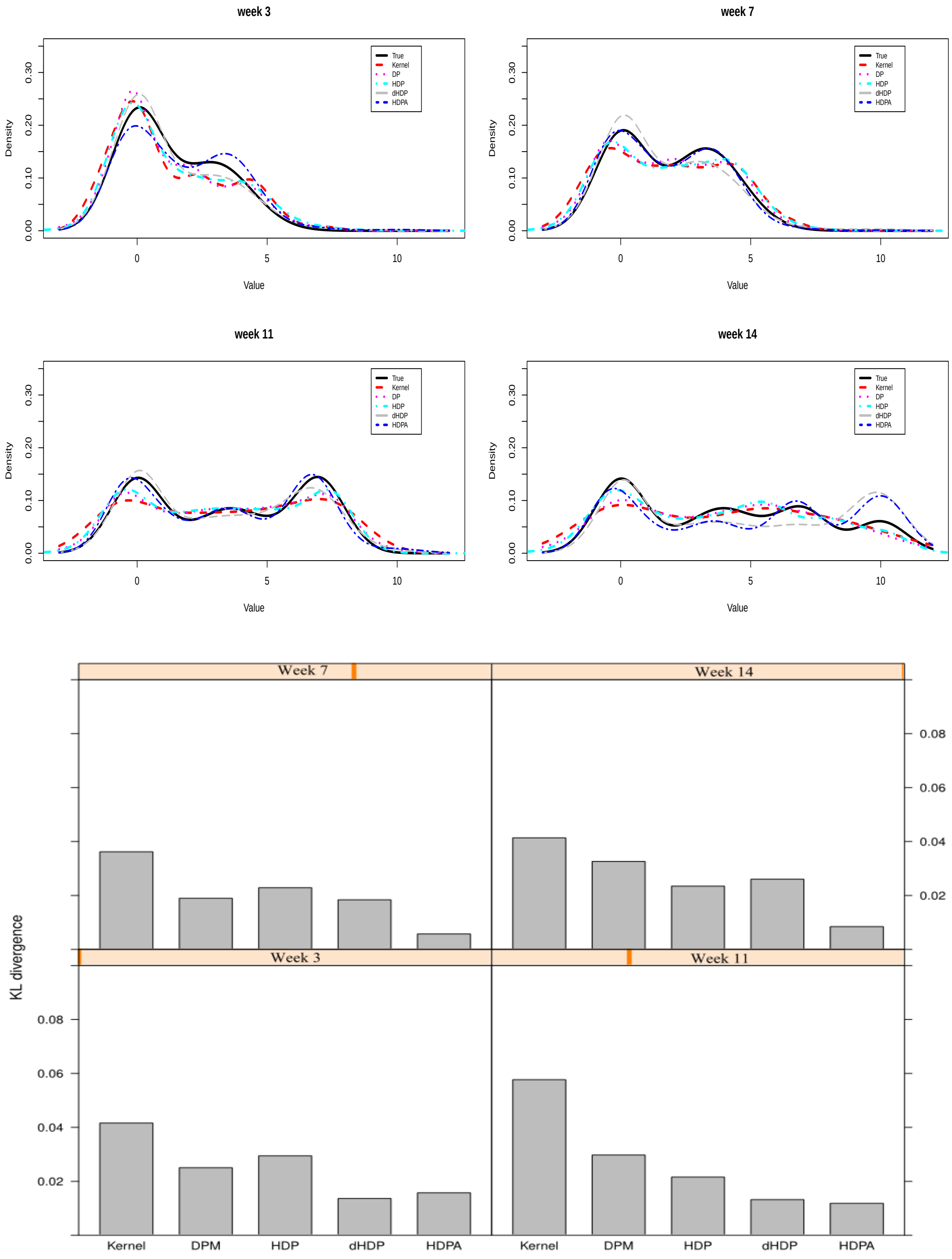


Figure 4.5: The density estimates and Kullback-Leibler divergence measures for weeks 3, 7, 11 and 14

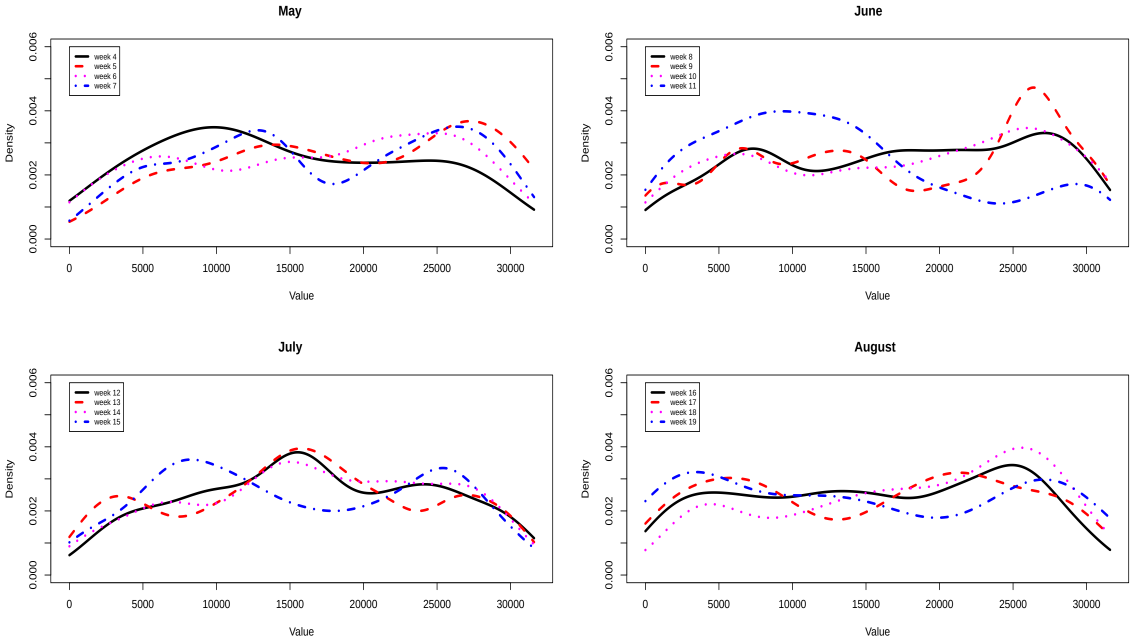


Figure 4.6: Kernel density estimates of loggerhead nesting along Juno Beach, 1998

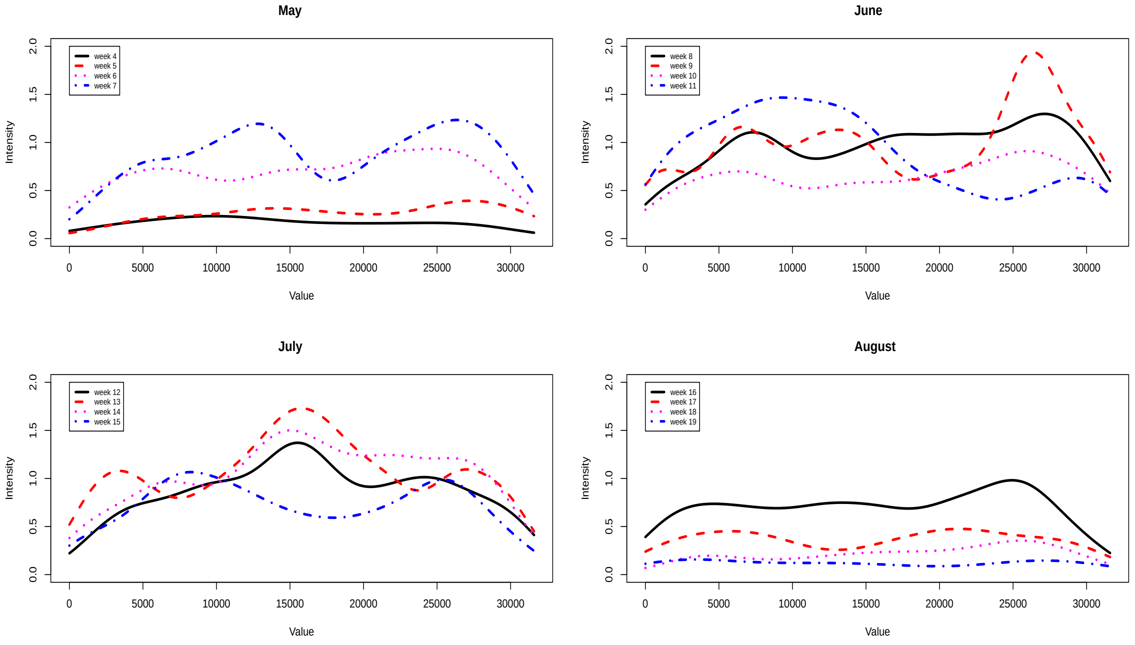


Figure 4.7: Kernel intensity estimates of loggerhead nesting along Juno Beach, 1998

and weeks 16-19 August. Although the kernel estimation curve in Figure 4.6 looks smoother, but peaks in the density estimate are vague in some weeks, e.g., week 6. The DPM model assumes an infinite mixture Gaussian distribution, and the model fitting is conducted independently among weeks. From Figure 4.8, the estimates for week 9 look similar to that in Figure 4.1, whilst there are several weeks having smoother estimates than kernel estimates, e.g., week 5, where peaks are more noticeable. Compared with the DPM model, the HDP model performs better because it allows different groups to share the same atoms of the mixture distribution. As a result, the estimates in Figure 4.9 more readily identify multi-modality. Moreover, some week's density estimates look smoother, but lose vital local information. For instance, in week 9, the HDP estimate has local peaks, but ignores temporal dependency by assuming exchangeability among week groups. To avoid this limitation, we apply the HDPA model to incorporate the temporal (first-order) autoregressive correlation in density estimation. From the results shown in Figure 4.10, we can see the impact of temporal dependency. For weeks 3-10, the peak in the southern (right) side of the beach stays quite stable, and only slight changes in density estimates appear on the northern (left) side of the beach. There exist tremendous change of pattern in week 11, and more nesting locations appear on the north side. Later on, the cluster pattern move to the middle, and thus back to the south side again. Within each plot, due to small changes of temperature or other environmental factors within each month, the shape of those density estimates looks similar, and only the magnitude varies. To better learn the difference in density estimation based on these four methods, we randomly choose four weeks, i.e., week 7, 9, 14, and 19, to compare their performances, and the results are presented in Figure 4.11. We see that the kernel estimates tend to smooth the estimate more, while DP models allow local detail. However, independent infinite countable clusters tend to raise more local peaks as in week 9, and it is not clear if these are appropriate. HDP models allow sharing information across groups, thus smoothing results; and HPA models not only share the same atoms across groups but also incorporate time-dependency, thus linking mixture components within the same month through the hierarchial structure.

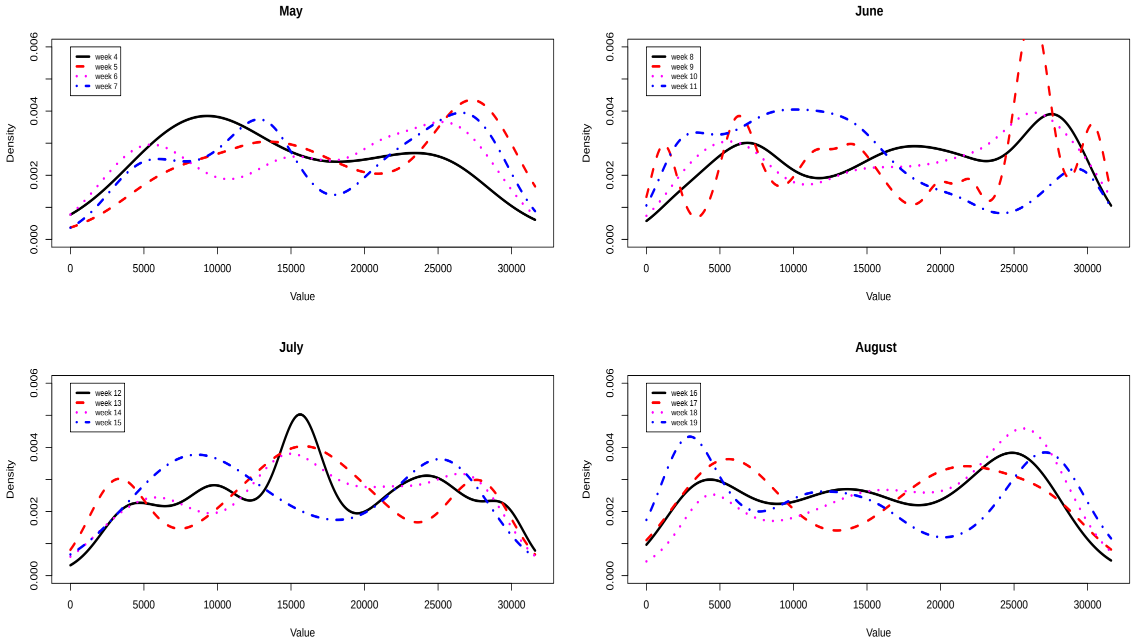


Figure 4.8: DPM density estimates of loggerhead nesting along Juno Beach, 1998

4.7 Conclusion and Discussion

We proposed a Bayesian non-parametric modeling formulation for density estimation by incorporating sequential temporal dependency and hierarchical structure, and compare our formulation with existing Bayesian non-parametric models as well as kernel methods via simulation and a data application. Simulation shows that HDPA models perform best in terms of detecting the mixture components accurately and provide the smallest KL divergence measures. We find that dHDP models do not perform as well as HDPA models without considering a higher hierarchical structure (month-level) although it performs comparably with HDPA models in some weeks. We also

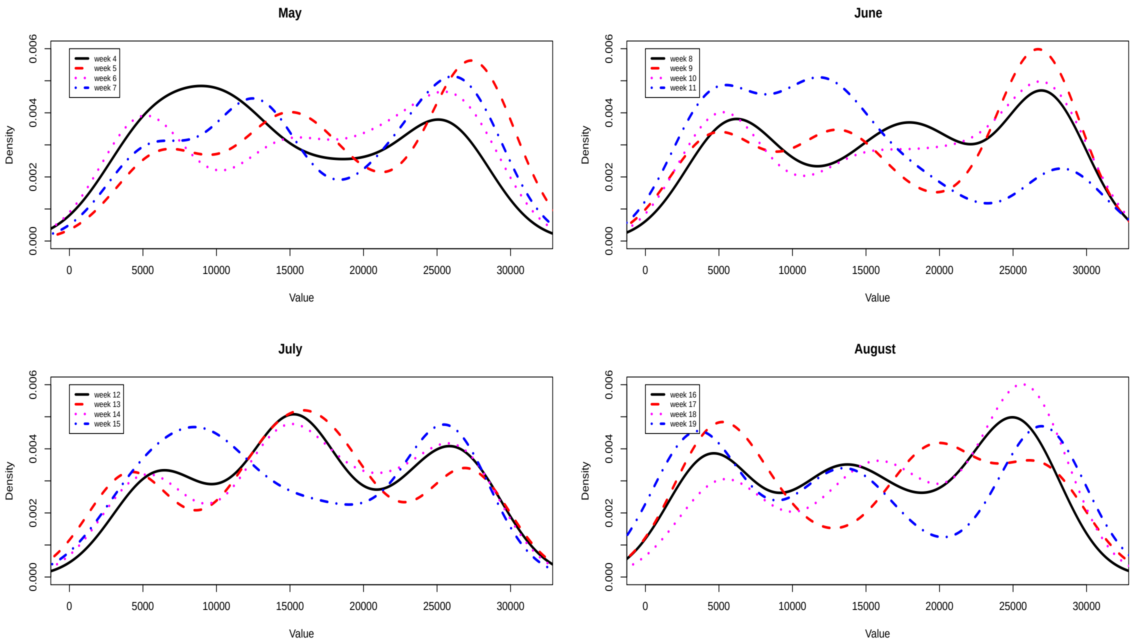


Figure 4.9: HDP intensity estimates of loggerhead nesting along Juno Beach, 1998

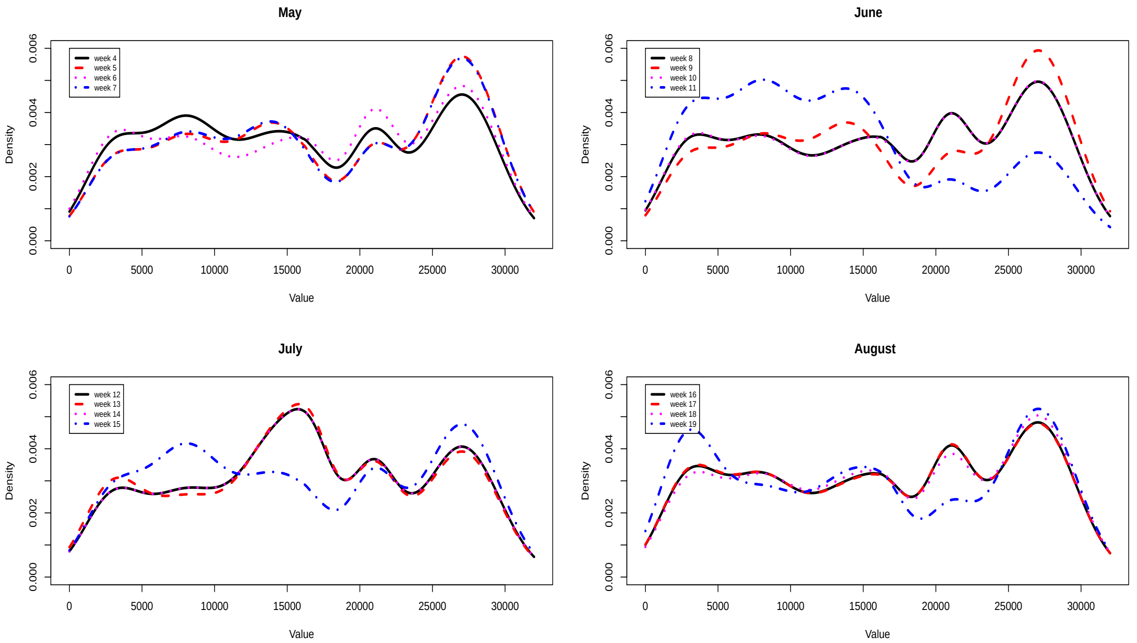


Figure 4.10: HDPA intensity estimates of loggerhead nesting along Juno Beach, 1998

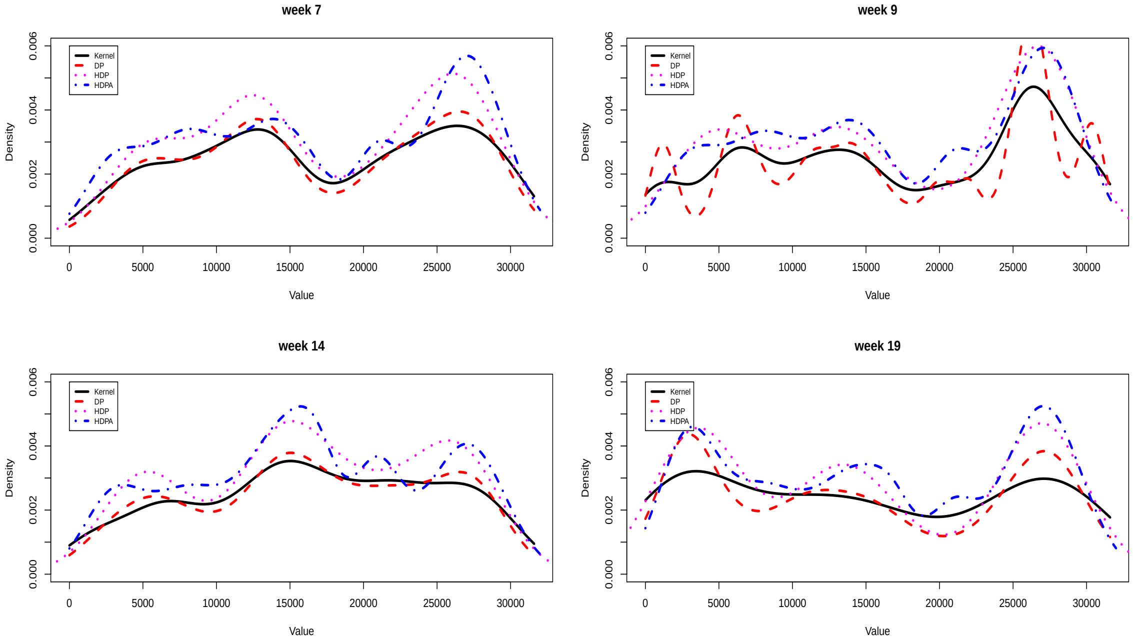


Figure 4.11: Comparing density estimates of loggerhead nesting for weeks 7, 9, 14, and 19

find HDP models gain better performance than either DPM models or kernel approaches by allowing groups share the same set of Gaussian atoms. Kernel methods are influenced by bandwidth selection, and do not retain flexibility as DPM models.

In our application, we estimate the density of loggerhead turtle (CC) nesting locations in the year 1998, allowing similar patterns between weeks within the same month via HDPA models, and provide a clearer picture of pattern change across the nesting season. We observe cluster patterns in the southern beach area in May to late June, and these move to the northern part in week 11, and then slowly back to the southern part in July, with little further change in August.

For future work, we may consider more complex simulation settings to further evaluate our proposed model's performance in the identification of evolutionary mixture components over time. We may consider mixture beta distributions to replace Gaussian distributions if we assume the data observation window is fixed. The non-parametric Bayesian approaches with Beta process prior could handle the boundary estimation issues, but may also involve considerable computational complexity. In addition, we may include temporally evolving mixture atoms to the model to gain generality and efficiency. All of those questions are of research interest and may be applied to other similar data applications.

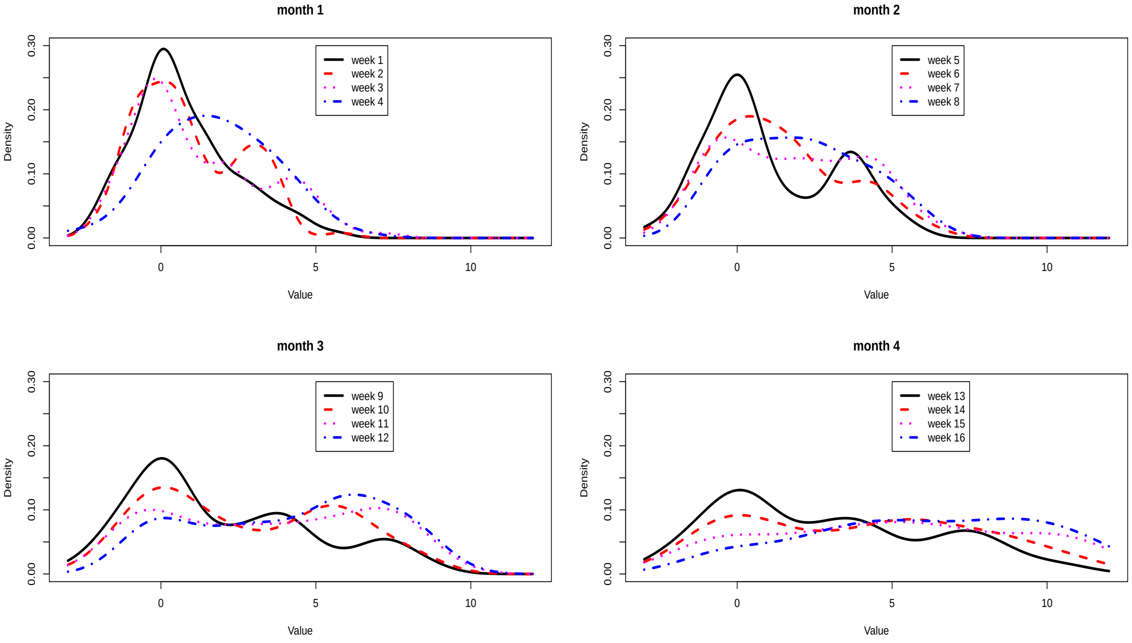


Figure 4.12: Kernel density estimates for each week by month in simulation

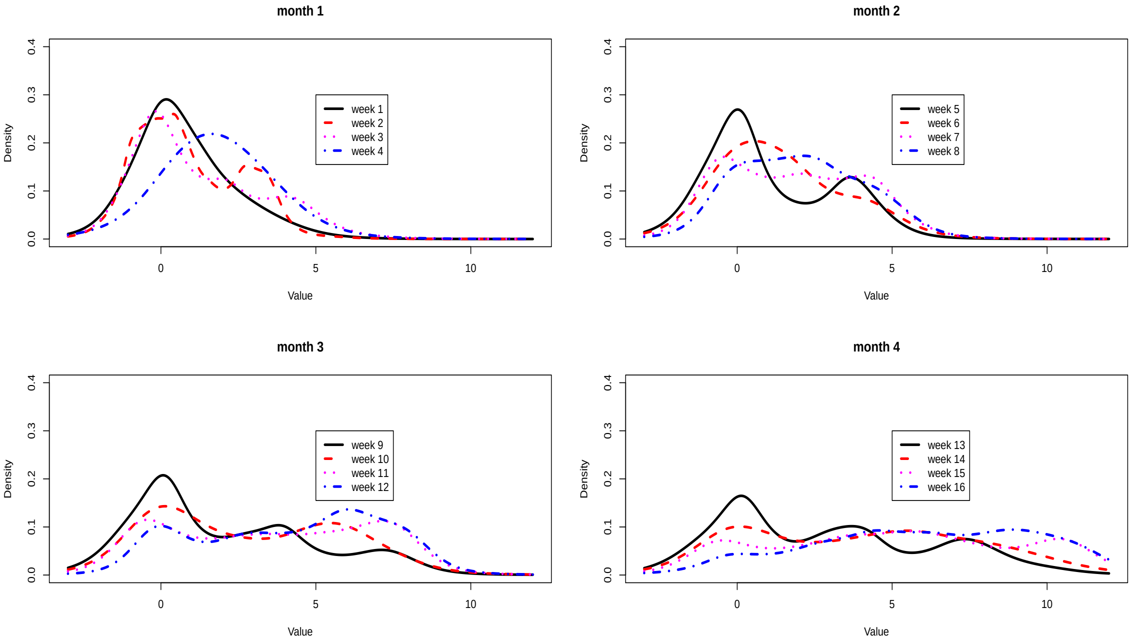


Figure 4.13: DPM density estimates for each week by month in simulation

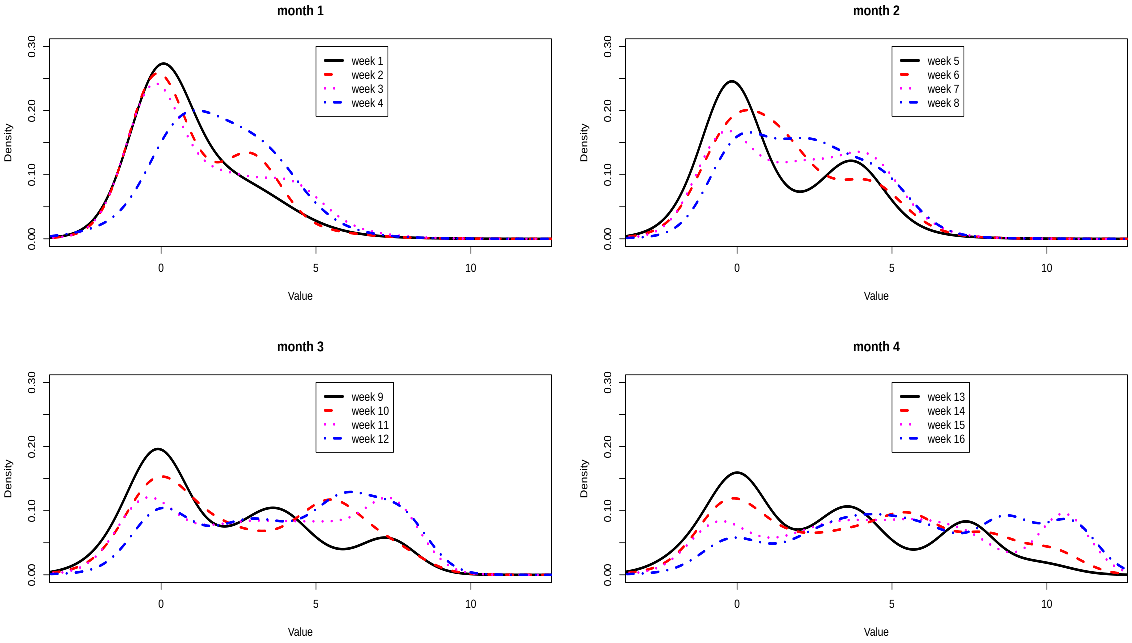


Figure 4.14: HDP density estimates for each week by month in simulation

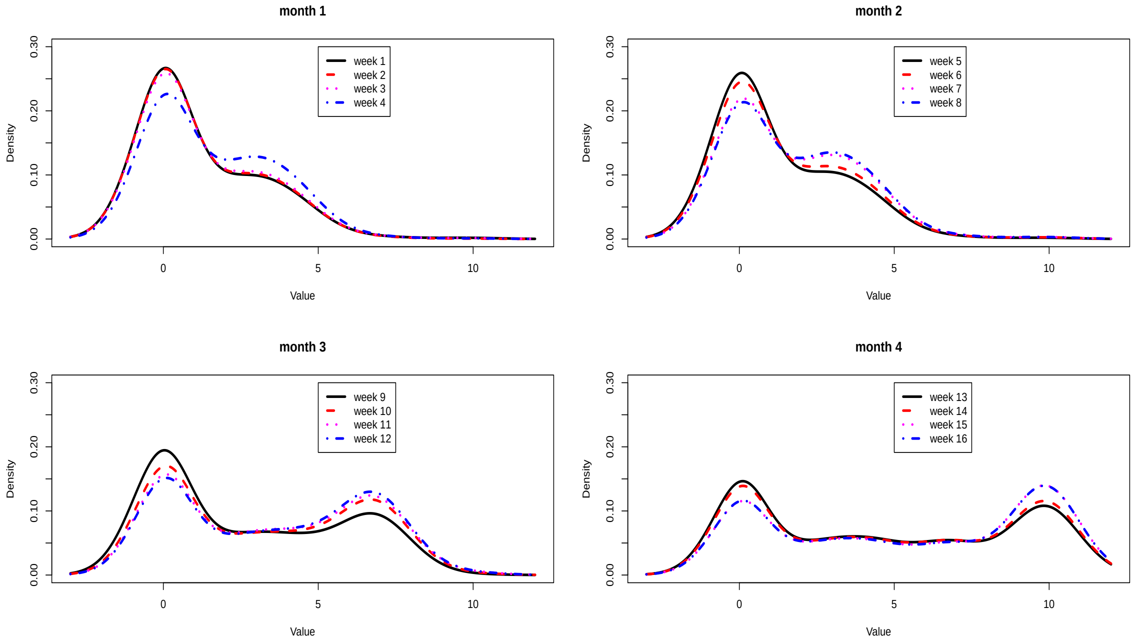


Figure 4.15: dHDP density estimates for each week by month in simulation

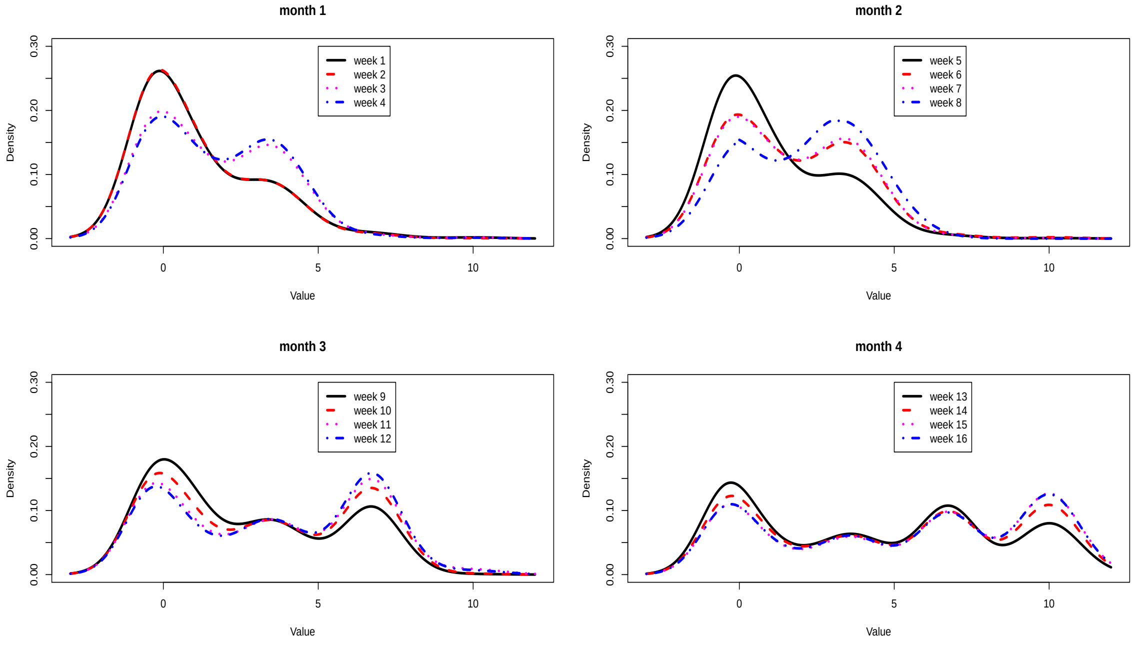


Figure 4.16: HDPA density estimates for each week by month in simulation

Chapter 5

Summary and future research

Spatial-temporal point processes have wide applications in epidemiology, social science, environmental health, neuroimaging analysis and so on. There exists a growing body of methods for detecting clusters and clustering patterns based on non-parametric or parametric approaches. Our current work proposes new methods and evaluates their performance for one-dimensional spatial(-temporal) point process analysis, including non-parametric estimation of $K-$function and PCF as well as algorithms for simulating realizations from different types of point processes.

Specifically, we have (i) evaluated non-parametric pair correlation function estimate for log-Gaussian Cox processes under infill asymptotics; (ii) analyzed spatial-temporal point patterns of sea turtle nesting locations; (iii) conducted Bayesian non-parametric modeling for density estimation by incorporating temporal dependency and hierarchical structure. Future work may include (i) derivation of asymptotic properties for non-parametric PCF estimates, especially for one-dimensional point processes; (ii) investigation of marked point process for sea turtle nesting data to explore species interaction; (iii) consideration of Dirichlet process models with mixture beta distributions to handle the fixed boundaries in real data applications.

Chapter 6

Appendix

6.1 R program for Chapter 2

```
####################################################################################
##########     Non-parametric estimates of spatial K-function             ##########
####################################################################################
### Note: data: the points observed along the line;
###       lag: the distance lag which corresponds to K(r);
###       Edge correction factor (inside matrixS): 1/2 based on Diggle's correction method
one_khat <- function(data, lag, status){
        label <- switch(status,
                  homo = 1,
                  inhomo = 2,
                  )
        pts <- data$points
        len <- length(pts)
        pairdis <- as.matrix(dist(pts))
        K.space <- c()
        for (k in 1:length(lag)) {
           C <- matrix(1, nrow=len, ncol=len)
           matrixS <- (pairdis<lag[k])*((pairdis>pts)*1+(pairdis>(max(pts) - pts))*1+C)
           if (label == 1){
               diag(matrixS) <- 0
               K.space[k] <- sum(matrixS)*max(pts)/(len*(len-1)) }
           else {
               space.ins <- data$lambda_s
               matrixS <- matrixS/matrix(kronecker(space.ins,space.ins), ncol=len, byrow=TRUE)
               diag(matrixS) <- 0
               K.space[k] <- sum(matrixS)/diff(range(pts)) }
        }
        return(list(lag = lag, k_est = K.space))
}
####################################################################################
##########     Non-parametric estimates of spatial-temporal K-function    ##########
####################################################################################
### Note: data: the points observed along the line and dates;
###       s: the distance lag which corresponds to K(s,t);
###       t: the temporal lag which corresponds to K(s,t);
###       Edge correction factor (inside matrixS and matrixT): 1/2 based on Diggle's correction method
```

```
one.stkhat <- function (data, s, tm, status) {
            label <- switch(status,
                   sep = 1,
                   nonsep = 2,
                   homo =3,
                   )
            pts <- data$points
            times <- data$date
            len <- length(pts)
            ### Read into the intensity estimates;
            if(!is.null(data$lambda_st)) { st.ins <- data$lambda_st }
            if(!is.null(data$lambda_s)) {space.ins <- data$lambda_s }
            if(!is.null(data$lambda_t)) {time.ins<- data$lambda_t }

            slimits <- range(pts)
            tlimits <- range(times)
            tlow <- min(tlimits)
            thigh <- max(tlimits)
            A <- diff(range(pts))
            T <- diff(range(times))

            K.st <- matrix(0,length(s),length(tm))
            K.space <- rep(0,length(s))
            K.time <- rep(0,length(tm))

            pairdis <- as.matrix(dist(pts))
            pairt <- as.matrix(dist(times))

            for (si in 1:length(s)) {
                for (ti in 1:length(tm)) {
                     C <- matrix(1, nrow=len, ncol=len)
                     matrixS <- (pairdis<s[si])*((pairdis>pts)*1+(pairdis>(max(pts) - pts))*1+C)
                     matrixT <- (pairt>0)*(pairt<tm[ti])*((pairt>=times)*1+(pairt>=(thigh - times))*1+C)
                     diag(matrixS) <- 0
                     diag(matrixT) <- 0

                  if (label == 1){
                     matrixS1 <- matrixS/matrix(kronecker(space.ins,space.ins), ncol=len, byrow=TRUE)
                     matrixT1 <- matrixT/matrix(kronecker(time.ins,time.ins), ncol=len, byrow=TRUE)
                     K.time[ti] <- sum(matrixT1)/T
                     K.space[si] <- sum(matrixS1)/A
                     matrixW <- matrixS1*matrixT1
                     K.st[si,ti] <- sum(matrixW)/(A*T)*len^2  }
                  else if (label ==2){
                     matrixS1 <- matrixS/matrix(kronecker(space.ins,space.ins), ncol=len, byrow=TRUE)
                     matrixT1 <- matrixT/matrix(kronecker(time.ins,time.ins), ncol=len, byrow=TRUE)
                     K.time[ti] <- sum(matrixT1)/T
                     K.space[si] <- sum(matrixS1)/A
                     matrixW <- matrixS*matrixT/matrix(kronecker(st.ins,st.ins), ncol=len, byrow=TRUE)
                     K.st[si,ti] <- sum(matrixW)/(A*T) }
                  else {
                     K.time[ti] <- sum(matrixT)*T/(len^2)
                     K.space[si] <- sum(matrixS)*A/(len^2)
                     matrixW <- matrixS*matrixT
                     K.st[si,ti] <- sum(matrixW)*(A*T)/(len^2) }
               } # ti
          } # si
    return(list(s_lag = s, t_lag = tm, ks = K.space, kt = K.time, kst = K.st))
}
######################################################################################
##########      Non-parametric estimates of pair correlation function       ##########
######################################################################################
### Note: data: the points observed along the line;
###       lag: the distance lag which corresponds to g(r);
###       Edge correction factor (matrixB): 1/2 based on Diggle's correction method
one.ghat <- function(data, lag, bw, max.win){
       pts <- data$points
       len <- length(pts)
       if (is.null(max.win)) {
           max.win <- max(pts)  }
       pairdis <- as.matrix(dist(pts))
       ## edge-correction matrix below
       matrixB <- (pairdis>pts)*1+(pairdis>(max.win - pts))*1+matrix(1, nrow=len, ncol=len)
       diag(matrixB) <- 0
       g.space <- c()
       for (k in 1:length(lag)) {
            matrixA <- 1/sqrt(2*pi*bw)*exp(-((lag[k]-pairdis)/bw)^2/2)
            diag(matrixA) <- 0
            g.space[k] <- sum(matrixA*matrixB)*max.win/(2*len*(len-1))
       }
       return(list(lag=lag, g_est=g.space))
}
####################################################################################
##########     Non-parametric estimates of pair correlation function      ##########
####################################################################################
### Note: data: the points observed along the line;
###       lag: the distance lag which corresponds to g(r);
###       Edge correction factor (inside matrixS): 1/2 based on Diggle’s correction method
one.stghat <- function(data, lag, bw, max.win){
       unique.date_subset <- unique(data$date)
       if (is.null(max.win)) {
           max.win <- max(data$points) }
       maxt_subset <- diff(range(data$date))

       # Get the summary;
       g_est <- c()
       for (si in 1:length(lag)){
           total <- c()
       for (i in unique.date_subset){
           data_sub <- data[data$date==i,]
           pts <- data_sub$distance
           space.ins <- data_sub$lambda_s
           time.ins <- data_sub$lambda_t[1]
           len <- length(pts)
           pairdis <- as.matrix(dist(pts))
           matrixA <- 1/sqrt(2*pi*bw)*exp(-((lag[si]-pairdis)/bw)^2/2)
           matrixS <- ((pairdis>pts)*1+(pairdis>(max.win - pts))*1+matrix(1, nrow=len, ncol=len))
           diag(matrixA) <- 0
           diag(matrixS) <- 0
           matrixW <- matrixA*matrixS
           for (k in 1:len){
                for (j in 1:len){
                   matrixW[k,j] <- matrixW[k,j]/(space.ins[k]*space.ins[j]*(time.ins)^2)
                } #j
           } #k
         total <- c(total, sum(matrixW))
         }
       g_est[si] <- sum(total)/(2*max.win*maxt_subset)
       }
       return(list(lag=lag, gst_est=g_est, total=total))
}
####################################################################################
##########    The existing parametric forms of pair correlation function  ##########
####################################################################################
### Note: u: distance lags
pcf_para <- function(u, sigma, phi, model, additionalparameters){
      if (model=="exponential") { g_fit <- exp(sigma*exp(-(u/phi))) }
      else if (model=="gaussian") { g_fit <- exp(sigma*exp(-(u/phi)^{2})) }
      else if (model=="cardinal") { g_fit <- exp(sigma*(sin(u/phi)/(u/phi)))  }
      else if (model=="stable") { g_fit <- exp(sigma*exp(-sqrt(u/phi)))  }
      else if (model=="hyperbolic") { g_fit <- exp(sigma/(1+u/phi))  }
      else if (model=="bessel1") { g_fit <- exp(sigma*exp(-(u/phi)^{2})*besselJ(u/phi, 0))  }
      else if (model=="bessel2") { g_fit <- exp(sigma*exp(-(u/phi)^{2})*besselJ(u/phi, 1))  }
      else if (model=="spherical") {
         index <- as.numeric(u/beta<1)
         g_fit <- ifelse(index==1, exp(sigma*((u/phi)^3/2+1-(3*u/(2*phi)))), 1)  }
      else if (model=="card2") {
         g_fit <- exp(sigma*(sin(u/(beta*phi))/(u/(beta*phi))))  }
      return(g_fit=g_fit)
 }
```

6.2 R program for Chapter 3

```
############################################################################################
########## Simulate One-dimensional Spatial-temporal Homogeneous Poisson Process  ##########
############################################################################################
### Note: lambda: a constant (fixed) value;
###       npoints: the total number of points to be generated;
###       nsim: the number of realizations to be generated.
One_rpp_homo <- function (lambda, s.region, t.region, npoints = NULL, nsim, replace = TRUE)
{
    if (missing(s.region))
        s.region <- c(0, 1)
    if (missing(t.region))
        t.region <- c(1, 30)
    s.region <- sort(s.region)
    t.region <- sort(t.region)
    s.area <- s.region[2] - s.region[1]
    t.area <- t.region[2] - t.region[1]
    pattern <- list()
    index.t <- list()
    ni <- 1
    if (is.numeric(lambda) & length(lambda) == 1) {
       while (ni <= nsim) {
          if (is.null(npoints)==TRUE) {
             if (t.area==0)
                { npoints <- round(rpois(n=1,lambda=lambda * s.area),0) }
             else
                { npoints <- round(rpois(n=1,lambda=lambda * s.area * t.area),0) }
          }
          x <- runif(npoints, min=s.region[1], max=s.region[2])
          times <- runif(npoints,min=t.region[1],max=t.region[2])
          samp <- sample(1:npoints,npoints,replace=replace)
          times <- times[samp]
          times <- sort(times)
          index.times <- sort(samp)
          pattern.interm <- list(x=x,t=times,index.t=index.times)
          pattern[[ni]] <- cbind(x=pattern.interm$x,t=pattern.interm$t)
          index.t[[ni]] <- pattern.interm$index.t
          ni <- ni+1
       }
    }
    else stop("lambda must be a single positive value.")
    invisible(return(list(xt = pattern, index.t = index.t, lambda = lambda)))
}
##############################################################################################
########## Simulate One-dimensional Spatial-temporal Inhomogeneous Poisson Process  ##########
##############################################################################################
### Note: lambda: a function of space and time or an array;
###       npoints: the total number of points to be generated;
###       nsim: the number of realizations to be generated.
One_rpp_inhomo <- function (lambda, s.region, t.region, npoints, nsim, replace, lmax, nx=100, nt=100)
{
    if (missing(s.region))
        s.region <- c(0, 1)
```

```
if (missing(t.region))
    t.region <- c(1, 30)
s.region <- sort(s.region)
t.region <- sort(t.region)
s.area <- s.region[2] - s.region[1]
t.area <- t.region[2] - t.region[1]
lambdamax <- lmax
pattern <- list()
index.t <- list()
if (is.function(lambda)) {
     xrang <- range(s.region, na.rm = TRUE)
     xmin <- xrang[1]
     xmax <- xrang[2]
     xinc <- (xmax-xmin)/nx
     xc <- xmin-xinc/2
     xgrid <- rep(0,nx)
     xgrid[1] <- xc + xinc
     for (i in 2:nx) { xgrid[i] <- xgrid[i-1]+xinc }
     tinc <- t.area/(nt - 1)
     t.grid <- list(times = seq(floor(t.region[1]), ceiling(t.region[2]),length = nt), tinc = tinc)
     Lambda <- array(NaN, dim = c(nx, nt))
     for (it in 1:nt) {
         L <- lambda(xgrid, t.grid$times[it])
         Lambda[, it] <- L }
     ni <- 1
     while (ni <= nsim) {
            if (is.null(npoints)) {
             en <- sum(Lambda,na.rm=TRUE)*xinc*t.grid$tinc
               npoints <- round(rpois(n=1,lambda=en),0) }
            if (is.null(lambdamax)) { lambdamax <- max(Lambda,na.rm=TRUE) }
            npts <- round(lambdamax*s.area*t.area,0)
            x <- runif(npts, min=s.region[1], max=s.region[2])
            times.init <- runif(nt,min=t.region[1],max=t.region[2])
            samp <- sample(1:nt,npts,replace=replace)
            times <- times.init[samp]
            prob <-  lambda(x,times)/lambdamax
            u <- runif(npts)
            retain <- u <= prob
            if (sum(retain==FALSE)==length(retain)) {
               lambdas <- rep(0,nx)
               for(ix in 1:nx){
                  lambdas[ix] <- median(Lambda[ix,],na.rm=TRUE) }
               lambdamax <- max(lambdas,na.rm=TRUE)
               prob <-  lambda(x,times)/lambdamax
               retain <- u <= prob
               if (sum(retain==F)==length(retain)) stop ("Please check the parameters")
            }
            x <- x[retain]
            samp <- samp[retain]
            samp.remain <- (1:nt)[-samp]
            times <- times[retain]
            neffec <- length(x)
            while(neffec < npoints) {
                x2 <- runif(npoints-neffec, min=s.region[1], max=s.region[2])
                wx <- x2
               if(replace==FALSE)
                 { wsamp <- sample(samp.remain,npoints-neffec,replace=replace)
               } else
               { wsamp <- sample(1:nt,npoints-neffec,replace=replace) }
               wtimes <- times.init[wsamp]
               prob <-  lambda(wx,wtimes)/lambdamax
               u <- runif(npoints-neffec)
               retain <- u <= prob
               x <- c(x,wx[retain])
               times <- c(times,wtimes[retain])
               samp <- c(samp,wsamp[retain])
               samp.remain <- (1:nt)[-samp]
               neffec <- length(x)
            }
            interm <- data.frame(x=x, times=times, index.times=samp)
            interm <- interm[order(interm$t),]
            pattern.interm <- list(x=interm$x,t=interm$times,index.t=interm$index.times)
            pattern[[ni]] <- cbind(x=pattern.interm$x,t=pattern.interm$t)
            index.t[[ni]] <- pattern.interm$index.t
            ni <- ni+1
        }
}
else if (is.array(lambda)) {
    if (length(dim(lambda)) != 2)
        stop("lambda must be a 2D-array")
    Lambda = lambda
    nx <- dim(Lambda)[1]
    nt <- dim(Lambda)[2]
    xrang <- range(s.region, na.rm = TRUE)
    xmin <- xrang[1]
    xmax <- xrang[2]
    xinc <- (xmax-xmin)/nx
    xc <- xmin-xinc/2
    xgrid <- rep(0,nx)
    xgrid[1] <- xc + xinc
    for (i in 2:nx) { xgrid[i] <- xgrid[i-1]+xinc }
    tinc <- t.area/(nt - 1)
    t.grid <- list(times = seq(floor(t.region[1]), ceiling(t.region[2]),length = nt), tinc = tinc)
    ni <- 1
    while (ni <= nsim) {
            if (is.null(npoints)) {
             en <- sum(Lambda,na.rm=TRUE)*xinc*t.grid$tinc
               npoints <- round(rpois(n=1,lambda=en),0) }
            if (is.null(lambdamax)) { lambdamax <- max(Lambda,na.rm=TRUE) }
            npts <- round(lambdamax*s.area*t.area,0)
            times.init <- runif(nt,min=t.region[1],max=t.region[2])
            samp <- sample(1:nt,npts,replace=replace)
            times <- times.init[samp]
            retain.eq.F <- FALSE
            while(retain.eq.F==FALSE) {
            x <- runif(npts, min=s.region[1], max=s.region[2])
            prob <- NULL
            for(dx in 1:length(x)) {
               nix <- findInterval(vec=xgrid,x=x[dx])
               nit <- findInterval(vec=t.grid$times,x=times[dx])
           if (nix==0 | nit==0) {
            prob=c(prob,NA)
           } else {
                  prob <- c(prob,Lambda[nix,nit]/lambdamax) }
             }
             M <- which(is.na(prob))
             if (length(M)!=0) {
                x <- x[-M]
                times <- times[-M]
                samp <- samp[-M]
                prob <- prob[-M]
                npts <- length(x)
              }
             u <- runif(npts)
             retain <- u <= prob
             if (sum(retain==F)==length(retain)) retain.eq.F <- FALSE
             else retain.eq.F <- TRUE
            }
           if (sum(retain==FALSE)==length(retain)) {
               lambdas <- rep(0,nx)
               for(ix in 1:nx){
                  lambdas[ix] <- median(Lambda[ix,],na.rm=TRUE) }
```

```
                   lambdamax <- max(lambdas,na.rm=TRUE)
                   prob <-  lambda(x,times)/lambdamax
                   retain <- u <= prob
                   if (sum(retain==F)==length(retain)) stop ("Please check the parameters")
                }
                x <- x[retain]
                samp <- samp[retain]
                samp.remain <- (1:nt)[-samp]
                times <- times[retain]
                neffec <- length(x)
                while(neffec < npoints) {
                    x2 <- runif(npoints-neffec, min=s.region[1], max=s.region[2])
                    wx <- x2
                   if(replace==FALSE)
                     { wsamp <- sample(samp.remain,npoints-neffec,replace=replace)
                   } else
                   { wsamp <- sample(1:nt,npoints-neffec,replace=replace) }
                   wtimes <- times.init[wsamp]
                   prob <- NULL
                   for(nx in 1:length(wx)) {
                       nix <- findInterval(vec=xgrid,x=wx[nx])
                       nit <- findInterval(vec=t.grid$times,x=wtimes[nx])
               if (nix==0 | nit==0)
                { prob=c(prob,NA)
               } else
                     { prob <- c(prob,Lambda[nix,nit]/lambdamax) }
                   }
                   M <- which(is.na(prob))
                   if (length(M)!=0) {
                      wx <- wx[-M]
                      wsamp <- wsamp[-M]
                      wtimes <- wtimes[-M]
                      prob <- prob[-M]
                   }
                    if (neffec > 0) {
                       u <- runif(length(prob))
                       retain <- u <= prob
                       x <- c(x,wx[retain])
                       times <- c(times,wtimes[retain])
                       samp <- c(samp,wsamp[retain])
                       samp.remain <- (1:nt)[-samp]
                       neffec <- length(x)
                    }
                }
                interm <- data.frame(x=x, times=times, index.times=samp)
                interm <- interm[order(interm$t),]
                pattern.interm <- list(x=interm$x,t=interm$times,index.t=interm$index.times)
                pattern[[ni]] <- cbind(x=pattern.interm$x,t=pattern.interm$t)
                index.t[[ni]] <- pattern.interm$index.t
                ni <- ni+1
            }
    } else
      stop("lambda must be a function or an array.")
    invisible(return(list(xt = pattern, index.t = index.t, lambda = lambda)))
}
#########################################################################################
########## Simulate One-dimensional Spatial-temporal Log-Gaussian Cox Process  ##########
#########################################################################################
### Note: the function GaussRF is used to generate Gaussian process (RandomFields library);
###       npoints: the total number of points to be generated;
###       nsim: the number of realizations to be generated.
One_rlgcp <- function(s.region, t.region, replace, npoints, nsim, nx=100, nt=100, model,lmax)
{
  if (missing(s.region)) s.region <- c(0,1)
  if (missing(t.region)) t.region <- c(1,30)
```

```
s.region <- sort(s.region)
t.region <- sort(t.region)
s.area <- s.region[2] - s.region[1]
t.area <- t.region[2] - t.region[1]
tau <- c(start=t.region[1],end=t.region[2],step=(t.region[2]-t.region[1])/(nt-1))

lambdamax <- lmax
pattern <- list()
index.t <- list()
Lambdafin <- list()
ni <- 1

xrang <- range(s.region, na.rm = TRUE)
xmin <- xrang[1]
xmax <- xrang[2]
xinc <- (xmax-xmin)/nx
xc <- xmin-xinc/2
xgrid <- rep(0,nx)
xgrid[1] <- xc + xinc
for (i in 2:nx) { xgrid[i] <- xgrid[i-1]+xinc }
t.grid <- list(times=seq(floor(t.region[1]), ceiling(t.region[2]),length=nt), tinc=t.area/(nt-1))

while(ni<=nsim)
  {
    S <- GaussRF(x=xgrid, T=c(t.region, 1), grid=TRUE, model = model) #Please specify your own model;
    Lambda <- exp(S)
    mut <- rep(0,nt)
    for (it in 1:nt) {
        mut[it] <- sum(Lambda[,it],na.rm=TRUE) }

   if (is.null(npoints)) {
       en <- sum(Lambda,na.rm=TRUE)*xinc*t.grid$tinc
       npoints <- round(rpois(n=1,lambda=en),0) }
       if (is.null(lambdamax)) { lambdamax <- max(Lambda,na.rm=TRUE) }
   npts <- round(lambdamax*s.area*t.area,0)
   if (npts==0) stop("there is no data to thin")
   times.init <- runif(nt,min=t.region[1],max=t.region[2])
   samp <- sample(1:nt,npts,replace=replace,prob=mut/max(mut,na.rm=TRUE))
   times <- times.init[samp]

   retain.eq.F <- FALSE
   while(retain.eq.F==FALSE) {
        x <- runif(npts, min=s.region[1], max=s.region[2])
        prob <- NULL
        for(njx in 1:length(x)) {
            nix <- findInterval(vec=xgrid,x=x[njx])
            nit <- findInterval(vec=t.grid$times,x=times[njx])
            if (nix==0 | nit==0)
    prob=c(prob,NA)
    else
                prob <- c(prob,Lambda[nix,nit]/lambdamax)
        }
        M <- which(is.na(prob))
        if (length(M)!=0) {
             x <- x[-M]
             times <- times[-M]
             samp <- samp[-M]
             prob <- prob[-M]
             npts <- length(x)
        }
        u <- runif(npts)
        retain <- u <= prob
        if (sum(retain==F)==length(retain))  retain.eq.F <- FALSE
        else  retain.eq.F <- TRUE
   }
   x <- x[retain]
```

```
     samp <- samp[retain]
     samp.remain <- (1:nt)[-samp]
     times <- times[retain]
     neffec <- length(x)
     if (neffec > npoints)
        {
          retain <- 1:npoints
          x <- x[retain]
          samp <- samp[retain]
          samp.remain <- (1:nt)[-samp]
          times <- times[retain]
        }
     while(neffec < npoints) {
          x2 <- runif(npoints-neffec, min=s.region[1], max=s.region[2])
          wx <- x2
          if (isTRUE(replace)) {
            wsamp <- sample(1:nt,npoints-neffec,replace=replace,prob=mut/max(mut,na.rm=TRUE))
          } else
          { prob <- mut[samp.remain]/max(mut[samp.remain]
            wsamp <- sample(samp.remain,npoints-neffec,replace=replace,prob=prob,na.rm=TRUE)) }
          wtimes <- times.init[wsamp]
          prob <- NULL
          for(nx in 1:length(wx)) {
              nix <- findInterval(vec=xgrid,x=wx[nx])
              nit <- findInterval(vec=t.grid$times,x=wtimes[nx])
              if (nix==0 | nit==0)
                prob=c(prob,NA)
               else
                      prob <- c(prob,Lambda[nix,nit]/lambdamax)
          }
          M <- which(is.na(prob))
          if (length(M)!=0) {
              wx <- wx[-M]
              wtimes <- wtimes[-M]
              wsamp <- wsamp[-M]
              prob <- prob[-M]
         }
         if (neffec > 0) {
             u <- runif(length(prob))
             retain <- u <= prob
             x <- c(x,wx[retain])
             times <- c(times,wtimes[retain])
             samp <- c(samp,wsamp[retain])
             samp.remain <- (1:nt)[-samp]
             neffec <- length(x)
         }
     }
     interm <- data.frame(x=x, times=times, index.times=samp)
     interm <- interm[order(interm$t),]
     pattern.interm <- list(x=interm$x,t=interm$times,index.t=interm$index.times)
     pattern[[ni]] <- cbind(x=pattern.interm$x,t=pattern.interm$t)
     index.t[[ni]] <- pattern.interm$index.t
     ni <- ni+1
  }
  invisible(return(list(xt=pattern,Lambda=Lambdafin,index.t=index.t)))
}
```

Bibliography

Antworth, R.L., Pike, D.A. and Stiner, J.C. (2006) Nesting ecology, current status, and conservation of sea turtles on an uninhabited beach in Florida, USA. *Biological Conservation* **130**: 10-15.

Baddeley, A. (2010) Analysing spatial point patterns in "R". *Workshop notes. Version 4.1. CSIRO online technical publication. URL: www.csiro.au/resources/pf16h.html.*

Baddeley, A., Møller, J. and Waagepetersen, R. (2000) Non- and semi-semiparametric estimation of interaction in inhomogeneous point patterns. *Statistica Neerlandica* **54(3)**: 329-350.

Baddeley, A. and Turner, R. (2000) Practical maximum pseudolikelihood for spatial point patterns. *Australian and New Zealand Journal of Statistics* **42(3)**: 283-322.

Baddeley, A. and Turner, R. (2005) Spatstat: an R package for analyzing spatial point patterns. *Journal of Statistical Software* **12(6)**: 1-42; URL: www.jstatsoft.org, ISSN: 1548-7660.

Barnard, G.A. (1963) Comment on "The spectral analysis of point processes" by M. S. Barlett. *Journal of the Royal Statistical Society Series B* **25**: 294.

Berman, M. and Diggle, P. (1989) Estimating weighted integrals of the second-order intensity of a spatial point process. *Journal Of The Royal Statistical Society B* **51**: 81-92.

Besag, J.E. (1977) Comments on Ripley's paper. *Journal of the Royal Statistical Society B* **39**: 193-195.

Besag, J.E. and Diggle, P. (1977) Simple Monte Carlo tests for spatial pattern. *Applied Statistics* **26(3)**: 327-333.

Brix, A. (1999) Generalized gamma mesures and shot-noise Cox processes. *Advances in Applied Probability* **31(4)**: 929-953.

Brix, A. and Diggle, P. (2001) Spatio-temporal prediction for log-Gaussian Cox processes. *Journal of Royal Statistical Society B* **63(4)**: 823-841.

Brix, A. and Møller, J. (2001) Space-time multitype log Gaussian Cox processes with a view to modelling weed data. *Scandinavian Journal of Statistics* **28(3)**: 471-488.

Caron, F., Davy, M., Doucet, A., Duflos, E. and Vanheeghe, P. (2008) Bayesian inference for linear dynamic models with Dirichlet process mixtures. *IEEE Transactions on Singnal Processing* **56(1)**: 71-84.

Chung, Y. and Dunson, D.B. (2011) The local Dirichlet process. *Annals of the Institute of Statistical Mathematics* **63**: 59-80.

Cox, D.R. (1972) The statistical analysis of dependencies in point processes. *In P. A. W. Lewis, editor, Stochastic Point Processes.* Wiley, New York.

Christakos, G. (1992) Random field models in earth sciences. *New York: Academic press.*

Clemencon, S. (2000) Adaptive estimation of the transition density of a regular Markov chain. *Math. Methods Statist* **9(4)**: 323-357.

Daley, D.J. and Vere-Jones, D. (2003) An introduction to the theory of point processes. *Springer.*

Diggle, P. (1985) A kernel method for smoothing point process data. *Applied Statistics* **34(2)**: 138-147.

Diggle, P. (2003) Statistical analysis of spatial point patterns. *Oxford University Press Inc., New York.*

Diggle, P., Rowlingson, B. and Su, T.L. (2005) Point process methodology for online spatio-temporal disease surveillance. *Environmetrics* **16(5)**: 423-434.

Doguwa, S.I. (1990) On edge-corrected Kernel-based pair correlation function estimators for point processes. *Biometrical Journal* **32(1)**: 95-106.

Doss, H. (1989) On estimating the dependence between two point processes. *The Annals of Statistics* **17(2)**:749-763.

Dunson, D.B. (2006) Bayesian dynamic modeling of latent trait distributions. *Biostatistics* **7(4)**: 551-568.

Escobar, M. and West, M. (1994) Bayesian density estimation and inference using mixtures. *Journal of American Statistical Association* **90**: 577-588.

Fan, J., Yao, Q. and Tong, H. (1996) Estimation of conditional densities and sensitivity measure in nonlinear dynamical systems. *Biometrika* **83(1)**: 189-206.

Fiksel, T. (1988) Edge-Corrected density estimators for point processes. *Statistics* **19(1)**: 67-75.

Gabriel, E. and Diggle, P. (2009) Second-order analysis of inhomogeneous spatio-temporal point process data. *Statistica Neerlandica* **63(1)**: 43-51.

Gatrell, A.C., Bailey, T.C., Diggle, P.J. and Rowlingson, B.S. (1996) Spatial point pattern analysis and its application in geographical epidemiology. *Transactions of the Institute of British Geographers* **21(1)**: 256-274.

Geyer, C.J. (1999) Likelihood inference for spatial point processes. *In Stochastic Geometry: Liklihood and Computaiton*; *London: Chapman and Hall.*

Geyer, C.J. and Møller, J. (1994) Simulation procedures and likelihood inference for spatial point processes. *Scandinavian Journal of Statistics* **21**: 359-373.

Geyer, C.J. and Thompson, E.A. (1992). *Journal Of The Royal Statistical Society B* **54(3)**: 657-699.

Gneiting, T. (2002) Nonseparable, stationary covariance function for space-time data. *Journal of American Statistical Association* **97 (458)**: 590-600.

Goreaud, F. and Pelissier, R. (1999) On explicit formulas of edge effect correction for Ripley's $K-$function. *Journal of Vegetation Science* **10**: 433-438.

Griffin, J.E. and Steel, M.F.J. (2006) Order-based dependent Dirichlet processes. *Journal of the American Statistical Association* **101(473)**: 179-194.

Griffin, J.E. and and Steel, M.F.J. (2009) Time-dependent stick-breaking processes. *Working paper. Univeristy of Warwick. Center for Research in Statitical Methodology, Coventry.*

Gu, C. and Wang, J. (2003) Penalized likelihood density estimation: direct cross-validation and scalable approximation. *Statist. Sinica* **13**: 811-826.

Gu, M.G. and Zhu, H.T. (2001) Maximum likelihood estimation for spatial models by Markov chain Monte Carlo stochastic approximation. *Journal Of The Royal Statistical Society B* **63(2)**: 339-355.

Guan, Y. (2008) On consistent nonparametric intensity estimation for inhomogeneous spatial point processes. *Journal of the American Statistical Association* **103(483)**: 1238-1247.

Guan, Y., Waagepetersen, R. and Beale, C.M. (2010) Nonparametric and parametric estimation of the second-order structure of inhomogeneous spatial point processes using case-control data. *Techinical report.*

Lahiri, S.N. (1996) On inconsistency of estimators based on spatial data under infill asymptotics. *Sankhya: the Indian Journal of Statistics* **58(3)**: 403-417.

Lawson, A. (2003) Issues in the space-time analysis of public health surveillance data in monitoring the health of populations: Statistical methods for public health surveillance. *eds Brookmeyer, R. and Stroup, D. Oxford University Press.*

Møller, J. and Ghorbani, M. (2010) Second-order analysis of structrued inhomogeneous spatio-temporal point processes. *CSGB Research Reports.*

Møller, J., Syversveen, A.R. and Waagepetersen, R. (1997) Log Gaussian Cox Process: A statistical model for analyzing stand strcuture heterogeneity in forestry. *Proceedings of first European Conference for Information Technology in Agriculture*; 339-342.

Møller, J., Syversveen, A.R. and Waagepetersen, R. (1998) Log Gaussian Cox Process. *Scandinavian Journal of Statistics*;**25(3)**: 451-482.

Møller, J. and Waagepetersen, R. (2003) An introduction to simulation-based inference for spatial point processes. *Spatial Statistics and Compuational Methods* **173**: 143-198.

Møller, J. and Waagepetersen, R. (2004) Statistical inference and simulation for spatial point processes. *Chapman and Hall/CRC.*

Ohser, J. (1983) On estimators for the reduced second moment measure of point processes. *Math. Operationsforsch. Statist., Series Statistics* **14(1)**: 63-71.

Ohser, J. and Stoyan, D. (1981) On the second-order and Orientation analysis of planar stationary point processes. *Biometrical Journal* **23(6)**: 523-533.

Penttinen, A. (1984) Modelling interactions in spatial point patterns: parameter estimation by the maximum likelihood method. *Jyvaskyla Studies in Computer Science, Economics and Statistics* **7**.

Penttinen, A. and Stoyan, D. (2000) Recent applications of point process methods in forestry statistics. *Statistical Science* **15(1)**: 61-78.

Ren, L., Dunson, D. B. and Carin, L. (2008) The dynamic hierarchical Dirichlet process. *ICML 2008*: 768-775.

Rodriguez, A. and Horst, E.T. (2008) Bayesian dynamic density estimation. *Bayeisan Analysis* **3(2)**: 339-366.

Rosenblatt, M. (1969) Conditional probability density and regression estimates. *Multivariate Analysis II. Ed. P.R. Krishnaiah. New York: Academic Press*; 25-31.

Richardson, S. and Green, P.J. (1997) On Bayesian analysis of mixtures with an unknown number of components. *Journal of Royal Statistical Society B* **59(4)**: 731-792.

Ripley, B.D. (1976) The second-order analysis of stationary point processes. *Journal of Applied Probability* **13**: 255-266.

Ripley, B.D. (1977) Modelling spatial patterns. *Journal of the Royal Statistical Society B* **39(2)**: 172-212.

Ripley, B.D. (1981) Spatial statistics. *Wiley, New York.*

Ripley, B.D. (1982) Edge effects in spatial stochastic processes. *In: Ranneby, B ed. Statistic in theory and practice: Umea: Swedish University of Agricultural Sciences* **39(2)**: 242-262.

Ripley, B.D. (1988) Statistical inference for spatial processes. *Cambridge University Press..*

Sheather, S.J. and Jones, M.C. (1991) A reliable data-based bandwidth selection method for kernel density estimation. *Journal Of The Royal Statistical Society B* **53(3)**: 683-690.

Stoyan, D., Bertram, U. and Wendrock, H. (1993) Estimation variances for estimators of product densities and pair correlation functions of planar point processes. *Journal Annals of the Institute of Statistical Mathematics* **45(2)**: 211-221.

Stoyan, D. and Stoyan, H. (1994) Fractals, random shapes and point fields: methods of geometrical statistics. *New York: John Wiley and Sons.*

Stoyan, D. and Stoyan, H. (1996) Estimating pair correlation functions of planar cluster processes. *Biometrical Journal* **38(3)**: 259-271.

Tang, Y.Q. and Ghosal, S. (2007) A consistent nonparametric Bayesian procedure for estimating autoregressive conditional densities. *Computational Statistics and Data Analysis* **51(9)**: 4424-4437.

Teh, Y.W., Jordan, M.I., Beal, M.J. and Blei, D.M. (2006) Hierarchical Dirichlet Processes. *Journal of the American Statistical Association* **101(476)**: 1566-1581.

Waagepetersen, R. (2007) An estimating function approach to inference for inhomogeneous Neyman-Scott processes. *Biometrics***63(1)**: 252-258.

Waagepetersen, R. and Guan, Y. (2009) Two-step estimation for inhomogeneous spatial point processes. *Journal Of The Royal Statistical Society Series B* **71(3)**: 685-702.

Waller, L.A. and Gotway, C.A. (2004) Applied Spatial Statistics for Public Health Data. *New York: John Wiley and Sons.*

Waller, L.A. and Leong, T. (2007) Evaluating local spatial nesting impacts of a beach nourishment project, Juno Beach, Florida, 1999-2002. *Poster Presentation. Sea Turtle Symposium.*

Weishampel, J.F., Bagley, D.A., Ehrhart, L.M., and Rodenbeck, B.L. (2003) Spatiotemporal patterns of annual sea turtle nesting behaviors along an East Central Florida beach. *IBiological Conservation* **110**: 295-303.

Wiegand, T. and Moloney, K.A. (2004) Rings, circle, and null models for point pattern analysis in ecology. *OIKOS* **104(2)**: 209-229.

Zhu, Z. and Zhang, H. (2006) Spatial sampling design under the infill asymptotic framework. *Environmetrics* **17(4)**: 323-337.