# Bayesian Analysis of Spatial Point Patterns

by

Thomas J. Leininger

Department of Statistical Science
Duke University

Date: _____

Approved:

_____
Alan E. Gelfand, Supervisor

_____
Robert L. Wolpert

_____
Merlise A. Clyde

_____
James S. Clark

Dissertation submitted in partial fulfillment of the requirements for the degree of
Doctor of Philosophy in the Department of Statistical Science
in the Graduate School of Duke University
2014

# Abstract

## Bayesian Analysis of Spatial Point Patterns

by

Thomas J. Leininger

Department of Statistical Science
Duke University

Date: _____

Approved:

_____
Alan E. Gelfand, Supervisor

_____
Robert L. Wolpert

_____
Merlise A. Clyde

_____
James S. Clark

An abstract of a dissertation submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy in the Department of Statistical Science in the Graduate School of Duke University
2014

# Abstract

We explore the posterior inference available for Bayesian spatial point process models. In the literature, discussion of such models is usually focused on model fitting and rejecting complete spatial randomness, with model diagnostics and posterior inference often left as an afterthought. Posterior predictive point patterns are shown to be useful in performing model diagnostics and model selection, as well as providing a wide array of posterior model summaries. We prescribe Bayesian residuals and methods for cross-validation and model selection for Poisson processes, log-Gaussian Cox processes, Gibbs processes, and cluster processes. These novel approaches are demonstrated using existing datasets and simulation studies.

To my wife, Alyse, and to my parents, Greg and Jackie

# Contents

# List of Tables

# List of Figures

# List of Abbreviations and Symbols

## Symbols

| | |
|---|---|
| $N(A)$ | The number of points falling in a set $A$ |
| $\lvert A\rvert$ | The size of a set $A$ |
| $\lambda(s)$ | The intensity function at a location $s \in D$ |
| $\gamma(s, s')$ | The second-order moment measure, evaluated at $s$ and $s'$ |
| $\tilde{g}(s, s')$ | The pair correlation function, evaluated at $s$ and $s'$ |

## Abbreviations

| | |
|---|---|
| CSR | Complete spatial randomness |
| HPP | Homogeneous Poisson process |
| GP | Gaussian process |
| LGCP | Log-Gaussian Cox process |
| MCMC | Markov chain Monte Carlo |
| MLE | Maximum likelihood estimate |
| MPLE | Maximum pseudolikelihood estimate |
| NHPP | Nonhomogeneous Poisson process |
| PCF | Pair correlation function |
| PGP | Poisson-gamma process |
| RPS | Ranked probability score |
| SNCP | Shot noise Cox process |

# Acknowledgements

First, I would like to thank my advisor, Alan Gelfand, for his dedication to my research and to my development as a statistician. He has provided wonderful guidance and constant encouragement over the last four years. I would also like to thank the rest of my committee, Robert Wolpert, Merlise Clyde, and Jim Clark for their time and feedback. I am also thankful for the chance to work on other projects with David Holland, Jenica Allen, and John Silander, Jr.

I am very grateful to the faculty, students, and staff who all contribute to the great environment in the department. I am thankful to have so many great fellow students who have carried me through measure theory, pulled the occasional prank on me, and provided a great source of friendship over the years.

I would like to thank my parents for their support and encouragement over my lifetime and all 20+ years of school. I would also like to thank my son, Dane, for always putting a smile on my face and providing me with fun-filled escapes from writing my dissertation. Lastly, and most importantly, I would like to thank my wife, Alyse, for her constant support and encouragement and for patiently waiting all this time for me to finally have a real job and a decent income.

This work was supported by NIH award 2R01-ES014843-04A1. I am also grateful to the International Society for Bayesian Analysis, American Statistical Association, Duke University Graduate School, and Duke University Department of Statistical Science for conference travel support.

# 1

# Introduction

As discussed in Banerjee et al. (2014), spatial data sets are generally classified into three categories: point-referenced data, areal data, and point pattern data. Point-referenced data refers to situations where the outcome of interest $y(s)$ varies continuously over locations $s$ within some region $D$. Data is collected at a finite set of locations, from which the continuous surface is then estimated. Examples of such data include estimating pollution or temperature levels across some region, such as the United States, or collecting house price information to understand the average house price in some city, state, etc. The region of interest is generally taken to be some subset of $\mathbb{R}^2$ or $\mathbb{R}^3$, though higher dimensions or more abstract spaces can be employed. Models for such data, often termed geostatistical models due to their use in many geological applications, generally employ Gaussian processes or splines as a flexible model for the continuous response surface.

Areal data differs from point-referenced data in that the spatial locations are discrete partitions of the region of interest or points on a lattice. For example, the data could consist of grayscale levels at each pixel of an image or cancer incidence rates across counties in a state. The outcome $y(s_i)$ is assumed to be similar at nearby

locations, though the definition of closeness must be defined for each application. Generally a local structure is defined, often employing Markov random fields, to account for the spatial correlation.

Point pattern data, the subject of this work, describes data in which random events are observed over some domain, with the number and locations of these events being random. Analysis of this category of data involves understanding the underlying process generating these events, which includes learning whether events (also called points) are more likely to occur in certain regions of the domain and whether the existence of an event affects the locations of subsequent events. For example, a point pattern may consist of the locations of trees in a forest or the locations of fast food locations in a city. There may be extra information attached to the event, such as the type of tree or the diameter of the tree, which can enrich the analysis.

The first two classes of spatial data problems are well-studied, but the complexities introduced in point pattern analysis leave many open problems, such as model diagnostics and model selection for point patterns. This thesis will explore some of these issues and suggest some methods for analyzing point pattern models and enriching the inference available from these models, primarily from a Bayesian perspective.

## 1.1 Point patterns

Before describing our contributions, some notation and theoretical development will be given. Many resources can provide a lengthier and more rigorous development of point pattern theory and related topics; see, e.g., Cressie (1993); Banerjee et al. (2014); Gelfand et al. (2010); Illian et al. (2008); Diggle (1983); Møller and Waagepetersen (2007); Daley and Vere-Jones (1998).

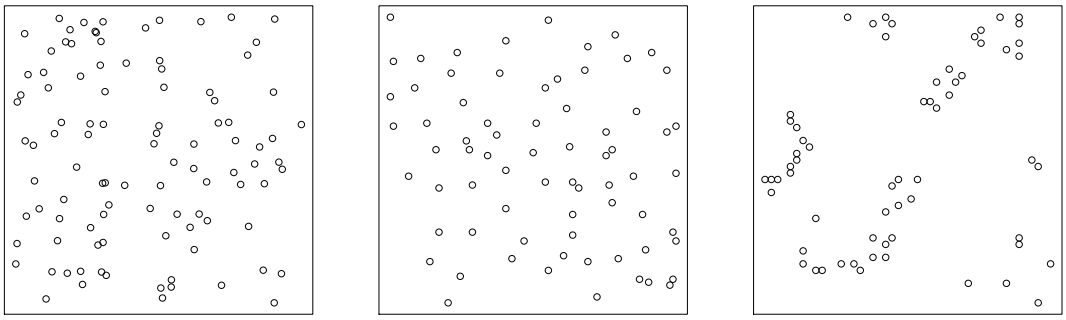
As previously described, a point pattern is a collection of points or events observed over some region, with the locations and number of events both being random. Point

patterns can represent cancer cases in a region, trees in a forest, or crimes in a city. Point pattern analysis is concerned with understanding the underlying process generating the events. This involves learning about the number of events expected to occur, how likely points are to occur over different areas of the domain, and whether event locations are independent of each other. A point process will denote the underlying process which generates the observed point patterns.

The locations of the points in the point pattern will be denoted by $s_i$ and the domain of interest will be denoted by $D$. The collection of points makes up the point pattern, $S$, with $S = \{s_i\}_{i=1}^n$, where $n \equiv N(D)$ is the number of points observed in $D$. In general, the number of points in any set $A \subseteq D$ will be denoted by $N(A)$. The treatment here will generally take $D$ to be a subset of $\mathbb{R}^2$, though other forms for $D$ are common. Time series point patterns will often use $D \equiv (0, T) \subset \mathbb{R}^+$ and spatiotemporal point patterns will take $D \subset \mathbb{R}^2 \times \mathbb{R}^+$. Ang et al. (2012) model crimes on the streets in Chicago, where $D$ is taken to be the linear network of streets in a neighborhood of Chicago.

The least complex point patterns exhibit complete spatial randomness (CSR), a property under which point locations occur independently and uniformly over $D$. Complete spatial randomness implies that points occur with equal likelihood over each region in $D$ and that the points do not cluster nor repel each other. Figure 1.1 shows an example of a point pattern which exhibits CSR and two examples which violate CSR due to regularity and clustering. The clustered point pattern in 1.1c is easily distinguished from the point pattern exhibiting CSR in 1.1a. The regular point pattern in 1.1b is harder to distinguish from 1.1a, especially for untrained eyes. The main difference is that, under CSR, points will sometimes randomly occur very close to each other, while under regularity, such occurrences are rare.

Building a probabilistic model for a point pattern $S$ requires specifying distributions for $N(D)$ and the locations of the points. The distribution for $N(D)$ must



(a) CSR (b) Regularity (c) Clustering

FIGURE 1.1: Examples of point patterns exhibiting (a) CSR, (b) regularity, and (c) clustering.

cover the set $\{0, 1, \ldots, \infty\}$ and is usually taken to be the Poisson distribution. The distribution for the locations of the points must have a valid density $f_n^\theta$ for any $n$ and point process parameters $\theta$. Since the points are unordered and, for now, unlabeled, the location density $f_n^\theta(s_1, s_2, \ldots, s_n)$ must be symmetric in its inputs. Combining these two pieces, the density for $S$, $f_S$, will take the form

$$f_S(S;\theta) = \Pr\big[N(D) = n \,|\, \theta\big]\, n!\, f_n^\theta(s_1, s_2, \ldots, s_n), \tag{1.1}$$

where the factorial $n!$ comes from the exchangeability of the events $s$ within $S$.

Under CSR, the location density $f_n^\theta$ is uniform and points occur independently, leading to $f_n^\theta(s_1, s_2, \ldots, s_n) = \prod_i f_1^\theta(s_i) = \prod_i 1/|D| = |D|^{-n}$, where $|A|$ denotes the size of a set $A$. Complete spatial randomness implies *stationarity*, which means that $f_n^\theta(s_1, \ldots, s_n) = f_n^\theta(s_1 + h, \ldots, s_n + h)$ for all $n$, $s \in D \subseteq \mathbb{R}^d$ and $h \in \mathbb{R}^d$. One implication of stationarity is that the first-order trend is constant.

The homogeneous Poisson process (HPP) is a point process that is built upon CSR. HPPs have a single parameter $\lambda$ which relates to the total number of points expected to be observed in $D$. The key property of the homogeneous Poisson process is that for any region $A \subseteq D$, the number of points expected in $A$, denoted by $N(A)$, follows a Poisson distribution with expectation $\lambda|A|$. This implies that $n$, the total

number of points observed in $D$, has expectation $\lambda|D|$. The independence of locations arising CSR property of HPPs implies that for two disjoint subsets, $A, B \subset D$, the number of points occurring in $A$ and $B$ are independent Poisson variables, again with expectations $\lambda|A|$ and $\lambda|B|$, respectively. The HPP is clearly a stationary process, though other stationary processes do exist.

The likelihood for a homogeneous Poisson process is composed of the two pieces discussed previously. The random number of events observed, $n$, is modeled as a Poisson random variable with expectation $\lambda|D|$. The random locations of these points given $n$ are distributed independently over $D$ with density $f_n^\theta(s_1, \ldots, s_n) = |D|^{-n}$. Combining these two pieces with (1.1) gives the HPP likelihood function

$$f_S(S; \lambda) = \frac{e^{-\lambda|D|}(\lambda|D|)^n}{n!} \times \frac{n!}{|D|^n} = e^{-\lambda|D|}\lambda^n. \tag{1.2}$$

The parameter $\lambda$ from above is called the intensity and controls the rate at which events occur. The intensity can be written more generally as a function $\lambda(s)$ for any location $s \in D$, where regions with higher $\lambda(s)$ have a higher expected number of events. The general definition for the intensity function is that the intensity $\lambda(s)$ is the function satisfying $\mathbb{E}[N(A)] = \int_A \lambda(s)ds$ for any subset $A \subseteq D$. The intensity can equivalently be defined as $\lambda(s) \equiv \lim_{|\partial s|\to 0} \mathbb{E}\big[N(\partial s)\big]/|\partial s|$. The intensity function may not always be tractable, e.g., as in Gibbs processes, which will be discussed later.

Relaxing the homogeneous Poisson process to have a spatially varying intensity $\lambda(s)$ results in the nonhomogeneous Poisson process (NHPP), also called the inhomogeneous Poisson process. Under the NHPP model, which is no longer stationary, the quantity $N(A)$ is distributed as Poisson($\lambda(A)$) where $\lambda(A) \equiv \int_A \lambda(s)ds$. As before, $N(A)$ and $N(B)$ are still independent, conditional on $\lambda(s)$, if $A$ and $B$ are disjoint subsets of $D$. The spatially varying intensity may include a regression component, often specified as $\lambda(s) = \lambda_0 \exp\{x^T(s)\beta\}$. For the HPP, $\theta$ consisted only of

$\lambda$, whereas now $\theta$ may be comprised of $\lambda_0$, several $\beta_k$, and possibly other parameters. Therefore, $\lambda(s)$ implicitly depends on $\theta$, so we will generally write $\lambda(s)$ but could more explicitly write $\lambda^\theta(s)$ instead.

The location density for an NHPP is easily developed first from considering a single point $s^*$. The likelihood of the location of $s^*$ is relative to the height of $\lambda(s)$ at each $s \in D$. Therefore, $\lambda(s)$ can be seen as the unnormalized location density, implying that $f_1^\theta(s^*) = \lambda(s^*)/\lambda(D)$, where $\lambda(D) = \int_D \lambda(s)ds$ is the normalizing constant. Since the NHPP still preserves independence among the locations of its points, $f_n^\theta(s_1, s_2, \ldots, s_n) = \prod_{s_i \in S} \left[\lambda(s_i)/\lambda(D)\right]$. Using this and the fact that $N(D)$ is distributed as Poisson$(\lambda(D))$, the NHPP likelihood builds on (1.2) to become

$$f_S(S;\theta) = \frac{\exp\{-\lambda^\theta(D)\}\left(\lambda^\theta(D)\right)^n}{n!} \times n! \prod_{s_i \in S} \frac{\lambda^\theta(s_i)}{\lambda^\theta(D)} = \exp\{-\lambda^\theta(D)\} \prod_{s_i \in S} \lambda^\theta(s_i) \quad (1.3)$$

Continuing to relax the assumptions of the Poisson process results in more complex, and perhaps more useful, point processes. For example, a Cox process results by taking the inhomogeneous Poisson process and letting $\lambda(s)$ be a realization of a random process. Gibbs processes, cluster processes, and others result when a dependence structure is introduced among the points. For example, saplings tend to cluster around the parent tree, resulting in a cluster process. Of course, some processes may exhibit both a nonconstant intensity as well as a dependence structure among the points, though it is known to be difficult to clearly separate these two influences, as noted in, e.g., Baddeley et al. (2000).

Point processes can be characterized by moment measures, with the intensity function $\lambda(s)$ defining the first-order moment measure of a point process. The second-order moment measure $\gamma(s, s')$, also called the second-order intensity, addresses the covariance structure, just as a Gaussian process used in a geostatistical model employs a covariance function. $\gamma(s, s')$ is defined as the function satisfy-

ing $\mathbb{E}[N(A)N(B)] = \int_A \int_B \gamma(s,s')ds'ds$, which provides a sense of the covariation between two sets $A, B \subseteq D$. The pair correlation function (PCF), also called the reweighted second-order intensity, provides a standardized version of the second-order measure, which is useful in assessing the range of correlation in point patterns (see, e.g., Illian et al., 2008, p. 220). The PCF is defined as $\tilde{g}(s,s') = \gamma(s,s')/(\lambda(s)\lambda(s'))$. Many processes have closed forms for $\gamma(s,s')$ and $\tilde{g}(s,s')$, but they will not be given here.

## 1.2 Frequentist Inference for Spatial Point Processes

We shall primarily operate within the Bayesian paradigm but it will be useful to briefly highlight a few aspects of frequentist inference for point patterns. As discussed in Møller and Waagepetersen (2007), maximum likelihood estimates (MLEs) are not always computationally feasible for point patterns. For example, Gibbs processes (see Section 4.1), have unknown normalizing constants, though path sampling (Gelman and Meng, 1998) and MCMC methods (Ogata and Tanemura, 1981) have developed to achieve MLE estimates. The MLEs of spatial point processes do not enjoy the usual asymptotic properties, making them less dominant over other methods. See Møller and Waagepetersen (2007, 2003) and references therein for a more thorough discussion of maximum likelihood estimates for point processes. In some cases, however, the MLE is simple to obtain. For example, the MLE does exist for the intensity parameter $\lambda$ of an HPP, and is simply $\hat{\lambda} = n/|D|$.

Since point process MLEs do not enjoy the usual asymptotic theoretical support for MLEs (Baddeley and Turner, 2000), other estimation methods enjoy wide use. For some point processes, minimum contrast estimates provide robust, computationally-efficient estimates through matching higher-order properties of the process to the observed data. Let $T(r;\theta)$ be some summary statistic of the point process with parameters $\theta$ and $\hat{T}(r)$ be the empirical estimate of the statistic. Typically $T$ is taken

to be the pairwise correlation function or the $K$-function, which will be introduced in Chapter 2. If the PCF $\tilde{g}(s, s')$ is just a function of the distance $r = ||s - s'||$ and then we can take $T(r; theta)$ to be the PCF evaluated at $r$. Then the minimum contrast estimate $\hat{\theta}$ is the parameter (or set of parameters) $\theta$ which minimize

$$\int_{r_{\min}}^{r_{\max}} \left(\hat{T}(r)^q - T(r;\theta)^q\right)^p dr, \tag{1.4}$$

where $p, q > 0$ and $0 \leq r_{\min} < r_{\max}$ specifies a reasonable range of values for $r$. See Waagepetersen (2007) for further discussion.

For processes with a likelihood containing an intractable normalizing constant, such as Gibbs processes, the pseudolikelihood is often employed because it removes the need to estimate the normalizing constant. Besag (1977) defines the point process pseudolikelihood to be

$$PL(\theta; S) = \exp\Big\{ - \int_D \lambda^\theta(s; S)\, ds \Big\} \prod_{s_i \in S} \lambda^\theta(s_i; S), \tag{1.5}$$

where $\lambda^\theta(s; S)$ is the (Papangelou) conditional intensity of at $s \in D$ and depends on process parameters $\theta$. The conditional intensity is defined as

$$\lambda^\theta(s; S) = \begin{cases} \dfrac{f^\theta(S \cup \{s\})}{f^\theta(S)} & s \notin S, \text{ and} \\ \\ \dfrac{f^\theta(S)}{f^\theta(S \setminus \{s\})} & s \in S, \end{cases} \tag{1.6}$$

where $f^\theta$ is the location density and $S \setminus \{s\}$ denotes $S$ with the singleton point $s$ removed. For Poisson processes, where point locations occur independently, the conditional intensity is equal to the intensity, or $\lambda^\theta(s; S) = \lambda^\theta(s)$. For processes with a second-order dependence, the conditional intensity will differ from the intensity according to the nature of interactions among points specified by the process. For

example, in a cluster process, $\lambda^\theta(s; S)$ may be larger than $\lambda^\theta(s)$ if $s$ is close to some of the points in $S$.

The real benefit to using the pseudolikelihood is that the normalizing constant cancels out in the fraction in (1.6) when computing the conditional intensity. This is because both the numerator and denominator have the same intractable normalizing constant because the parameters in $f^\theta$ are the same. The point process model can then be written as a generalized linear model and fit using a Berman-Turner device (Berman and Turner, 1992; Baddeley and Turner, 2000). The fitted model parameters are the maximum pseudolikelihood estimates (MPLEs). Baddeley and Turner (2000) notes that MPLEs are consistent and asymptotically normal under suitable conditions. Huang and Ogata (1999) suggest obtaining the MPLEs and then taking a single Newton-Raphson step toward maximizing the likelihood.

## 1.3 Bayesian Inference for Spatial Point Processes

In brief, Bayesian modeling seeks to take prior belief about model parameters $\theta$, quantified by a prior distribution $\pi(\theta)$, and combine this prior belief with the observed data $y$ to provide an updated belief about $\theta$. This updated belief, called the posterior distribution $\pi(\theta|y)$, is a probability distribution which accounts for the information about $\theta$ in both the prior and the data using Bayes rule. The posterior distribution is calculated as

$$\pi(\theta|y) = \frac{p(y|\theta)\pi(\theta)}{\int_\Theta p(y|\theta)\pi(\theta)}, \tag{1.7}$$

where $p(y|\theta)$ is the data model, which is equivalent to the likelihood $L(\theta; y)$, and $\theta$ can take on values over some domain $\Theta$. Though the integral in the denominator is generally intractable, methods such as Gibbs sampling and the Metropolis-Hastings algorithm can be utilized to obtain posterior samples from $\pi(\theta|y)$ through Markov

chain Monte Carlo (MCMC). More development on the basics of Bayesian inference is given in, e.g., Gelman et al. (2013).

Bayesian modeling can be challenging for spatial point processes, due to the same issues with an intractable normalizing constant in the point process likelihood as discussed above, but also due to poor mixing properties and inefficiency when applying standard MCMC algorithms. Fortunately, most point process models have at least one working method for obtaining posterior distributions for the model parameters, often involving an advanced MCMC algorithm. These methods will not be discussed here, but rather presented as they are used in the ensuing chapters.

Our focus will be less on how to fit Bayesian models and more on what to do once we've fit them. We will often use the Bayesian framework to generate posterior predictive distributions of point patterns. The posterior predictive distribution takes the posterior distribution $\pi(\theta|y)$ and generates simulated data $y^*$ using the model. The posterior predictive distribution is written as $p(y^*|y)$ where $p(y^*|y) = \int_\Theta p(y^*|\theta)\,\pi(\theta|y)\,d\theta$. Drawing from the posterior predictive distributions provides replicates $y^*$ which are directly comparable to the original data $y$.

## 1.4 Contributions to Spatial Point Process Analysis

Typical point pattern analysis usually begins by exploring whether such a point pattern exhibits complete spatial randomness (i.e., whether the point pattern arose from an HPP). Complete spatial randomness is violated if events exhibit a dependence structure and/or they occur with nonconstant intensity. Once CSR has been rejected for a given point pattern, however, the next model chosen should similarly be subjected to scrutiny and compared with other valid models. This second set of analyses is usually not carried out quite as thoroughly as the initial analysis which rejected CSR.

The gap in analysis here is more attributable to a lack of powerful diagnostic and comparison methods than to a lack of effort. Testing goodness of fit is not straightforward for point patterns, nor is there a widely applicable, easy-to-use method for model selection. The challenge with point patterns, as shall be presented in this work, is that a point pattern contains limited information about the process which generated it. For example, learning about intensity function is difficult because the smoothness in the intensity estimate is largely determined by the imposed model, since the data provide little indication of the smoothness of the original process. The user must either have prior knowledge of the smoothness or must employ some metric to choose an optimal smoothness parameter, as is done in kernel density estimation.

Chapter 2 describes point pattern analysis for Poisson processes and log-Gaussian Cox processes. Details about fitting Bayesian models for each are given, followed by a discussion of many posterior summaries which can be generated for a richer analysis of the posterior distribution. Much of this relies on generating posterior predictive point patterns from the posterior distributions of the model parameters to create *model-based* summaries of interest. Obtaining model-based estimates of the $F$-, $G$-, and $K$-functions are discussed and compared to the usual nonparametric estimates.

Chapter 3 builds on the model-based summaries of Chapter 2 to discuss ideas for model diagnostics based on the posterior predictive point patterns. The proposed predictive residuals are shown to illuminate regions of $D$ where the model fits poorly and, if the regions are substantial enough, can indicate overall lack of fit. For models that seem to adequately fit the data, a similar approach allows us to apply proper scoring rules to compare models. Cross-validation ideas presented herein allow for model comparison on data not used to fit the model. A large simulation study illuminates the extent of learning available when comparing point process models.

Chapter 4 extends the ideas of Chapters 2 and 3 to more complex processes, such as Gibbs processes and cluster processes. First, adaptations for Gibbs processes

are presented to overcome the inherent dependence structure among points. Since cross-validation is not viable for Gibbs processes, posterior predictive checks (Gelman et al., 1996) are utilized to determine model fit. The discussion then moves on to cluster processes, for which the methods of Chapter 3 apply. Ideas are then given for other complex point processes, with some discussion of posterior inference and model assessment.

Finally, Chapter 5 will summarize the work presented herein and discuss potential paths for future research and improvement of these methods.

# 2

# Bayesian Point Pattern Analysis

This chapter details Bayesian model-fitting for many standard point processes and introduces methods for extensive posterior inference. Beginning with homogeneous point processes, we illustrate how the posterior distribution for the model provides a rich variety of options for posterior inference using posterior predictive point patterns. Later in the chapter, models for nonhomogeneous point processes will be introduced and these posterior inference methods will naturally lead themselves to further analysis.

The goal with our posterior inference is to provide *model-based* inference of posterior quantities of interest. For example, Ripley's $K$-function (Ripley, 1976; Dixon, 2002) is a common exploratory tool in point pattern analysis which describes the expected number of points within a distance $d$ of a typical point in the point pattern. It is commonly used to criticize the CSR hypothesis by comparing the observed distribution of the $K$-function to its theoretical distribution under CSR. The usual $K$-function estimate employs a nonparametric estimate of the intensity surface, but we will take a model-based approach and use our posterior draws for the intensity surface instead. Our methods will demonstrate how to provide a whole posterior

distribution for the $K$-function, so that a comparison to the theoretical value can be done with a knowledge of the uncertainty involved.

## 2.1 Homogeneous Poisson Processes

As noted in the previous chapter, the most basic point process is a homogeneous Poisson process (HPP). This process implies that events occur independently over the domain with constant intensity. This model has a single parameter $\lambda$ which defines the number of events in any region $A \subseteq D$ to be distributed as $N(A) \sim \text{Poisson}(\lambda|A|)$. With one parameter, fitting an HPP model is very simple. From a frequentist perspective, the maximum likelihood estimate (MLE) is just the number of observed events divided by the area of $D$, or $\hat{\lambda} = n/|D|$. This is easily derived from the HPP likelihood, given in (1.2).

A Bayesian model requires a prior distribution for $\lambda$. The gamma distribution, a flexible distribution over $\mathbb{R}^+$, provides a conjugate prior for $\lambda$. Taking the prior for $\lambda$ to be

$$\lambda \sim \text{Gamma}(a_\lambda, b_\lambda), \tag{2.1}$$

with prior expectation $\mathbb{E}[\lambda] = a_\lambda/b_\lambda$, the posterior is given by

$$\lambda|S \sim \text{Gamma}(a_\lambda + n, b_\lambda + |D|). \tag{2.2}$$

Since the posterior distribution for $\lambda$ has a closed form, there is no need for MCMC. With little prior knowledge about the process, one could also use the Jeffreys prior, which would set the prior for $\lambda$ as $p(\lambda) \propto 1/\lambda$. The Jeffreys prior results in the posterior $\text{Gamma}(n, |D|)$. We imagine that informative prior knowledge is generally available, whether it be an expected number of trees per hectare or cancer cases per geographic region.

Other prior distributions for $\lambda$, such as the log-normal distribution, may also sensible and their posteriors can be sampled from via the Metropolis-Hastings algo-

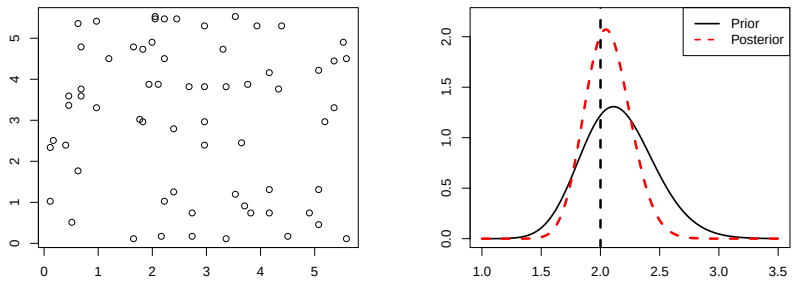
rithm. In fact, one could alternatively reparameterize from $\lambda$ to $\exp\{\beta_0\}$ and employ a prior on $\beta_0$ that takes values over $\mathbb{R}$. For example, a normal prior could be used for $\beta_0$, which induces a log-normal prior for $\lambda$. No matter the prior for $\lambda$, we can easily obtain posterior draws for $\lambda$ and use them in posterior analysis.

### *2.1.1 Japanese Pines Data*

To illustrate the HPP model, we turn to a well-studied dataset consisting of the locations of 65 black pine saplings in a 5.7m $\times$ 5.7m square patch of forest. This dataset was first studied by Numata (1961) but has seen several follow-up analyses (Diggle, 1983; Ogata and Tanemura, 1981; Baddeley and Turner, 2000). A plot of the data is given in Figure 2.1a. The data seem to be evenly spread over the domain, suggesting that a homogeneous intensity is reasonable.

To fit the HPP model, we use a gamma prior for $\lambda$ as suggested in (2.1). The prior distribution for $\lambda$ can then be specified directly or can be induced by first specifying a prior over the expected number of trees $\mathbb{E}[N(D)] = \lambda|D|$. Suppose our prior belief is that the expected number of trees in the region is about 70 trees with a prior variance of 100. This puts most of the prior mass for $\mathbb{E}[N(D)]$ between about 45 trees and 95 trees. Using a gamma distribution for $\mathbb{E}[N(D)]$, the expected value and variance choices imply that $\mathbb{E}[N(D)] \sim \text{Gamma}(49, 0.7)$. Since $\mathbb{E}[N(D)] = \lambda(D) = \lambda|D|$ for an HPP, this prior for the expected number of trees can be easily converted into a prior for $\lambda$ itself. The prior $\lambda|D| \sim \text{Gamma}(49, 0.7)$ implies the prior distribution for $\lambda$ is $\lambda \sim \text{Gamma}(a_\lambda = 49, b_\lambda = 0.7|D| = 22.743)$. This gives a prior mean for $\lambda$ of 2.154 with variance of 0.095.

This prior for $\lambda$ provides the posterior distribution for $\lambda$ as Gamma(114, 55.233), with posterior mean $\mathbb{E}[\lambda|S] = 2.064$. The prior and posterior distributions for $\lambda$ for the Japanese pines data are given in Figure 2.1b, with the vertical line marking the




(a) (b)

FIGURE 2.1: Plots of (a) the Japanese pines data and (b) the prior and posterior distributions for $\lambda$. The vertical line in (b) denotes the MLE $\hat{\lambda} = 2.001$.

MLE $\hat{\lambda} = 2.001$. For any subregion of interest $A \subseteq D$, the posterior distribution for $\lambda(A)$ is $\lambda(A) \sim \text{Gamma}(a_\lambda + n, (b_\lambda + |D|)/|A|)$.

### *2.1.2 Posterior Analysis for HPPs*

We now show that the Bayesian modeling framework lends itself naturally to a rich class of posterior model summaries. Not only do we have posterior draws of our parameters, which we can use to recreate the intensity surface, but we can also use these posterior draws to simulate posterior predictive point patterns, denoted by $\{S_l^*\}_{l=1}^L$. The posterior predictive point patterns will reflect our uncertainty in our model parameters and will be helpful in summarizing the model's fit to the data. In Chapter 3, we will discuss model diagnostics and model selection for point process models.

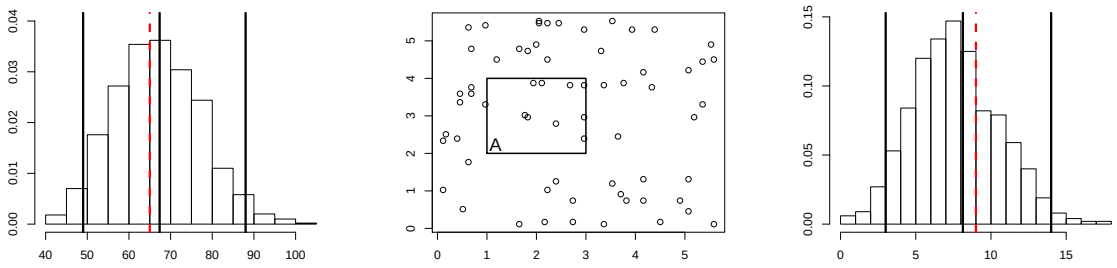
The first basic question that might be asked after fitting the model is how many points should be expected in the domain $D$. As noted previously, the expected number of points in D, denoted by $\mathbb{E}[N(D)]$ (or, equivalently, $\mathbb{E}[n]$), is given by the quantity $\lambda(D)$, which is equal to $\lambda|D|$ for an HPP. The posterior distribution of $\lambda(D)$ is Gamma$(a_\lambda + n, (b_\lambda + |D|)/|D|)$ distribution with posterior mean $|D|(a_\lambda + n)/(b_\lambda + |D|) = 67.06$. Though $\lambda(D)$ has a nice closed form because of the conjugate prior

specification, this distribution can in general be approximated using the posterior draws for $\lambda$ and multiplying them each by $|D|$.

A more useful answer to this question, however, can be given by finding the posterior predictive distribution of $N(D)$ itself, rather than the posterior for its expected value. Denoting the posterior draws of $\lambda$ by $\{\lambda^{(l)}\}_{l=1}^{L}$, the posterior predictive draws for $N(D)$, which we'll denote by $N^{(l)}(D)$, can be easily simulated from the model. For an HPP model this only requires drawing $N^{(l)}(D) \sim \text{Poisson}(\lambda^{(l)}|D|)$ for $l = 1, \ldots, L$. $L$ should be a large number to fully capture the variability in the parameters and the point process itself. We will generally take $L$ to be 1000, but since the MCMC chain will be run for much longer than 1000 iterations, we can take the $\lambda^{(l)}$ draws from the posterior to actually be thinned samples from the MCMC chain.

The expected value of the $N^{(l)}(D)$ is $\mathbb{E}[N(D)|S]$ as desired, but one can now also provide credible intervals to quantify the posterior variability of $N(D)$. Figure 2.2a shows the posterior distribution for the number of points in $D$, constructed using the $N^{(l)}(D)$ draws. We see that the mean number of points in the predictive point patterns is 67.16, which is close to the theoretical value of 67.06 and the observed $n = 65$, with a 95% credible interval of (48, 88). With a full posterior distribution, other potential summaries of interest can also be calculated, such as $Pr[N(D) \geq 70] = 0.415$.

Figure 2.2b shows a subregion $A$ for which we might want to know the distribution of the number of points. Again, we could either look at the posterior distribution of $\lambda(A)$, which gives the expected number of points in $A$, or of $N(A)$, the number of points itself. We prefer to look at the posterior distribution of $N(A)$ as it is more tangible, though the posterior of $\lambda(A)$ has a nice closed form as given above. Posterior draws for $N(A)$ are drawn from $N^{(l)}(A) \sim \text{Poisson}(\lambda^{(l)}|A|)$ as shown previously. The posterior distribution for $N(A)$ is given in Figure 2.2c, where we see that $N(A)$ has




(a) (b) (c)

FIGURE 2.2: (a) The posterior distribution for $N(D)$, (b) the subset $A \subset D$ of interest, and (c) the posterior distribution for $N(A)$. The solid vertical lines in (a) and (c) represent the posterior mean and 95% credible intervals and the dashed lines represent the observed values.

a posterior mean of 8.21 with a 95% credible interval of (3, 14). The observed $N(A)$ is 9, so our posterior interval seems reasonable.

The posterior distribution of the second-order intensity $\gamma(s, s')$ is also available to us. For an HPP, $\gamma(s, s') = \lambda^2$ due to the independence among the points. Draws from this posterior distribution are obtained by simply squaring the draws of the intensity, $\lambda^{(l)}$. The pairwise correlation function $\tilde{g}(s, s')$ is equal to 1 for an HPP, again because of the independence, but a posterior distribution for the PCF could be obtained for more complex point patterns, as will be shown later. For two subregions $A, B \subseteq D$, we can also obtain the posterior distribution of $[N(A)N(B)]$. Draws from this distribution are obtained by first taking draws from the posterior distributions for $N(A)$ and $N(B)$ as described above and then multiplying the draws from each: $N^{(l)}(A) \times N^{(l)}(B)$. In fact, the posterior distribution any function of subsets $A_1, \ldots, A_k$ is available to us, whether it is $[N(A_1)N(A_2)\ldots N(A_k)]$ or $[N(A_1) + \cdots + N(A_k)]$, by simply using our posterior predictive point patterns.

### *2.1.3 Homogeneous F-, G-, and K-functions*

So far, we have been calculating these posterior quantities of interest without much theoretical justification. The main theoretical tool we can employ here is Campbell's Theorem (see, e.g., Banerjee et al., 2014), which gives the expectation of a $D$-measurable function $g$ of points in a point pattern $S$. Campbell's Theorem gives the equality

$$\mathbb{E}\Big[\sum_{s_i \in S} g(s_i)\Big] = \int_D g(s)\lambda(s)\,ds. \tag{2.3}$$

For a feature of interest, say $g(s) = \mathbf{1}(s \in A)$ for some set $B \subset D$, Campbell's Theorem says that $\sum_{s_i \in S} \mathbf{1}(s_i \in A)$ is an unbiased estimator for $\int_D \mathbf{1}(s \in A)\lambda(s)\,ds = \int_A \lambda(s)\,ds = \lambda(A)$, which is just the expected number of points falling in a region $A$. All that is required is to choose an appropriate function $g$ such that the right-hand side gives a quantity of interest, then the left-hand side becomes the unbiased estimate of the quantity of interest. This theorem is typically used to construct estimators as functions of the observed point pattern $S$. However, this theorem is also useful when applied to posterior predictive point patterns, as we will now discuss.

With the posterior draws for $\lambda$, posterior predictive point patterns are trivially generated by first drawing the number of points $n^{(l)} \equiv N^{(l)}(D)$ in the posterior predictive point pattern $S_l^*$ from $n^{(l)} \sim \text{Poisson}(\lambda^{(l)}|D|)$. The locations of each $s_{li}^* \in S_l^*$ are then randomly generated with uniform probability over $D$. For irregular $D$, the location sampling can be performed by repeatedly drawing points within a bounding box for $D$, retaining any points which fall inside $D$ also, and continuing until $n^{(l)}$ locations have been sampled.

These posterior predictive point patterns provide an alternate, more general, method for constructing the posterior distributions of $N(D)$ and $N(A)$. For each $S_l^*$, counting the total number of points, $n^{(l)}$, gives $N^{(l)}(D)$ and counting the number of

points inside $A$ gives $N^{(l)}(A)$. Previously, we simply drew $N^{(l)}(D)$ and $N^{(l)}(A)$ from their marginal (Poisson) distribution. The two methods are comparable, though this counting method using posterior predictive point patterns will also work for more complex point patterns, such as those with spatially varying intensities and second-order dependence. Each $N^{(l)}(A)$, for example, then takes the form of a sum which becomes the inside part of the expectation on the left-hand side (2.3). Each of these has expectation $\int_D \mathbf{1}(s \in A)\lambda^{(l)}(s)\, ds = \lambda^{(l)}(A)$ using Campbell's Theorem. Integrating over our posterior samples, we find that the expected value of the left side $\mathbb{E}[N(A)|S]$ is equal to $\mathbb{E}[\lambda(A)|S]$ as expected. Though the result may not be surprising, we see how Campbell's Theorem begins to prove useful in conjunction with these posterior predictive point patterns.

Campbell's Theorem also has a bivariate form for a $(D \times D)$-measurable function $g$ of two points in $S$:

$$\mathbb{E}\Big[\sum_{\substack{s_i,\, s_j \in S \\ i \neq j}} g(s_i, s_j)\Big] = \int_D \int_D g(s, s')\gamma(s, s')\, ds\, ds', \tag{2.4}$$

where $\gamma(s, s')$ is the second-order intensity defined previously. The bivariate form is useful for exploring second-order properties of a point process. Another useful result is the Georgii-Nguyen-Zessin (GNZ) formula (Georgii, 1976; Nguyen and Zessin, 1979), which gives the equality

$$\mathbb{E}\big[\sum_{s_i \in S} g(s_i, S\backslash\{s_i\})\big] = \mathbb{E}\Big[\int_D g(s, S)\lambda(s; S)ds\Big], \tag{2.5}$$

where $\lambda(s; S)$ is the Papangelou conditional intensity and $g$ is a non-negative function.

We can now use these results to obtain inference for the homogeneous $F$-, $G$-, and $K$-functions. These three functions are standard procedures in point pattern analysis for exploring the distribution of interpoint distances to determine whether

complete spatial randomness is a reasonable assumption for the current dataset. If CSR seems reasonable, an HPP model is employed. Otherwise, the $F$-, $G$-, and $K$-functions provide insight into whether the points appear to be more clustered or dispersed than would be expected under CSR. Typically, nonparametric empirical estimates of these functions are used, but we demonstrate how to use these theoretical tools along with Monte Carlo integration to provide model-based expectations and posterior distributions for these functions.

The $F$-function, denoted by $F(d)$, is the cumulative distribution function (CDF) of the nearest neighbor distance $d$ from a random point in $D$ to an event in $S$. It is often called the "empty space function" because it measures the gaps or empty space in the point pattern. Under CSR, $F(d) = 1 - \exp(-\lambda\pi d^2)$. The usual estimator for $F(d)$ is obtained by randomly sampling a large number of points uniformly over $D$, call this set of points $T = \{t_j\}_{j=1}^{J}$, and calculating the proportion of $t_j$ having a point of $S$ within distance $d$.

The $G$-function, denoted by $G(d)$, is the CDF of the nearest neighbor distance from one observed event to another. Under CSR, $G(d) = F(d) = 1 - \exp(-\lambda\pi d^2)$. Using the notation of Banerjee et al. (2014), define $N(s, d, S)$ to be the number of events in $S \setminus s$ inside a ball of radius $d$ centered around an arbitrary observed event $s$, where $S \setminus s$ denotes the point pattern $S$ with the event at $s$ removed. In this notation, we can express $G(d)$ as $Pr[N(s, d, S) > 0]$.

The $K$-function, also called Ripley's $K$-function, gives a scale-free description of the expected number of points within distance $d$ of an arbitrary event in $S$ (Ripley, 1976; Dixon, 2002). For a first-order stationary process, meaning that the intensity is constant over $D$, the $K$-function is equal to $\mathbb{E}\left[N(s, d, S)\right]/\lambda$. Dividing by $\lambda$ adjusts for the overall intensity of the HPP and allows $K(d)$ to be scale-free. Under CSR, $K(d) = \pi d^2$.

Appendix A contains the standard empirical estimators for these functions. These estimators also include an edge correction to compensate for the bias in the naive estimators. The bias arises because these point patterns are only observed over some finite domain $D$, but could potentially exist on a much larger, possibly infinite, domain (such as $\mathbb{R}^2$). Looking at $K(d)$, for example, when counting the number of neighbors within distance $d$ of some point $s_i \in S$, it will often be the case that $s_i$ is close to the boundary of $S$. When that happens, $s_i$ may have neighbors that are outside of $D$ yet are within distance $d$ of $s_i$. These neighbors were not observed and therefore we have no idea of knowing how many there might be. The edge corrections proposed in the literature address this bias by adjusting the estimates to handle points near the boundary of $D$ differently. For example, the "reduced sample" or border correction estimates these functions using only the points in $D$ that are at least distance $d$ from the boundary of $D$. Since the remaining points are far enough from the edge of $D$, their $d$-close neighbors will all be observed. This correction provides an unbiased estimate though it sacrifices useful information. Other approaches adjust differently and are able to retain more of the data.

As noted previously, typical point pattern analysis involves obtaining the empirical estimate of these three functions, as well as other similar functions such the $J$-function, etc. Typically this is done as an exploratory analysis, investigating what second-order trends are suggested by the data. We now present methods for obtaining *model-based* posterior estimates of these functions. In the following development, these functions describe *a posteriori* features of our model as opposed to just highlighting trends in the data.

For some models, such as the HPP model, the theoretical forms for the $F$-, $G$- and $K$-functions are known and the posterior mean for model parameters, such as $\lambda$ for the HPP, could be used as plug-in estimates. However, our method employs the posterior draws rather than just the posterior mean, integrating over all the

uncertainty in the posterior. Further, our method will generalize to other models without requiring that the theoretical forms of $F(d)$, $G(d)$, etc., be known.

We first begin by developing a model-based summary of $G(d)$. We have been somewhat relaxed in our notation thus far by assuming that each $s_i$ in $S$ is within $D$. In order to think correctly about edge effects, however, we will assume that the point pattern might have points outside of $D$ and the notation will explicitly restrict $S$ to $D$ when desired for the rest of this section. It is often the case that $D$ represents an observation window inside which events of interest are recorded, yet these events exist as part of a much larger point pattern. A common example of this is recording tree locations within some small subset of a large forest.

Let $N_D(s_i, d, S) \equiv N(s_i, d, S \cap D)$ denote the number of points in $S \cap D$ that are $d$-close to $s_i$. Referring to the discussion above about edge corrections, we only know quantities such as $N_D(s_i, d, S)$ but we would like to adjust this quantity to be unbiased for $N(s_i, d, S)$.

Recalling that $G(d)$ can be written as $Pr[N(s, d, S) > 0]$, consider the calculation

$$\begin{aligned}
\mathbb{E}_S \sum_{s_i \in S \cap D} \mathbf{1}(N_D(s_i, d, S) > 0) &= \mathbb{E}_{N(D)} \mathbb{E}_{S|N(D)} \sum_{s_i \in S \cap D} \mathbf{1}(N_D(s_i, d, S) > 0) \\
&= \mathbb{E}_{N(D)} \big[ \sum_{s_i \in S \cap D} \mathbb{E}_{S|N(D)} \mathbf{1}(N(s_i, d, S) > 0) \big] \\
&= \mathbb{E}_{N(D)} \big[ N(D) Pr[N_D(s, d, S) > 0] \big] \\
&= \lambda(D) Pr[N_D(s, d, S) > 0]. \qquad (2.6)
\end{aligned}$$

We can adjust the left-hand side of (2.6) and consider the quantity

$$\mathbb{E}_S \sum_{s_i \in S \cap D} \frac{\mathbf{1}(N_D(s_i, d, S) > 0)}{N(D)} \qquad (2.7)$$

which has expected value

$$\mathbb{E}_S \sum_{s_i \in S \cap D} \frac{\mathbf{1}(N_D(s_i, d, S) > 0)}{N(D)} = \mathbb{E}_{N(D)} \left[ \frac{1}{N(D)} \sum_{s_i \in S \cap D} \mathbb{E}_{S|N(D)} \mathbf{1}(N_D(s_i, d, S) > 0) \right]$$

$$= \mathbb{E}_{N(D)} \big[ Pr[N_D(s, d, S) > 0] \big]$$

$$= Pr[N_D(s, d, S) > 0]. \tag{2.8}$$

The quantity $Pr[N_D(s, d, S) > 0]$ is close to $G(d)$. In fact, removing the expectation from the left-hand side of (2.7) gives a naive estimator for $G(d)$. However, we are still only estimating $Pr[N_D(s, d, S) > 0]$, i.e., the probability that the count is greater than 0 under the restriction of $S$ to $D$. Evidently, $Pr[N(s, d, S) > 0]$ applies to the countable point pattern $S$ over $\mathbb{R}^2$ but we can only make the previous calculations by restriction to a bounded set $D$. Of course $N_D(s, d, S) \leq N(s, d, S)$ for any $s$ and any $S$, so $G(d) = Pr[N(s, d, S) > 0] \geq Pr[N_D(s, d, S) > 0]$ which clarifies the need for edge correction.

The edge correction for the usual empirical estimate considers summing over only those $s_i \in S \cap D$ such that $c_d(s_i) \subset D$, where $c_d(s_i)$ is the interior of the circle of radius $d$ at $s_i$. This is also often written as only considering those $s_i \in S \cap D$ with $b_i \leq d$, where $b_i$ is the distance from $s_i$ to the nearest boundary of $D$. The edge correction is needed because when $c_d(s_i) \cap D^C \neq \emptyset$, or equivalently $b_i > d$, $s_i$ may have a $d$-close neighbor just outside of $D$ that we didn't observe.

In this spirit, suppose we look at

$$\mathbb{E}_S \left[ \sum_{s_i \in S \cap D} \frac{\mathbf{1}(N_D(s_i, d, S) > 0,\ c_d(s_i) \subset D)}{N(D)} \right]. \tag{2.9}$$

Note that the sum in the numerator of (2.9) is less than that in the numerator of (2.7). Now the expectation in (2.9) is

$$\mathbb{E}_S \left[ \sum_{s_i \in S \cap D} \frac{\mathbf{1}(N_D(s_i, d, S) > 0,\ c_d(s_i) \subset D)}{N(D)} \right] = Pr[N_D(s, d, S) > 0,\ c_d(s) \subset D]. \tag{2.10}$$

Again, this describes the probability that, for a random $S$, and a random $s \in S$ with $c_d(s) \subset D$, there is at least one $s' \in (S \backslash \{s\}) \cap D$ in $c_d(s)$.

Let $\tilde{G}(d) = Pr[N_D(s, d, S) > 0 \,|\, c_d(s) \subset D]$. The usual empirical estimate obtains an estimate of this probability and calls it $\hat{G}(d)$. For us, we would say that $\tilde{G}(d) = G(d)$ for small values of $d$, such that it is possible for $c_d(s) \subset D$. The empirical estimate will have a denominator decreasing in $d$, eventually becoming 0 when $d$ is too big, but this degenerate case is usually handling by defining $\hat{G}(d) = 1$ for such cases.

We can easily create a Monte Carlo estimate of (2.10), so if we estimate $Pr[c_d(s) \subset D]$, the ratio will provide a Monte Carlo estimate of $\tilde{G}(d)$, an edge-corrected estimate. Since our estimate of $Pr[c_d(s) \subset D]$ will be less than 1, we will increase (2.11), which makes sense given that $Pr[N_D(s, d, S) > 0, c_d(s) \subset D] \leq Pr[N_D(s, d, S) > 0] \leq Pr[N(s, d, S) > 0] = G(d)$. Our estimate for $Pr[c_d(s) \subset D]$ will be based on the equality

$$\mathbb{E}_S \sum_{s_i \in S \cap D} \frac{\mathbf{1}(c_d(s_i) \subset D)}{N(D)} = Pr[c_d(s) \subset D], \tag{2.11}$$

and the left-hand side of (2.11) naturally invites a Monte Carlo integration using our posterior predictive point patterns.

In summary, we will construct $\tilde{G}(d) = Pr[N_D(s, d, S) > 0 \,|\, c_d(s) \subset D]$ using Monte Carlo integration with our posterior predictive point patterns (which we have already restricted to $D$), using the following formulas:

$$Pr[N_D(s,d,S)>0,\ c_d(s)\subset D] \approx \frac{1}{L}\sum_{l=1}^{L}\left[\frac{\sum_{s_{li}^*}\mathbf{1}(N(s_{li}^*,d,S_l^*)>0,\ c_d(s_{li}^*)\subset D)}{N^{(l)}(D)}\right] \tag{2.12}$$

$$Pr[c_d(s)\subset D] \approx \frac{1}{L}\sum_{l=1}^{L}\left[\frac{\sum_{s_{li}^*}\mathbf{1}(c_d(s_{li}^*)\subset D)}{N^{(l)}(D)}\right] \tag{2.13}$$

$$\Rightarrow \tilde{G}(d) \approx \frac{\frac{1}{L}\sum_{l=1}^{L}\left[\frac{\sum_{s_{li}^*}\mathbf{1}(N(s_{li}^*,d,S_l^*)>0,\ c_d(s_{li}^*)\subset D)}{N^{(l)}(D)}\right]}{\frac{1}{L}\sum_{l=1}^{L}\left[\frac{\sum_{s_{li}^*}\mathbf{1}(c_d(s_{li}^*)\subset D)}{N^{(l)}(D)}\right]}. \tag{2.14}$$

The $F$-function is very similar to the $G$-function, so we can use a similar argument to construct our posterior estimate of $F(d)$, or rather $\tilde{F}(d) = Pr[N_D(t,d,S) > 0 \,|\, c_d(t)\subset D]$ for any random location $t\in D$. The only difference here is that we will look at the grid points $t_l$ in $T$ and the distance to their nearest neighbors in each $S_l^*$. The posterior estimate for $\tilde{F}(d)$ is then constructed as

$$\begin{aligned}\tilde{F}(d) &\approx \frac{\frac{1}{L}\sum_{l=1}^{L}\left[\frac{\sum_{j=1}^{J}\mathbf{1}(N(t_j,d,S_l^*)>0,\ c_d(t_j)\subset D)}{J}\right]}{\frac{1}{L}\sum_{l=1}^{L}\left[\frac{\sum_{j=1}^{J}\mathbf{1}(c_d(t_j)\subset D)}{J}\right]} \\ &= \frac{\frac{1}{L}\sum_{l=1}^{L}\sum_{j=1}^{J}\mathbf{1}(N(t_j,d,S_l^*)>0,\ c_d(t_j)\subset D)}{\sum_{j=1}^{j}\mathbf{1}(c_d(t_j)\subset D)}.\end{aligned} \tag{2.15}$$

Turning now to $K(d)$, we consider the term $\mathbb{E}_S\big[\sum_{s_i\in S\cap D} N_D(s_i,d,S)\big]$. Following a similar argument to our calculation in (2.6), we get

$$\mathbb{E}_S\sum_{s_i\in S\cap D} N_D(s_i,d,S) = \lambda(D)\,\mathbb{E}_S\big[N_D(s,d,S)\big]. \tag{2.16}$$

As before, we can adjust (2.16) to get

$$\mathbb{E}_S \sum_{s_i \in S \cap D} \frac{N_D(s_i, d, S)}{N(D)} = \mathbb{E}_S\big[N_D(s, d, S)\big]. \tag{2.17}$$

If we call $\mathbb{E}\big[N_D(s, d, S)\big] = \lambda K_D(d)$, we have an immediate Monte Carlo integration for $K_D(d)$, i.e.,

$$K_D(d) \approx \frac{1}{L} \sum_{l=1}^{L} \left[ \frac{\sum_{s_{li}^*} N(s_{li}^*, d, S_l^*)}{\lambda^{(l)}\, N^{(l)}(D)} \right] \tag{2.18}$$

using the posterior draws $\lambda^{(l)}$ and posterior point patterns $S_l^*$, each drawn from an HPP($\lambda^{(l)}$).

Again, we see the need for edge correction. We are estimating $K_D(d)$ rather than $K(d)$. In fact, since, again $N_D(s, d, S) \leq N(s, d, S)$, we see that $K_D(d) \leq K(d)$.

Now consider that $\sum_{s_i \in S \cap D} N_D(s_i, d, S) = \sum_{s_i \in S \cap D} \sum_{j \neq i} \mathbf{1}(s_j \in c_d(s_i) \cap D)$. Given $s_i$, $\mathbb{E}\,\mathbf{1}(c_d(s_i) \cap D) = Pr[c_d(s_i) \cap D]$. We want $Pr[c_d(s_i)]$, but again we are restricted to only observing $S \cap D$, so we can only observe $\mathbf{1}(s_j \in c_d(s_i) \cap D)$. Instead, note that

$$Pr[c_d(s_i)] = Pr[c_d(s_i) \cap D] / Pr[D \,|\, c_d(s_i)]. \tag{2.19}$$

The denominator provides the appropriate inflation of the probability to give us $Pr[c_d(s_i)]$.

In the literature, the empirical estimators employ an edge-correction factor $w_{s_i,s_j}$ which is similar to $Pr[D \,|\, c_d(s_i)]$. The adjustment $w_{s_i,s_j}$ proposed in Ripley (1977) calculates, for a given $s_j$, the proportion of the circumference of the circle centered at $s_i$ with radius $||s_i - s_j||$ which is contained in $D$. In other words, $w_{s_i,s_j}$ is a rough approximation to $Pr[D \,|\, c_{||s_i - s_j||}(s_i)]$. Exact expressions for $w_{s_i,s_j}$ are available for $D$ of special shape (in 2 dim, essentially a circle; see Illian et al., Appendix B). Yet we seek to estimate $Pr[D \,|\, c_d(s_i)]$, which for a homogeneous intensity is equal to

$|D \cap c_d(s_i)|/|c_d(s_i)|$, the proportion of $c_d(s_i)$ that is inside $D$. To handle arbitrary regions $D$, we must perform a Monte Carlo integration for each $s_i \in S \cap D$, to perform a Monte Carlo integration, i.e., draw points uniformly in $c_d(s_i)$ and obtain the proportion which also fall in $D$. This proportion is the Monte Carlo estimate $\tilde{w}_{s_i} \approx |D \cap c_d(s_i)|/|c_d(s_i)| = |D \cap c_d(s_i)|/(\pi d^2)$, which can be approximated within arbitrary precision.

The resulting edge-adjusted estimator for $N(s, d, S)$ would become

$$\sum_{s_i \in S \cap D} \frac{\sum_{j \neq i} \mathbf{1}(s_j \in c_d(s_i) \cap D)}{\tilde{w}_{s_i} N(D)} \tag{2.20}$$

which, with posterior predictive patterns $S_l^*$, would yield a Monte Carlo estimator for

$$\begin{aligned} \mathbb{E}_S \left[ \sum_{s_i \in S \cap D} \frac{\sum_{j \neq i} \mathbf{1}(s_j \in c_d(s_i) \cap D)}{N(D) Pr[D \,|\, c_d(s_i)]} \right] &= \mathbb{E}_S \left[ \sum_{s_i \in S \cap D} \frac{N_D(s_i, d, S)}{N(D) Pr[D \,|\, c_d(s_i)]} \right] \\ &= \mathbb{E}_S \left[ \frac{N_D(s, d, S)}{Pr[D \,|\, c_d(s)]} \right]. \end{aligned} \tag{2.21}$$

Given $s$, we can think of $N_D(s, d, S)$ as the number of successes in $N(s, d, S)$ Bernoulli trials with success probability $Pr[D \,|\, c_d(s)]$. So,

$$\mathbb{E}\big[N_D(s, d, S) \,|\, N(s, d, S), Pr[D \,|\, c_d(s)]\big] = N(s, d, S)\, Pr[D \,|\, c_d(s)] \tag{2.22}$$

$$\text{or } \mathbb{E}\left[ \frac{N_D(s, d, S)}{Pr[D \,|\, c_d(s)]} \,|\, N(s, d, S), Pr[D \,|\, c_d(s)] \right] = N(s, d, S). \tag{2.23}$$

Hence

$$\mathbb{E}\left[ \frac{N_D(s, d, S)}{Pr[D \,|\, c_d(s)]} \right] = \mathbb{E}\,\mathbb{E}\left[ \frac{N_D(s, d, S)}{Pr[D \,|\, c_d(s)]} \,|\, N(s, d, S), Pr[D \,|\, c_d(s)] \right] = \mathbb{E}\big[N(s, d, S)\big]. \tag{2.24}$$

As above, since we want $\mathbb{E}\big[N(s, d, S)/\lambda\big]$, we will put $\lambda^{(l)}$ in the denominator of the Monte Carlo integration as in (2.18). Note also that we are not employing the

"finite" $K$-function $K_{\text{fin}}(d)$ in (3.5.7) in Illian et al. Our resulting posterior estimator for $K(d)$ is

$$\tilde{K}(d) \approx \frac{1}{L} \sum_{l=1}^{L} \sum_{s_{li}^* \in S_l^* \cap D} \frac{\sum_{j \neq i} \mathbf{1}(s_{lj}^* \in c_d(s_{li}^*) \cap D)}{\tilde{w}_{s_{li}^*} \lambda^{(l)} N^{(l)}(D)}, \tag{2.25}$$

where $\tilde{w}_{s_{li}^*}$ is estimated through a Monte Carlo integration.

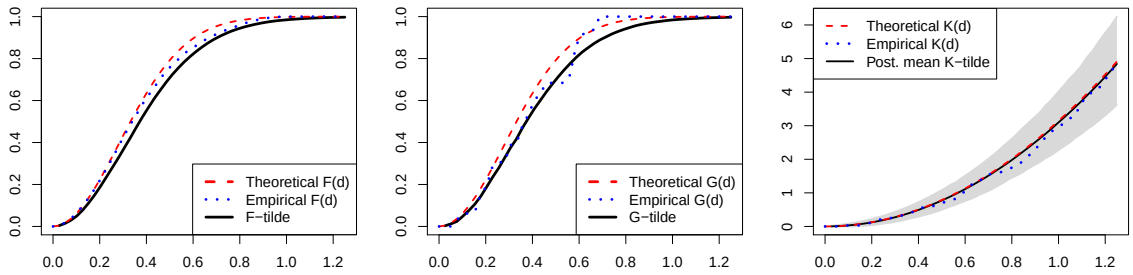
The form of the $K$-function also allows a posterior distribution due to the single Monte Carlo integration taking place outside the ratio. By removing the averaging over $l = 1, \ldots, L$, we also obtain the posterior draws for $K(d)$, which we will denote by $\tilde{K}_l(d)$ and calculate as

$$\tilde{K}_l(d) = \sum_{s_{li}^* \in S_l^* \cap D} \frac{\sum_{j \neq i} \mathbf{1}(s_{lj}^* \in c_d(s_{li}^*) \cap D)}{\tilde{w}_{s_{li}^*} \lambda^{(l)} N^{(l)}(D)}. \tag{2.26}$$

This approach for generating posterior samples of the $K$-function is valid for any point process with a constant intensity of a known form, such as an HPP. For a process where the intensity function is not known, such as a Strauss process, then another approach must be taken. This is one area of ongoing research.

### *2.1.4 F-, G-, and K-functions for Japanese Pines Data*

Figure 2.3 shows the posterior estimates for $\tilde{F}(d)$, $\tilde{G}(d)$, and $K(d)$ using the HPP model for the black pines data from Figure 2.1a. The empirical $F(d)$, $G(d)$, or $K(d)$ functions represent the standard nonparametric empirical estimates with appropriate edge corrections. The theoretical $F(d)$ and $G(d)$ represent the theoretical values using $\hat{\lambda} = n/|D|$ and the theoretical $K(d)$ is equal to $\pi d^2$. For $F(d)$ and $G(d)$, the posterior estimates $\tilde{F}(d)$ and $\tilde{G}(d)$ are given using equations (2.15) and (2.11), respectively, in combination with the posterior predictive point patterns. Figure 2.3c shows the posterior mean for $K(d)$ using equation (2.25) and a 95% pointwise credible interval for $K(d)$ using the posterior draws of $K(d)$ from equation (2.26).



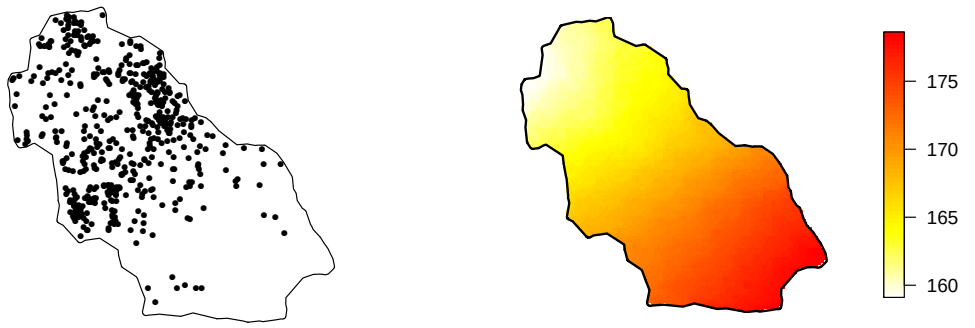

(a) (b) (c)

FIGURE 2.3: Posterior estimates for (a) $F(d)$, (b) $G(d)$, and (c) $K(d)$ under the HPP model for the Japanese black pines data. The theoretical forms use the MLE for $\hat{\lambda}$ and the empirical estimates are the standard nonparametric estimates. The shaded area represents the 95% pointwise credible intervals for $K(d)$.

$\tilde{F}(d)$ and $\tilde{G}(d)$ are generally lower than both their theoretical values and the empirical estimate. The $K$-function provides a notion of the uncertainty attached, and we see that the posterior mean for $K(d)$ was very similar to the theoretical and empirical curves. The credible interval contains both the theoretical and empirical curves, suggesting that there is no reason to reject the HPP model for this dataset. This dataset has often been used in the literature with more complex models, such as in Ogata and Tanemura (1981) and Baddeley and Turner (2000), yet the HPP model does not appear to be inadequate.

## 2.2 Nonhomogeneous Poisson Processes

Many point processes of interest are not expected to have a constant intensity over space or time. A nonhomogeneous Poisson process (NHPP) is a generalization of the HPP in which the intensity $\lambda(s)$ varies deterministically over space, though the points still occur independently over $D$. The number of points occurring in the subset $A$, denoted by $N(A)$, is still a Poisson random variable, but now with expectation $\lambda(A) = \int_A \lambda(s)ds$. For disjoint $A$ and $B$, $N(A)$ and $N(B)$ are still independent Poisson random variables.




(a) (b)

FIGURE 2.4: (a) The locations of 530 American sweetgum trees in a tract of Duke forest and (b) the elevation in meters over the same region.

### *2.2.1 NHPP Model and Duke Forest Data*

Covariate data can be informative about the intensity of the point process, leading to a regression model for $\lambda(s)$. A common specification for an inhomogeneous intensity is $\lambda(s) = \lambda_0 \exp\{x^T(s)\beta\}$, where $\lambda_0$ is the baseline intensity and $x(s)$ is a point-specific set of covariates providing a local adjustment to the intensity function. Many other forms for the NHPP intensity are possible and widely used, but we will only use this form here. Adapting the NHPP likelihood given in (1.3), the likelihood function becomes

$$\begin{aligned} f_S(S;\theta) &= \exp\big\{ -\textstyle\int_D \lambda(s)ds \big\} \prod_{s_i \in S} \lambda(s_i) \\ &= \exp\big\{ -\lambda_0 \textstyle\int_D \exp\{x^T(s)\beta\}ds \big\} (\lambda_0)^n \exp\big\{ \textstyle\sum_{s_i \in S} x^T(s_i)\beta \big\}. \end{aligned} \tag{2.27}$$

We now consider a specific point pattern consisting of the locations of American sweetgum trees (*Liquidambar styraciflua*) in a subplot of Duke forest, which surrounds Duke University in North Carolina, USA. This dataset was prepared by James S. Clark and Kai Zhu at Duke University. Figure 2.4a shows the locations of these trees within the tract of forest. The elevation is also available across a fine grid over the region, as shown in Figure 2.4b.

For this data, the elevation may be useful in estimating the intensity. Trees may be more likely to grow at certain elevations or maybe elevation will act as a surrogate for other significant, yet unobserved, covariates. In fact, the ecologists who collected this data suggest that the moisture levels in the soil exhibits a similar trend to elevation, where high elevation levels have lower soil moisture. A spatial trend might also be included, but we include only a linear and quadratic trend in elevation for now. The regression model we propose is written as

$$\log \lambda(s) = \log \lambda_0 + \beta_1 \, \texttt{elev}(s) + \beta_2 \, \texttt{elev}^2(s). \tag{2.28}$$

A Gamma($a_\lambda$, $b_\lambda$) prior distribution for $\lambda_0$ provides a conjugate prior distribution as before. The full conditional becomes $\lambda_0|\beta, S \sim \text{Gamma}(a_\lambda + n, b_\lambda + \int_D \exp\{x^T(s)\beta\}ds)$. In applications with little prior information, the Jeffreys prior for $\lambda_0$ would again be valid here. We use the prior $\lambda_0 \sim \text{Gamma}(a_\lambda = 1.3, b_\lambda = 50)$, which gives $\mathbb{E}[\lambda_0] = 0.026$. It may be simplest to expect *a priori* each $\beta_j = 0$ and then specify the prior for $\lambda_0$ by first specifying the prior for $\mathbb{E}[N(D)] = \lambda(D) = \lambda_0|D|$ and then calculating the implied prior for $\lambda_0$ as we did for the HPP model. This gives $\mathbb{E}[N(D)] \approx 500$ with a wide variance. Alternatively, the reparameterization from $\lambda_0$ to $\exp\{\beta_0\}$ is also an option here as before. No conjugate prior specifications exist for the regression coefficients $\{\beta_j\}$ due to the integral in the likelihood, but a normal distribution seems reasonable. Specifically, we use $\beta_j \overset{\text{ind}}{\sim} \text{Normal}(0, s^2)$ for $j = 1, 2$ and a large value $s^2$ (e.g., $s^2 = 1000$).

Fitting the model now requires Markov Chain Monte Carlo (MCMC) with a Metropolis-Hastings step for the $\beta_j$. We find that a random walk Metropolis-Hastings step for each $\beta_j$ is sufficient. The variance of the proposal distribution was adaptively tuned during the burn-in period to achieve a reasonable acceptance rate.

An important issue that now arises is the integral in the exponent of the likelihood function (2.27) and again in the complete conditional distribution for $\lambda_0$. Evaluating

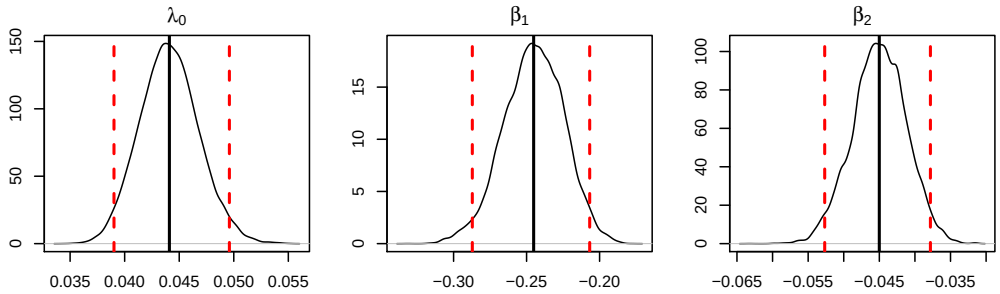


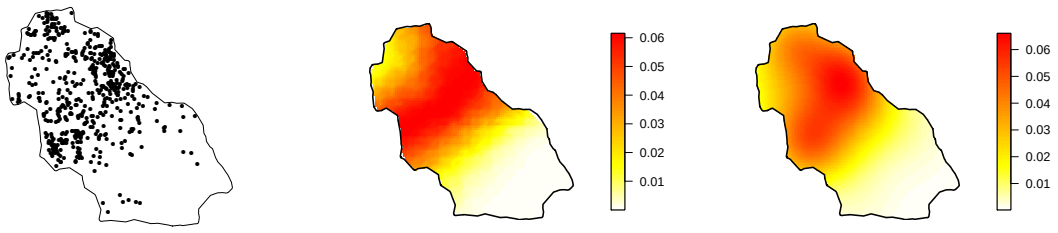

FIGURE 2.5: Posterior distributions for the parameters of the NHPP model. The posterior mean is marked by the solid vertical line and the 95% credible intervals are marked by the dashed lines.

the likelihood now involves calculating the integral $\int_D \exp\{x^T(s)\beta\}\, ds$ which has no analytical solution. Typically, Monte Carlo integration is used by discretizing the domain $D$ and evaluating the function $\exp\{x^T(s)\beta\}$ at the centroids of the grid cells. This is the ecological fallacy discussed in Banerjee et al. (2014), but from Figure 2.4b we can see that elevation can be reasonably assumed to be constant over small regions.

### *2.2.2 Posterior Inference for NHPPs*

We ran our MCMC scheme for 10,000 iterations of burn-in and then collected 20,000 posterior samples for each of our model parameters. MCMC convergence was monitored using standard techniques, such as the Gelman-Rubin diagnostic (Gelman and Rubin, 1992). Posterior draws of the intensity $\lambda(s)$ at any $s \in D$ can then be constructed using the posterior draws of $\lambda_0$ and each $\beta_j$.

Figure 2.5 shows the posterior distributions of $\lambda_0$ and each $\beta_j$. The regression coefficients all have 95% confidence intervals which do not contain zero, suggesting that they are significant to the model. The $X$ matrix was centered prior to fitting the model so that $\lambda_0$ is roughly interpretable as the average intensity across $D$. That is to say that at a point $s^*$ with average elevation, the intensity $\lambda(s^*)$ is about 0.044. A location that is 5 meters higher in elevation than the average has an intensity that




(a) (b) (c)

FIGURE 2.6: (a) The Duke forest data, (b) the posterior mean of the intensity surface for the NHPP model, and (c) the kernel intensity estimate.

is around $\exp\{5\beta_1 + 5^2\beta_2\} \approx 0.095$ percent of the intensity at the mean elevation. Note that many other types of trees exist on this tract of land and so this elevation effect is may also be affected by many other factors, e.g., competition in sweetgum tree presence.

### *2.2.3 Domain-level Inference*

Figure 2.6a gives the posterior mean of the intensity surface. Comparing this with Figure 2.4, we see that the intensity is higher where more points are observed. The intensity is also low in the bottom right region of the domain where few sweetgum trees were observed. We can compare our fitted intensity with the empirical kernel intensity estimate in Figure 2.6c, where we see that the empirical estimate looks similar to our posterior mean. Kernel intensity estimates are easily computed and are available in the R package `spatstat` (Baddeley and Turner, 2005), but since they are nonparametric they provide little illumination as to the underlying process and no uncertainty quantification.

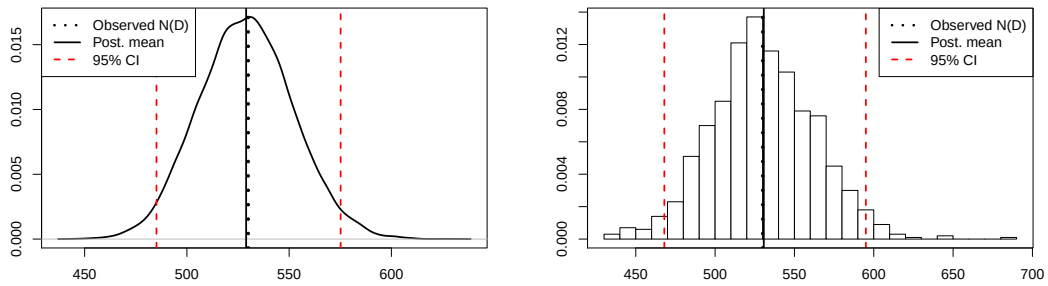
As before, our model can be used to provide many interesting posterior summaries of interest. For an NHPP model, the opportunities are more expansive and interesting than for the HPP model due to the addition of a spatially varying intensity. The HPP model had some nice properties, such as $\lambda(A) = \lambda|A|$ for any

$A \subseteq D$, which allowed us to directly calculate the posterior distribution for $\lambda(A)$. For inhomogeneous processes we will rely more heavily on our posterior predictive point patterns.

Generating these predictive point patterns is done using the Lewis-Shedler thinning approach (Lewis and Shedler, 1979). Their approach draws a point pattern from an HPP with intensity $\lambda_{max} \equiv \sup_{s \in D} \lambda(s)$ and then thins the sampled points using rejection sampling, where each point is retained independently with probability $\lambda(s)/\lambda^{max}$. The resulting point process can be shown to come from a nonhomogeneous Poisson process with intensity $\lambda(s)$. We employ this algorithm to generate $L$ posterior predictive point patterns, with each point pattern $S_l^*$ arising from an NHPP with intensity $\lambda^{(l)}(s)$, where the $l^{\text{th}}$ posterior samples of $\lambda_0$, $\beta_1$, and $\beta_2$ are used to construct $\lambda^{(l)}(s)$. We again assume that $L = 1000$ predictive patterns will be more than sufficient, so we actually thin our posterior samples to get $L$ thinned $\lambda^{(l)}(s)$ surfaces to generate the $S_l^*$ point patterns.

How many points should be expected in the domain $D$, given our model and the data we have observed? Again, we can start by looking at the posterior for $\lambda(D) = \int_D \lambda(s)ds$, which is the posterior for the *expected* number of points we should expect in $D$. This integral had to approximated to evaluate the integral in (2.27) during the model fitting, so these posterior samples have already been computed. Figure 2.7a shows the posterior distribution of $\lambda(D)$, which has a posterior mean of 528.97 with a 95% credible interval of (485.12, 575.18).

A more intuitive answer to this question, however, is again given by obtaining the posterior for the actual number of trees in $D$, rather than the posterior distribution for the expected number of trees. This is where the posterior predictive point patterns begin to prove useful. Counting the number of points in each $S_l^*$ is in fact also the posterior distribution for the number of points observed in $D$. Figure 2.7b shows the posterior distribution for $N(D)$, the number of trees arising in $D$, which has a




(a) (b)

FIGURE 2.7: The posterior distributions for (a) $\lambda(D)$ and (b) $N(D)$ in the Duke forest NHPP model.

posterior mean of 530.70 and a 95% credible interval of (468, 595). This distribution is centered around the posterior mean of $\lambda(D)$, since $\mathbb{E}[N(D)] = \lambda(D)$, but with wider spread due to integrating over the Poisson variability.

### *2.2.4 Point-level Inference*

With a spatially varying intensity $\lambda(s)$, we now have local posterior distributions for the intensity $\lambda(s)$ at any point $s \in D$. The posterior distribution not only provides us with a point estimate and quantification of the uncertainty in the intensity at each point, but it also allow comparisons between intensities at different points over the domain. For example, it may be of interest to test whether tree density is significantly different at two points in the forest tract. Researchers analyzing a point pattern of cancer cases may wish to test whether cancer rates are significantly higher in regions of interest, such as factories or large cities. These comparisons are more often of interest at the block level, with counties or plots of land in mind, as will be demonstrated in the next section.

Figure 2.8 shows the posterior distributions of $\lambda(s)$ at three points in the study region. All three points have distinctly different intensities with non-overlapping credible intervals. The second point, $s_2$, falls where the intensity is near its highest,

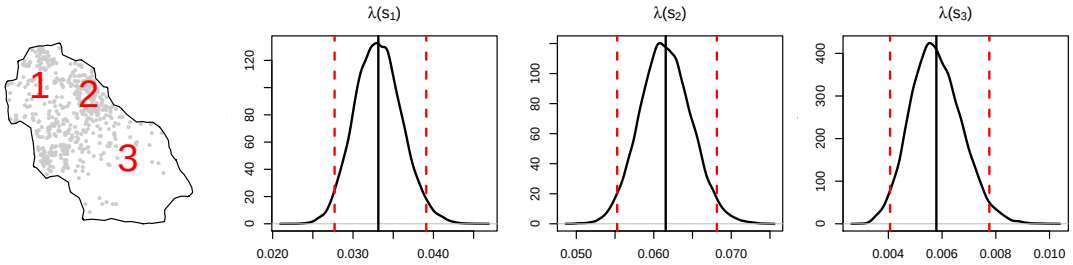



FIGURE 2.8: The posterior distributions for $\lambda(s)$ at three points in Duke forest. The solid vertical lines represent the posterior means and the dashed vertical lines represent the 95% credible intervals.

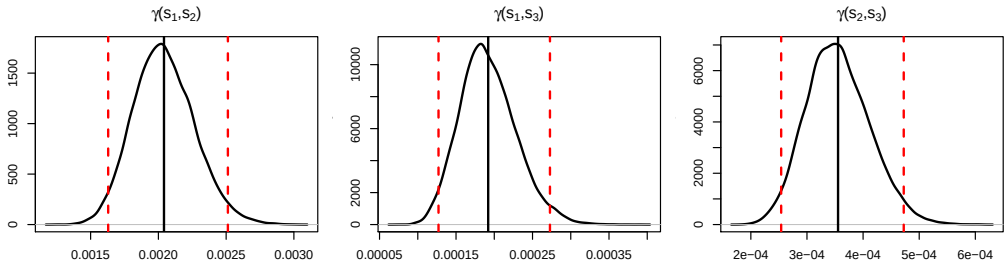



FIGURE 2.9: The posterior distributions for $\gamma(s, s')$ at three points in Duke forest. The locations of the three points are given in Figure 2.8. The solid vertical lines represent the posterior means and the dashed vertical lines represent the 95% credible intervals.

resulting in an intensity that is an order of magnitude higher than for the third point, $s_3$. With these posterior draws, we can also calculate quantities such as $Pr[\lambda(s_2) \geq \lambda(s_1) \,|\, S_{obs}] \approx 1$ or $Pr[\lambda(s_3 \geq 0.0075 \,|\, S_{obs}] = 0.042$.

Since the independence of points in a Poisson process holds for the NHPP model, we can decompose the second-order intensity as $\gamma(s, s') = \lambda(s)\lambda(s')$. Figure 2.9 shows the posterior distributions for $\gamma(s, s')$ at each combination of the same three points used in Figure 2.8. With the clear relation between $\gamma$ and $\lambda$ for Poisson processes, the results in this figure are not surprising. The independence also implies that the pair correlation function is equal to one, or $\tilde{g}(s, s') = 1$.

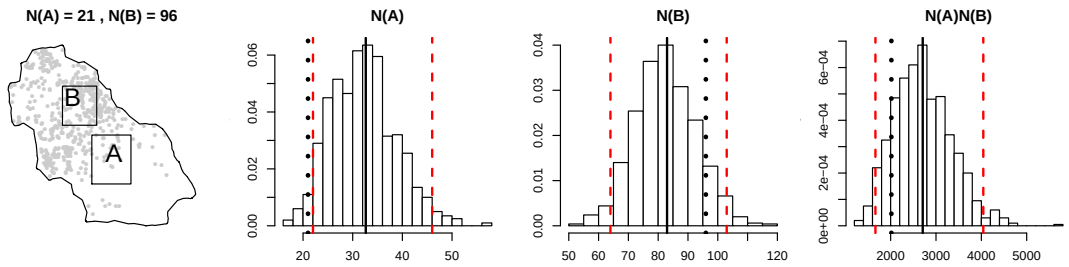



FIGURE 2.10: The posterior distributions for $N(A)$, $N(B)$, and $N(A)N(B)$ under the NHPP model for the Duke forest data. The solid lines give the posterior means, the dashed lines give the 95% credible intervals, and the dotted lines give the observed values.

### *2.2.5 Block-level Inference*

Block-level inference is arguably more interesting than point-level inference, in that the point-level intensities are only useful for comparing the intensity at one point to another, whereas block-level inference allows one to look at the distribution of intensities and counts over blocks. For example, one can now compare counts of trees in different tracts of land or counts of lung cancer in different counties and learn whether the underlying intensities are different. As before, for two regions $A$ and $B$, we can compare the posterior distributions of $\lambda(A)$ and $\lambda(B)$ , but looking at the distributions of $N(A)$ and $N(B)$ seems more intuitive and therefore more useful.

Figure 2.10 shows the posterior distributions for $N(A)$ and $N(B)$ for the regions $A$ and $B$ shown in the figure. Of course, there is nothing special about $A$ and $B$ being squares, besides ease of computation. We see that the posterior distribution for $N(A)$ is centered around 33, with a 95% posterior credible interval of (22, 46), which barely excludes the observed $N(A) = 21$. The plot for $N(B)$ is much better and shows that on average we slightly underestimate $N(B)$ using the fitted model. We observed $N(B) = 96$ and the posterior mean was 83 with a 95% credible interval of (64, 103). Since $A$ is a low-intensity area and $B$ is a high-intensity area, it seems

that the NHPP model intensity was overly smooth and shrunk towards the mean, though these are only two regions out of many possible choices.

We might also be interested in the joint distribution of two regions of the domain, just as we previously looked at the joint distribution of two points in the domain. For disjoint $A$ and $B$, the Poisson process implies that the distributions of $N(A)$ and $N(B)$ are independent conditional on $\lambda(s)$. But what about for non-disjoint $A$ and $B$? Clearly the answer again is easily found using these posterior predictive point patterns. For any $A, B \subseteq D$, whether disjoint or not, the $S_l^*$ allow us to learn about the joint distribution or any function of any of $N(A)$ and $N(B)$, such as $N(A) \times N(B)$, $N(A) + N(B)$, etc. This extends of course to joint distributions of more than two sets just as easily. All that is required is counting the function of interest for each $S_l^*$.

Figure 2.10 shows the posterior distribution of $N(A)N(B)$. The distribution $N(A)N(B)$ has a posterior mean of 2708 with a 95% credible interval of (1664, 4043), so most of its mass is above the observed $N(A)N(B) = 2016$. This is not too surprising since we already saw that we tended to overpredict $N(A)$ significantly. Variants of these plots will be useful as model diagnostics, which we discuss later.

### *2.2.6 Inhomogeneous K-function*

For an inhomogeneous point process, the typical nonparametric estimates for the inhomogeneous $F$- and $G$-functions (van Lieshout, 2011) and the inhomogeneous $K$-function (Baddeley et al., 2000) take on different forms from the homogeneous case.

From these two papers, the standard edge-corrected estimates are:

$$\hat{F}_{\text{inhom}}(d) = 1 - \frac{\sum_{t_j \in T \cap D_{\ominus d}} \prod_{s_i \in S \cap B(t_j,d)} [1 - \tilde{\lambda}/\hat{\lambda}(s_i)]}{N(T \cap D_{\ominus d})} \tag{2.29}$$

$$\hat{G}_{\text{inhom}}(d) = 1 - \frac{\sum_{s_i \in S \cap D_{\ominus d}} \prod_{s_j \in S \setminus \{s_i\} \cap B(s_i,d)} [1 - \tilde{\lambda}/\hat{\lambda}(s_j)]}{N(S \cap D_{\ominus d})} \tag{2.30}$$

$$\hat{K}_{\text{inhom}}(d) = \frac{1}{|D|} \sum_{s_i \in S \cap D} \sum_{s_j \in S \cap D \setminus \{s_i\}} \frac{\mathbf{1}(||s_i - s_j|| \leq d)}{w_{s_i,s_j} \hat{\lambda}(s_i) \hat{\lambda}(s_j)} \tag{2.31}$$

where $T$ is a set of points $\{t_k\}$ over $D$ as used before, $D_{\ominus d}$ is an erosion of $D$ defined by the set $\{s \in D : ||s - \partial D|| \geq d\}$, $\partial D$ denotes the boundary of $D$, $B(s,d)$ denotes a ball of radius $d$ centered around $s$, $\tilde{\lambda}$ is defined as $\tilde{\lambda} \equiv \inf_{s \in D} \hat{\lambda}(s)$, and $w_{s_i,s_j}$ is Ripley's edge-correction factor previously introduced. These estimates for $F(d)$, $G(d)$, and $K(d)$ all require an intensity estimate $\hat{\lambda}(s)$ which typically comes from an empirical kernel intensity estimate.

The homogeneous $F$-, $G$-, and $K$-functions are generally defined using the notion of a *typical point* in the point pattern, or a point which can be taken to be representative of the other points in the pattern. For such a point, we have looked at quantities such as $N(s,d,S)$, but we could equivalently write $N(0,d,S-s)$, where $S-s$ denotes shifting each point in $S$ by the vector $-s$, so that $S - s \equiv \{s_i - s : s_i \in S\}$. However, with a spatially varying intensity function, the notion of a typical point is lost. The estimates in (2.29)–(2.31) attempt to construct a notion of a typical point by adjusting for the intensity function, though (2.29) and (2.30) are not very intuitive.

Further, the homogeneous $F$- and $G$-functions are defined such that they are not scale-free. Therefore, the $F$- and $G$-functions do not seem to be well-defined for a spatially varying process. One option is to ignore the fact that the intensity varies and use the same estimators as in the homogeneous case, which will essentially provide a weighted average of the nearest neighbor distances. The areas of high intensity will

have relatively smaller nearest neighbor distances, while the areas of lower intensity will exhibit larger nearest neighbor distances. However, this approach will provide estimates that are strictly tied to the observation window $D$ over which they are estimated. In other words, they will not provide a good estimate for $F(d)$ and $G(d)$ in some other region $D'$. For a finite point pattern, this option may make sense, however.

The second option for the inhomogeneous $F(d)$ and $G(d)$ is to attempt to decouple the spatial variation from the process, as is done in (2.29) and (2.30). Since the NHPP produces points with independent locations, it seems reasonable that accounting the spatial trend may result in nearest neighbor distances that are HPP-like. From the equations above and in van Lieshout (2011), it appears that they should be similar to those under an HPP($\tilde{\lambda}$). This interpretation may be useful in exploratory data analysis when looking for signs of clustering or repulsion, but this does not make sense when looking for posterior features of our model that illuminate our understanding of the process. The model we have used, an NHPP in this case, already makes the assumption of independent locations, so this interpretation doesn't provide a meaningful summary of the posterior distribution over our model.

The $K$-function, however, is defined to be scale-free and has a much clearer interpretation for a spatially varying process. Baddeley et al. (2000) propose that $K(d)$ can also be defined, rather than using the typical point notation, by using the pair correlation function $\tilde{g}$, which becomes useful for inhomogeneous point processes. Their definition assumes that the process is second-order reweighted stationary, meaning that the pair correlation function $\tilde{g}(s, s') = \gamma(s, s')/\lambda(s)\lambda(s')$ is just a function of $||s - s'||$, or $\tilde{g}(s, s') = \tilde{g}_0(||s - s'||)$ for some function $\tilde{g}_0$. This allows us to use the alternate definition of K-function given in equation (4) of Baddeley et al. (2000) as

$$K_{0,\text{inhom}}(d) \equiv 2\pi \int_0^d \tilde{g}_0(t)dt. \tag{2.32}$$

The equivalence between (2.31) and (2.32) can be seen when applying the bivariate form for Campbell's Theorem to (2.31). Evidently, one simple way to construct posterior draws of $K_{\text{inhom}}(d)$ would be to first construct posterior draws of $\tilde{g}_0(t)$ and integrate it over $B(0,d)$. For an NHPP, $\tilde{g}_0(t) = 1$ so this would not be very illuminating. For other inhomogeneous processes, however $\tilde{g}_0(t)$ will not be constant and so may be of greater worth.

Another option for constructing posterior draws of $K_{\text{inhom}}(d)$ is to start with the uncorrected form of the estimator in (2.31). It can then be calculated that

$$\mathbb{E}_S \frac{1}{|D|} \sum_{s_i \in S \cap D} \sum_{s_j \in S \cap D \setminus \{s_i\}} \frac{\mathbf{1}(||s_i - s_j|| \leq d)}{\lambda(s_i)\lambda(s_j)} \tag{2.33}$$

$$= \mathbb{E}_S \frac{1}{|D|} \sum_{s_i \in S \cap D} \frac{1}{\lambda(s_i)} \left[ \sum_{s_j \in S} \frac{\mathbf{1}(s_j \in c_d(s_i) \cap D)}{\lambda(s_j)} \right]. \tag{2.34}$$

Note that the right-hand side cannot be collapsed into a form involving $N_D(s_i, d, S)$ as we did earlier in the homogeneous case.

Again, the need for an edge correction becomes apparent. We would like to modify (2.34) to get

$$\mathbb{E}_S \frac{1}{|D|} \sum_{s_i \in S \cap D} \frac{1}{\lambda(s_i)} \left[ \sum_{s_j \in S} \frac{\mathbf{1}(s_j \in c_d(s_i))}{\lambda(s_j)} \right], \tag{2.35}$$

where the inner sum is not restricted to points in $D$. To make the correction, we only need $\mathbb{E}\big[\mathbf{1}(s_j \in c_d(s_i) \cap D)/\lambda(s_j)\big]$ and $\mathbb{E}\big[\mathbf{1}(s_j \in c_d(s_i))/\lambda(s_j)\big]$. Given $s_i$, we can

compute that

$$\frac{\mathbb{E}\left[\frac{\mathbf{1}(s_j \in c_d(s_i) \cap D)}{\lambda(s_j)}\right]}{\mathbb{E}\left[\frac{\mathbf{1}(s_j \in c_d(s_i))}{\lambda(s_j)}\right]} = \frac{\int_D \frac{\mathbf{1}(s \in c_d(s_i) \cap D)}{\lambda(s)} \lambda(s)\, ds}{\int_D \frac{\mathbf{1}(s \in c_d(s_i))}{\lambda(s)} \lambda(s)\, ds}$$

$$= \frac{|c_d(s_i) \cap D|}{|c_d(s_i)|}$$

$$= \frac{|c_d(s_i) \cap D|}{\pi d^2}$$

$$= Pr[D|c_d(s_i)]$$

$$= \tilde{w}_{s_i}, \tag{2.36}$$

the same edge correction proposed previously for the homogeneous $K$-function.

So we can create a Monte Carlo integration of (2.34) and correct it, for each $s_i$, using the same edge correction we used in the homogeneous case: $\tilde{w}_{s_i} = (|c_d(s_i) \cap D|)/(\pi d^2)$. Putting all this together, we can create posterior draws for $K_{\text{inhom}}(d)$

$$\tilde{K}^{(l)}_{\text{inhom}}(d) = \frac{1}{|D|} \sum_{s^*_{li} \in S^*_l} \frac{1}{\tilde{w}_{s^*_{li}} \lambda^{(l)}(s^*_{li})} \left[ \sum_{j \neq i} \frac{\mathbf{1}(s^*_{lj} \in c_d(s^*_{li}))}{\lambda^{(l)}(s^*_{lj})} \right], \tag{2.37}$$

which allows a Monte Carlo integration to give the expected value

$$\tilde{K}_{\text{inhom}}(d) = \frac{1}{L\,|D|} \sum_{l=1}^{L} \sum_{s^*_{li} \in S^*_l} \frac{1}{\tilde{w}_{s^*_{li}} \lambda^{(l)}(s^*_{li})} \left[ \sum_{j \neq i} \frac{\mathbf{1}(s^*_{lj} \in c_d(s^*_{li}))}{\lambda^{(l)}(s^*_{lj})} \right]. \tag{2.38}$$

### *2.2.7 Inhomogeneous K-function for Duke Forest Data*

We now give the posterior distributions for the inhomogeneous $K$-function for the Duke forest data in Figure 2.11. $\tilde{K}(d)$ is quite similar to the theoretical value of $K(d) = \pi d^2$ for an NHPP, but that is also expected since we used an NHPP model. $\hat{K}_{\text{inhom}}(d)$ also produced as estimate that was quite similar to $\tilde{K}(d)$.

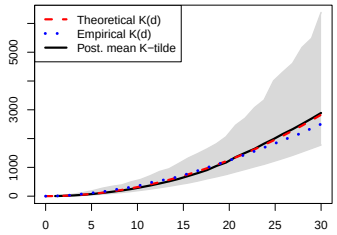



FIGURE 2.11: The posterior distributions for the inhomogeneous $K$-function under the NHPP model for the Duke forest data. The theoretical and empirical estimates, as computed in `spatstat`, are also given.

## 2.3 Log-Gaussian Cox Processes

A common extension of the nonhomogeneous Poisson process is the doubly stochastic Cox process (Cox, 1955). Cox processes are generated as NHPPs with a random intensity function $\Lambda(s)$. That is, if $X$ is a Cox process with intensity process $\Lambda(s)$, then conditional on $\Lambda(s) = \lambda(s)$, $X|\Lambda(s)$ is an NHPP with intensity $\lambda(s)$. Two important properties of Cox processes are a) if $\Lambda(s)$ is stationary then $X$ is stationary and b) it is impossible to distinguish a Cox process from an NHPP when only one realization of the point process is available (Møller and Waagepetersen, 2007).

We wish to focus here on the log-Gaussian Cox process (LGCP), a Cox process characterized by the log of the intensity surface arising from a Gaussian process (Møller et al., 1998). Gaussian processes are widely used in many applications, such as Gaussian process regression and geostatistics, for their simplicity and flexibility. Hence log-Gaussian processes provide a natural and flexible framework for modeling both density and intensity functions. In fact, Tokdar and Ghosh (2007) show posterior consistency for using log-Gaussian processes in density estimation, which is very similar to intensity estimation for point processes.

We note that from a Bayesian modeling standpoint, putting a prior on the parameters of $\lambda(s)$ for the NHPP in the previous sections made $\lambda(s)$ itself a random

process already. Thus, the distinction between NHPPs and Cox processes may be less meaningful in Bayesian models, but the extra flexibility provided by the Gaussian process prior on $\log \lambda(s)$ can be very beneficial. We will therefore continue to use the notation $\lambda(s)$ when discussing the intensity of the LGCP, rather than the more correct form $\Lambda(s)$.

### *2.3.1 LGCP Model*

Let $Z$ arise from a Gaussian process with mean $m(s)$ and covariance function $c(s, s')$, denoted by $Z \sim \text{GP}(m, c)$. We assume the covariance function can also be written as $c(s, s') = \sigma^2 c_0(s, s')$, where $c_0(s, s')$ is the correlation function. The model for the intensity of an LGCP can now be written as

$$\lambda(s) = \lambda_0 \exp\{x^T(s)\beta + Z(s)\}. \tag{2.39}$$

This leads to the LGCP likelihood taking the form

$$\begin{aligned} f_S(S;\theta) &= \exp\{-\textstyle\int_D \lambda(s)ds\} \prod_{s_i \in S} \lambda(s_i) \\ &= \exp\big\{ -\lambda_0 \textstyle\int_D \exp\{x^T(s)\beta + Z(s)\}ds\big\} (\lambda_0)^n \exp\big\{ \textstyle\sum_{s_i \in S} x^T(s_i)\beta + Z(s_i)\big\}. \end{aligned} \tag{2.40}$$

Møller et al. (1998) provide some discussion about the choice of covariance function. Though the covariance function can be specified to be of the same form as are common to Gaussian processes, some care is needed in specifying the priors for the hyperparameters. Looking back at (2.40), the likelihood function is maximized when $\lambda(s)$ is high at each $s_i \in S$ and close to zero everywhere else. In other words, it seems that the data prefer a peaked intensity function with high peaks at the data points and low values everywhere else. With diffuse prior specification, the parameter values will almost exclusively be determined by the likelihood of the observed data given those parameters. We therefore take the view that modeling the intensity

function is a matter of smoothing, similar to using a conditional autoregressive model for modeling the spatial random effects for areal data.

The common approach of using a kernel intensity estimate inherently gives the intensity estimate a certain level of smoothness determined by the kernel bandwidth. Choosing the bandwidth parameter is nontrivial and is often chosen to reflect the user's conception of the smoothness of the true intensity function. The decision for choosing a default kernel bandwidth is usually made through leave-one-out cross-validation, as introduced in Diggle and Marron (1988) and developed for density estimation by Bowman (1984). Diggle and Marron's method chooses an optimal bandwidth $\hat{h}_{CV}$ as the minimizer of the cross-validation score

$$\int [\hat{\lambda}_h(s)]^2 \, ds - (2/n) \sum_{j=1}^{n} \hat{\lambda}_{h,j}(s_j) \tag{2.41}$$

where $\hat{\lambda}_h(s)$ is the kernel intensity at $s$ with bandwidth $h$ and $\hat{\lambda}_{h,j}(s_j)$ is the leave-one-out kernel intensity estimate with $s_i$ removed. It is clear that this score is chosen to guard against overfitting the observed data.

Another consideration in prior specification for the LGCP is that of parameter identifiability. In the geostatistical setting, a Gaussian process is commonly used to model spatial random effects. The model might look something like $Y(s) = \mu + \omega(s) + \epsilon(s)$, where $Y(s)$ is some observed outcome at location $s$, $\omega(s)$ is a spatial random effect, and $\epsilon(s)$ is the error. In this setting, a rough estimate of the spatial effect can be constructed as $\hat{\omega}(s) = Y(s) - \hat{\mu}$, for some estimate $\hat{\mu}$ such as $\overline{y}$. In this setting, Zhang (2004) shows that the parameters of the Gaussian process $\omega$ are only fully identified up to some function of the parameters, such as $\sigma^2\phi^{2\nu}$ for the Matérn covariance function. In the point process setting, there is no rough estimate of the spatial effect, but rather we only observe locations of points with no explicit indication of the intensity at that point. In this setting with less information about the

spatial random effects, the ability to identify the parameters of the Gaussian process is further diminished. Therefore it seems that good prior information is needed for the parameters of the latent Gaussian process, otherwise an informative prior should be specified to give the desired amount of smoothing. This latter option can be explored by drawing prior predictive intensity surfaces to visualize the smoothness implied by the current prior.

In the absence of good prior knowledge about the hyperparameters, our preference is to estimate $\phi$ at its minimum contrast estimate using the pairwise correlation function (Møller et al., 1998), which we denote by $\tilde{\phi}$. In our experience, based on extensive simulation, this estimate seemed to be more robust than the $K$-function minimum contrast estimate. By optimizing the hyperparameters for this second-order functional, the minimum contrast estimate at least partially overcomes the issue explained above of likelihood-based methods trying to overfit the observed data with a highly peaked intensity. With $\phi$ fixed, $\sigma^2$ will now be better identified. We suggest using either a log-normal or gamma distribution for $\sigma^2$, preferably centered around its minimum contrast estimate $\tilde{\sigma}^2$.

The priors for our model are now given by

$$\lambda_0 \sim \text{Gamma}(a, b) \tag{2.42}$$

$$\beta_j \overset{iid}{\sim} \text{Normal}(0, s_\beta^2), j = 1, \ldots, p \tag{2.43}$$

$$Z \sim \text{GP}(-c/2, c), \text{ with } c(s, s') = \sigma^2(1 + \phi||s - s'||)\exp\{-\phi||s - s'||\} \tag{2.44}$$

$$\sigma^2 \sim \text{Log-Normal}(\tilde{\sigma}^2, s_\sigma^2) \tag{2.45}$$

$$\phi = \tilde{\phi}. \tag{2.46}$$

The covariance function used in (2.44) corresponds to a Matérn covariance function with smoothness $\nu = 3/2$, which was chosen after discussions with ecologists involved in the project. For $Z \sim \text{GP}(m, c)$, $\mathbb{E}[\exp\{Z(s)\}] = \exp\{m(s) + c(s, s)/2\}$, so setting $m(s) = -c(s, s)/2 = -\sigma^2/2$ as in (2.44) gives $\mathbb{E}[\exp\{Z(s)\}] = \exp\{-\sigma^2/2 + \sigma^2/2\} =$

1. This means that the expected spatial adjustment is 1, which along with a centered $X$ matrix tries to preserve $\lambda_0$ as the baseline intensity.

Sampling $\lambda_0$ and the $\beta_j$ can be done as discussed previously. Sampling $Z$ cannot be done through Gibbs sampling as in the geostatistical setting, and simple Metropolis-Hastings samplers seem to get stuck easily in local modes. Thus, more advanced MCMC methods are required to efficiently sample $Z$. The most common approach in literature is to use a Metropolis-adjusted Langevin algorithm (MALA), as discussed in Møller et al. (1998) and Christensen et al. (2005). Girolami and Calderhead (2011) provide some extensions, including Hamiltonian Monte Carlo methods, for a more robust method which requires less tuning of the algorithm. Murray et al. (2010) and Murray and Adams (2010) develop a slice sampling algorithm for latent Gaussian fields and their hyperparameters, called Elliptical Slice Sampling (ESS). Simpson et al. (2011) show that approximation of the Gaussian field by a Gaussian Markov random field allows the use of integrated nested Laplace approximation (INLA) to provide a computationally attractive alternative to fitting LGCPs. The Poisson-gamma process (Wolpert and Ickstadt, 1998) and Dirichlet process mixture of Beta processes (Kottas, 2006) are other flexible alternatives to LGCPs.

Each of these algorithms has different benefits and different complexities involved. We employ elliptical slice sampling here, which has the benefits of being intuitive, easy to implement, and requires no matrix inversions or estimation of the Fisher information matrix. We found Algorithm 2 in Murray and Adams (2010) to work well for updating the hyperparameters, which in our case will just be $\sigma^2$, and then employs elliptical slice sampling for updating $Z$. A brief description of elliptical slice sampling and how it is used in our specific context is given in the next section.

It is important to note that each of the algorithms for fitting LGCPs requires discretizing $Z$ to a finite-dimensional grid over the domain $D$ to evaluate the integral

in the exponent of the likelihood function (2.40). Typically, Monte Carlo integration is used by discretizing the domain $D$ and evaluating the function $\exp\{x^T(s)\beta + Z(s)\}$ at the centroids of the grid cells, similar to what was done for the NHPP model. This may always not provide an accurate approximation as discussed in Banerjee et al. (2014), but Waagepetersen (2004) shows that the approximation converges to the exact value as the size of the discretized grid cells goes to zero. Other approaches which avoid this integral approximation have been investigated in Wolpert and Ickstadt (1998), Kottas (2006), Adams et al. (2009) and Simpson et al. (2011).

### *2.3.2 Elliptical Slice Sampling for LGCPs*

We now briefly summarize the use of elliptical slice sampling for LGCPs, based on the work in Murray et al. (2010) and Murray and Adams (2010). Let $\mathbf{f}$ denote the finite-dimensional discretization of $Z$. We can express the distribution of $\mathbf{f}$ as a multivariate normal distribution with mean zero and covariance matrix $\Sigma$, or $\mathbf{f} \sim \text{Normal}(\mathbf{0}, \Sigma)$, where $\Sigma_{ij} = \sigma^2 \exp\{-\phi||s_i - s_j||\}$. Murray and Adams (2010) discuss how the dependence between $\mathbf{f}$ and its hyperparameters can make it difficult to efficiently sample the parameters. They suggest a transformation, as also discussed in Christensen et al. (2005), to "whiten the prior," or remove some of the dependence between our parameters. This is done by transforming $\mathbf{f}$ to $\boldsymbol{\nu}$, where $\mathbf{f} = L_{\Sigma_\theta}^T \boldsymbol{\nu}$, $\Sigma = L_{\Sigma_\theta}^T L_{\Sigma_\theta}$, and $\theta$ denotes the hyperparameters $(\sigma^2, \phi)$ of our Gaussian process. It is clear that $\boldsymbol{\nu} \sim \text{Normal}(\mathbf{0}, \mathbf{I})$ and that $\boldsymbol{\nu}$ has no dependence on $\sigma^2$ and $\phi$. It will be necessary to compute $\mathbf{f}$ often when evaluating the likelihood, but $\boldsymbol{\nu}$ is now the actual model parameter.

Algorithm 1 shows how to apply the Elliptical Slice Sampling algorithm of Murray et al. (2010) under this "whitened" prior transformation to using $\boldsymbol{\nu}$. The algorithm works by drawing a new multivariate normal random variable $\boldsymbol{\eta}$ from the same distri-

bution as $\boldsymbol{\nu}$. Since these two variables are independent, they can be quite different. A proposal $\boldsymbol{\nu}'$ for $\boldsymbol{\nu}$ comes as a random point along the elliptical curve connecting $\boldsymbol{\nu}$ and $\boldsymbol{\eta}$. The proposal is evaluated using the standard Metropolis-Hastings acceptance ratio, which requires calculating the inferred proposal $\mathbf{f}' = L_{\Sigma_\theta}^T \boldsymbol{\nu}'$. If the proposal is not accepted, then the proposal region along the elliptical curve is shrunk and a new value is proposed. This process continues until a candidate $\boldsymbol{\nu}'$ is accepted.

**Algorithm 1** "Whitened" Variant of Elliptical Slice Sampling — Murray et al. (2010)

1: Choose ellipse $\boldsymbol{\eta} \sim N(0, \Sigma)$
2: Draw $u \sim \text{Uniform}(0, 1)$
3: Compute log-likelihood threshold $\log y \leftarrow \log \mathcal{L}(\mathbf{f}) + \log u$
4: Draw initial proposal $\omega \sim \text{Uniform}(0, 1)$
5: Calculate proposal bracket $[\omega_{\min}, \omega_{\max}] \leftarrow [\omega - 2\pi, \omega]$
6: $\boldsymbol{\nu}' \leftarrow \boldsymbol{\nu} \cos\omega + \boldsymbol{\eta} \sin\omega$
7: $\mathbf{f}' \leftarrow L_{\Sigma_\theta}^T \boldsymbol{\nu}'$
8: **if** $\log \mathcal{L}(\mathbf{f}') > \log y$ **then**
9: **return** $\boldsymbol{\nu}'$
10: **else**
11: Shrink the bracket and try a new point on the ellipse:
12: **if** $\omega < 0$ **then**
13: $\omega_{\min} \leftarrow \omega$
14: **else**
15: $\omega_{\max} \leftarrow \omega$
16: $\omega \sim \text{Uniform}(\omega_{\min}, \omega_{\max})$
17: **Go to** step 6

Algorithm 2 below describes how to employ algorithm 2 in Murray and Adams (2010) to update the hyperparameters of our latent Gaussian process. Since we have fixed $\phi$, we only consider updating $\sigma^2$ here, though both could updated jointly or through two separate uses of this algorithm. The update is really just a standard Metropolis-Hastings update, but it demonstrates how this fits in with $\boldsymbol{\nu}$ and $\mathbf{f}$. Murray and Adams (2010) discuss other options to further improve the efficiency of sampling the hyperparameters through introducing auxiliary variables, but we find this simpler method to be sufficient for our use.

**Algorithm 2** Metropolis-Hastings for GP hyperparameters $\theta$ — Murray and Adams (2010)

Propose $\theta' \sim q(\theta'; \theta)$
Compute implied value $\mathbf{f}' = L_{\Sigma_{\theta'}} \boldsymbol{\nu}$
Draw $u \sim \text{Uniform}(0, 1)$
**if** $u < \dfrac{\mathcal{L}(\mathbf{f}')\pi(\theta')q(\theta; \theta')}{\mathcal{L}(\mathbf{f})\pi(\theta)q(\theta'; \theta)}$ **then**
    **return** $\theta'$
**else**
    **return** $\theta$

### *2.3.3 Posterior Inference for LGCPs*

We now fit the model, running 10,000 iterations of burn-in and then taking 100,000 posterior samples. Elevation and squared elevation were again used as covariates. MCMC convergence was monitored as discussed previously. We can now employ all the same methods for posterior inference as before. Since we only require $L$ samples for our posterior analysis, we found it convenient to again thin the posterior parameter samples and only retain $L = 1000$ posterior samples, which also helps maintain the memory requirements at a reasonable level.

Figure 2.12 shows the posterior distributions for $\lambda_0$, $\beta_1$, $\beta_2$, and $\sigma^2$. The minimum contrast estimate $\tilde{\phi}$ was 0.0427. It appears that the linear effect for elevation was again significant here, judging by its credible interval not overlapping 0, with a posterior mean similar to that estimated by the NHPP model. The quadratic effect of elevation has a credible interval that does overlap 0, however, and the posterior mean is much closer to 0 than we saw for the NHPP. We also note that the posterior mean for $\lambda_0$ is essentially the same as the HPP MLE estimate $\hat{\lambda} = n/|D| = 0.0273$.

### *2.3.4 Domain-level Inference*

Figure 2.13 shows the posterior mean and kernel intensity estimate for $\lambda(s)$. The posterior mean is more peaked than the kernel intensity estimate, but has the same general trend. We also tried other covariance functions for this dataset, as will be

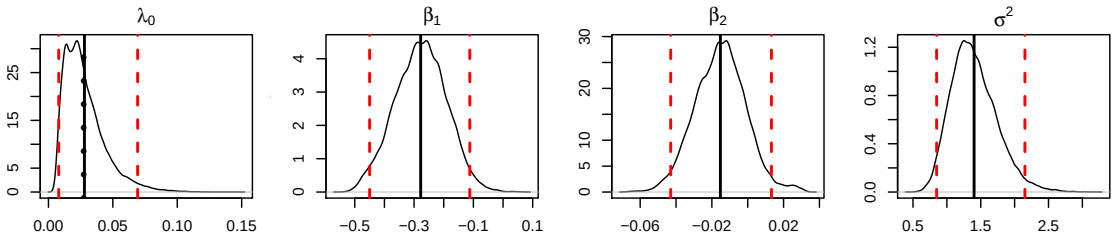
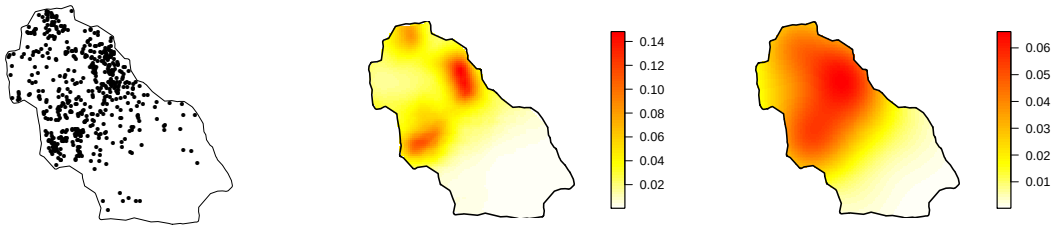



FIGURE 2.12: Posterior distributions for the parameters of the LGCP model. The posterior mean is marked by the solid vertical line and the 95% credible intervals are marked by the dashed lines. The HPP MLE $\hat{\lambda}$ is given by the dotted line.

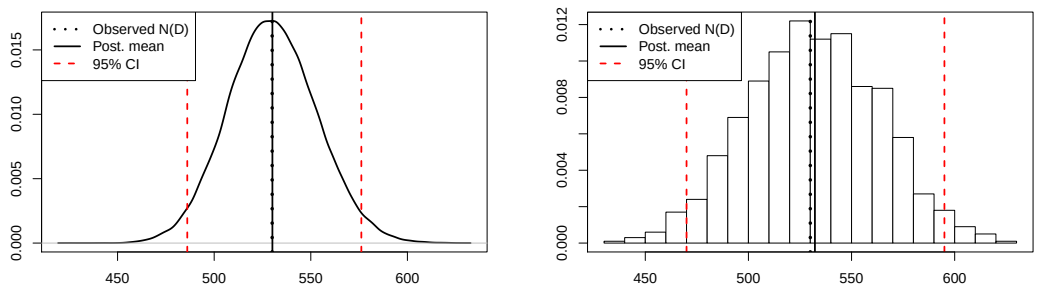



(a) (b) (c)

FIGURE 2.13: (a) The posterior mean of $\lambda(s)$ for the LGCP model and (b) the kernel intensity estimate for the Duke forest data.

discussed in the next chapter. The model using an exponential covariance function, for example, provided a posterior mean intensity that was even more peaked than we see in Figure 2.13b.

Figure 2.14 shows the posterior distributions for $\lambda(D)$ and $N(D)$. We see that both are centered around the observed value $n = 530$ and the distribution for $N(D)$ is more variable than $\lambda(D)$ as expected. $\lambda(D)$ has a posterior mean of 530.13 with a 95% credible interval of (486.07, 576.16). $N(D)$ has a posterior mean of 532.17 with a 95% credible interval of (471, 596). Both of these posterior distributions are very similar to those obtained using the NHPP model.




(a) (b)

FIGURE 2.14: The posterior distributions for (a) $\lambda(D)$ and (b) $N(D)$ in the Duke forest LGCP model. The observed value of $n = 530$ is denoted by the dotted line.

### *2.3.5 Point-level Inference*

The LGCP provides even finer spatial resolution in the intensity function, making point-wise comparisons more interesting. Figure 2.15 shows the posterior distributions for the intensity at the same three locations used earlier in Figure 2.8. The posterior distribution of the intensity at $s_2$ is notably higher here, due to the extra flexibility provided by the LGCP. Since $s_2$ is in a region of high intensity, the NHPP smoothed the intensity here more than the LGCP did, providing very different posterior distributions. The posteriors for $s_1$ and $s_3$ are more similar, though both are slightly shifted downwards from those obtained under the NHPP model. Finally, all three distributions here have a slight right-skew that was not present in the posteriors from the NHPP model, which again is likely due to the added flexibility in the LGCP model.

Point-level distributions can be useful in at least a few situations. For example, when monitoring a spatiotemporal point process, such as cancer cases over time, creating a posterior distribution for $\lambda(s)$ at some $s \in D$ of interest would allow comparisons over time to detect significant changes in the intensity over time. Similarly, if one were monitoring some process, it may be of interest to monitor when

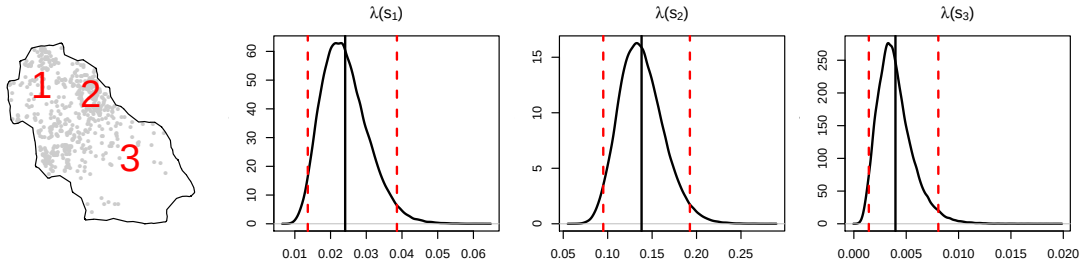



FIGURE 2.15: The posterior distributions for $\lambda(s)$ at three points in Duke forest under the LGCP model.

the intensity surges over some prespecified threshold, indicating abnormal behavior of the process or hazardous conditions. For example, telecommunications providers might be interested in monitoring dropped call rates over a city and intervening if a significant surge in the dropped call intensity is noted, possibly indicating a system malfunction. These same monitoring techniques could be used on regions of interest, but depending on the application, a point-level summary may be more useful.

For LGCPs, since $\lambda(s)$ is just a realization of the random process $\Lambda(s)$, the theoretical intensity function can be written as $\rho(s) = \mathbb{E}[\Lambda(s)] = \lambda_0 \exp\{x^T(s)\beta\}\,\mathbb{E}[\exp\{Z(s)\}]$. The intensity and pair correlation functions for LGCPs are given in Møller et al. (1998), though they present the simple stationary case of $\mathbb{E}[\Lambda(s)] = \mathbb{E}[\exp\{Z(s)\}] = \rho$. Adapting to the non-stationary case, the $n^{th}$-order intensity and pair correlation function can be written as

$$\rho^{(n)}(s_1, \ldots, s_n) = \exp\left\{ \sum_{i=1}^{n} m(s_i) + \frac{n\sigma^2}{2} + \sum_{1 \le i < j \le n} c(s_i, s_j) \right\} \tag{2.47}$$

$$\tilde{g}(s, s') = \frac{\rho^{(2)}(s, s')}{\rho^{(1)}(s)\rho^{(1)}(s')} = \exp\{c(s, s')\}, \tag{2.48}$$

where $m(s) = \mathbb{E}[Z(s)]$ and $c(s, s') = \sigma^2 c_0(s, s') = \mathrm{Cov}(Z(s), Z(s'))$ as introduced previously. We had previously used $\gamma(s, s')$ for the second-order notation, but here $\rho^{(2)}(s, s')$ will denote the second-order intensity for LGCPs.

Under our current model specification, with $m(s) = -\sigma^2/2$, we can also move $\lambda_0$ and $\exp\{x^T(s)\beta\}$ into $m(s)$ and then use (2.47) and (2.48) to calculate

$$\begin{aligned}\rho^{(1)}(s) &= \exp\left\{\log\lambda_0 + x^T(s)\beta - \frac{\sigma^2}{2} + \frac{\sigma^2}{2}\right\} \\ &= \lambda_0 \exp\left\{x^T(s)\beta\right\} \qquad (2.49)\end{aligned}$$

$$\rho^{(2)}(s) = (\lambda_0)^2 \exp\left\{x^T(s)\beta + x^T(s')\beta + c(s,s')\right\} \qquad (2.50)$$

$$\tilde{g}(s,s') = \exp\left\{c(s,s')\right\}. \qquad (2.51)$$

The posterior distribution for $\lambda(s)$ is arguably of more interest for us, so we do not show the posterior for $\rho^{(s)}$. Figure 2.16 shows the posterior distributions of $\rho(2)(s,s')$ and $\tilde{g}(s,s')$ for each combination of the tree points from Figure 2.15. We see that the second-order intensities are highly right-skewed and have much wider credible intervals. The PCF posterior distributions are less skewed and essentially all of the mass is above 1. Since $s_1$ and $s_2$ are the closest pair, they are the most correlated and $\tilde{g}(s_1,s_2)$ is the largest. Apparently $s_1$ and $s_3$ are far enough away that there is little correlation between the two, since $\tilde{g}(s_1,s_3)$ is very close to 1, which implies independence.

### *2.3.6 Block-level Inference*

As before, we generally are most interested in the intensities integrated over some region of interest. Figure 2.17 shows the posterior distributions for $N(A)$, $N(B)$, and $N(A)N(B)$ as was done for the NHPP model. We see that the distributions match up much better with the observed values than under the NHPP model, as shown in Figure 2.10. Under the LGCP model, the posterior distribution for $N(A)$ has a posterior mean of 21.97, which is close to the observed $N(A)$=21, with a 95% credible interval of (11, 34). For $N(B)$, we observed $N(B) = 96$ and the posterior

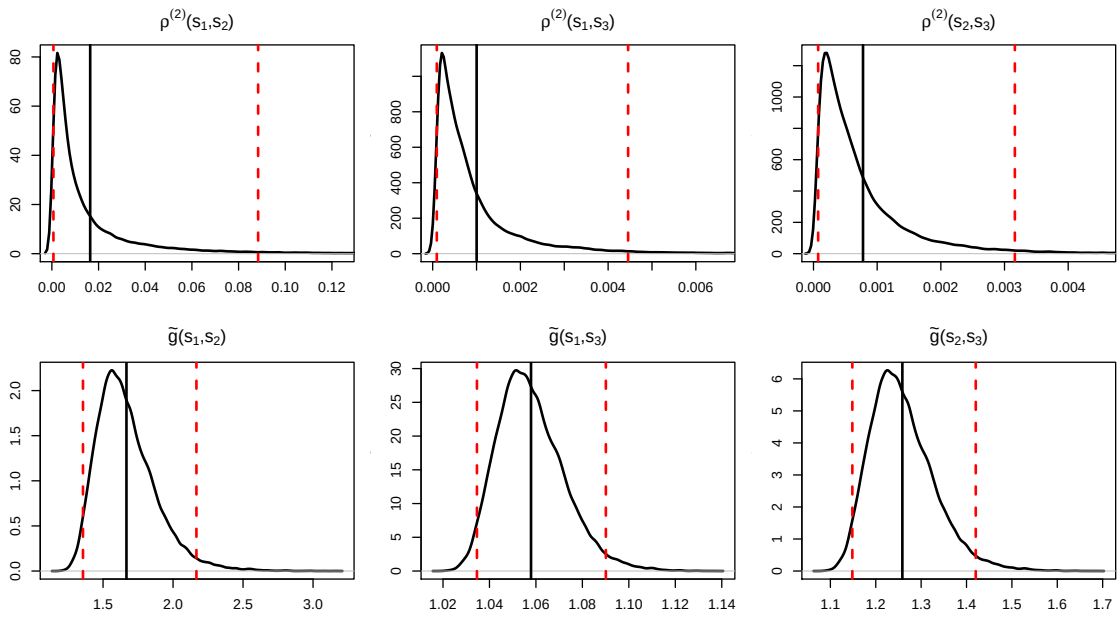



FIGURE 2.16: The posterior distributions for the second-order intensity $\rho^{(2)}(s, s')$ and the PCF $\tilde{g}(s, s')$ at each combination of the three points in Duke forest used in Figure 2.15.

mean is 90.93 with a 95% credible interval of (68, 117). For $N(A)N(B)$, we observed $N(A)N(B) = 2016$ and the posterior mean is 1996.11 with a 95% credible interval of (966, 3276). These three posterior distributions have roughly the same or smaller variability than under the NHPP model, but under the LGCP model they are also centered much closer to the actual observed values, indicating a better fit to the data.

### *2.3.7 Inhomogeneous K-function for Duke Forest Data*

The posterior distribution for the inhomogeneous $K$-function is obtained here using the same method as introduced previously for the NHPP model. We see in Figure 2.18 that the posterior mean for $K(d)$ is again very close to the theoretical value and the empirical estimate. This is again just as we expected, since the locations of points are still independent conditional on the observed intensity $\lambda(s)$, so no additional clustering or repulsion should be observed.

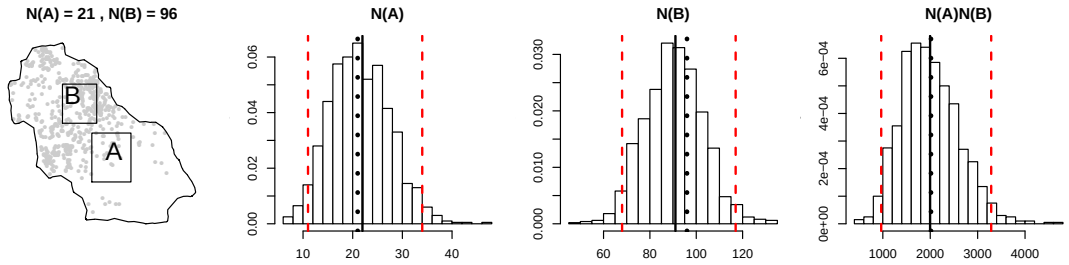



FIGURE 2.17: The posterior distributions for $N(A)$, $N(B)$, and $N(A)N(B)$ under the LGCP model for the Duke forest data. The solid lines, dashed lines, and dotted lines represent the posterior means, 95% credible intervals, and observed values, respectively.

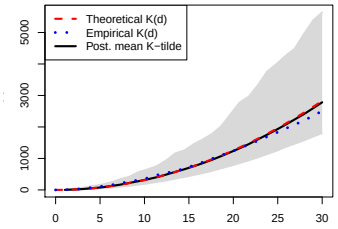



FIGURE 2.18: The posterior distribution for the inhomogeneous $K$-function under the LGCP model for the Duke forest data. The theoretical and empirical estimates, as computed in `spatstat`, are also given.

## 2.4 Summary

As has been demonstrated in this chapter, Bayesian models for spatial point patterns allow a rich array of posterior summaries. Specific applications may suggest specific points or regions to generate posterior summaries, or perhaps even suggest different summaries from those presented here. Generating predictive point patterns from the posterior samples of the intensity surface permits studying posterior distributions of counts in regions of interest, rather than just learning about the expected number of counts. The predictive point patterns also provide posterior distributions over joint distributions of counts in subregions of interest. Finally, these predictive point

patterns can be used to construct posterior distributions for more complex model characteristics, such as the $F$-, $G$-, and $K$-functions or the pair correlation function.

Berthelsen and Møller (2008) are a rare example in the literature of generating posterior predictive point patterns for posterior analysis. They generate a few of the quantities we have discussed here, yet in some cases they also chose to employ the usual nonparametric empirical estimates to these predictive point patterns rather than seeking to construct posterior distributions themselves. For example, they compared the empirical intensity and PCF estimates on their observed data to the empirical estimates applied to their generated data. With posterior distributions available, we find it more meaningful to compare the nonparametric estimate on the observed data to the *posterior distributions* of useful summaries using the predictive point patterns.

# 3

# Model Diagnostics and Model Choice

The posterior analyses of the previous chapter provide many ideas for model features which can be examined using posterior predictive point patterns. This is not the end of the analysis, however, nor should they be the first thing examined after fitting a model. As with any statistical model, the validity of the model needs to be checked before the other inferences can be used. Ideally, the model diagnostics presented here would be evaluated prior to performing all the inference ideas presented in the previous chapter, but we have reversed the order in order to build up the framework for model diagnostics. In this chapter, we discuss model validation and present a framework for cross-validation and model selection for Poisson and Cox processes.

## 3.1 Residual Diagnostics

We first present ideas for residual diagnostics for Bayesian spatial point processes. Residuals are a common model diagnostic in many statistical settings, so properly defining residuals can also be helpful as a model diagnostic here. Baddeley et al. (2005) develop many notions of residuals with more theoretical details provided by

the follow-up paper of Baddeley et al. (2008). These residuals are adapted from the innovation processes in time series settings and residuals for Poisson regression.

The first type of residual they present is a raw residual, similar to the standard residual from a regression model. It is defined on a set $B \subseteq D$ as

$$R_{\hat{\theta}}(B) \equiv N(B) - \int_B \hat{\lambda}(s; S)ds, \tag{3.1}$$

where $\hat{\lambda}(s; S) \equiv \lambda_{\hat{\theta}}(s; S)$ is the estimated Papangelou conditional intensity introduced in Chapter 1. For Poisson and Cox processes, the Papangelou conditional intensity is equal to the intensity function, so for now we can just think of it as $\hat{\lambda}(s)$. Again, Baddeley et al. (2005) employs a plug-in estimate of $\hat{\lambda}(s; S)$, where we would prefer to explore the properties of our model by building a posterior distribution of residuals.

They next present a class of scaled residuals, akin to the standardized residuals of linear regression, in which the raw residuals are scaled by a chosen function $h$. For an appropriate function $h$, the $h$-scaled residuals are defined as

$$R(B, \hat{h}, \hat{\theta}) \equiv \sum_{s_i \in S \cap B} \hat{h}(s_i, S \backslash \{s_i\}) - \int_B \hat{h}(s, S)\hat{\lambda}(s; S)ds, \tag{3.2}$$

where $\hat{h}(s, S) \equiv h_{\hat{\theta}}(s, S)$. They provide a few suggestions of meaningful functions for $h$. Setting $h(s, S) = 1/\lambda(s; S)$ defines the inverse $\lambda$ residuals, which are essentially the exponential energy mark diagnostics of Stoyan and Grabarnik (1991). The exponential energy marks, defined as $m_i = 1/\lambda(s_i, S)$, were proposed by Stoyan and Grabarnik (1991) as the first model diagnostics for point patterns. They proposed that locations $s_i$ with extreme values of $m_i$ could be seen as outliers and hence regions with many extreme values can be indicative of poor model fit.

The next residual we consider is the Pearson residual, which are scaled residuals with $h(s, S) = 1/\sqrt{\lambda(s; S)}$. These are analogues of the Pearson residuals from Poisson regression. A final residual, which we shall not consider here, is the pseudoscore

residual, which sets $h(s, S) = \frac{\partial}{\partial\theta} \log\{\lambda(s; S)\}$, where $\theta$ denotes the parameters of the intensity function $\lambda$.

In each of these cases, it is expected that the residuals should be close to 0 when the model is true. For the forms given above, the unknown $\lambda(s; S)$ must be estimated using maximum likelihood or minimum contrast techniques. These techniques, however, actually provide parameter estimates $\hat{\theta}$ which are then used in the intensity function. However, Baddeley et al. (2008) note that for an NHPP, their plug-in estimate for $\hat{\lambda}(s, S) \equiv \lambda(s; S; \hat{\theta})$ will in general be biased, causing the residuals to not have expectation 0.

The Bayesian equivalent of their residuals is to use the posterior mean of each parameter in the intensity function, or $\hat{\lambda}(s, S) \equiv \lambda(s; S; \mathbb{E}[\theta|S])$ in equations (3.1) and (3.2). From a Bayesian perspective, however, the more proper quantity to use, with (3.2) for example, is the posterior distribution of $\int_B h(s, S)\lambda(s; S)ds$ and use the posterior mean $\mathbb{E}[\int_B h(s, S)\lambda(s; S)ds \mid S]$ as a point estimate. In the terminology used in Baddeley et al. (2005), this would be an analogue to looking at the innovation measures rather than the residual measures. For example, they define the $h$-weighted innovation measure as

$$I(B, h, \lambda) \equiv \sum_{s_i \in S \cap B} h(s_i, S\backslash\{s_i\}) - \int_B h(s, S)\lambda(s; S)ds. \tag{3.3}$$

The innovations have mean 0, as can be calculated using the GNZ formula in (2.5). Baddeley et al. (2005) and Baddeley et al. (2008) provide formulas for the variance calculations of residuals and innovations.

Figure 3.1 shows the posterior distributions of the raw, inverse $\lambda$, and Pearson innovations on $D$, $A$, and $B$ under the NHPP model from the previous chapter, using the subregions $A$ and $B$ as introduced in Figure 2.10. We see that while the coverage is good for $D$, the innovations for $B$ are slightly off and the innovations for $A$ seem

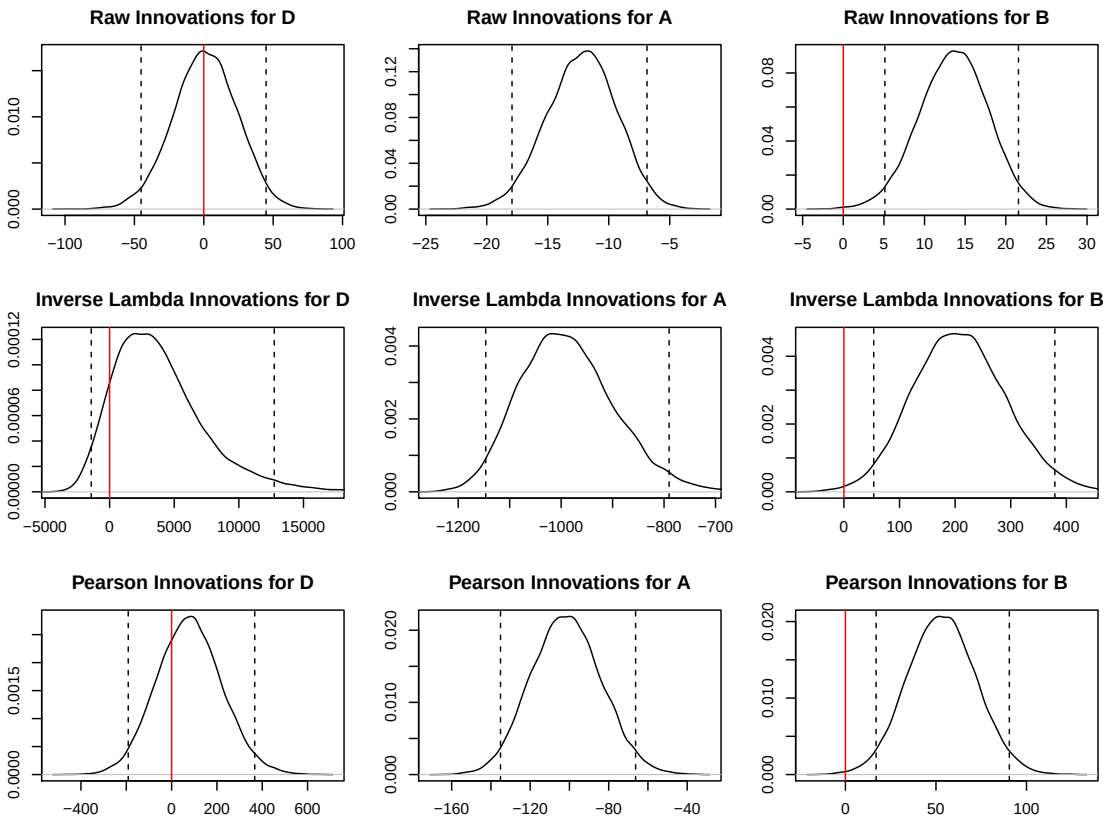



FIGURE 3.1: The raw, inverse $\lambda$, and Pearson residuals for $D$, $A$, and $B$ under the NHPP model for the Duke forest data, with regions $A$ and $B$ as shown in Figures 2.10 and 2.17. The dashed lines indicate the 95% credible intervals, with 0 marked by a solid line.

to indicate a severe lack of fit. Is the lack of fit over $A$ and $B$ enough to invalidate the NHPP model entirely? The intensity plot in Figure 2.6b seemed reasonable, but it's possible that the posterior distribution of the intensity is badly biased in some regions due to the parametric form and lacks enough posterior uncertainty to cover the truth.

Figure 3.2 shows the same panel of innovation plots for $D$, $A$, and $B$ under the LGCP model for the Duke forest data. The coverage of the residual distributions is much better here, with each of the innovation distributions containing 0. It appears

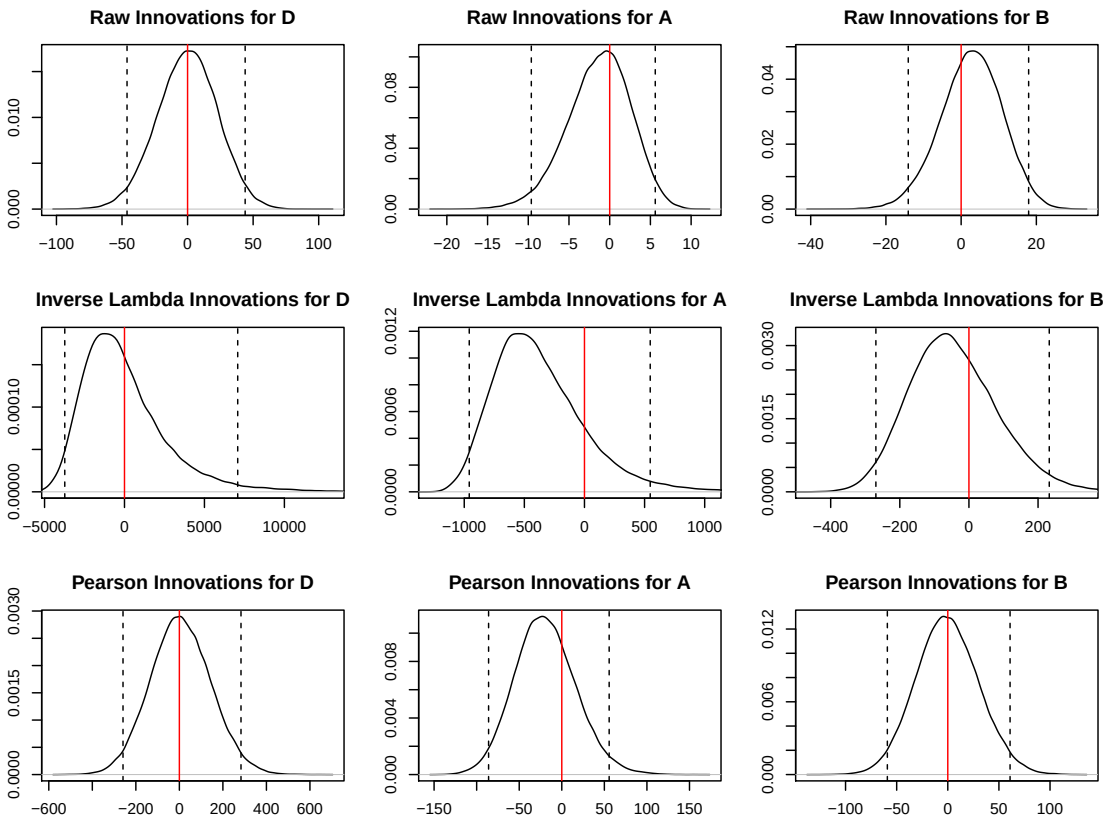



FIGURE 3.2: The raw, inverse $\lambda$, and Pearson residuals for $D$, $A$, and $B$ under the LGCP model for the Duke forest data, with regions $A$ and $B$ as shown in Figures 2.10 and 2.17. The dashed lines indicate the 95% credible intervals, with 0 marked by a solid line.

the extra flexibility provided by the LGCP was very beneficial in adapting to the local variation in the intensity function.

This brings up the issue of whether these innovation distributions should be expected to contain 0, or at least provide the nominal level of coverage. As noted earlier, the innovations have mean 0 with a variance that can be calculated. For the raw innovations at least, the answer is clearly no. The raw innovations compare an observed count with the posterior distribution for the *expectation* of that count. Though we hope the raw innovations are close to their expectation, which is 0, the credible intervals provide coverage for the expected counts rather than the counts

themselves. In other words, the raw innovations compare the distribution of $[y - \mu_y|S]$, when it seems more natural to compare the distribution of $[y - y_{\text{pred}}|S]$. In a linear regression setting, confidence intervals for $y_i$ using $x_i^T\hat{\beta}$ should have mean 0, yet it is the prediction intervals that allow us to investigate whether the observed coverage is on par with the nominal coverage level.

We therefore recommend the use of what we will term *predictive residuals*:

$$R_{\text{pred}}(B) = N(B) - N_{\text{pred}}(B), \tag{3.4}$$

where the draws $N^{(l)}(B)$ from the posterior predictive point patterns will be used as the posterior distribution of $N(B)$. The predictive residuals should similarly be centered around zero for an adequate model. Further, when looking at many subregions $B_k$, we can expect to see the nominal level of coverage if the model is adequate. Now we can assess whether the predictive residuals for the NHPP are significantly poor as to decide that the NHPP model does not fit the data.

### *3.1.1 Monte Carlo Residual Test*

The residual and innovation diagnostics given so far have required the specification of a set of windows over which to evaluate the residuals. For a more formal testing procedure, we start with the idea suggested in section 11.1 of Baddeley et al. (2005) to analyze the residuals over disjoint partitions $B_k$ of the domain, similar to quadrat counting. With an irregular domain $D$, however, dividing the domain into disjoint subregions of similar size can be time-consuming. Rather, we prefer to draw random subregions uniformly over $D$ and then evaluate the residuals or innovations in each subregion. There is no reason to require the $B_k$ be disjoint, and allowing them to overlap allows us to draw as many $B_k$ as we like. For consistency, it seems that each $B_k$ should have equal area. Denote the area of each $B_k$ by $q|D|$ where $q$ will represent the size of each $B_k$ relative to $D$. After partitioning the domain $D$ into subregions,

Table 3.1: Coverage of the various innovations and residuals in the Monte Carlo test targeting a 90% coverage rate.

| Model | Raw Inn. | Inverse $\lambda$ Inn. | Pearson Inn. | Predictive Res. |
|---|---|---|---|---|
| NHPP | 0.17 | 0.14 | 0.16 | 0.69 |
| LGCP | 0.84 | 0.81 | 0.84 | 1.00 |

we can evaluate the innovation or residual measures on each of the $B_k$ subregions and evaluate the observed coverage.

Though we take the shape of each $B_k$ to be a square, there may be some reason to choose the shape more carefully. We observed that the use of the square sometimes limited the placement of the $B_k$ when $q$ was large, due to the irregular region. For large $q$, the $B_k$ do not fit close to any of the edges of $D$, and hence the sampling of these boxes is very rare near the boundary of $D$. Work by Sherman and Carlstein (1994), Lahiri (1999), and Lahiri (2003) suggests that there may be good reason to let the shape of $B_k$ mimic the shape of $D$. Even though the full results developed in these papers do not seem to apply here, using the same shape as $D$ would seem to allow the $B_k$ to be placed closer to the boundary of $D$.

Table 3.1 shows the empirical coverage of the raw, inverse $\lambda$, and Pearson innovations and the predictive residuals for the Duke forest data. We used $K = 200$ squares of size $0.05 \times |D|$ and calculated 90% credible intervals for the raw, inverse $\lambda$ and Pearson innovations and 90% prediction intervals for the predictive residuals.

Integrating over the domain, we see that the NHPP model has poor coverage, especially under the Baddeley residuals. For the predictive residuals, which are the only residuals should achieve the nominal 90% rate, the NHPP still underperforms slightly. Recalling the posterior mean intensity for the NHPP model from Figure 2.6b, we saw that the intensity was perhaps overly smooth, but seemed reasonable. The predictive residuals similarly convey the conclusion that the NHPP model is

only slightly ill-fitting, whereas the other residuals seem to conclude that the NHPP model is severely misspecified.

The LGCP model appears to have achieved fairly decent coverage, even for the Baddeley residuals. In fact, the predictive residual coverage appears to be overly optimistic, indicating either very large variability in the posterior or overfitting to the observed data. Comparing the LGCP residuals in Figure 3.2 to the NHPP residuals in Figure 3.1, the spread of the residual distributions seem fairly comparable, with the LGCP residuals being of similar or slightly larger spread. This suggests that the LGCP does not have significantly larger uncertainty, enough to make up the difference in the performance of the residuals, but rather that the LGCP exhibits less bias. It remains then that the LGCP appears to be a more adequate model for the data, though it may also be more susceptible to overfitting the data.

With the overlapping $B_k$, it can be hard to identify specific regions where the model fits poorly, unless the results in each $B_k$ are plotted sequentially. One alternative is to use disjoint $B_k$, as is demonstrated in Illian et al. (2009), but we prefer the sharper resolution provided by the smoothed residual plots in Baddeley et al. (2005). They define the smoothed residual field $r(u)$ at location $u \in D$ as

$$\begin{aligned} r(u) &= e(u) \int_D k(u-v) dR(v, \hat{h}, \hat{\theta}) \\ &= e(u) \Bigg[ \sum_{s_i \in S} k(u-s_i) \hat{h}(s_i, S \backslash \{s_i\}) - \int_D k(u-v) \hat{h}(v, S) \hat{\lambda}(v, S) dv \Bigg], \end{aligned} \tag{3.5}$$

where $k(h)$ is a probability density on $\mathbb{R}^2$ used as a smoothing kernel and $e(u)^{-1} \equiv \int_D k(u-v)dv$ is an edge correction. An equivalent definition using innovations is also given. The smoothed residual field puts positive atoms at each $s_i \in S$ and a negative value elsewhere and then uses the kernel smoother to provide a smooth field. Briefly, positive values in the smoothed raw residual field indicate locations where

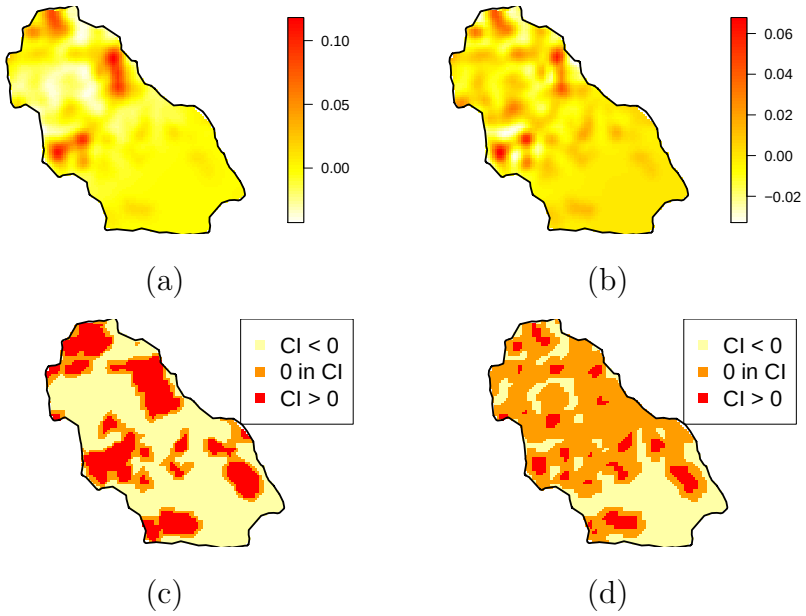



FIGURE 3.3: Posterior mean of the smoothed raw innovation fields for the (a) NHPP and (b) LGCP models and posterior coverage plots for the smoothed raw innovation fields for the (c) NHPP and (d) LGCP models. The coverage plots describe whether a pointwise credible interval (CI) contains 0 or whether the interval is completely above or below 0.

the empirical intensity was higher than our model's fitted intensity, while negative values indicate areas where the model's intensity was higher.

With a posterior distribution over $\lambda(s)$, we can calculate a posterior distribution over the smoothed raw innovation fields for the NHPP and LGCP models. Then a pointwise posterior mean for the smoothed raw innovation field can be calculated, as shown in the top row of Figure 3.3. The smoothed raw residual plot looks similar to the posterior mean for the smoothed raw innovation field, yet the innovations provide a sense of uncertainty. A bivariate Gaussian kernel was used with a bandwidth chosen using cross-validation. We see that the smoothed innovation field for the NHPP model has more extreme negative and positive values than the LGCP model. The NHPP intensity was too low in areas where a lot of data was observed (the high positive values in the smoothed residual field). We also note that in the upper

left part of the domain, the NHPP intensity was also too high in the areas of low intensity, resulting in negative values in the smoothed residual field. The LGCP residual field is generally much closer to 0 and has lower, yet still large and positive, values where the NHPP field was very high.

We propose a companion plot to the smoothed residual/innovation field plots above which describes the coverage of the smoothed residual/innovation field. The bottom row of Figure 3.3 describe which locations have a pointwise credible interval over the smoothed raw innovation surface that contains 0. For the NHPP model, about 60% of the locations in $D$ have a raw innovation posterior 95% credible which is contained below 0 and 25% of the locations have a credible interval that is contained above 0. This leaves only about 15% of the domain being covered by the residuals, which is about what we found in Table 3.1. From the figure, we also note that the only places where the smoothed innovation credible intervals contain 0 are the areas where the intensity function transitions from a high peak to a lower level. Thus, the smoothed innovation surface can be described to just be passing through the region around 0 on its way to a large positive or negative value.

In contrast, the LGCP model tends to have large regions where the smoothed innovation surface stays closer to 0. For the LGCP model, 33% of the domain have an raw innovation credible interval below 0, 58% had an interval containing zero, and about 9% had an interval contained above 0. We saw a higher percent of coverage in Table 3.1 for the LGCP model, but the difference can most likely be attributed to the difference between the pointwise comparisons done here and the blockwise comparisons used previously. We again note that these credible intervals for the smoothed innovation field are not necessarily expected to provide the nominal level of coverage for these smoothed innovations, so the poor coverage is not necessarily disconcerting, though it does identify areas where each model may not fit well.

We propose the Monte Carlo test using predictive residuals as a general method for testing overall model fit. If the observed coverage is on par with (or greater than) the nominal level, then the model and the attached uncertainty in model parameters appear to adequately fit the point pattern. Later, we will discuss model comparison for choosing among adequate models. The smoothed residual or innovation fields and their corresponding coverage plots (proposed above) can be used to discover areas where model fit is lacking, yet they do not give a good sense of overall model fit. Baddeley et al. (2005) also suggest added variable and lurking variable plots for identifying specific ways in which the model can be improved, such as adding a spatial trend, etc.

## 3.2 Cross-validation for Point Patterns

Thus far we have looked at the posterior distributions of residuals, innovations, and other summaries and then compared them to the data we used to get our posterior. Investigating residuals on data used to fit the model is a common concern with posterior predictive checks (Gelman et al., 1996; Gelman and Shalizi, 2013). These checks will highlight features of our model which do not fit the data well, but they will not be able to expose overfitting, which is important when considering model comparison or prediction of future data (e.g., for space-time point processes).

Cross-validation is a very useful tool in model assessment and model comparison which can provide model assessment without encouraging overfitting. There is limited discussion of cross-validation methods for point processes, however. As noted previously, Diggle and Marron (1988) adapted leave-one-out cross-validation from Bowman (1984) for bandwidth-selection for kernel smoothing in intensity estimation. Arguing that the choice of the kernel is less important than the choice of the bandwidth, both papers focus on choosing an optimal bandwidth using cross-validation.

This approach makes sense for kernel intensity estimates, which are simple and can be quickly computed.

For a more model-based approach, especially a Bayesian model requiring MCMC, the extra computational burden required by leave-one-out cross-validation makes this approach impractical for comparing models. One could then turn to using training and test data, where the training data is used to fit the model and the test model is used to critique the model's performance. The question arises of how to choose the training data. Is it proper to simply remove 10% of the data? We propose that the $p$-thinning approach of Illian et al. (2008) can be applied to create proper training and test data for a coherent analysis.

Letting $p$ denote the retention probability, $p$-thinning proceeds by independently deleting each point $s_i \in S$ with probability $1 - p$. This thinning is applied point-by-point with the decision to remove or keep each point being independent of other points. This produces a training test point pattern $S^{\text{train}}$ and a test point pattern $S^{\text{test}}$, where $S^{\text{train}}$ and $S^{\text{test}}$ contain roughly $p \times 100\%$ and $(1-p) \times 100\%$ of the data in $S$, respectively. This independent, stochastic thinning ensures coherence between the model we fit to $S^{\text{train}}$ and using that model to explain $S^{\text{test}}$. Conveniently, the training and test datasets are independent conditional on $\lambda(s)$.

We can now define model comparison and diagnostic methods using $S^{\text{train}}$ to fit our model and then applying our residual analysis or any model-comparison metrics to $S^{\text{test}}$. We must first discuss the bias we have introduced into the model, however. The main issue with removing $(1 - p) \times 100\%$ of the data to be used for cross-validation is that the number of points observed in $D$ is itself a parameter in the model, so $S^{\text{train}}$ actually has intensity $p\lambda(s)$, while the full point pattern $S$ has intensity $\lambda(s)$. In the leave-one-out approach to bandwidth selection describe above, we can roughly think of $p$ being equal to $(n-1)/n$, though leave-one-out drops each point deterministically. In any case, the bias for leave-one-out cross-validation is

fairly minimal and will converge in probability to the truth. Here, however, the bias will not decrease with a larger sample size and will need to be accounted for.

To be a little more explicit, let $\lambda^{\text{train}}(s) = p\lambda(s)$ be the intensity estimated under the model using $S^{\text{train}}$. To use our fitted model for cross-validation purposes, one need only convert the posterior draws of $\lambda^{\text{train}}(s)$ to predictive draws of $\lambda^{\text{test}}(s)$ using

$$\lambda^{\text{test}}(s) = \left(\frac{1-p}{p}\right)\lambda^{\text{train}}(s). \tag{3.6}$$

Residuals, predictions, posterior summaries, etc., can now all be made on $S^{\text{test}}$ using $\lambda^{\text{test}}(s)$ in the methods described previously.

We noted previously that the predictive residuals for the LGCP model had 100% coverage. To investigate how this might change when doing cross-validation, we applied $p$-thinning to the dataset to create training and test datasets for $p = 0.5$, 0.8. We then performed the Monte Carlo residual test with the various innovations and residuals used previously, performing the test on both the training and test data for each level of $p$.

Table 3.2 shows the coverage of the various residual metrics for different thinning levels, specifically for $p = 0.5$ and 0.8 and with $q = 0.05$ still. We see that the LGCP model provides uniformly better coverage than the NHPP model. The coverage for the hold-out data is generally lower than the training data, with the exception of the predictive residuals for the NHPP model in the $p = 0.8$ case. In that one case, the test data actually lined up better with the model than did the training data. The NHPP model coverage for the predictive residuals is closer to the nominal level of 90% than its coverage according to the three innovation measures, yet still far enough to be of concern. The LGCP predictive residuals still provided coverage that was better than nominal for the training data, but the coverage was very close to the nominal 90% for the test data. For the LGCP model with $p = 0.8$, we also see

Table 3.2: Coverage of the 90% credible intervals for the innovations and residuals in the Monte Carlo test for thinning levels $p = 0.5, 0.8$ and $q = 0.05$. The coverage on the training dataset is given before the forward slash and the coverage on the test dataset is given after the forward slash.

| Model | $p$ | Raw Inn. | Inverse $\lambda$ Inn. | Pearson Inn. | Predictive Res. |
|---|---|---|---|---|---|
| NHPP | 0.5 | 0.29 / 0.18 | 0.24 / 0.13 | 0.28 / 0.15 | 0.84 / 0.76 |
| LGCP | 0.5 | 0.76 / 0.44 | 0.69 / 0.44 | 0.72 / 0.41 | 0.98 / 0.90 |
| NHPP | 0.8 | 0.14 / 0.09 | 0.18 / 0.08 | 0.17 / 0.07 | 0.74 / 0.82 |
| LGCP | 0.8 | 0.76 / 0.29 | 0.78 / 0.27 | 0.77 / 0.29 | 0.99 / 0.88 |

that the coverage on the test data is much lower for the raw, inverse $\lambda$, and Pearson residuals, possibly due to the smaller sample size. There were 267 observations in the training data and 263 observations in the test data for $p = 0.8$ and 434 observations in the training data and 96 observations in the test data when $p = 0.8$.

Figure 3.4 shows the training data, test data, and posterior mean intensities for $\lambda^{\text{train}}(s)$ for the training subset of the Duke forest data, thinned using $p = 0.5$. Comparing these with Figures 2.6b and 2.13b, the intensity estimates look very similar, though of course the intensity estimates here should be about one-half the values of those previously given since $\lambda^{\text{train}}(s) = p\lambda(s) = 0.5\lambda(s)$. Also, since $p = 0.5$ implies $\lambda^{\text{train}}(s) = \lambda^{\text{test}}(s)$, the intensity estimates are valid for both the training and test data.

## 3.3 Model Selection for Point Patterns

One major challenge in point process methodology is the lack of useful model selection tools, especially for complex Bayesian models. The typical analysis will uses ad hoc tests designed to test the homogeneous and independence assumptions of CSR, but having decided which assumption to relax there is no clear procedure for comparing models. In some cases, there may be a natural process corresponding to the intensity function which can guide the choice of model, but in the absence of

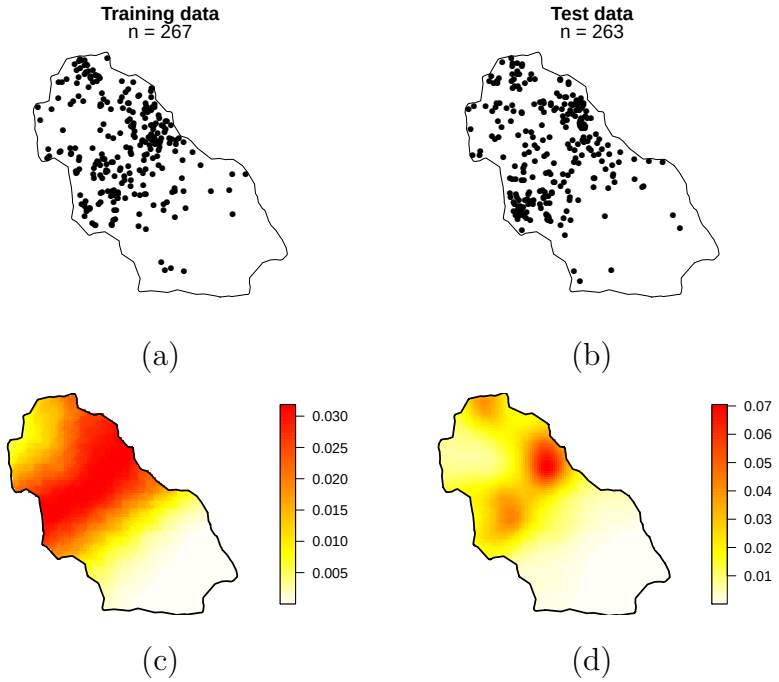



FIGURE 3.4: The (a) training and (b) test data for the $p = 0.5$ cross-validation data, along with the posterior means for $\lambda^{\text{train}}(s)$ under the (c) NHPP model and (d) LGCP model. Since $p = 0.5$, $\lambda^{\text{train}}(s) = \lambda^{\text{test}}(s)$.

a natural correspondence between the model and the underlying process, it seems natural to have a metric for comparing competing forms of models. Lack of model fit using the methods described previously in this chapter is one way to eliminate models, but this model choice metric will help when choosing among models which appear to fit well.

For frequentist models, the AIC or BIC can be computed, but these fail for more complex point processes where the likelihood is intractable. The first discussions of Bayesian model selection for point processes appear in Akman and Raftery (1986) and Raftery and Akman (1986), who discuss computing Bayes factors for NHPPs and change-point Poisson processes, respectively. Guttorp and Thorarinsdottir (2012) perform model choice via a reversible jump algorithm that allows movement between

two nested models. They can then use the work of Akman and Raftery (1986) to compute a Bayes factor.

To introduce a Bayesian model selection procedure which is more easily generalizable, we suggest using scoring rules with our posterior predictive point patterns. Scoring rules are a useful metric for comparing predictive distributions. For a given score function, the score obtained by the model describes the closeness of observed values and their predictive distributions under the model. Proper scoring rules are those where the score is maximized when the predictive distribution of interest matches the underlying distribution exactly. Strictly proper scoring functions are those for which the score is uniquely maximized when the predictive and underlying distributions match, providing an incentive to be both accurate and honest. Such scoring rules exist for discrete and continuous data, as discussed in, e.g., Gneiting and Raftery (2007).

Following the notation of Gneiting and Raftery (2007), competing models are compared using their average score

$$\mathcal{S}_n = \frac{1}{n}\sum_{i=1}^{n}\mathcal{S}(P_i, X_i), \tag{3.7}$$

where $\mathcal{S}(P_i, X_i)$ denotes the score of the predictions $P_i$ for some observable $X_i$ of interest. $\mathcal{S}(\cdot,\cdot)$ can take on many forms, though the ranked probability score (RPS) seems most fitting for this type of data. The RPS is an extension of the Brier score (BS), which was introduced by Brier (1950). For an observed outcome of interest $X_i$ taking on discrete values in $\{\omega_1, \ldots, \omega_R\}$, the Brier score uses $\mathcal{S}(P_i, x_i) = \sum_{j=1}^{R}(f_{ij} - o_{ij})^2$ where $f_{ij}$ gives the model's predicted probability $Pr[X_i = \omega_j]$ and $o_{ij} \equiv \mathbf{1}(x_i = \omega_j)$.

The ranked probability score (Epstein, 1969) builds on the Brier score by addressing ordered categorical data, penalizing poor predictions more heavily if pre-

dicted values are further away from the true value. The RPS can be written as $\mathcal{S}(P_i, x_i) = \sum_{j=1}^{R}(F_{ij} - \mathbf{1}(x_{ij} \geq \omega_j))^2$, where $F_{ij} = \sum_{l=1}^{j} f_{il}$ is the predictive distribution function for $X_i$ and the $\omega_j$ have a natural ordering. Without loss of generality, we assume here that $\omega_1 \leq \omega_2 \leq \cdots \leq \omega_R$. The RPS intuitively compares the predictive distribution function to the empirical distribution function and prefers models which provide predictions that are concentrated around the observed value $x_i$.

With the goal of assessing model fit and choosing between models, we can employ scoring rules on predictive distributions of features of our hold-out data. We could start by looking at any of the quantities of interest we have previously computed. Since it is hard to compare fitted intensities to held-out events, we feel that it makes most sense to compare observed counts to predicted counts in subregions of $D$, as is done with the raw residuals in section 3.1. Specifically, we propose choosing subregions $B_k$ uniformly over $D$, with each $B_k$ having the same size and potentially overlapping other $B_{k'}$. In fact, we use the same $B_k$ as used in the Monte Carlo residual test above. For each $B_k$, we can calculate $N_{\text{test}}(B_k)$ from the held-out data $S^{\text{test}}$ and compare it the predictive draws of $N_{\text{test}}^{(l)}(B_k \,|\, S^{\text{train}})$ using posterior predictive point patterns $S_{\text{test},l}^{*}$, which were generated using posterior draws of $\lambda^{\text{test}}(s)$. Since $\lambda^{\text{test}}(s)$ is the predictive intensity for $S^{\text{test}}$, the posterior predictive distribution for $N_{\text{test}}(B_k)$ should be close to the observed $N_{\text{test}}(B_k)$ in the held-out data.

In terms of the scoring rules proposed above, we define $X_l$ as the observed $N_{\text{test}}(B_k)$ and $P_k$ is the posterior predictive distribution for $N_{\text{test}}(B_k|S^{\text{train}})$ with distribution function $F_{N_{\text{test}}(B_k|S^{\text{train}})}$. We can write the RPS as

$$\text{RPS}(P_k, N_{\text{test}}(B_k)) = \sum_{n=0}^{\infty} \left[F_{N_{\text{test}}(B_k|S^{\text{train}})}(n) - \mathbf{1}[n \geq N_{\text{test}}(B_k)]\right]^2. \tag{3.8}$$

For any given model, we can compare the average $\text{RPS}_K = \frac{1}{K}\sum_{k=1}^{K} \text{RPS}(P_k, N_{\text{test}}(B_k))$ with that of other models. The scoring rules will provide us with a picture of how

close our predicted counts in the subregions are to the observed counts under different models.

We have discussed calculating the RPS on the held-out data $S^{\text{test}}$, but one could also calculate it for the training data $S^{\text{train}}$ also. We can denote these two different scores by $\text{RPS}_K^{\text{train}}$ and $\text{RPS}_K^{\text{test}}$. The notation $\text{RPS}_K^{\text{train}}$ and $\text{RPS}_K^{\text{test}}$ also suggests that we need not use the same number $K$ of subregions nor need they be the same subregions. However, for simplicity we use the same set of subregions when calculating both $\text{RPS}_K^{\text{train}}$ and $\text{RPS}_K^{\text{test}}$.

For the Duke forest data, we compute $\text{RPS}_K^{\text{train}}$ and $\text{RPS}_K^{\text{test}}$ when holding out roughly 50% of the data by using independent $p$-thinning with $p = 0.5$. We set $K = 200$ and sample the $B_k$ locations uniformly over $D$, with each $B_k$ being a square of size $q|D|$ with $q \in (0, 0.1]$. It is convenient to choose the shape of the $B_k$ to be a box or circle, but any shape (or a variety of shapes) is allowed. We only require that the size of each $B_k$ is constant. It is important to note that for $q > 0.1$, the size of $B_k$ became prohibitively large such that the $B_k$ were only allowed to fit certain places over the domain. With a more regular domain $D$, such as a rectangular $D$, larger $B_k$ could be explored. The average RPS and predictive residual coverage were calculated for both the training and test data. We replicated this analysis three times, meaning that we applied $p$-thinning to the dataset three separate times and performed this analysis for each set of training and test data.

We used the NHPP and LGCP models as previously described. We also used two other LGCP specifications using different covariance functions. The LGCP covariance function used thus far has been a Matérn covariance function with $\nu = 3/2$. We now also consider LGCPs with exponential and Gaussian covariance functions. The exponential covariance function, equivalent to a Matérn covariance function with $\nu = 1/2$, takes the form $c(s, s') = \sigma^2 \exp\{-\phi||s - s'||\}$. The Gaussian covariance function, also called the squared exponential function and equivalent to a Matérn

covariance function as $\nu \to \infty$, takes the form $c(s, s') = \sigma^2 \exp\{-\phi^2 ||s - s'||^2\}$. The parameter $\nu$ describes the smoothness prescribed by the Matérn covariance function, with larger $\nu$ implying greater smoothness in the Gaussian process and therefore in the LGCP intensity function itself. Therefore, the Gaussian covariance function will result in the smoothest intensity surfaces and the exponential covariance function will result in the least smooth surfaces and will be the most susceptible to overfitting. One goal of this analysis is to see if the data give preference to the fit of a certain covariance function in the LGCP model.

Figure 3.5 compares the RPS for each set of training and test datasets for the four models under consideration and for different values of $q$, which determines the size of the $B_k$. The same set of $B_k$ were used for each set of training and test data. We see that the three LGCP models performed almost identically, as evidenced by the three dashed lines matching up almost perfectly in each plot. The LGCP models clearly outperformed the NHPP model (solid line) on the training datasets, yet they didn't always show an advantage on the test data. For the first cross-validation set, the LGCP models seemed to clearly outperform the NHPP model on both the training and test data. For the second replication, the NHPP had scores that were about four times larger than the LGCP models on the training data, but only slightly larger on the test data. The last replication shows a clear advantage for the LGCP models on the training data and a smaller yet constant advantage on the test data.

These plots also provide some other interesting insights. For the second cross-validation replicates, the RPS for the training data under the NHPP model were larger than for the test data. It appears that the NHPP model was more robust to overfitting the training data due to its smooth, parametric form. On the other hand, the LGCP models generally exhibited slightly worse performance on the test data than on the training data, as expected, since the flexibility of the LGCP model allows more adaptation to local structure observed in the training data which may or

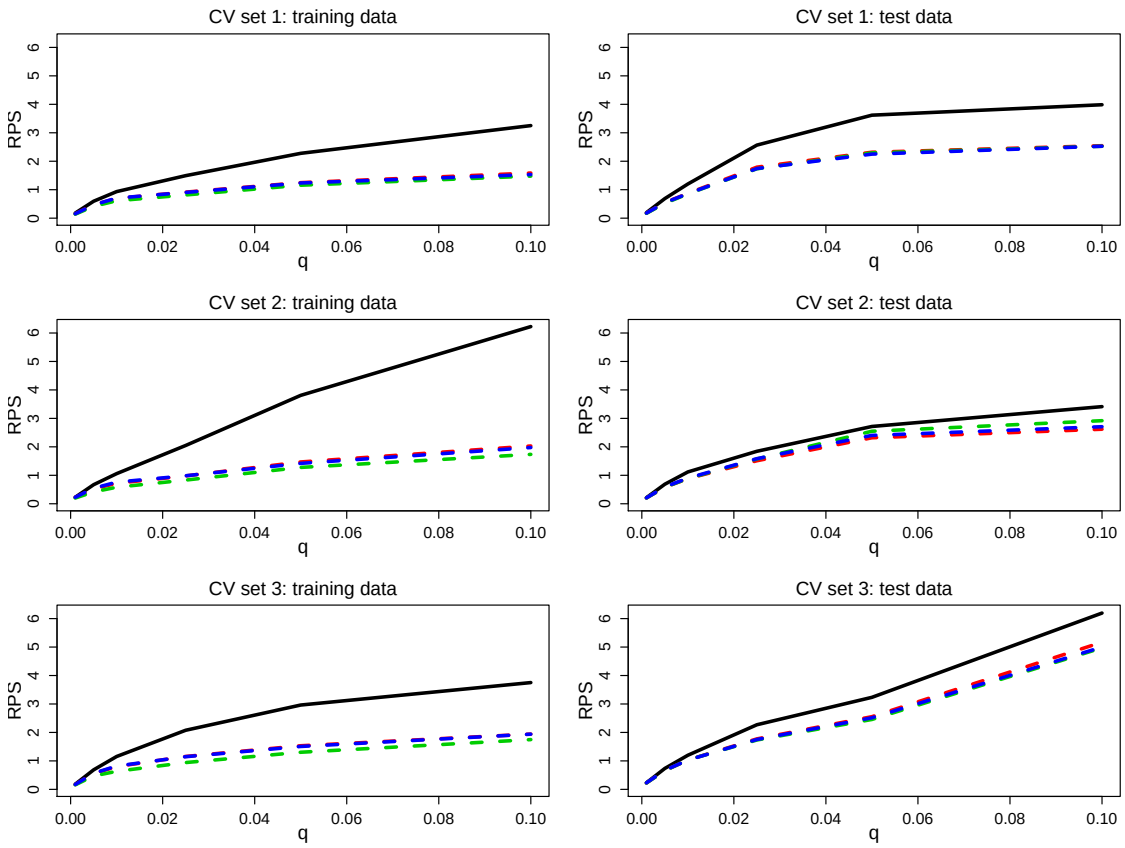



FIGURE 3.5: Ranked probability scores for the NHPP model (solid black line) and the three LGCP models (dashed lines) fitted to the Duke forest test data for three cross-validation sets with $p = 0.5$.

may not be replicated in the test data. Finally, with the three sets of cross-validation data, there is a fair amount of variation in the results, so using a few different cross-validation sets as was demonstrated here may be necessary to provide an accurate representation of model performance.

Figure 3.6 presents the coverage of the predictive counts for the same set of $K$ boxes. For each box, we calculate a 90% predictive credible interval for the number of counts in that box. The proportion of credible intervals which contain the observed count is the coverage. A good model will provide coverage that is close to 90% on the test data. We see that the LGCP models are again almost indistinguishable

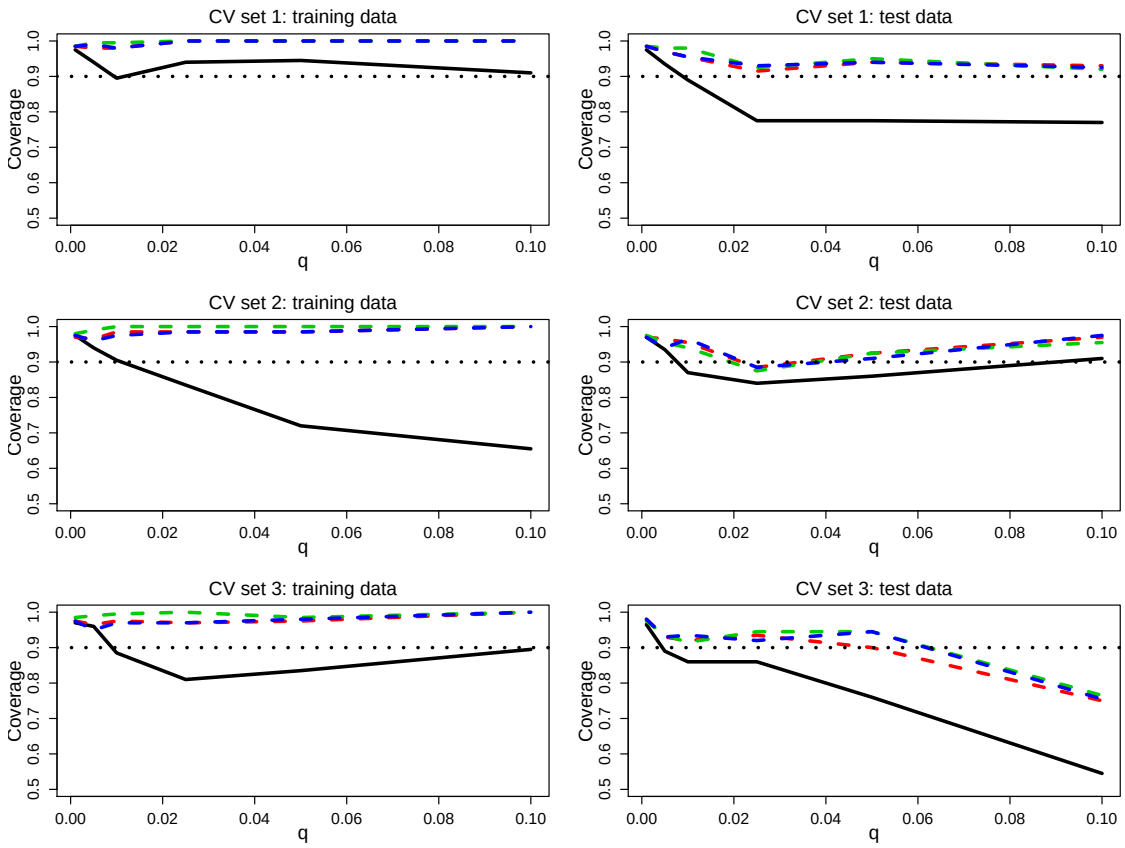



FIGURE 3.6: 90% predictive interval coverage for the NHPP model (solid black line) and the three LGCP models (dashed lines) fitted to the Duke forest for three cross-validation sets with $p = 0.5$. The black dotted line indicates the 90% nominal level.

and provide the nominal level (or better) of coverage for both the training and test datasets. The LGCP models also provide uniformly better coverage than the NHPP model for the training and test data. The NHPP model exhibits slightly lower coverage than the nominal level most of the time, though the coverage is still generally better than 75%. The NHPP model exhibited poor RPS for the second set of training data, which is mirrored here by poor coverage. For the third set of test data, all of the models had RPS trends that increased constantly with $q$. The coverage plots here similarly show coverage levels that drop with $q$, though the NHPP model exhibits much worse coverage than the LGCP models.

Using both the RPS and coverage results, it seems like the LGCP models are preferable to the NHPP model, though virtually indistinguishable from each other. The ranked probability scores indicate that the extra flexibility gives the LGCP models an advantage. Even when the RPS results were similar, the coverage for the LGCP models was higher, possibly indicating that the posterior distributions were centered closer to the right location.

## 3.4 Simulation Study

The results of the previous section raise some questions regarding cross-validation, ranked probability scores, and our suggested model selection methods for point patterns. It is of interest to understand the limits of the learning available for point patterns. In other words, how certain can we be in our decisions about which model is preferred? For example, if data truly arose from an NHPP, might we still tend to prefer the more flexible LGCP? Or if the point pattern truly arose from a LGCP with a Matérn covariance function with $\nu = 3/2$, would our methods choose the Matérn ($\nu = 3/2$) LGCP model over the LGCP models with exponential or Gaussian covariance functions? We expect that there will be fairly limited learning available, especially without a large number of observations. Our results thus far have certainly failed to show much difference in performance between LGCPs with different covariance functions.

We now present results from a simulation study designed to provide some insight into these questions concerning model choice. For each of the four models used previously as well as an HPP, we generate data from each of the models in turn and fit all of the models to the generated point pattern. We then employ the RPS on held-out data to provide a preferred model in each scenario, though we will also look at the coverage rates. After many simulations for each scenario, we obtain an estimate for the proportion of time that the correct model was chosen in each

scenario. Finally, we perform this for a setting where the expected number of events is small ($\mathbb{E}[n] \approx 100$) and a setting where the expected number of events is large ($\mathbb{E}[n] \approx 1000$). It is hoped that the larger datasets will provide more information and better separation among the performance of the LGCP models. Each time a simulated dataset is generated, using a specific model and intensity surface, we will generate a training dataset and a separate test dataset, giving a valid set of hold-out data. Each dataset will be generated to target the same desired $\mathbb{E}[n] \approx 100$ or 1000, so this will be equivalent to generating one dataset with $\mathbb{E}[n] \approx 200$ or 2000 and applying $p$-thinning with $p = 0.5$. We will evaluate the results on the test data for several values of $q$ so we can study these results over varying sizes of $B_k$.

The domain $D$ will be the unit square $[0, 1] \times [0, 1]$ so that larger boxes $B_k$ can be easily placed anywhere in the region, as opposed to the domain in the Duke forest example which limited the placement of the $B_k$. The HPP scenario will use $\lambda_0 = 100$ for the small data scenario and $\lambda_0 = 1000$ for the larger scenario. For the NHPP model, the intensity will be of the form $\lambda(s) = \lambda_0 \exp\{x_1(s)\beta_1 + x_2(s)\beta_2\}$ with $\beta_1 = 2$, $\beta_2 = 4$, and $\lambda_0$ is chosen to solve $\mathbb{E}[n] \approx \int_D \lambda(s)$ for each setting of $\mathbb{E}[n]$. We will construct two covariates $x_1(s) = x(s) * y(s)$ and $x_2(s) = \cos(\pi\, x(s)) * \sin(\pi\, y(s))$, where $x(s)$ and $y(s)$ are the $x$- and $y$-coordinates of $s$ in the unit square. For the three LGCP models we have been using, we set $\sigma^2 = 1/2$ and $\phi = 5$. Each set of simulated data will use a random draw from the LGCP prior and use that realization of the intensity function to create the training and test data. The hyperparameters of the LGCP have been chosen so as to not drown out the covariate information, such that the NHPP model will still capture some of the varying trend in the intensity.

Appendix B contains the full set of output from the simulation study, but we will highlight a few of the figures here and discuss the general results. For each scenario, we calculate the relative RPS using 200 random boxes on the test dataset, where the relative RPS is calculated as $\text{RPS}_K^{\text{test}}$ of the particular model divided by $\text{RPS}_K^{\text{test}}$ for

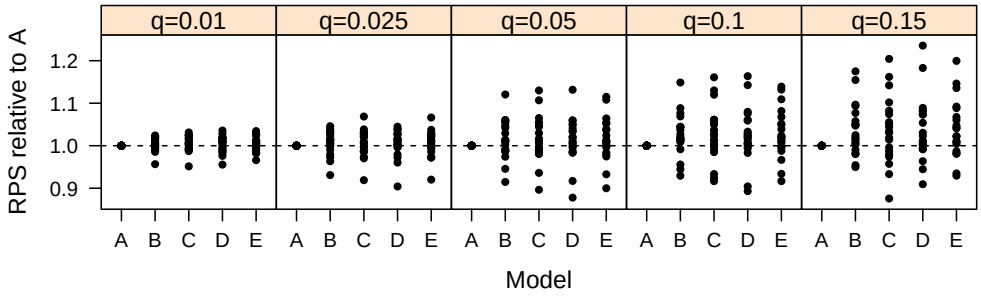



FIGURE 3.7: The relative RPS for the simulated HPP data with $\mathbb{E}[n] = 100$. The models are labeled as (A) HPP, (B) NHPP, (C) LGCP with exponential covariance, (D) LGCP with Matérn ($\nu = 3/2$) covariance, and (E) LGCP with Gaussian covariance.

the true underlying model. This means that in each scenario, one of the models will be the underlying model and will have relative RPS equal to 1. We then compared the coverage of the posterior predictive distributions on the same 200 random boxes using a 90% credible interval.

Figure 3.7 shows the relative RPS for the simulated HPP data with $\mathbb{E}[n] = 100$. The results shows that the models all perform similarly, with each model occasionally performing slightly better than the HPP and occasionally performing slightly worse. As shown in Figure B.1, the coverage is generally adequate, though occasionally drops below 80% for all of the models as $q$ gets large. The results when $\mathbb{E}[n] = 1000$ are similar for the relative RPS and slightly improved for the coverage.

For the simulated NHPP data, shown in Figure B.2 shows that the HPP model performs poorly and gets worse as $\mathbb{E}[n]$ gets large. The LGCP models performed similarly to the NHPP model, though they appear to slightly outperform the NHPP model for $\mathbb{E}[n] \approx 1000$ in terms of both RPS and coverage, even though the NHPP model is the underlying data-generating model. The NHPP model provides poor coverage in the $\mathbb{E}[n] \approx 1000$ case.

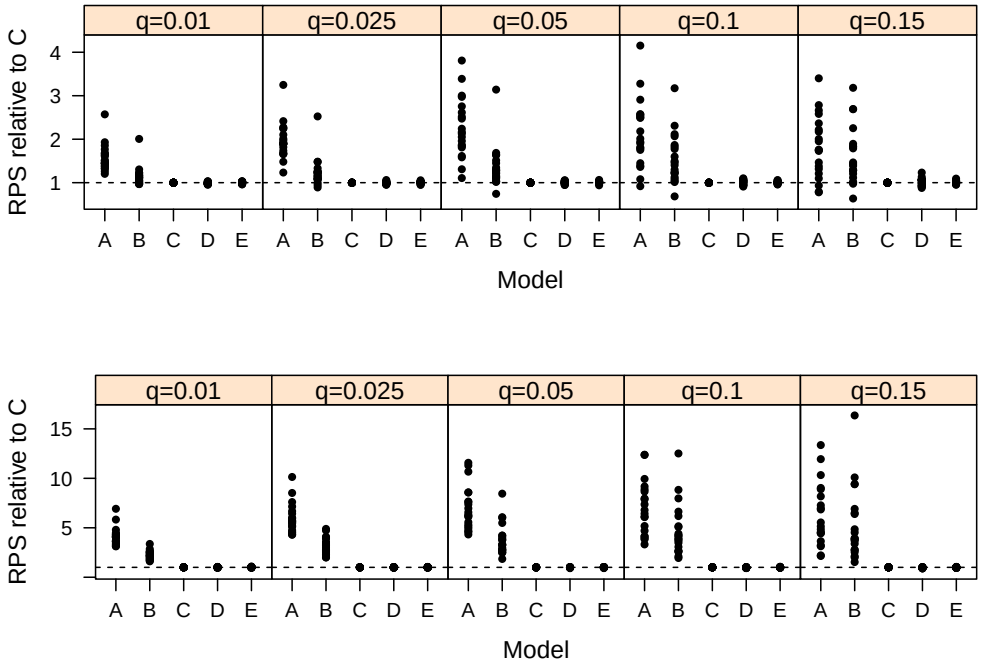



FIGURE 3.8: The relative RPS for the simulated LGCP (exponential covariance) data with $\mathbb{E}[n] \approx 100$ (top row) and $\mathbb{E}[n] \approx 1000$ (bottom row). The model labels are the same as those used in Figure 3.7.

The results for all of the simulated LGCP datasets are similar. Figure 3.8 shows the relative RPS for $\mathbb{E}[n] \approx 100$ and 1000. We see that the NHPP model outperforms the HPP model, yet both are generally worse than the LGCP models. For $\mathbb{E}[n] \approx 1000$ the LGCP models demonstrate a clear performance advantage of the HPP and NHPP models. The coverage levels, shown in Figure B.3, show that the HPP and NHPP models generally have poor coverage, while the LGCP models generally have coverage that is at least 70%.

The general results of these figures show that when the sample size is small, the models are not always distinguishable, regardless which model was used to generate the data. With only 100 or so training data points, it is not surprising that there is little to learn. However, as more information is available, such as in the high data setting, the models begin to distinguish themselves. Fitting the right model or a more

flexible one (such as using an LGCP model for the NHPP data) usually performed about the same. However, fitting a less flexible model than was used to generate the data (such as an HPP model for the NHPP data) gave higher RPS scores, with increasing significance as the number of data points got larger. The results also show that there is little difference in the performance of the different LGCP models under any of the scenarios, even in the high data setting. It appears that the learning for the covariance function is very limited and should chosen to match the conception of smoothness in the underlying intensity.

The coverage results are similar to the RPS results. With little data, the models generally provided similar coverage regardless of the data-generating model used, especially for small $q$. However, with more data the less flexible models began to exhibit poor coverage. Even when fitting the right model to a specific dataset, we see that the coverage results can be quite variable and sometimes quite poor, especially with a small dataset. Again, the LGCP models all performed fairly similarly, even when there was lots of data. It appears that the choice of a covariance function for a LGCP model cannot, in our experience, be informed from the data, leaving the user to choose the covariance based on either expert opinion or user preference.

## 3.5 Summary

In this chapter, we have discussed tools for model diagnostics and model selection for point process models. We have introduced predictive residuals as a natural Bayesian residual which uses posterior predictive point patterns to compare observed and predicted counts over random subsets of $D$. The predictive residuals can be expected to provide the nominal level of coverage, though the simulation study showed that even the true model can easily overfit the data and provide poor coverage.

We proposed $p$-thinning be used as a coherent method for providing cross-validation for point patterns. This provides the benefit that the intensity can be naturally ad-

justed to make predictions on test data. Finally, we suggested ranked probability scores as a powerful method for comparing posterior distributions over point patterns in the course of model selection. With little data, the simulation study showed that it can be hard to effectively choose a proper model, but with large amounts of data it appears that the data can provide some direction in choosing the best model.

# 4

# Analysis of Complex Spatial Point Processes

The point processes described and used in the previous chapters are among the simplest forms for point processes. Extensions to these basic forms include adding clustering or inhibition between points, modeling the intensities over time as well as space, and adding marks to the events at each location. We first demonstrate how to adapt our previous methods to Gibbs processes, which are generally used to describe points which exhibit inhibition or regularity. Section 4.2 will introduce cluster point processes and describe relevant posterior inference procedures. Finally, Section 4.3 will introduce ideas for further research, such as applying these methods to marked point patterns and other processes.

Poisson and Cox processes imply that, given the intensity function, the locations of the events of a point process are independent. More general models relax the independence assumption and allow points to cluster together or spread apart. For example, Harkness and Isham (1983) study locations of ant nests to learn whether ant nests tend to cluster around or spread apart from nests of ants of the same or other species. They find evidence that one species, *Messor wasmanni*, tends to spread their nests apart from each other more than would be expected under

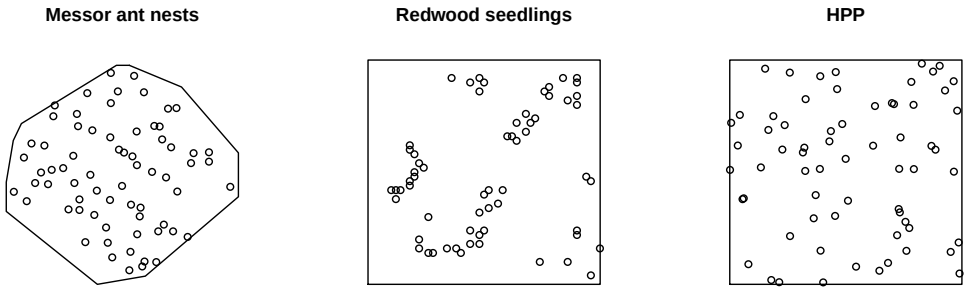



FIGURE 4.1: Plots of *Messor* ant nests (inhibitive), redwood seedlings (clustered), and simulated homogeneous Poisson point patterns (completely spatially random).

the independence assumption. Conversely, Strauss (1975) and subsequent analyses have looked at redwood seedlings which appeared to be more clustered than would normally happen under independence. This is sensible given that the seedlings likely did not travel far from their parent tree.

Figure 4.1 shows shows both datasets as well as a draw from a homogeneous Poisson process to demonstrate the difference between inhibition (also called regularity), clustering, and complete spatial randomness. The ant nests exhibit a regular pattern with some space between neighboring nests, whereas the redwood data has clusters with lot of empty space between the clusters. The homogeneous Poisson process is somewhere in between the two, with some large gaps between points and some points that are very close together.

The distinction between an inhomogeneous intensity and dependence among locations is a complex matter often discussed in the literature. Looking at the redwood data, it may not be clear whether the point pattern arises from a process with a spatially varying intensity or a cluster process. It is, of course, possible for point patterns to exhibit spatial inhomogeneity (first-order characteristic) as well as inhibition or clustering (second-order characteristic), though the distinction can be hard to detect (see, e.g., Baddeley et al., 2000). It is known to be especially difficult to distinguish between clustering and a spatially varying intensity. Consultation with

those familiar with the point pattern in question may be required to specify valid models. We will not delve further into this issue, however, and will assume that we are able to use our knowledge of the natural process to suggest reasonable models.

## 4.1 Gibbs Processes

Gibbs processes constitute a broad class of models for point patterns. A point process is a Gibbs process if its location density can be written as

$$f_n(S) = \exp(-Q(S)) \tag{4.1}$$

$$\text{where } Q(S) = c_0 + \sum_{i=1}^{n} h_1(s_i) + \sum_{i \neq j} h_2(s_i, s_j) + \cdots + h_n(s_1, \ldots, s_n). \tag{4.2}$$

The Gibbs process is characterized by the interactions among the points in $S$, as specified through the function $Q$. $c_0$ is an unknown constant making the density integrate to 1 and $h_k$ represents a potential of order $k$. The first potential $h_1$ controls the intensity, resulting in an NHPP with intensity $\lambda(s) = e^{-c_0 - h_1(s)}$ if no higher-order potentials are used (this of course is the only case with a known normalizing constant). The higher order potentials control higher order interactions, such as clustering or inhibition. To ensure integrability, it is usually required that $h_k \geq 0$ for $k \geq 2$. This in turn implies that we will capture inhibition with these models, so the point patterns will be more regular than those from a Poisson process.

Usually only pairwise interaction processes are considered, meaning that only $h_1$ and $h_2$ are used, with $h_2$ usually being a function of the distance $||s_i - s_j||$. Such processes usually also assume a Markov property, in that the interaction potential is only nonzero within a distance $R$. In other words, if $h_2(s_i, s_j) = h_2(||s_i - s_j||)$, then the Markov property implies that $h_2(||s_i - s_j||) = 0$ if $||s_i - s_j|| > R$. Under this property, points only interact if they lie within distance $R$ of each other, just as Markov random fields exhibit a local dependence structure.

One of the most common forms for a pairwise interaction Gibbs process is the Strauss process (Strauss, 1975). The Strauss process sets $h_2(d) = -\log\gamma$ if $d \leq R$ and 0 otherwise. The restriction of $h_2 \geq 0$ for integrability implies that $0 \leq \gamma \leq 1$. Specifying $h_1(s) = \beta$ provides a constant first-order intensity, resulting in a homogeneous Strauss process. The location density for the homogeneous Strauss process is then

$$f_n(S) = e^{-c_0(\beta,\gamma)}\,\beta^n\,\gamma^{s_R(S)}, \tag{4.3}$$

where $s_R(S)$ counts the number of pairs of points $(s_i, s_j) \subset S$ with $||s_i - s_j|| \leq R$. Setting $\gamma = 0$ results in a hard core process, which prevents any two points in $S$ being within distance $R$ of each other.

The Strauss process has Papangelou conditional intensity

$$\lambda(s; S) = \beta\,\gamma^{s_R(S\cup\{s\})-s_R(S\setminus\{s\})}, \tag{4.4}$$

where the unknown normalizing constant has now been canceled out. Since the unknown normalizing constant makes the intensity function intractable, the Papangelou intensity is often used as a convenient substitute.

Generating Gibbs processes can be done using birth-death algorithms, as outlined in, e.g., Section 3.6.3 in Illian et al. (2008). These methods run an MCMC chain and propose changes to the current point pattern until the process has reached its stationary distribution, though as usual it is impossible to know exactly when the stationary distribution has been reached. Usually summaries such as $n$ or $\sum_{i=1}^{n} h_1(s_i) + \sum_{i\neq j} h_2(s_i, s_j)$ are monitored until convergence seems likely. An alternate method has been proposed by Berthelsen and Møller (2002) and Berthelsen and Møller (2003), who develop a perfect simulation algorithm to simulate from spatial point processes such as Strauss processes. Their method, using dominated coupling from the past, provides a simulation from the exact desired distribution, whereas the birth-death algorithms only provide an approximation.

### *4.1.1 Model Fitting for Gibbs processes*

The intractable normalizing constant in the Gibbs process likelihood (4.2) complicates fitting Gibbs process models. Frequentist estimation generally proceeds by maximizing the pseudolikelihood, which again has no unknown normalizing constant. The exponential family form of Gibbs processes make them convenient to analyze using typical generalized linear model methods. Baddeley and Turner (2000) describe how to use a Berman-Turner device to estimate the maximum pseudolikelihood estimates. Illian and Hendrichsen (2010) further generalize this model to provide more general interactions.

From a Bayesian standpoint, King et al. (2012) provide a Bayesian version of Illian and Hendrichsen (2010), in which the pseudolikelihood is again used. To avoid using the pseudolikelihood, however, the issue of the unknown normalizing constant must be addressed. Gelman and Meng (1998) propose a bridge sampling method to compute ratios of normalizing constants, which could be used in a Metropolis-Hastings algorithm to sample the model parameters $\theta$. Møller et al. (2006) discuss an auxiliary variable approach in which the auxiliary variable comes from the same state space as the point pattern. In their approach, the normalizing constant cancels out with that of the auxiliary variable in the Metropolis-Hastings ratio, removing the need for pseudo likelihood or bridge sampling. Berthelsen and Møller (2006) further study this approach and demonstrate its use for Strauss processes.

The auxiliary variable method proceeds as follows: Given data $y$ with likelihood $f(y|\theta) = q_\theta(y)/Z_\theta$, the goal is to simulate from the posterior distribution $\pi(\theta|y) \propto \pi(\theta)f(y|\theta)$. Taking the $y$ to be a point pattern and $\theta$ to be the parameters governing the point process, $Z_\theta$ is the unknown normalizing constant. The basic Metropolis-Hastings algorithm proceeds by proposing candidate values of $\theta$, labeled by $\theta'$, from

the density $p(\theta'|\theta)$. $\theta'$ is then accepted with probability $\alpha(\theta'|\theta) = \min(1, H(\theta'|\theta))$, where

$$H(\theta'|\theta) = \frac{\pi(\theta')q_{\theta'}(y)p(\theta|\theta')}{\pi(\theta)q_{\theta}(y)p(\theta'|\theta)} \times \frac{Z_\theta}{Z_{\theta'}}. \tag{4.5}$$

Equation (4.5) above has a ratio of normalizing constants, which could be estimated using the bridge sampling method of Gelman and Meng (1998), as noted previously. The auxiliary variable method gets around this by introducing an auxiliary variable $x$ defined on the same state space as $y$ with conditional density $f(x|\theta, y)$. The proposal distribution $p(\theta', x'|\theta, x)$ for both $\theta$ and $x$ can be chosen to factor as $p(\theta', x'|\theta, x) = p(x'|\theta')p(\theta'|\theta)$. Further, the proposal density $p(x'|\theta')$ is taken to be the same distribution as the likelihood for $y$. This introduces more normalizing constants, but in such a way that they cancel out all the existing normalizing constants and makes the Hastings ratio tractable. The Hastings ratio is then

$$H(\theta'|\theta) = \frac{f(x'|\theta', y)\pi(\theta')q_{\theta'}(y)q_{\theta}(x)p(\theta|\theta')}{f(x|\theta, y)\pi(\theta)q_{\theta}(y)q_{\theta'}(x')p(\theta'|\theta)}. \tag{4.6}$$

As noted in Berthelsen and Møller (2006), the critical issues then are to choose an appropriate auxiliary density $f(x|\theta, y)$ and proposal density $p(\theta'|\theta)$. For our purposes, we have generally experienced satisfactory results with a random walk proposal for $p(\theta'|\theta)$. For the auxiliary density, we employ the fixed Strauss density from Berthelsen and Møller (2006), which uses the MLE estimates $\hat{\beta}$ and $\hat{\gamma}$ in the Strauss likelihood. Though the fixed Strauss density still has an unknown normalizing constant $Z_{\hat{\theta}}$, the constant is determined by the MLE estimates $\hat{\theta}$ and is not tied to either $\theta$ or $\theta'$. Thus, the normalizing constants will cancel out in $f(x'|\theta', y)/f(x|\theta, y)$ from (4.6) and the Hastings ratio will be tractable. This auxiliary density is expected to work well if the posterior for $\theta$ is concentrated around $\hat{\theta}$.

### *4.1.2 Simulation Study*

With interest in being able to perform model diagnostics and model selection, we present a few results from a simulation study. The goal of the simulations was to see whether a Strauss model would exhibit a lack of fit when fitted to data from an HPP, and whether an HPP model fit to Strauss process data would likewise exhibit a lack of fit. Alternatively, if the incorrect model didn't exhibit a lack of fit, would the ranked probability scores give preference to the correct model? Of course, this also includes checking whether the correct model fits the corresponding data appropriately.

As was observed in our previous simulation studies, we expect that the Strauss data will need to be fairly different from the HPP data in order for model selection to be possible. Generating data from a Strauss process with $\gamma = 1$, or similarly, with $R = 0$, is equivalent to generating data from an HPP. This raises the question of how small $\gamma$ needs to be, in combination with how large $R$ needs to be, to distinguish between Strauss and HPP data? Answering this question fully will not be attempted, yet the results presented will provide insight these questions. As before, we also consider the effect of sample size in being able to distinguish between the processes.

The two data-generating processes used in the simulation study are an HPP with $\lambda = 100$ and a Strauss process with $(\beta = 250, \gamma = 0.05, R = 0.05)$. Two domains were used to provide a comparison between the learning available on a small domain versus a larger domain. By keeping the parameter values the same for each domain, using a small and large domain is the analogue to the low and high intensity settings used in the simulation study in the previous chapter. Here, the larger domain will provide more information about the process than the smaller domain. The small domain $D_1$ will be the unit square $[0, 1] \times [0, 1]$ and the larger domain $D_2$ will be the square $[0, \sqrt{10}] \times [0, \sqrt{10}]$, such that the larger domain is ten times larger than the smaller domain and should help to distinguish between the two processes.

The models fit to each simulated dataset will be an HPP model, with $\lambda$ unknown, and a Strauss model, with $R = 0.05$ fixed and $(\beta, \gamma)$ unknown. Several point patterns from each data-generating process will be simulated over each domain. Then both models will be fit to each simulated dataset, with the goal of learning whether the data are informative enough and whether we can adequately detect the true model. As before, we can compare the coverage and ranked probability scores of the predictive residuals over random subsets of the domain.

One issue that arises is that our cross-validation method using $p$-thinning from the previous chapter is not proper here. With the interaction among points, removing points will alter the interpoint distances. This will change the dependence structure in the data and therefore bias the parameter estimation. In our simulation study, we can overcome this by generating two realizations from the same process and using one as training data to fit our model and the other as test data for our model diagnostics and model selection. The RPS, predictive residuals, and two variance metrics can be computed on both the training and test data. With real data, however, the best that can be done is to use all the data to fit the model and then calculate these quantities on the full dataset. This follows the posterior predictive check ideas of Gelman et al. (1996).

The simulation study was run on a smaller scale than the simulation study in the previous chapter. We will present outcomes from one simulation which were typical of replications from the same data-generating process. The auxiliary variable approach used to fit the Strauss process model was found to often encounter very poor MCMC acceptance rates, making a mass simulation and comparison infeasible. The was likely due to the added complexity arising from the auxiliary variable $x$, which here is an auxiliary point pattern. The chain generally sampled well except for long, intermittent periods of staying on a single set of parameters. Berthelsen and Møller (2006) found partially ordered Markov models (POMMs) to be useful in alleviating

the stickiness of the chain, but we also found that fine-tuning the proposal densities and starting parameter values were useful as well. However, when $\mathbb{E}[n]$ is large, the MCMC chains still exhibited very poor mixing despite both of these strategies.

The ranked probability scores and coverages for each scenario were computed and analyzed as before. They are not given here because they showed little ability to consistently choose the correct model. Using RPS for model choice, the correct model was chosen only slightly more than 50% of the time, even in the high-data setting. The coverage of the predictive residuals in all cases were at or above the nominal 90% level, so our previous model fit diagnostics were incapable of distinguishing between processes with a constant first-order but different second-order characteristics.

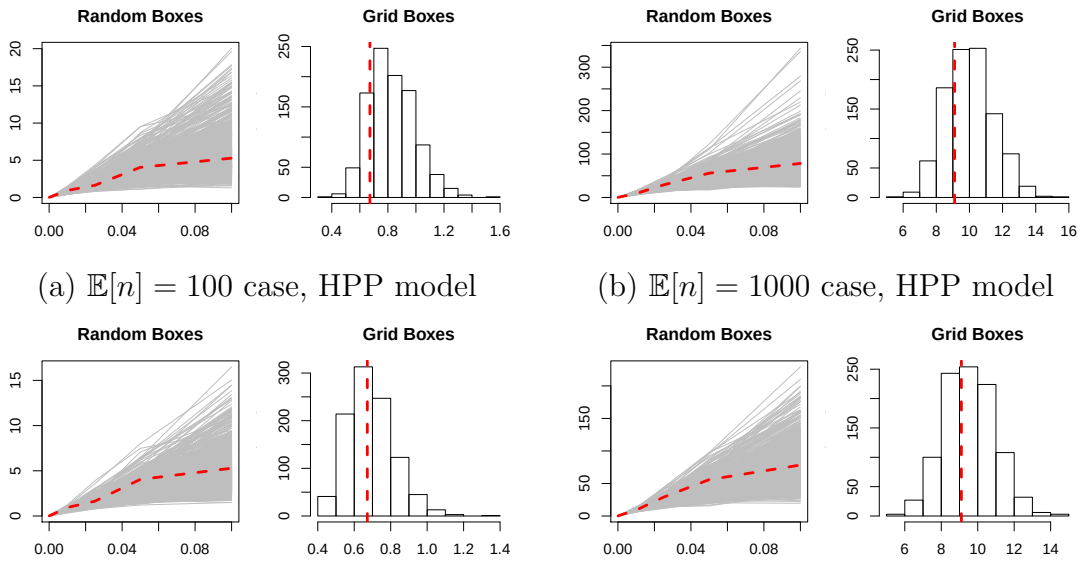
A few additional metrics we considered for model diagnostics and comparison are aimed at comparing the regularity of the observed and posterior predictive point patterns. The regularity of the point pattern is a second-order characteristic, whereas the predictive residuals and ranked probability scores are tied to the first-order characteristics. For example, we first consider dividing the domain into, say, 100 boxes of equal size and calculating the variance of the counts in each box. If the model fits well, then this variance metric on the observed point pattern $S$ should be similar to those obtained using the posterior predictive point patterns. For an irregular domain, an alteration could be made to discretize the domain according to some grid and then calculate the variance using only those counts where the entire box is inside the domain, discarding those regions near the borders of the domain.

A second, similar metric we also consider extracts the counts from random subsets of the domain, as is done for the predictive residuals, and calculates the variance of those counts. For each choice of $q$, the variance in the counts of the observed point pattern on the random subsets can be compared to those from the predictive point patterns. Again, a well-fitting model should provide values in a similar region to the observed value for $S$.

Berthelsen and Møller (2008) took a similar approach of using simple posterior predictive checks for model diagnostics, comparing the observed $N(D)$ and minimum interpoint distance to their distributions using posterior predictive point patterns. Their method fits in well with our methods thus far, though looking at the distribution of $N(D)$ did not help to distinguish between the HPP and Strauss models here. The minimum observed interpoint distance could also be a useful metric here, since it addresses the second-order structure.

Figure 4.2 provides comparisons of the two variance metrics discussed previously when fitting both models to the simulated HPP data. The red, dashed lines indicate the observed variance metrics for the observed point pattern, while the histograms and gray lines indicate the same variance metrics calculated on the posterior predictive point patterns from the fitted model. We see that the observed variance metrics lines up fairly well with the variance metrics calculated on the predictive point patterns, regardless of the model or value of $\mathbb{E}[n]$. This says that the variance of the counts are fairly similar in each case and neither model indicates a lack of fit. The variance metrics can only indicate lack of fit but are not able to definitively suggest that the model fits well. In these simulations, the posterior distribution for $\gamma$ in the Strauss model was close to 1, especially in the $\mathbb{E}[n] = 1000$ case, signifying that the Strauss model was converging to an HPP.

Figure 4.3 shows the same set of figures for the simulated Strauss data. We see that the HPP posterior predictive point patterns exhibit cell counts with a much higher variance than the observed point pattern when looking at the grid boxes. The random boxes metric provides similar evidence for small $q$ and when $\mathbb{E}[n]$ is large. The Strauss model, however, generates posterior predictive point patterns with similar variance metrics, giving no evidence of lack of model fit by either of these two metrics. These plots were typical of most of the simulated point patterns we looked at, though the variation in the Strauss simulated data for the $\mathbb{E}[n] \approx 100$




(a) $\mathbb{E}[n] = 100$ case, HPP model

(b) $\mathbb{E}[n] = 1000$ case, HPP model

(c) $\mathbb{E}[n] = 100$ case, Strauss model

(d) $\mathbb{E}[n] = 1000$ case, Strauss model

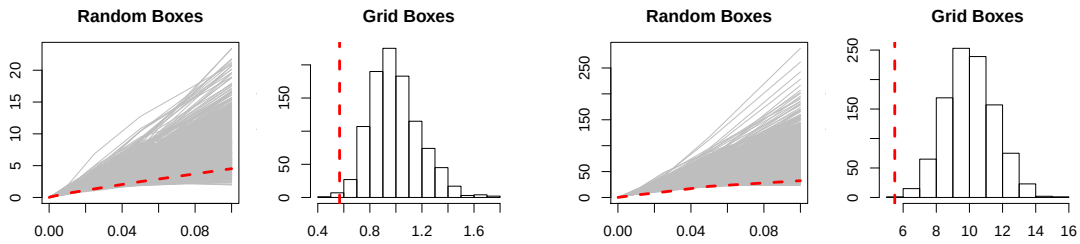
FIGURE 4.2: Variance metrics for simulated HPP($\lambda = 100$) data with $\mathbb{E}[n] \approx 100, 1000$. The top row shows the results when fitting the HPP model to the HPP data and the second row shows the results using the Strauss model. The dashed line indicates the observed variance metrics.

case resulted in a few cases where the HPP model didn't exhibit such a strong lack of fit.

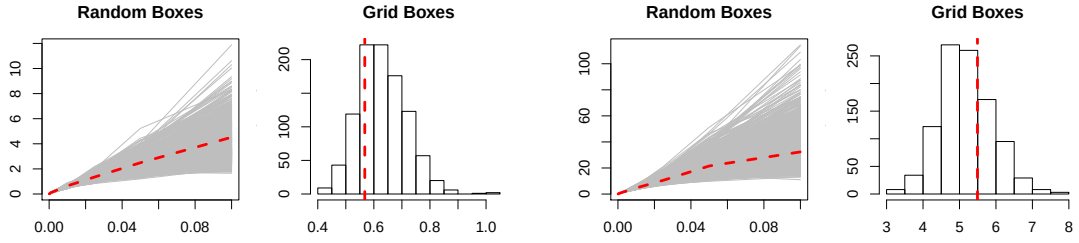
We also note that the auxiliary variable MCMC method for fitting the Strauss process had extremely poor acceptance rates for the Strauss data when $\mathbb{E}[n] \approx 1000$, even after tuning the proposal densities as best as was possible. This is likely due to the auxiliary variable being hard to update, given that the update involves jointly accepting an approximately 1000-dimensional auxiliary variable. This issue has not been investigated in the literature, but severely complicates Bayesian model fitting when the sample size is large.

### *4.1.3 Swedish Pines Data*

We now fit the HPP and Strauss models to the Swedish pines data from Baddeley and Turner (2000) and Ripley (1981). The data consists of the locations of 71 pine




(a) $\mathbb{E}[n] = 100$ case, HPP model (b) $\mathbb{E}[n] = 1000$ case, HPP model

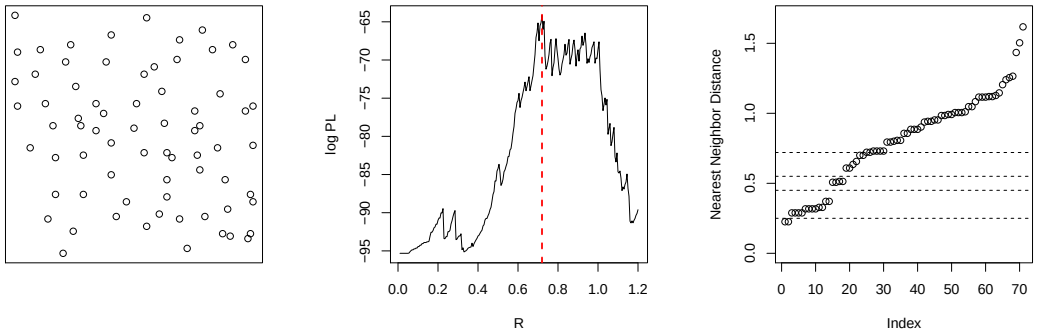



(c) $\mathbb{E}[n] = 100$ case, Strauss model (d) $\mathbb{E}[n] = 1000$ case, Strauss model

FIGURE 4.3: Variance metrics for simulated Strauss($\beta = 250, \gamma = 0.05, R = 0.05$) data with $\mathbb{E}[n] \approx 100, 1000$. The top row shows the results when fitting the HPP model to the Strauss data and the second row shows the results using the Strauss model. The dashed line indicates the observed variance metrics.

saplings within a 10m $\times$ 10m square. The data is shown in Figure 4.4, along with the profile pseudolikelihood of the Strauss model and the nearest neighbor distances for each $s_i \in S$.

We fit the HPP model and compared its performance with four Strauss models with different values for $R$. The minimum observed interpoint distance is 0.22 with most of the nearest neighbors being greater than 0.5, so the values of $R$ we initially consider are $R = 0.25$, 0.45, and 0.55. However, the profile maximum pseudolikelihood estimate of $\hat{R} = 0.72$ is much higher than our initial candidates, so we also include 0.72 as a candidate for $R$. These candidate values are shown as dashed horizontal lines in Figure 4.1.3. Each model was run for 1,000 iterations of burn-in and then 50,000 posterior samples were collected. Every $50^{\text{th}}$ sample was used to generate a posterior predictive point pattern.




(a) (b) (c)

FIGURE 4.4: Plots of (a) the Swedish pines data, (b) profile pseudolikelihood for the Strauss model as a function of $R$, and (c) the (sorted) nearest neighbor distances. The dashed line in (b) indicates the profile maximum pseudolikelihood estimate $\hat{R} = 0.72$. The dashed lines in (c) indicate the candidate $R$ values of 0.25, 0.45, 0.55, and 0.72.

Table 4.1: RPS scores for the HPP and Strauss models on the Swedish pines data. The coverage of the 90% intervals are given in parentheses.

| Model | $q = 0.005$ | 0.01 | 0.025 | 0.05 | 0.10 |
|---|---|---|---|---|---|
| HPP | 0.25 (0.98) | 0.34 (1.00) | 0.51 (1.00) | 0.73 (1.00) | 1.17 (1.00) |
| Strauss ($R = 0.25$) | 0.25 (0.98) | 0.34 (1.00) | 0.51 (1.00) | 0.72 (1.00) | 1.18 (1.00) |
| Strauss ($R = 0.45$) | 0.25 (0.98) | 0.34 (1.00) | 0.50 (1.00) | 0.71 (1.00) | 1.13 (1.00) |
| Strauss ($R = 0.55$) | 0.24 (0.98) | 0.33 (1.00) | 0.49 (1.00) | 0.69 (1.00) | 1.07 (1.00) |
| Strauss ($R = 0.72$) | 0.24 (0.98) | 0.33 (1.00) | 0.49 (0.98) | 0.69 (1.00) | 1.10 (0.96) |

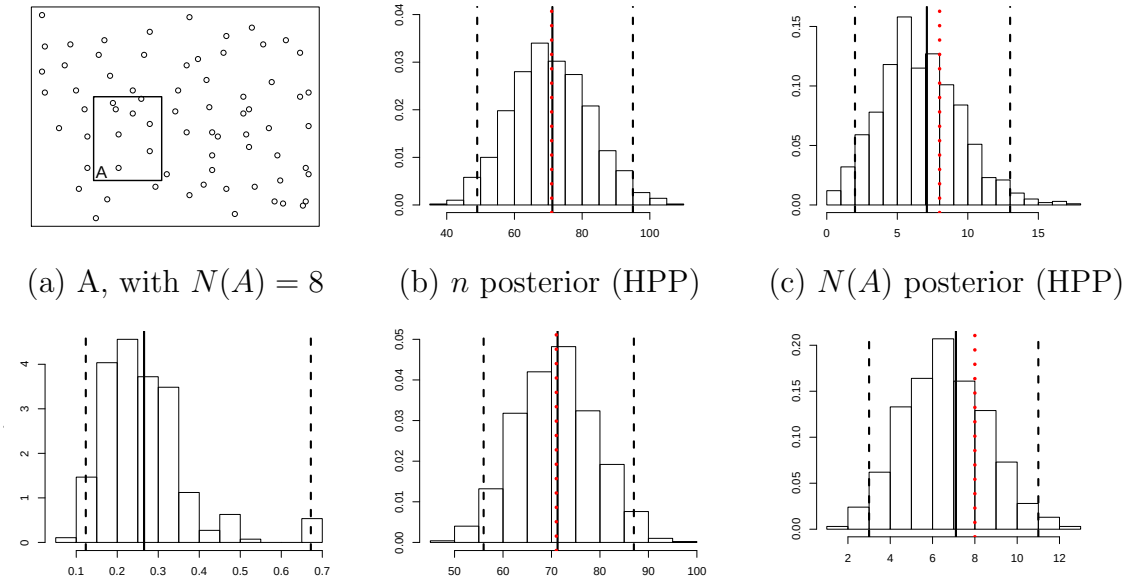
A full analysis will not be given here, but Table 4.1 shows the RPS and coverage results for the HPP and Strauss models using 200 random boxes placed in $D$ with size $q|D|$. These results are shown based on using the data used to fit the model since we cannot employ cross-validation here. The coverage rates show that all of the models give similar RPS and provide sufficient coverage using predictive residuals. In fact, the coverage percentages for all of the models are all well above the 90% nominal level, yet some inflation is not unexpected since this is calculated on the same data used to fit the model. The RPS results, show that the Strauss models are very similar, with the largest differences occurring for the largest box size ($q = 0.10$).

Taking the Strauss model with $R = 0.72$, Figure 4.5d shows the posterior distribution for $\gamma$, the interaction potential. Since the Strauss model simplifies to an HPP for $\gamma = 1$, the posterior for $\gamma$ suggests that some interaction is present, given that most of the mass is in the range (0.1, 0.4). The mass near 0.7 arises due to the MCMC chain getting stuck in an extreme location for a few hundred iterations. In practice, we would want to run this chain for much longer to smooth out over these aberrations, yet we left this in to demonstrate this tendency of the auxiliary variable method. Møller et al. (2006) noted this tendency, yet we found it to worsen as $\gamma$ get smaller, $R$ gets larger, or for datasets with large $n$.

We can also compare other posterior summaries of interest under this Strauss model and the HPP model. Figure 4.5 shows the posterior distributions for $n$ and $N(A)$ under both models, where $A = [2, 4.5] \times [2, 6]$ and $|A| = 0.1|D|$. The posterior summaries for both $n$ and $N(A)$ seem pretty similar under both models, though the posteriors under the Strauss model are slightly more concentrated around the observed values. This is consistent with the RPS results, which suggested that the Strauss model provided predictions that were closer to the observed values.

Figure 4.6 show the posterior estimates for the $F$, $G$, and $K$ under the Strauss model with $R = 0.72$. These are compared to their theoretical values under CSR, employing the estimate $\hat{\lambda} = n/|D|$. While $\tilde{F}(d)$ is close to the theoretical $F(d)$ under CSR, we see that $\tilde{G}(d)$ for the Strauss model stays lower than the theoretical $G(d)$ under CSR, with a smooth bend in the curve right around 0.72. This is expected given that using $R = 0.72$ in the Strauss model discourages neighbors within 0.72 feet.

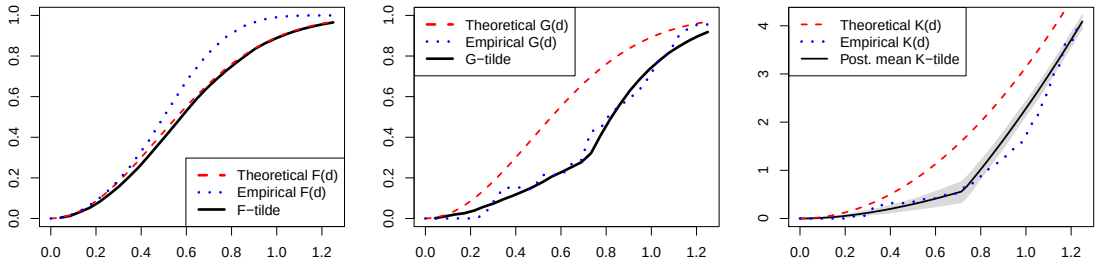
The $K$-function presented in Chapter 2 was only valid for models where the form for the intensity is known. For a Strauss process, the intensity is intractable, so we are currently developing general forms for $K$-functions for processes with a constant intensity function. Briefly, the new approach uses the equality $\lambda = \mathbb{E}[N(D)]/|D|$




(a) A, with $N(A) = 8$ (b) $n$ posterior (HPP) (c) $N(A)$ posterior (HPP)

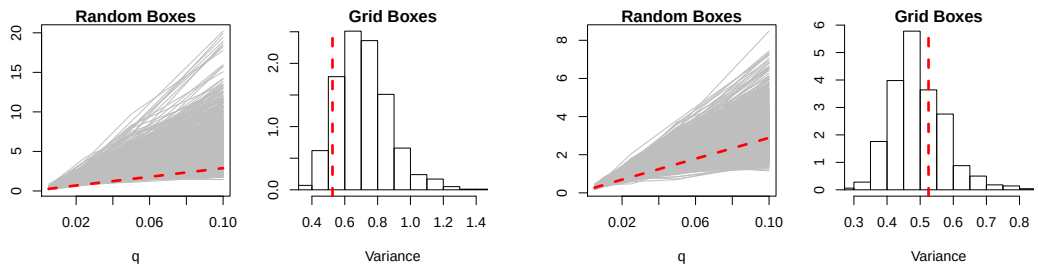
(d) $\gamma$ posterior (Strauss) (e) $n$ posterior (Strauss) (f) $N(A)$ posterior (Strauss)

FIGURE 4.5: Plots of (a) the Swedish pines data with subregion $A$ labeled and the posterior distributions for (b–c) $n$ and $N(A)$ under the HPP model, and (d–f) $\gamma$, $n$, and $N(A)$ under the Strauss($R$=0.72) model. The solid and dashed lines indicate the posterior means and 95% credible intervals, respectively, and the dotted lines indicate the observed values.




(a) $F(d)$ (b) $G(d)$ (c) $K(d)$

FIGURE 4.6: Plots of the $F$-, $G$-, and $K$-functions for the Strauss model with $R = 0.72$. The theoretical forms use the MLE for $\hat{\lambda}$ and the empirical estimates are the standard nonparametric estimates. The shaded area in (c) represents the 95% pointwise credible intervals for $K(d)$.




(a) HPP model (b) Strauss model

FIGURE 4.7: Plots of the variance of box counts for the HPP and Strauss ($R = 0.72$) model. The dashed lines indicate the observed variance, while the histogram and gray lines indicate predictive values under the model.

to replace the $\lambda$ in the denominator of (2.25) with $\mathbb{E}[N(D)]/|D|$. This provides a $K$-function for the Strauss process, as shown in Figure 4.6. We see that the posterior mean for $K(d)$ is lower than the theoretical $K(d)$ under CSR for all $d > 0.2$ or so. The credible interval does not contain the theoretical values, suggesting that CSR is not met here and an HPP model is inappropriate for this data. Since the Strauss process can simplify to being an HPP model, this shows that the Strauss model strongly preferred to include a significant amount of inhibition.

As one last way of comparing the models, we could look at the variance of counts in boxes over $D$. Under the HPP model, the counts would be expected to exhibit higher variability than under the Strauss model, which would encourage regularity and similar counts in the boxes. Figure 4.7 shows this variance comparison, both using 100 grid boxes and random boxes with varying $q$. The gray lines on the plots on the right side indicate the variance observed for each $q$ for a posterior predictive dataset. For the plots using the random boxes, a histogram could be drawn at each $q$, just as was done for the plots on the left side. This isn't shown but the histograms at each $q$ are similar to the histograms on the left-side plots.

In Figure 4.7, the predictive patterns under the HPP model slightly higher variance metrics than the observed values, but it is not clear whether this is significant.

Table 4.2: The posterior $p$-values for the posterior predictive variance metrics using random boxes of varying size $q|D|$.

| Model | $q = 0.005$ | 0.01 | 0.025 | 0.05 | 0.10 |
|---|---|---|---|---|---|
| HPP | 0.163 | 0.024 | 0.005 | 0.025 | 0.116 |
| Strauss($R = 0.72$) | 0.475 | 0.207 | 0.148 | 0.253 | 0.482 |

The Strauss model, however, clearly generates point patterns with similar variance to the observed values. While generating Strauss processes in the simulation study, the HPP model would sometimes produce results similar to what we see here, but most of the time the HPP was clearly different from the Strauss model.

If we look at the random box variance metric under the HPP model, we see that the dashed line is close to the lower edge of the predictive values. We can calculate, for each $q$, the posterior $p$-value for the predictive check, which is just the proportion of simulated variance metrics that are below the observed metric at that value of $q$. Table 4.2 shows the posterior $p$-values for these variance metrics under the HPP model and the Strauss ($R = 0.72$) model. We see that the HPP model exhibits some fairly small variances for $q \in (0.01, 0.05)$, enough so that one would have evidence that HPP model exhibits some lack of fit. This also agrees with the evidence from the $K$-function that the inhibition seems significantly different from complete spatial randomness at certain distances.

We have not discussed comparing a Strauss process model to, for example, an area-interaction process model (Baddeley and van Lieshout, 1995; Widom and Rowlinson, 1970). The area-interaction model looks at the area of the union of discs (or spheres in $\mathbb{R}^3$) centered at each $s_i \in S$ and either encourages the union of discs to be small or large, depending on whether points are encouraged to be clustered or regular. We do not expect the data alone to provide enough information to reliably distinguish between the two processes empirically, especially for small sample sizes. Rather, we expect that the model choice in such a situation will be guided by the

nature of the problem and how well each model matches with the belief of the underlying process. What we have shown in this section is that we can, with enough data and enough regularity, distinguish between processes with completely random locations and those encouraging regularity.

## 4.2 Cluster Processes

In contrast to Section 4.1, we will now discuss clustered point processes. Many point patterns exhibit clustering, where the nearest neighbor distances are much smaller than might be expected under complete spatial randomness. Sometimes there may be a natural process explaining the clustering, such as seedlings being clustered a one large parent tree.

Clustered point processes are usually thought of as a superposition of point patterns, or $S = \cup_k S_k$, with each $S_k$ being a cluster of points. This fundamental change arises from the typical assumption that a clustered point pattern arises first as a set of (unobserved) clusters, followed by the observed points occurring within each cluster. Models differ by the distributions assigned to the clusters and then to the points within each cluster.

### *4.2.1 Common Cluster Processes*

A very common cluster process is the Neyman-Scott process (Neyman and Scott, 1958). A Neyman-Scott process first draws the cluster locations, called *parent* events, according to an NHPP. Denote the locations of the parent events by $\{\mu_1, \ldots, \mu_K\}$. Then each parent produces $N_k$ offspring according to some distribution, usually specified as $N_k \overset{iid}{\sim} \text{Poisson}(\delta)$. Given $N_k$, the *children* points are drawn within each cluster with their location relative to the parent governed by some density $f^\theta(s; \mu_k)$. The density $f^\theta$ can be any proper density, centered at $\mu_k$ and potentially depending on extra parameters $\theta$. A few special forms arise from this specification, as discussed

in, e.g., Banerjee et al. (2014). Taking $f^\theta$ to a Normal$(\mu_k, \sigma^2 I)$ distribution results in the (modified) Thomas process. The Matérn process results from setting $f^\theta$ to be uniform on a disc of radius $R$, or a ball of radius $R$ if $D \subset \mathbb{R}^d$ with $d > 2$.

The Neyman-Scott process can also be written as a mixture of densities, much like a mixture modeling approach to density estimation. In this form and given then number of parents $K$ and their locations $\{\mu_k\}$, the total number of points $n = \sum_{k=1}^{K} N_k$ is drawn rather than an $N_k$ for each cluster, with the distribution for $n$ determined by the distributions for the $N_k$. If $N_k \overset{iid}{\sim}$ Poisson$(\delta)$, then $n \sim$ Poisson$(K\delta)$. Then the locations of the points in $S$ are drawn according to the mixture of cluster intensities, $\sum_{k=1}^{K} \frac{1}{K} f^\theta(s; \mu_k)$.

The Neyman-Scott process is just another way to construct Cox processes, which we used in previous chapters. As noted before, a Cox process is a Poisson process with an intensity $\lambda(s)$ arising as the realization of a random process $\Lambda(s)$. Neyman-Scott processes are part of a larger family of Cox processes called shot noise Cox processes. A shot noise Cox process (SNCP) is a Cox process where $\lambda(s)$ (or, technically, the random function $\Lambda(s)$) is of the form

$$\lambda(s) = \sum_k \gamma_k k^\theta(s; u_k), \tag{4.7}$$

where $\gamma_k \in \mathbb{R}^+$ and $k^\theta(s; u_k)$ is a kernel centered at location $u_k \in D$ potentially having other parameters $\theta$. The kernel $k^\theta$ is a valid density function for $\mathbb{R}^2$ (or $\mathbb{R}^d$ for processes in other dimensions), with potentially finite support. This representation for $\lambda(s)$ is again reminiscent of using a mixture model approach to density estimation.

Similar to the Neyman-Scott process, the shot noise process can be generated first as a marked point pattern $U = \{u_k\}$ with each $u_k$ having an associated random mark, or shot, called $\gamma_k$. The shot determines the influence that $u_k$ has on the resulting

intensity surface. Smoothing the shots at each $u_k$ using the kernel $k^\theta$ provides the $\lambda(s)$ in (4.7).

The intensity function $\rho(s) = \mathbb{E}[\lambda(s)]$ is given by

$$\rho(s) = \int \gamma k^\theta(s; u) d\zeta(u, \gamma) \tag{4.8}$$

where $\zeta$ is a "locally finite diffuse intensity measure" (Møller, 2003) over the product space for $(u, \gamma)$. The measure $\zeta$ and the kernel $k^\theta$ determine the nature of the process. If the kernel $k^\theta$ is invariant under translations, meaning $k^\theta(s; u) = k^\theta(s-u)$, then the point process itself is stationary. Then, for example, if $\zeta$ assigns all of its mass at one constant value for $\gamma$, then a Neyman-Scott process results. Other, more complex, specifications of $\zeta$ can result in the Dirichlet process mixture model approaches of Kottas (2006), Ji et al. (2009), and Taddy and Kottas (2012). Finally, a further generalization using gamma measures results in the Poisson-gamma process (Wolpert and Ickstadt, 1998). Further discussion and theoretical development of shot noise Cox processes is available in, e.g., Møller (2003).

Posterior analysis for cluster process models will now be discussed, starting with a description of the Poisson-gamma process. We detail how to generate data from the Poisson-gamma process and how to fit the model to an observed point pattern. We present a small simulation study and analysis of observed data to illuminate the potential for posterior analysis in clustered processes.

### *4.2.2 Poisson-Gamma Processes*

The Poisson-gamma process (PGP) was developed by Wolpert and Ickstadt (1998) and used further in, e.g., Ickstadt et al. (1998) and Best et al. (2000). The Poisson-gamma process is a special case of a shot noise Cox process which essentially uses a kernel convolution to provide a spatially varying intensity surface. At the first stage, counts arise as a Poisson random variable given the intensity function, making

this a Cox process. The intensity function is constructed using a Gamma process to provide a random positive spatial surface, which is then used in a kernel convolution to provide a random intensity function.

For the Poisson-gamma process, the random intensity can be written as

$$\lambda(s) = \int_{D_{ext}} k^{\theta}(s,u)\Gamma(du), \tag{4.9}$$

for some kernel function $k^{\theta}$ and $\Gamma(du)$ has the Gamma random field distribution, $\Gamma(du) \sim \text{Gamma}(\alpha(du), \beta(u)^{-1})$, over an auxiliary domain $D_{ext}$. As our notation suggests, we take $D_{ext}$ to be an extension or superset of $D$. The random measure $\Gamma$ provides extra flexibility in the specification of the intensity surface. Covariate information can also be included by modifying (4.9) to

$$\lambda(s) = \int_{D_{ext}} e^{x^T(s)\eta} k^{\theta}(s,u)\Gamma(du), \tag{4.10}$$

where $x(s)$ represents the covariate information at location $s$ and $\eta$ is the set of regression coefficients. Additive models for the covariate information are more sensible in some cases, as discussed in Best et al. (2000), though we'll assume the multiplicative model makes sense here.

Both the shape measure $\alpha(du)$ and inverse scale function $\beta(u)$ are assumed to depend on unknown parameters, written as $\alpha(ds) = \alpha^{\theta}(ds)$ and $\beta(s) = \beta^{\theta}(s)$. For example, Wolpert and Ickstadt (1998) employ a uniform shape measure $\alpha^{\theta}(ds) = \theta_1/\theta_2 ds$ and constant scale $\beta^{\theta}(s)^{-1} = \theta_2$. This implies that the mean of the gamma field is $\alpha^{\theta}(ds) * \beta^{\theta}(s)^{-1} = \theta_1 ds$. Thus, a third stage is introduced to include a prior distribution for the $\theta$. The full hierarchical model can now be written as

$$\theta \sim \pi(d\theta) \tag{4.11}$$

$$\Gamma(du)|_{\theta} \sim \text{Gamma}(\alpha^{\theta}(du), \beta^{\theta}(u)^{-1}) \tag{4.12}$$

$$N(ds)|_{\theta,\Gamma} \sim \text{Poisson}(\Lambda(ds)) \text{ where } \Lambda(ds) \equiv \int_D e^{x^T(s)\eta} k^{\theta}(s,u) ds \Gamma(du) \tag{4.13}$$

Model fitting for the Poisson-gamma process is explained in detail in Wolpert and Ickstadt (1998) and Ickstadt et al. (1998). The hybrid Gibbs/Metropolis MCMC algorithm described in the references is also provided in Appendix C with a few details on employing the algorithm. The Inverse Lévy Measure Algorithm of Wolpert and Ickstadt (1998) is used to approximate the Gamma random field to the desired precision, which admits a finite representation of $\Gamma(du)$ as

$$\Gamma(du) \approx \sum_{m \leq M} v_m \delta_{\sigma_m}(ds), \tag{4.14}$$

where $v_m$ is a coefficient or shot, $\sigma_m$ are realizations of the Gamma random field, and $M \gg n$ is the truncation threshold. The approximation error is minimized as $M \to \infty$, though alternative MCMC schemes avoid the truncation and, hence, the approximation error.

The method for generating posterior predictive point patterns is not explicitly given in the references, especially when covariates are included, but will be outlined here. Using the truncation described above, the intensity function is approximated using

$$\lambda(s) = \int_{D_{\text{ext}}} e^{x^T(s)\eta} k^\theta(s;u)\Gamma(du) \approx \sum_{m \leq M} e^{x^T(s)\eta} v_m k^\theta(s;\sigma_m). \tag{4.15}$$

This approximation can then be rewritten as

$$\sum_{m \leq M} e^{x^T(s)\eta} v_m k^\theta(s;\sigma_m) = \sum_{m \leq M} c_m f_D^\theta(s;\sigma_m), \tag{4.16}$$

where $f_D^\theta$ is a proper density on $D$ with $\int_D f_D^\theta(s;\sigma_m) = 1$. Both $c_m$ and $f_D^\theta$ incorporate the effect of the covariate information $x(s)$, with $c_m \equiv \int_D e^{x^T(s)\eta} v_m k^\theta(s;\sigma_m) ds$ and $f_D^\theta(s;\sigma_m) \equiv (v_m/c_m) e^{x^T(s)\eta} k^\theta(s;\sigma_m)$. Generating a posterior predictive point pattern $S^*$ then proceeds by the following process, which requires a set of posterior draws for $(\{v_m\}, \{\sigma_m\}, \theta)$:

1. Calculate $\lambda(D) \equiv \int_D \lambda(s)ds$ to obtain $\mathbb{E}[N(D)]$ using the approximation in (4.15),

2. Draw $n^* \sim \text{Poisson}(\lambda(D))$,

3. For $i = 1, \ldots, n^*$

   (a) Choose a cluster index $m^*$ with $Pr(m^* = m) = c_m / \big( \sum_{j=1}^{M} c_j \big)$,

   (b) Draw a predictive point $s_i^* \sim f_D^{\theta}(s; \sigma_{m^*})$

4. Output the point pattern $S^* = \{s_i^*\}_{i=1}^{n^*}$.

### *4.2.3 Simulation Study*

With the previously noted difficulty that exists in distinguishing between a spatially varying intensity and a second-order dependence structure, especially in the case of a cluster process, we now present a simulation study which addresses this issue. Clustering presents an especially difficult case because a cluster of points may have either come from a cluster process with a constant intensity, or the clustering may be due simply to a high intensity region under an NHPP or LGCP, with the points occurring in that region independently. To study this, we will again generate data from a few different models and then use each model to fit each simulated dataset.

We again consider a low-data ($\mathbb{E}[n] \approx 100$) and high-data setting ($\mathbb{E}[n] \approx 1000$) and generate 10 simulations of each type of process at the low-data and high-data settings. For each simulated point pattern, a replicate pattern from the same process with the same parameters and intensity was generated to allow cross-validation. The processes we consider here are an HPP (constant intensity and independent point locations), an LGCP (spatially varying intensity and conditionally independent point locations), and a Poisson-gamma process (cluster process). The LGCP will use a Matérn covariance function with $\nu = 3/2$, as used in previous chapters. The Poisson-

gamma process will use a disc of radius $R$ for the kernel $k^\theta$, with the intent of making the process easier to distinguish from the LGCP by employing a finite kernel. For the Poisson-gamma process, we use $\alpha(ds) = e^{\theta_1} ds$ and $\beta(s)^{-1} = e^{\theta_2} ds$, as suggested in Wolpert and Ickstadt (1998), and $\Pi(ds)$ will be uniform over $D_{\text{ext}}$.

Table 4.3 provides a comparison of the ranked probability scores for the low-data setting of the simulation study. For each simulated dataset, the model with the smallest RPS for each value of $q$ was recorded. The HPP data simulation shows that the true HPP model has a slight edge on the performance of the LGCP model, with the Poisson-gamma model rarely being favored. For the LGCP data, the LGCP model was generally preferred, with the HPP and Poisson-gamma models also being preferred on occasion. For the Poisson-gamma data, the Poisson-gamma was almost exclusively the preferred model, especially for small $q$ (small grid boxes). Since the kernel function used in generating the Poisson-gamma data is a finite disc, it seems reasonable that the Poisson-gamma model would outperform at these small ranges since it learns the cluster locations and employs the same finite disc. The other models would instead smooth over the region and not provide the same sharp contrasts in the intensity function.

Table 4.4 provides a similar comparison of the ranked probability scores for the high-data setting of the simulation study. The results are similar here for the simulated HPP data, except that the LGCP model seems to provide essentially equal performance to the HPP model. For the LGCP data, the LGCP model is strongly preferred, especially on small ranges. It is possible that with so much data, the contrasts in the intensity are clearer and thus the LGCP model is preferred over the Poisson-gamma model, which under the current specification would assign constant intensity over small regions. For the simulated Poisson-gamma data, the LGCP model appears to be at least as good as the Poisson-gamma model. The LGCP model, it appears, was able to use the available data to sharpen its estimate of the

Table 4.3: Number of times (out of 10) having best RPS score on a simulated dataset under each type of data-generating process, with $\mathbb{E}[n] \approx 100$.

| HPP Data, $\mathbb{E}[n] = 100$ | | | | | |
|---|---|---|---|---|---|
| Model | $q = 0.005$ | 0.01 | 0.025 | 0.05 | 0.1 |
| HPP | 6 | 4 | 6 | 8 | 7 |
| LGCP (Matérn) | 3 | 6 | 4 | 2 | 2 |
| PGP (disc) | 1 | 0 | 0 | 0 | 1 |
| LGCP Data, $\mathbb{E}[n] \approx 100$ | | | | | |
| Model | $q = 0.005$ | 0.01 | 0.025 | 0.05 | 0.1 |
| HPP | 2 | 2 | 2 | 1 | 1 |
| LGCP (Matérn) | 7 | 5 | 6 | 7 | 6 |
| PGP (disc) | 1 | 3 | 2 | 2 | 3 |
| Poisson-Gamma Data, $\mathbb{E}[n] \approx 100$ | | | | | |
| Model | $q = 0.005$ | 0.01 | 0.025 | 0.05 | 0.1 |
| HPP | 0 | 0 | 0 | 0 | 0 |
| LGCP (Matérn) | 0 | 0 | 2 | 2 | 3 |
| PGP (disc) | 10 | 10 | 8 | 8 | 7 |

intensity function between the areas with clusters and those without, mirroring the Poisson-gamma model.

The tables above are useful in assessing decisions made based on which model had the best RPS, but they indicate nothing about the sizes of the RPS values. In some cases, the ranked probability scores will be very close and inconclusive. Figure 4.8 shows the RPS for each model for each simulated HPP dataset and setting of $\mathbb{E}[n]$. The figures actually show the RPS relative to the true model, in this case comparing the LGCP and PGP models to the HPP model. The LGCP values are generally clustered tightly around 1, indicative of very similar RPS, whereas the PGP model generally gives slightly larger RPS than the HPP model. In fact, sometimes the Poisson-gamma RPS is twice as large or larger. No big differences seem to exist between the low- and high-data settings.

Figure 4.9 shows the same set of plots for the simulated LGCP data, now comparing the RPS of the HPP and Poisson-gamma models to that of the LGCP model.

Table 4.4: Number of times (out of 10) having best RPS score on a simulated dataset under each type of data-generating process, with $\mathbb{E}[n] \approx 1000$.

| HPP Data, $\mathbb{E}[n] = 1000$ | | | | | |
|---|---|---|---|---|---|
| Model | $q = 0.005$ | 0.01 | 0.025 | 0.05 | 0.1 |
| HPP | 6 | 5 | 5 | 3 | 6 |
| LGCP (Matérn) | 4 | 5 | 5 | 7 | 3 |
| PGP (disc) | 0 | 0 | 0 | 0 | 1 |
| LGCP Data, $\mathbb{E}[n] \approx 1000$ | | | | | |
| Model | $q = 0.005$ | 0.01 | 0.025 | 0.05 | 0.1 |
| HPP | 0 | 0 | 0 | 0 | 0 |
| LGCP (Matérn) | 10 | 10 | 9 | 7 | 5 |
| PGP (disc) | 0 | 0 | 1 | 3 | 5 |
| Poisson-Gamma Data, $\mathbb{E}[n] \approx 1000$ | | | | | |
| Model | $q = 0.005$ | 0.01 | 0.025 | 0.05 | 0.1 |
| HPP | 0 | 0 | 0 | 0 | 0 |
| LGCP (Matérn) | 7 | 4 | 6 | 6 | 6 |
| PGP (disc) | 3 | 6 | 4 | 4 | 4 |

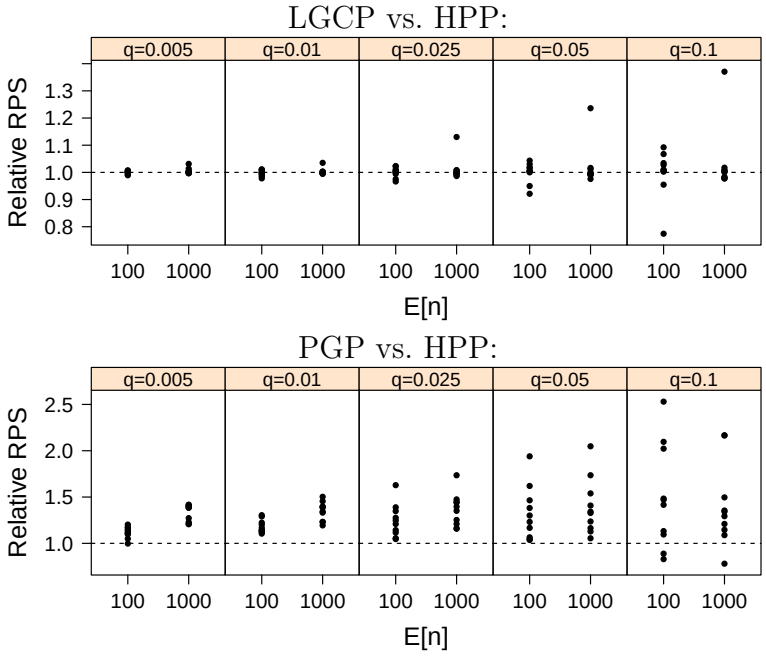



FIGURE 4.8: Relative RPS at each value of $q$ for simulated HPP data with $\mathbb{E}[n] = 100, 1000$. The LGCP (top row) and PGP (bottom row) models are compared to the HPP model, with the horizontal line at 1 indicating equivalent performance.

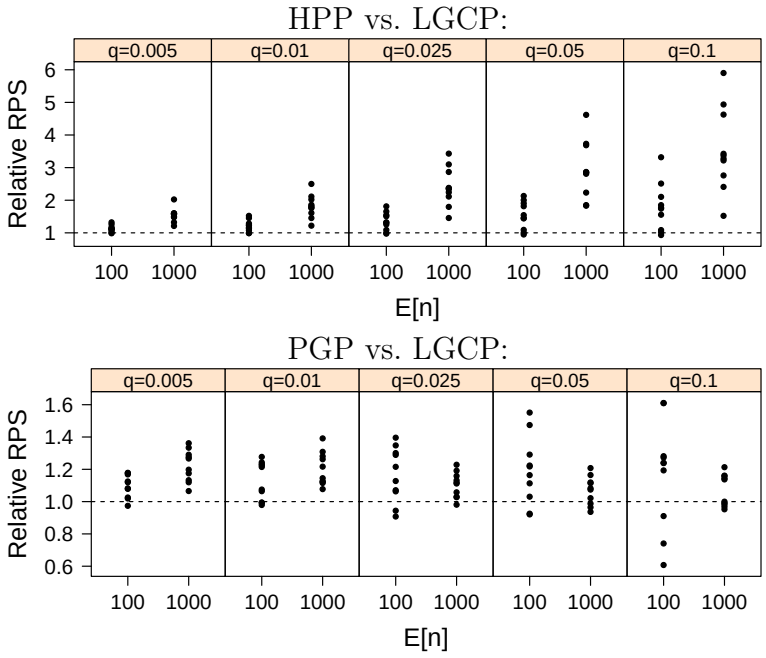



FIGURE 4.9: Relative RPS at each value of $q$ for simulated LGCP data with $\mathbb{E}[n] = 100, 1000$. The HPP (top row) and PGP (bottom row) models are compared to the LGCP model, with the horizontal line at 1 indicating equivalent performance.

The Poisson-gamma values are clustered around 1, but the HPP values are larger, especially for the high-data setting where the HPP gives RPS that is several times larger than the LGCP RPS.

Figure 4.10 shows the results for the simulated Poisson-gamma data. Despite the interesting results in Tables 4.3 and 4.4, it seems that the LGCP had similar RPS to the PGP model. For the low-data setting, Table 4.3 showed a strong preference for the Poisson-gamma model, yet the LGCP had RPS that were only 15–20% larger.

The predictive residuals were used as in the previous chapter to investigate the coverage of each model with each dataset. The results will not be given here, but the coverage rates were generally clustered around the nominal coverage rate (we used 90% again here). The only real problems were in the coverage of the HPP

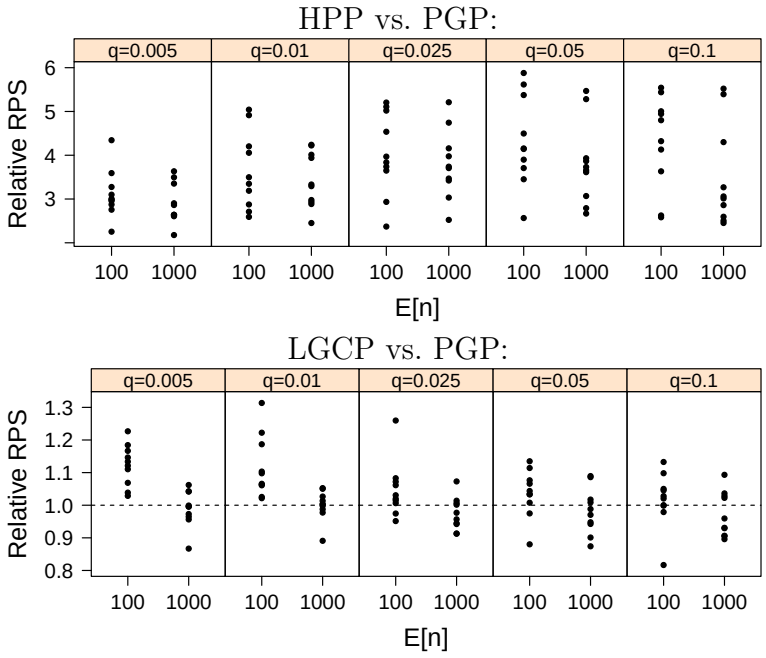



FIGURE 4.10: Relative RPS at each value of $q$ for simulated PGP data with $\mathbb{E}[n] = 100, 1000$. The HPP (top row) and LGCP (bottom row) models are compared to the PGP model, with the horizontal line at 1 indicating equivalent performance.

model when applied to the LGCP and PGP data. There, the HPP model generally exhibited less than 50% coverage on the test dataset.

With only ten simulations, these results are not conclusive, but they do suggest some general trends. The simulated Poisson-gamma process data results suggested that while the LGCP model is very flexible, it seemed to struggle when applied to the clustered data when $n$ was small. However, with enough data, the LGCP model was able to adapt to mirror the clustering and become a competitive model. This just again highlights that it can be difficult to distinguish between clustering and a spatially varying intensity, because a flexible model such as the LGCP will be able to perform reasonably well with enough data. Having an intuition of the true generating process will generally perform well and provide insight into the parameters governing the generative process, such as the size of clusters or the number of clusters.

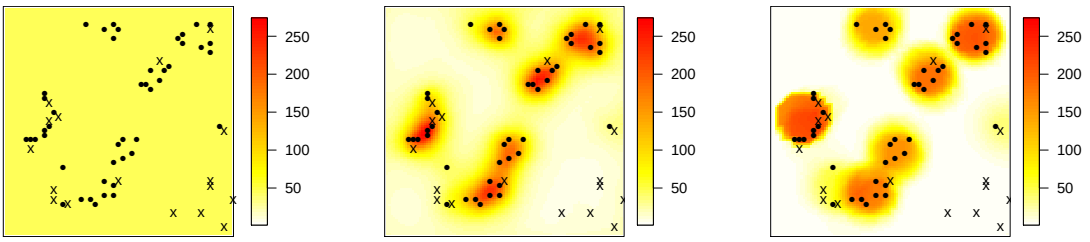
Finally, we also note that the Poisson-gamma process model is quite flexible and could also be made to compete well with the LGCP model in many circumstances. Our current specification using the finite disc as the mixing kernel is intentionally rigid so that the model would be best-suited for clustered data. This put the PGP model at a disadvantage when modeling the LGCP data, which has a smoothly varying intensity. Had a Gaussian kernel been used with the PGP model when being applied to the LGCP simulated data, the PGP model would be expected to provide RPS values comparable to the LGCP model on the simulated LGCP data.

### *4.2.4 Redwood Data*

The California redwood trees plotted in Figure 4.1 are a subset of a larger dataset collected by Strauss (1975). Ripley (1977) explored the clustered nature of this subset and many subsequent analyses have made this a standard dataset for clustered point pattern analysis. The 62 redwood trees are actually seedlings and saplings and the subset was mapped to the unit square.

To fit this data we considered comparing the HPP model, the PGP model with the finite disc kernel, and the LGCP with Matérn covariance function. For model validation and comparison purposes, cross-validation was used by employing $p$-thinning with $p = 0.5$ and $p = 0.75$. To average over the randomness inherent in the thinning process, ten sets of training and test data by applying $p$-thinning ten separate times. Each model was run for 50,000 MCMC iterations after a reasonable burn-in and 1,000 posterior predictive point patterns were created.

Figure 4.11 shows the posterior mean intensities for each model for the first set of training and test data for $p = 0.75$. We see that the training data contained only one point in the lower right corner, so the LGCP and PGP models gave those regions a very low intensity. This highlights the high variability in performing cross-validation, especially when the sample size is so small, and underscores the need




HPP LGCP PGP

FIGURE 4.11: The posterior mean intensity for the three models fit to the first cross-validation set from the redwood tree data for $p = 0.75$. The circles denoted points in the training data and the x's represent the test data.

to perform multiple rounds of cross-validation. The LGCP model and PGP model with a Gaussian kernel provide similar estimates of the intensity function, though the LGCP intensity stays slightly higher in the regions where no data was observed. The PGP model with a uniform disc kernel collapsed to essentially three large clusters. Experimentations with the data showed that if we take the data in the lower right corner, which do not appear to be clustered, and clump them into a single cluster in the bottom right corner, then the PGP model with uniform disc kernel employs more clusters and begins to look similar to other PGP model and the LGCP model.

Figure 4.12 the average RPS and coverage for each model on the held-out data for $p = 0.5$ and 0.75 (called 50% and 25% cross-validation, respectively). We see that the HPP performed worse on average than the other two models, with poor coverage for larger $q$ in the $p = 0.5$ case. For both thinning levels, the PGP model slightly outperformed the LGCP model, though not significantly.

Figure 4.13 shows the RPS and coverage results for four of the cross-validation replications of the redwood tree data at each thinning level. We see that the ranked probability scores are generally better for the PGP and LGCP models, though the HPP model performed equally well in some of the plots. One case in which the HPP performed equally well is the first set of cross-validation data for $p = 0.75$ which had

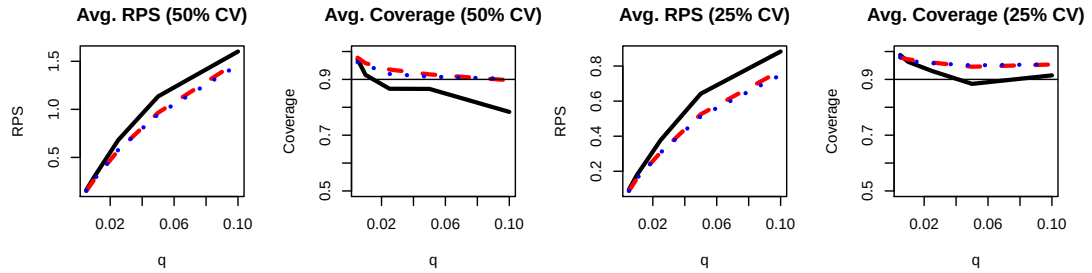



FIGURE 4.12: The average RPS and coverage for each model over 10 rounds of cross-validation. The solid line is the HPP model, the dashed line is the LGCP model, and the dotted line is the PGP model.

few points in the training data from the lower right corner. Here the PGP and LGCP models had a low intensity in the lower right corner (see Figure 4.11), so they would have been penalized for their poor performance in that region, despite performing well on capturing the other clusters.

These plots reiterate the importance of using several cross-validation sets in order to get an adequate representation of model fit and model performance, especially when the sample size is so small. For some cross-validation sets the coverage of all models dropped well below the nominal level, while for many other sets the coverage would be well above nominal. Similar variation in the RPS results is observed in Figure 4.13.

These results suggest that the PGP and LGCP models are outperforming the HPP model by the proposed metrics. The average RPS for the HPP model ranges from 10–25% higher than the PGP model, which we consider significant. The data does not seem to strongly differentiate between the PGP and the LGCP, but knowing that the data describes seedlings coming from common parent trees would suggest that a clustering model should be preferred here.

Given that the data visually exhibit a strong amount of clustering, it may seem disappointing that the HPP model is not doing relatively much worse. It would seem that the main culprit is the small sample size, which is even smaller after pulling out

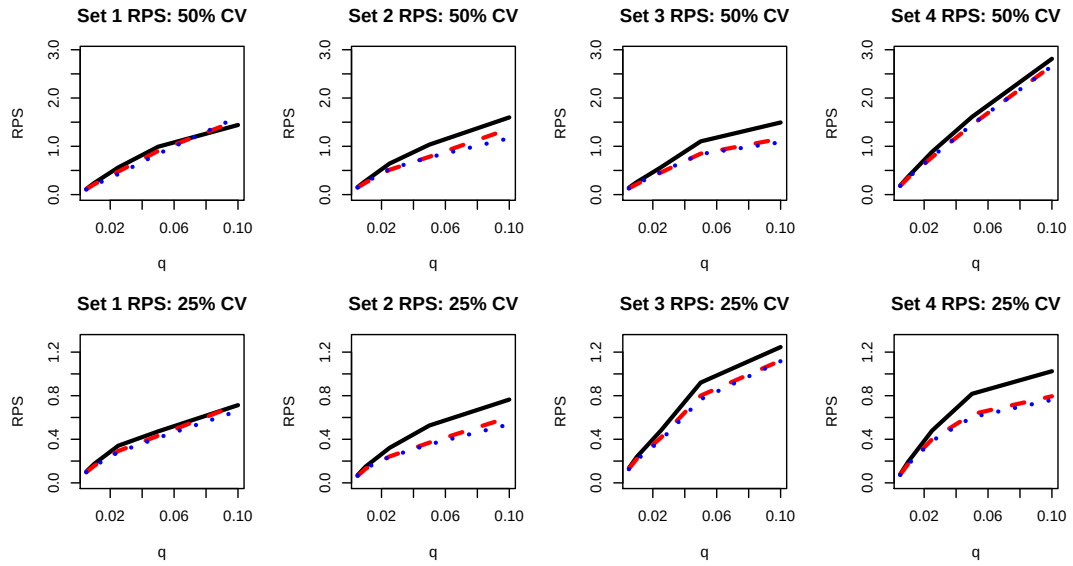



FIGURE 4.13: Ranked probability scores for test data in four cross-validation sets of the redwood data for $p = 0.5, 0.75$. The solid line is the HPP model, the dashed line is the LGCP model, the dotted line is the PGP model.

the test data. With such small samples sizes, it is hard to clearly distinguish between models. Another issue is that the points in the lower right corner are not clustered, making the model adjust for those points. A third issue is that the clusters appear to take on different shapes and different sizes, for which the finite disc kernel in the PGP is especially not well-suited.

To address the sample size question and the lack of clustering in the lower right corner, we considered adding additional points to the redwood dataset to bring the sample size to 152 seedlings. The additional points were added into the existing clusters by randomly selecting a cluster center and then locating each new point uniformly inside a disc centered at the chosen cluster center. Additionally, the points in the lower right corner of the original dataset, which were not tightly clustered, were modified to form a tighter cluster. This modified dataset is shown in Figure 4.14.

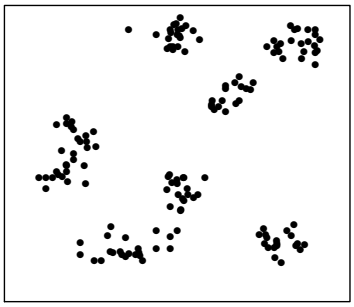


FIGURE 4.14: The modified redwood seedling dataset ($n = 152$), which was constructed by improving the clustering and randomly adding 90 data points to the clusters of the original dataset in Figure 4.14.

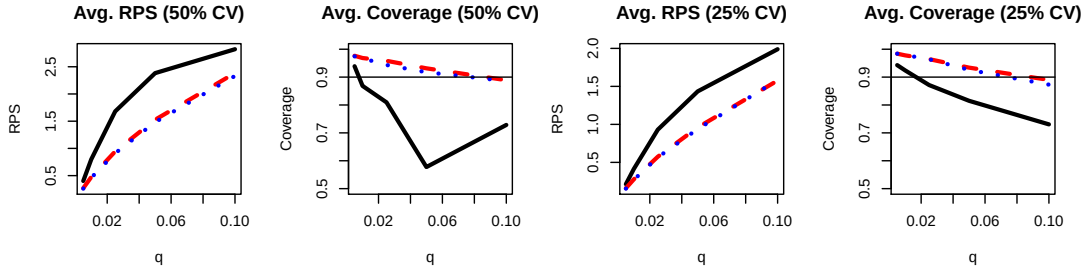



FIGURE 4.15: The average RPS and coverage for each model over 10 rounds of cross-validation for the modified redwood data. The solid line is the HPP model, the dashed line is the LGCP model, and the dotted line is the PGP model.

The same analysis was performed on the modified redwood dataset. Figure 4.15 shows the average RPS over the 10 simulations using cross-validation with $p = 0.5$ and 0.75. We see that the PGP and LGCP models perform very similarly and both are strongly preferred to the HPP model. The HPP model shows poor coverage, especially for $p = 0.5$.

## 4.3 Other Point Processes

There are yet many other varieties of point processes which have not been discussed thus far in this dissertation. Future research will investigate the application of the methods presented herein for these other, more complex processes. More complex

processes will allow richer inference once a Bayesian model has been fit, requiring also the development of further ideas to fully explore the posterior inference available in these complex models. We now give some brief remarks on some possible extensions of this work.

The Poisson-gamma process model was used as a flexible form for specifying shot noise Cox processes, yet other models are available. For example, Neyman-Scott models are a simpler option which specify a normal mixture model for the intensity function, as discussed previously. Fitting a Neyman-Scott model can be done by fixing the number of clusters and employing Bayesian model averaging to assign weights to each number of clusters, or by employing a reversible-jump MCMC algorithm (Green, 1995) that allows the number of clusters to vary (see, e.g., Guttorp and Thorarinsdottir, 2012). The other model-fitting details are straightforward. The PGP model can also be made more flexible by employing Lévy adaptive regression kernels (Wolpert et al., 2011), which allow the parameters in the kernel function to vary locally. Looking back at the redwood data of the previous section, it can be seen that the clusters appear to have varying shapes and sizes, which can be naturally incorporated using Lévy adaptive regression kernels.

The Gibbs processes used earlier in this chapter assumed that the process was stationary. Relaxing this assumption to allow a spatially varying intensity results in inhomogeneous Gibbs processes. Baddeley et al. (2000) discuss such processes and provide some ideas for estimating the spatial dependence structure for inhomogeneous processes. They assess model fit by comparing the empirical $L$-function on the observed data to empirical $L$-functions calculated on simulated point patterns from the fitted model. This can be easily re-created using our posterior predictive point patterns, but we could also naturally apply the techniques discussed previously and compute residuals, innovations, ranked probability scores, etc. With an appropriate

summary (or discrepancy) function, posterior predictive checks could again provide a sense of model fit.

Apart from a spatially varying intensity, it might also be possible that the dependence structure of the Gibbs process varies locally, such as a Strauss process with a soft core radius that varies over the domain. The model used previously could specify a regression model or even a Gaussian process prior on the radius $R$ over the domain. Comparing a model with a spatially varying $R$ to a stationary model with a fixed $R$ may be feasible by again using the posterior predictive point patterns. For example, one might specify a grid over the domain and then calculate the minimum or average nearest neighbor distance in each grid box, comparing the observed distances to their posterior distribution using the posterior predictive point patterns. Poor coverage would indicate lack of model fit. Of course, the Strauss process model was not always well-behaved for fixed $R$, so fitting a model with spatially varying $R$ may be prohibitive.

Marked point process constitute another rich class of processes that deserve more attention in regards to model diagnostics and model selection. A marked point process consists of pairs $(s_i, m_i)$ where $m_i$ is a mark attached to the event at $s_i$. The marks can be discrete or continuous, allowing a variety of options. A spatiotemporal point pattern would use $m_i \equiv t_i$, where $t_i$ denotes the time of the events. The marks might denote the diameter at breast height of the tree, the species of the tree, the type of crime committed at location $s_i$, etc. Taddy (2010) and Taddy and Kottas (2012) present extensions of the spatial Dirichlet process mixture model for marked point patterns, modeling violent crimes over time in Cincinnati and tree diameters across a forest in Georgia.

For a marked point pattern with continuous marks, an LGCP model naturally allows the inclusion of the marks by employing a Gaussian process over both locations and continuous marks. Denote the marked point pattern as $Y$ and denote each event

$(s_i, m_i)$ with $y_i \equiv (s_i, m_i)$ with $m_i \in \mathcal{M}$. The intensity function associated with $Y$ can be written as $\lambda(s, m)$. A common form for $\lambda(s, m)$ would employ a separable Gaussian process prior for $\log \lambda(s, m)$, with the option to also include a regression component. A separable covariance function means that the correlation over space is independent from the correlation over the mark space $\mathcal{M}$. Thus, the covariance function $c(y, y')$ can be factored as $c(y, y') = \sigma^2 c_{0,S}(s, s')c_{0,M}(m, m')$ with $c_{0,S}$ denoting the spatial correlation function and $c_{0,M}$ denoting the correlation function over the mark space. Though separable covariance functions may not always be appropriate, they do provide the necessary distinction between distance in $D$ and distance in $\mathcal{M}$, which cannot be assumed to be comparable.

The LGCP model for continuous marks is straightforward to fit and to simulate data from. Fitting the model occurs just as before, requiring integrals of $\lambda(s, m)$ over $D \times \mathcal{M}$. Simulating point patterns is again performed using the Lewis-Shedler approach of finding $\lambda_{max} = \max_{s,m} \lambda(s, m)$, drawing from an HPP over $D \times \mathcal{M}$ with intensity $\lambda_{max}$, and then thinning each point according to the ratio $\lambda(s, m)/\lambda_{max}$.

For such processes, model diagnostics and model choice can proceed as before by thinning the data and creating a test dataset. Residuals and ranked probability scores can be used to assess model fit and performance relative to other models. Posterior distributions for the intensities or predicted counts over subsets of $D \times \mathcal{M}$ can be calculated using discrete approximations as previously discussed.

If the marks are event times, implying that $\mathcal{M}$ is just a time interval $(a, b]$, then the posterior distribution for the intensity over time could be calculated for points or regions of interest inside $D$. This could be summarized by plotting a credible band for the intensity over time, allowing simple identification of temporal trends in the intensity function. Another possibility with a spatiotemporal point pattern is the prediction of future events. This suggests that model selection could also be performed by using only the data up to some time $t$ and than predicting the outcomes

in the time window $(t, t+h]$. Ranked probability scores could then be calculated on the predicted data, with the possibility of performing multiple sets of predictions over sequential windows.

Another possibility here would be to compare separable and non-separable space-time LGCP models, detecting whether there is an interaction between the intensities for the marks and space. Simulation studies would be necessary here to determine whether making the distinction between separable and non-separable models is even a possibility. In either case, one could also imagine integrating over the mark intensity to provide a marginal intensity for space. Similarly, one could then integrate over space to provide a marginal intensity for the marks.

Marked point patterns with discrete marks require a slightly different approach. With continuous marks, a joint model was directly provided though incorporation into the covariance function of the LGCP model. As discussed in Gelfand et al. (2010) and Banerjee et al. (2014), marked point process models for discrete marks require consideration of the order of conditioning used to create a joint model over locations and marks. One way to view such processes is to consider the point pattern $S$ as the superposition of $K$ point patterns, written as $S = \cup S_k$, where $S_k \equiv \{s_i : m_i = \omega_k\}$. Each $S_k$ would have an associated intensity function $\lambda_k(s)$, providing the cumulative intensity function as $\lambda(s) = \sum_{k=1}^{K} \lambda_k(s)$. Each $\lambda_k(s)$ can take on the form of any of the models previously discussed, such as an LGCP. With a prior over marks, the model is now fully specified with a prior over locations given marks and a prior over marks. Simulating from this model is done by drawing a label $\omega_l$ and then drawing a location given the mark using $\lambda_k(s)$.

One interesting way to specify this model would be through coregionalization. Rather than specifying independent Gaussian processes for $\log \lambda_k(s)$, one can learn about the correlation between each $\lambda_k(s)$ by using a linear model of coregionalization to provide a joint prior over all $\lambda_k(s)$. This model could be specified as the following,

assuming for simplicity that $K = 2$:

$$
\begin{aligned}
& Pr(m_i = \omega_k) = \alpha_k, \ \ l = 1, 2 \\
& \lambda_k(s) = \lambda_{0,k} \ \exp\{x^T(s)\beta_k + Z_k(s)\} \\
& \begin{bmatrix} Z_1(s) \\ Z_2(s) \end{bmatrix} = A \begin{bmatrix} U_1(s) \\ U_2(s) \end{bmatrix} \\
& U_k(s) \overset{iid}{\sim} \mathrm{GP}(-\sigma_k^2/2, c_k) \\
& A = \begin{bmatrix} 1 & 0 \\ \rho & 1 \end{bmatrix} \\
& \rho \sim \pi(\rho) \text{ on } (-1, 1).
\end{aligned} \tag{4.17}
$$

This model puts a multinomial probability over the marks and provides each intensity function with its own set of regression coefficients $\beta_k$ and baseline intensities $\lambda_{0,k}$. The Gaussian process piece ($\{Z_k(s)\}$) comes through mixing two independent Gaussian processes ($\{U_k(s)\}$) by a matrix $A$. This specification set $Z_1(s) = U_1(s)$ and $Z_2(s) = \rho U_1(s) + U_2(s)$. If $\rho \to 1$ and $\sigma_2^2 \to 0$, then $Z_2(s) \approx U_1(s) = Z_1(s)$, implying that $Z_1(s)$ and $Z_2(s)$ are essentially the same. Thus, this parameterization provides meaningful interpretations of its parameters and allows the model to inform on the difference between the two intensities.

One could again perform cross-validation to assess model fit and compare competing models. For each model, one can also compare posterior distributions of counts or intensities over a region for each mark/label. Plots of quantiles of the posterior distribution of $\lambda_k(s) - \lambda_{k'}(s)$ could also be created, providing useful comparison of where each intensity function differs.

The other option for specifying models for marked point patterns with discrete marks is to first model the locations of events and then model the mark given the location. This specification uses an overall intensity function $\lambda(s)$ and requires a model to provide $Pr[m(s) = \omega_k]$ to provide the probabilities that the mark $m(s)$ at

location $s$ is equal to $\omega_k$, for any $s \in D$ and $k$. Logit or probit models can be used to provide $Pr[m(s) = \omega_k]$.

These two approaches to conditioning provide inherently different processes, as discussed in Gelfand et al. (2010) and Banerjee et al. (2014). However, both cases provide interesting comparisons between the occurrence of events for different marks. Cross-validation can be applied in either case, reducing $\lambda_k(s)$ to $p\lambda_k(s)$ in the first case and $\lambda(s)$ to $p\lambda(s)$ in the second case. Residuals and innnovations could be computed on a mark-specific level as well as the aggregate level, allowing assessment for each mark as to whether the model fits. Similarly, models could be compared using RPS on a mark-specific level or on the aggregate level.

Extensions to these marked point patterns include spatiotemporal point patterns with discrete time and Gibbs or cluster processes with discrete marks. Taddy (2010) proposes a flexible dynamic model providing an autoregressive model for Poisson processes. Here time is taken to be a discrete mark and the intensity at time $t$ uses a log-Gaussian dynamic linear model framework. Högmander and Särkkä (1999) analyzed the ant nest data from Harkness and Isham (1983), shown in Figure 4.1. They propose two bivariate Gibbs process models to model the dependence structure between nests from ants of the same species and between nests of the other species. Their hierarchical model also addresses the asymmetry in the relationship between the species, allowing the locations of the nests of one species to affect the other species' nests but not vice versa.

These extensions of basic point patterns are just a few examples of the rich class of models available for point processes. More attention is needed to continue developing and assessing methods for evaluating model fit and selecting models in these more complex cases.

# 5

# Discussion

This dissertation focused on extending Bayesian inference for spatial point process models. Much work along these lines has been developed from a frequentist view, leaving a need for further discussion about how such methods apply to Bayesian models. This work has demonstrated how the posterior distribution from a Bayesian model provides many avenues for inference, diagnostics, and even model selection.

Chapter 2 discussed simple processes, such as the homogeneous Poisson process, the nonhomogeneous Poisson process, and the log-Gaussian Cox process. Besides fitting this models, the discussion detailed how the posterior distribution for the model parameters provided useful summaries of the intensity function on the point level, region level, or the whole domain level. The posterior also allowed drawing posterior predictive point patterns, which were shown to be extremely useful in summarizing our posterior belief in summaries of the point process. Posterior inference for the $F$-, $G$-, and $K$-functions was also explored using these posterior predictive point patterns.

Chapter 3 delved more deeply into the ideas of model diagnostics and model selection. It was shown that, for many point processes, cross-validation using $p$-

thinning allow the creation of a valid, independent set of test data. This test data, a thinned version of the original point pattern, can then be used for calculating residuals and comparing model predictions. The predictive residual was presented as a more meaningful metric in comparing model coverage, where low coverage would signify poor model fit. Looking at these residuals across the domain also provides insight into specific areas where the model under- or overpredicts event counts. The ranked probability score was shown to be a useful metric in comparing predictions on test data between competing models. The simulation study in this chapter showed that RPS could clearly detect when a model was not flexible enough, though sometimes a large amount of data was required. Our method was not able to distinguish between covariance functions in the LGCP even with large amounts of data, which we attribute more to the limited information that a point pattern can give about the intensity function than as a failing of our method.

Chapter 4 discussed applying these ideas to more complex point processes. For repulsive processes, many of the previous ideas apply, though cross-validation is not feasible due to the dependence in the point locations. In addition to residual diagnostics, looking at second-order characteristics of the posterior predictive point patterns, such as the variance of grid cell counts, were found to be more useful in detecting lack of model fit. For clustered point patterns, cross-validation is again feasible and all of the methods of Chapter 3 apply. However, the well-known ambiguity between a nonhomogeneous intensity and clustering makes it difficult to distinguish between models. The chapter ends with ideas for extending these ideas to more complex point processes, such as space-time or marked point processes.

Besides extending these ideas to more complex processes, one promising avenue for future work involves exploring the use of approximate Bayesian computation (ABC) for point patterns. ABC is a very general approach which can at times prove very useful in fitting Bayesian models when traditional MCMC methods are

ill-behaved or even impossible. For example, when fitting the Strauss process model to the simulated Strauss process data in Chapter 4, the acceptance rate of the model was extremely low. With an intractable likelihood, few other options for model fitting exist. Since Strauss processes are easy to simulate, ABC appears to be a good candidate for obtaining posterior distributions of model parameters, though some care will be needed to choose appropriate summary statistics.

The methods for model analysis and selection presented in this work do not attempt to replace, nor can they, the need for expert involvement in model building and analysis. The simulation studies presented throughout the dissertation demonstrated many cases in which it is difficult, if not impossible, to distinguish between two processes, especially with a small sample size. Combined with the guidance of expert opinion, however, these methods provide a general sense of model fit and a reasonable approach to choosing between two equally plausible models.

# Appendix A

## Formulas for $F$-, $G$-, and $K$-functions

### A.1 Standard Empirical Estimates

For homogeneous point patterns, the standard edge-corrected estimates for the homogeneous $F$-, $G$-, and $K$-functions are given below. Small variations exist, mainly in the form of using different edge corrections. $\hat{F}(d)$ and $\hat{G}(d)$ below are the simple "reduced sample" estimates, which only consider points that are further than distance $d$ from the boundary of $D$. Other proposed variations of these functions aim to be more efficient in using more of the data. These formulas, or similar variants thereof, can be found in, e.g., Banerjee et al. (2014); Illian et al. (2008); Gelfand et al. (2010); Cressie (1993); Diggle (1983).

$\hat{F}(d)$ uses a fine grid of points, each denoted by $t_j$, to approximate the probability that a random location in $D$ has a point in $S$ within distance $d$. Below, we use the notation $d_j \equiv \min_{s_i} ||t_j - s_i||$, $d_i \equiv \min_{s_{i'}} ||s_i - s_{i'}||$, $b_j$ and $b_i$ are the nearest-boundary distances for $t_j$ and $s_i$, respectively, $\hat{\lambda} \equiv n/|D|$, and $w_{s_i,s_{i'}}$ is the proportion of the circle centered at $s_i$ with radius $||s_i - s_{i'}||$ that is contained in $D$.

$$\hat{F}(d) = \frac{\sum_j \mathbf{1}(d_j \leq d < b_j)}{\sum_j \mathbf{1}(b_j > d)} \tag{A.1}$$

$$\hat{G}(d) = \frac{\sum_i \mathbf{1}(d_i \leq d < b_i)}{\sum_i \mathbf{1}(b_i > d)} \tag{A.2}$$

$$\hat{K}(d) = \frac{1}{n\hat{\lambda}} \sum_{i=1}^{n} \sum_{i' \neq i} \mathbf{1}(||s_i - s_{i'}|| \leq d)/w_{s_i, s_{i'}} \tag{A.3}$$

For inhomogeneous point patterns, extensions of the above functions exist for the inhomogeneous case and are given below. $\hat{F}_{\text{inhom}}(d)$ and $\hat{G}_{\text{inhom}}(d)$ are taken from equations 6 and 7 in van Lieshout (2011) and $\hat{K}_{\text{inhom}}(d)$ is taken from Baddeley et al. (2000). Below, $\overline{\lambda} \equiv \inf_{s \in D} \lambda(s)$.

$$\hat{F}_{\text{inhom}}(d) = 1 - \frac{\sum_j \left( \mathbf{1}(b_j > d) \left[ \prod_i \mathbf{1}\big(||s_i - t_j|| \leq d\big) \big(1 - \overline{\lambda}/\lambda(s_i)\big) \right] \right)}{\sum_j \mathbf{1}(b_j > d)} \tag{A.4}$$

$$\hat{G}_{\text{inhom}}(d) = 1 - \frac{\sum_i \left( \mathbf{1}(b_i > d) \left[ \prod_{i' \neq i} \mathbf{1}\big(||s_{i'} - s_i|| \leq d\big) \big(1 - \overline{\lambda}/\lambda(s_{i'})\big) \right] \right)}{\sum_i \mathbf{1}(b_i > d)} \tag{A.5}$$

$$\hat{K}_{\text{inhom}}(d) = \frac{1}{|D|} \sum_{i=1}^{n} \sum_{i' \neq i} \frac{\mathbf{1}(||s_i - s_{i'}|| \leq d)}{w_{ii'}\, \hat{\lambda}(s_i)\, \hat{\lambda}(s_{i'})} \tag{A.6}$$

## A.2 Proposed Bayesian $F$-, $G$-, and $K$-functions

A full discussion of our proposed Bayesian analogues of the $F$-, $G$-, and $K$-functions is given in Chapter 2, but their forms are given below. These are *model-based* quantities, expressing features of the joint posterior distribution, rather than the more exploratory, nonparametric estimates above. Our proposed functions require posterior predictive point patterns, $S_l^*$ and, for the $K$-function, posterior draws of the parameters.

Only point estimates are available for $\tilde{F}$ and $\tilde{G}$, but a full posterior distribution is available for $\tilde{K}(d)$ by removing the outer sum over $l$ and the division by $L$ to get a draw of $\tilde{K}(d)$ for each $S_l^*$. Below, $\tilde{w}_{s_{li}^*} = |c_d(s_{li}^*) \cap D|/|c_d(s_{li}^*)|$ is estimated through a Monte Carlo integration by drawing points uniformly inside $c_d(s_{li}^*)$ and calculating the proportion which are also inside $D$. For a first-order stationary process, $\tilde{w}_{s_{li}^*} = |c_d(s_{li}^*) \cap D|/|c_d(s_{li}^*)| = Pr[D \,|\, c_d(s_{li}^*)]$.

$$\tilde{F}(d) = \frac{\frac{1}{L}\sum_{l=1}^{L}\left[\frac{\sum_{t_j}\mathbf{1}(N(t_j,d,S_l^*)>0,\, c_d(t_j)\subset D)}{N^{(l)}(D)}\right]}{\frac{1}{L}\sum_{l=1}^{L}\left[\frac{\sum_{t_j}\mathbf{1}(c_d(t_j)\subset D)}{N^{(l)}(D)}\right]} \tag{A.7}$$

$$\tilde{G}(d) = \frac{\frac{1}{L}\sum_{l=1}^{L}\left[\frac{\sum_{s_{li}^*\in S_l^*}\mathbf{1}(N(s_{li}^*,d,S_l^*)>0,\, c_d(s_{li}^*)\subset D)}{N^{(l)}(D)}\right]}{\frac{1}{L}\sum_{l=1}^{L}\left[\frac{\sum_{s_{li}^*\in S_l^*}\mathbf{1}(c_d(s_{li}^*)\subset D)}{N^{(l)}(D)}\right]} \tag{A.8}$$

$$\tilde{K}(d) = \frac{1}{L}\sum_{l=1}^{L}\sum_{s_{li}^*\in S_l^*}\frac{\sum_{j\neq i}\mathbf{1}(s_{ji}^*\in c_d(s_{li}^*)\cap D)}{\tilde{w}_{s_{li}^*}\,\lambda^{(l)}\,N^{(l)}(D)} \tag{A.9}$$

For inhomogeneous point patterns, we discussed that the $F$- and $G$-functions don't have a clear definition or interpretation. Our proposed inhomogeneous $K$-function takes the form

$$\tilde{K}_{\text{inhom}}(d) = \frac{1}{L\,|D|}\sum_{l=1}^{L}\sum_{s_{li}^*\in S_l^*}\frac{1}{\tilde{w}_{s_{li}^*}\lambda^{(l)}(s_{li}^*)}\left[\sum_{j\neq i}\frac{\mathbf{1}(s_{lj}^*\in c_d(s_{li}^*))}{\lambda^{(l)}(s_{lj}^*)}\right]. \tag{A.10}$$

# Appendix B

## Simulation Study Plots from Chapter 3

This appendix gives the full results of the simulation study from Chapter 3. As explained in that chapter, data was generated from five point process models with a low intensity setting ($\mathbb{E}[n] \approx 100$) and a high intensity setting ($\mathbb{E}[n] \approx 1000$). For each simulated dataset, we fit the same five models and compared the relative ranked probability scores using 200 random boxes (each with area $q|D|$ for various levels of $q$) and the coverage of 90% prediction intervals over the same boxes. The relative RPS is calculated as the RPS for the particular model divided by the RPS for the data-generating model. Thus, in each scenario, one of the models will show a relative RPS of 1 for every replicate. The relative RPS and coverage results are calculated on the test dataset, which was an independent replicate from the same intensity surface with an equivalent $\mathbb{E}[n]$. We generated 20 training and test replications for each data-generating model and intensity level to provide a sense of the variation involved.

In general, we see that the large sample size allows better separation between adequate and inadequate models. We also see that using a model which is more

flexible than the underlying process does not seem to incur a large penalty, though using a model without enough flexibility will be penalized heavily. Finally, there appears to be an inability to identify the correct covariance function in an LGCP model from the data alone.

The five models used in the simulation study are assigned a label to facilitate the limited space on the plots. The labels and corresponding models are given in the table below.

| Label | Model |
|---|---|
| A | Homogeneous Poisson Process (HPP) |
| B | Nonhomogeneous Poisson Process (NHPP) |
| C | Log-Gaussian Cox Process (LGCP) with Exponential covariance function |
| D | LGCP with Matérn($\nu = 3/2$) covariance function |
| E | LGCP with Gaussian covariance function |

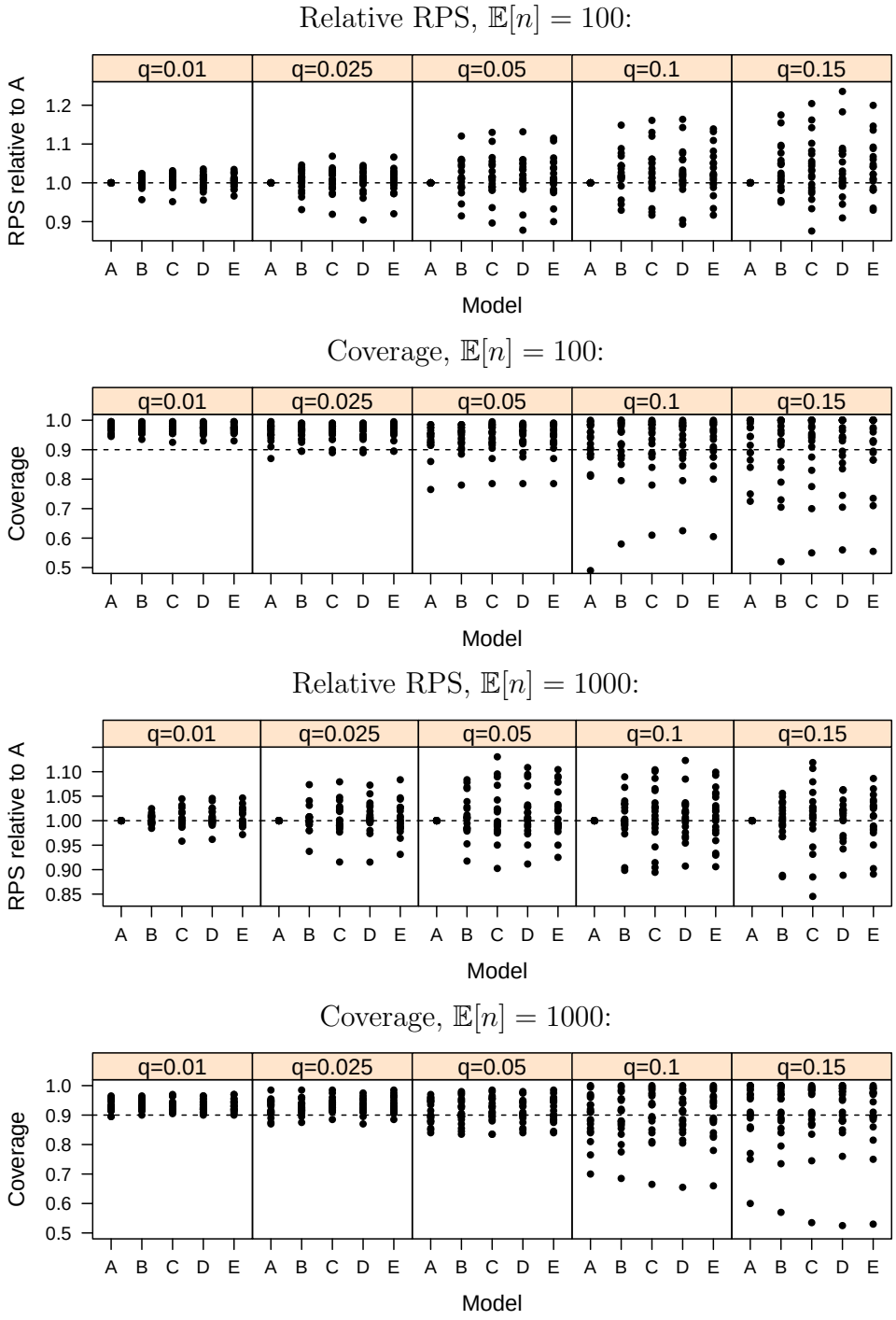



FIGURE B.1: The RPS and coverage results for the simulated HPP data. All the models perform fairly similarly to the HPP model. The coverage relative RPS plots show more variability as $q$ gets larger. The coverage levels are all close to the nominal 90% level, though with more variability in the $\mathbb{E}[n] = 100$ case.

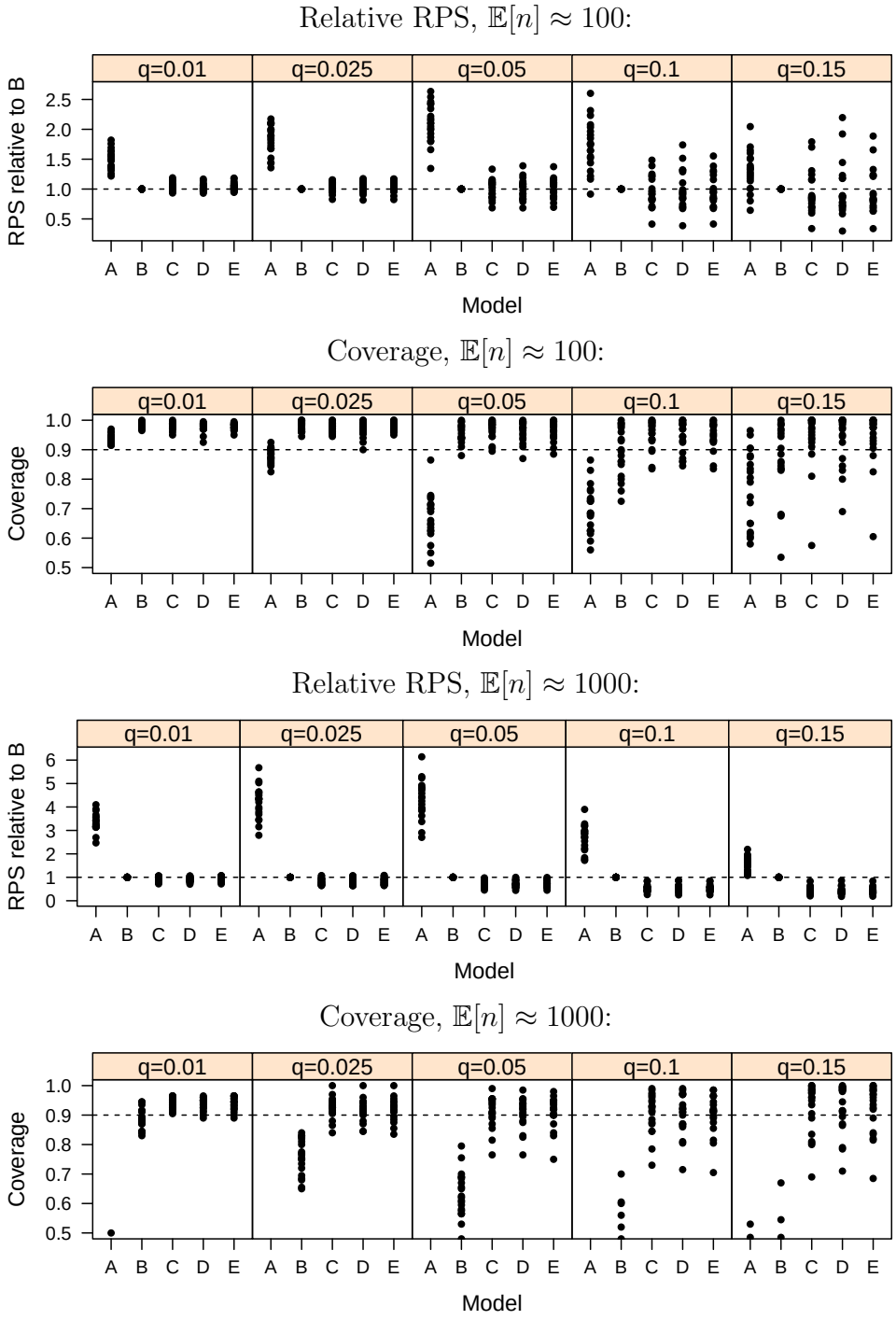



FIGURE B.2: The RPS and coverage results for the simulated NHPP data. The HPP performs poorly, but the other models perform similarly. For $\mathbb{E}[n] \approx 1000$, the LGCP models outperform the true NHPP model both in RPS and coverage, and the NHPP coverage is largely inadequate despite being the true underlying model.

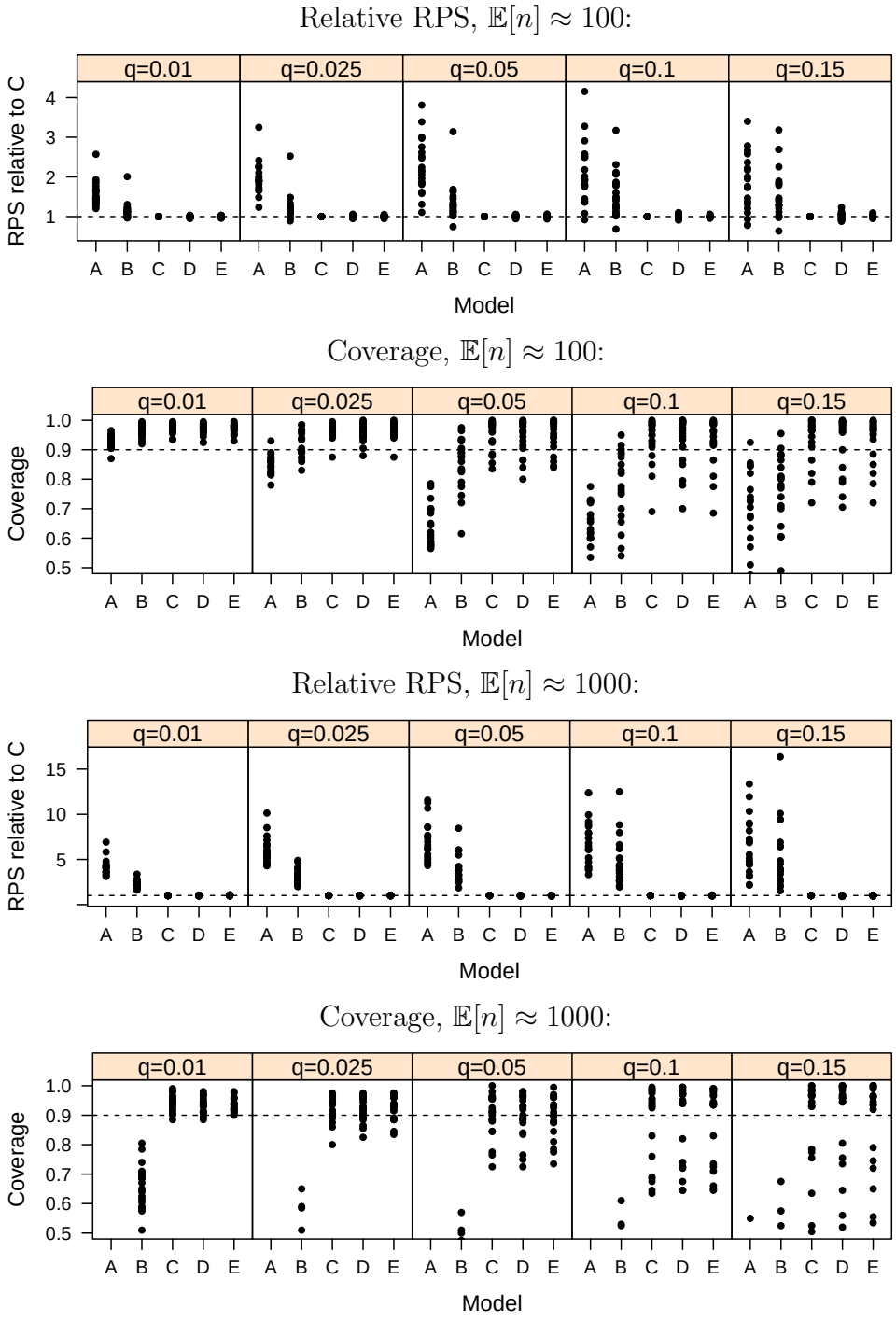



FIGURE B.3: The RPS and coverage results for the simulated LGCP (Exponential covariance) data. The HPP and NHPP models performed worse than the LGCP models, especially for $\mathbb{E}[n] \approx 1000$. The LGCP models all performed very similarly, with the coverage levels sometimes dropping close to 50%.

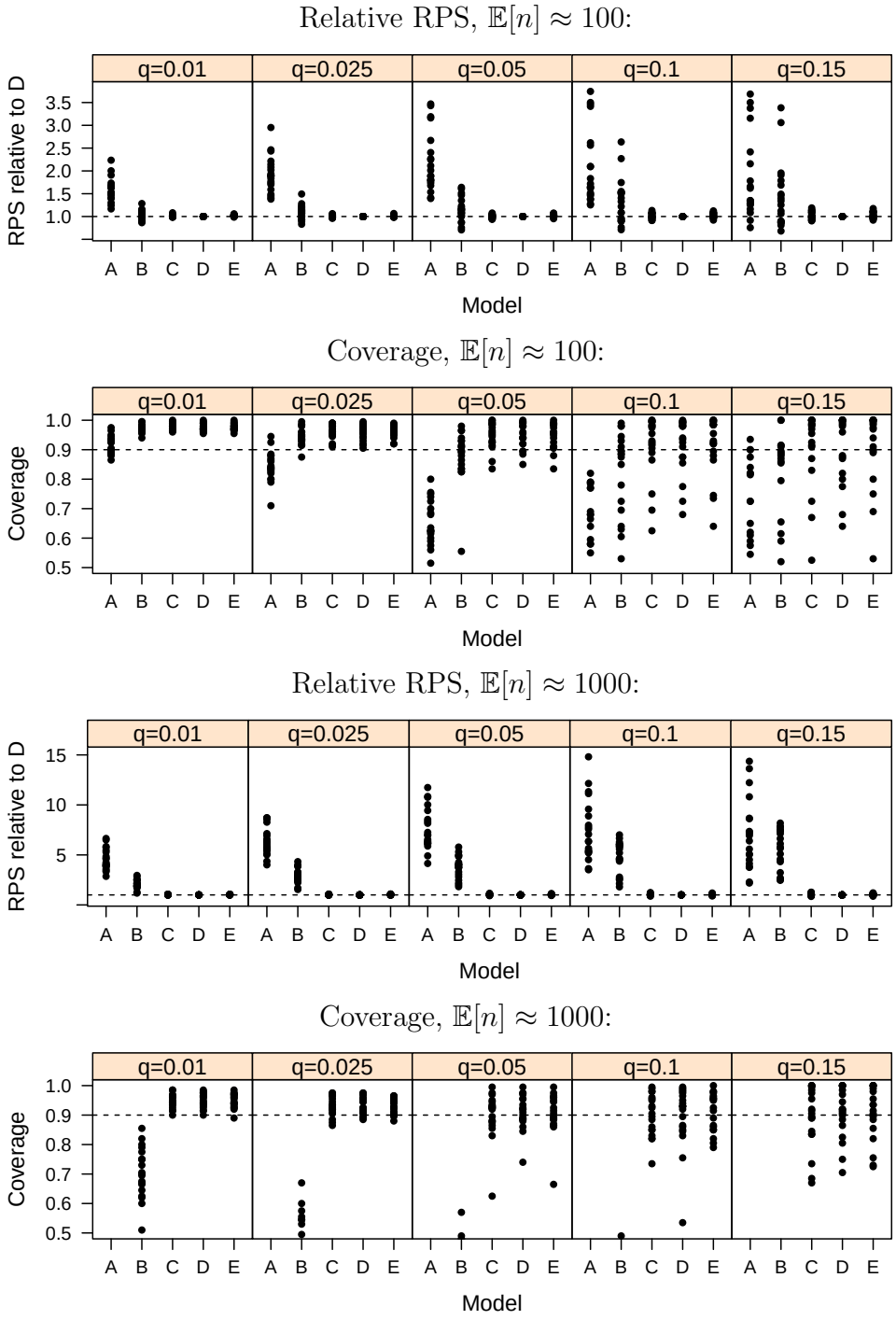



FIGURE B.4: The RPS and coverage results for the simulated LGCP (Matérn $\nu = 3/2$ covariance) data. The results are similar to Figure B.3.

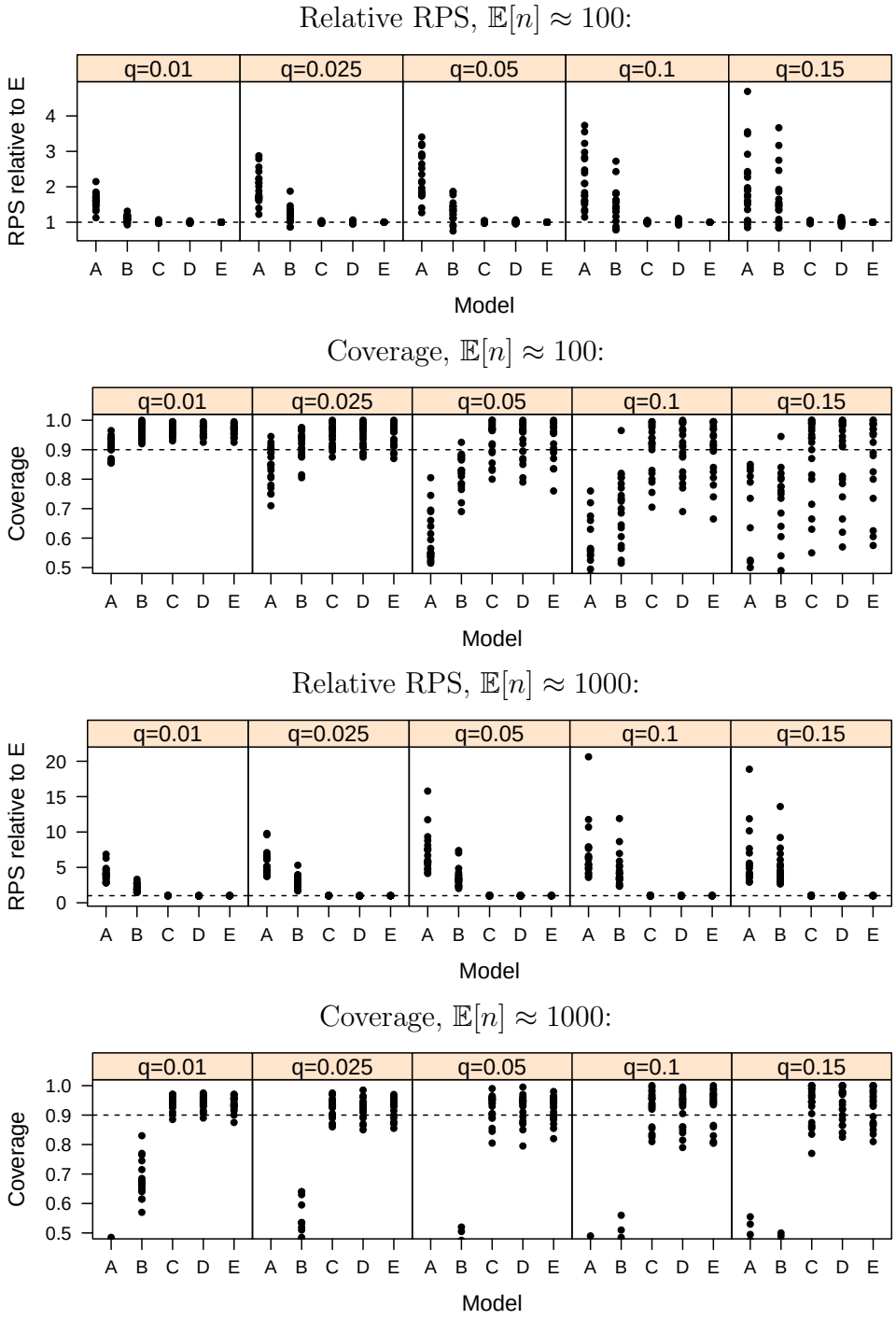



FIGURE B.5: The RPS and coverage results for the simulated LGCP (Gaussian covariance) data. The results are similar to Figure B.3.

# Appendix C

## MCMC Algorithm for the Poisson-Gamma Model

This algorithm, taken almost verbatim from Wolpert and Ickstadt (1998) and Ickstadt et al. (1998), describes how to fit the Poisson-gamma process model used in Chapter 4. The notation here is slightly adapted to fit the notation used in this dissertation. $E_1(t)$ is the exponential integral function $E_1(t) \equiv \int_t^\infty e^{-u}u^{-1}du$. $N \equiv N(D)$ will be used to denote the number of observations in $S$ and will have a corresponding index $n = 1, \ldots, N$. $M$ is a truncation on the Gamma random field so that only a finite number of realizations $\sigma_m$ from the Gamma random field are used, with $m = 1, \ldots, M$ and $M \gg N$. $U^t = \{u_n^t\}_{n \leq N}$ denotes the augmentation points, introduced by the Inverse Lévy Measure algorithm, at iteration $t$. $D_{\text{ext}}$ is an auxiliary space, taken here to be a superset of $D$. $\Pi(ds)$ is a distribution generating the samples $\sigma_m \in D_{\text{ext}}$ and is used in the shape measure $\alpha(s) \equiv \alpha(ds)/\Pi(ds)$. The parameters $\theta$ are given the prior distribution $\pi(d\theta)$. $Q(\theta, \theta^*)$ denotes the Markov transition kernel for proposing new values of $\theta$. The term $k^\theta(D, u)$ is defined as $k^\theta(D, u) \equiv \int_D k^\theta(s, u)ds$.

Given initial values for $\theta^0$ and $U^0$, the following Gibbs/Metropolis algorithm generates samples from the conditional distributions of each parameter, starting at iteration $t = 1$.

1. Gibbs step to update the gamma random field: given $\theta^{t-1}$ and $U^{t-1} = \{u_n^{t-1}\}_{n \le N}$,

   (a) Set $\sigma_m^t \leftarrow u_m^{t-1},\ 1 \le m \le N$, and generate independent $\sigma_m^t \sim \Pi(ds),\ N < m \le M$;

   (b) Set $\alpha_m^t \leftarrow \alpha^{\theta^{t-1}}(\sigma_m^t),\ \beta_m^t \leftarrow \beta^{\theta^{t-1}}(\sigma_m^t) + k^{\theta^{t-1}}(D, \sigma_m^t)$, and $i_m^t \leftarrow 0$ if $u_n^{t-1} = \sigma_m^t$ for some $n < m$, otherwise $i_m^t \leftarrow 1$;

   (c) Generate successive jumps $\{\tau_m\}_{m \le M}$ of a standard Poisson process[1];

   (d) Set $v_m^t \leftarrow (\tau_m - \tau_{m-1})/\beta_m^t$, for $1 \le m \le N$, and $v_m^t \leftarrow E_1^{-1}\big((\tau_m - \tau_N)/\alpha_m^t\big)/\beta_m^t$, for $N < m \le M$;

   (e) Set $\Gamma^t(du) \leftarrow \sum_{m \le M} v_m^t \delta_{\sigma_m^t}(du)$.

2. Gibbs step to update the augmentation points: given $\theta^{t-1}$ and $\Gamma^t$,

   (a) Generate independent $U^t = \{u_n^t\}_{n \le N}$ with

$$Pr[u_n^t = \sigma_m^t] \propto v_m^t k^{\theta^{t-1}}(s_n, \sigma_m^t).$$

3. Metropolis/Hastings step to update the parameter $\theta$: given $\theta^{t-1}$ and $U^t = \{u_n^t\}_{n \le N}$,

   (a) Set $\theta^- \leftarrow \theta^{t-1}$ and generate a new candidate $\theta^+ \sim Q(\theta^-, \theta^+)$;

   (b) Set $k_n^- \leftarrow k^{\theta^-}(s_n, u_n^t)$ and $k_n^+ \leftarrow k^{\theta^+}(s_n, u_n^t)$;

   (c) Set $\alpha_n^- \leftarrow \alpha^{\theta^-}(\sigma_n^t)$ and $\alpha_n^+ \leftarrow \alpha^{\theta^+}(\sigma_n^t)$;

   (d) Set $\beta_n^- \leftarrow \beta^{\theta^-}(\sigma_n^t) + k^{\theta^-}(D, \sigma_n^t)$ and $\beta_n^+ \leftarrow \beta^{\theta^+}(\sigma_n^t) + k^{\theta^+}(D, \sigma_n^t)$;

[1] This is done by letting $\tau_m = \sum_{j=1}^m \omega_j$ and $\omega_j \overset{\text{iid}}{\sim}$ Exponential(1).

(e) Calculate the Metropolis/Hastings acceptance probability

$$
\begin{aligned}
P^t =& \frac{\pi(\theta^+)}{\pi(\theta^-)} \times \frac{Q(\theta^+, \theta^-)}{Q(\theta^-, \theta^+)} \times \left[ \prod_{n \le N} \frac{k_n^+}{k_n^-} \right] \\
& \times \exp \left\{ \sum_{m \le M} \left[ i_m^t \log \left( \frac{\alpha_m^+}{\alpha_m^-} \right) - v_m^t (\beta_m^+ - \beta_m^-) \right] \right\} \\
& \times \exp \left\{ - \int_{D_{\text{ext}}} \log \left( 1 + \frac{k^{\theta^+}(D,u)}{\beta^{\theta^+}(u)} \right) \alpha^{\theta^+}(du) - E_1(v_M \beta_M^+) \alpha_M^+ \right. \\
& \left. + \int_{D_{\text{ext}}} \log \left( 1 + \frac{k^{\theta^-}(D,u)}{\beta^{\theta^-}(u)} \right) \alpha^{\theta^-}(du) + E_1(v_M \beta_M^-) \alpha_M^- \right\}.
\end{aligned}
$$

(f) Set $\theta^T \leftarrow \theta^+$ with probability $\min(1, P^t)$, otherwise set $\theta^t \leftarrow \theta^- = \theta^{t-1}$.

4. Set $t \leftarrow t + 1$ and return to step 1.

If a regression component is specified in the intensity function, then the regression coefficients $\beta$ are included in $\theta$ and updated as described above. As $\theta$ grows large, updating the entire vector $\theta$ at once will become inefficient, so we suggest updating the components individually or in small blocks. For further details on the algorithm, see the references mentioned above.

# Bibliography


Adams, R. P., Murray, I., and MacKay, D. J. C. (2009), "Tractable nonparametric Bayesian inference in Poisson processes with Gaussian process intensities," in *Proceedings of the 26th Annual International Conference on Machine Learning - ICML '09*, New York, New York, USA, ACM Press.

Akman, V. E. and Raftery, A. E. (1986), "Bayes Factors for Non-homogeneous Poisson Process with Vague Prior Information," *Journal of the Royal Statistical Society. Series B (Methodological)*, pp. 322–329.

Ang, Q. W., Baddeley, A., and Nair, G. (2012), "Geometrically Corrected Second Order Analysis of Events on a Linear Network, with Applications to Ecology and Criminology," *Scandinavian Journal of Statistics*, 39, 591–617.

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

Baddeley, A. and Turner, R. (2005), "Spatstat: an R package for analyzing spatial point patterns," *Journal of Statistical Software*, 12, 1–42.

Baddeley, A., Turner, R., Møller, J., and Hazelton, M. (2005), "Residual analysis for spatial point processes," *Journal of the Royal Statistical Society. Series B*, 67, 617–666.

Baddeley, A., Møller, J., and Pakes, A. G. (2008), "Properties of residuals for spatial point processes," *Annals of the Institute of Statistical Mathematics*, 60, 627–649.

Baddeley, A. J. and van Lieshout, M. N. M. (1995), "Area-interaction point processes," *Annals of the Institute of Statistical Mathematics*, 47, 601–619.

Baddeley, A. J., Moller, J., and Waagepetersen, R. (2000), "Non- and semi-parametric estimation of interaction in inhomogeneous point patterns," *Statistica Neerlandica*, 54, 329–350.

Banerjee, S., Carlin, B. P., and Gelfand, A. E. (2014), *Hierarchical modeling and analysis for spatial data*, Chapman and Hall/CRC Press, Boca Raton, FL, 2 edn.

Berman, M. and Turner, T. R. (1992), "Approximating point process likelihoods with GLIM," *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 41, 31–38.

Berthelsen, K. K. and Møller, J. (2002), "A primer on perfect simulation for spatial point processes," *Bulletin of the Brazilian Mathematical Society*, 33, 351–367.

Berthelsen, K. K. and Møller, J. (2003), "Likelihood and Non-parametric Bayesian MCMC Inference for Spatial Point Processes Based on Perfect Simulation and Path Sampling," *Scandinavian Journal of Statistics*, 30, 549–564.

Berthelsen, K. K. and Møller, J. (2006), "Bayesian analysis of Markov point processes," in *Case Studies in Spatial Point Processes*, eds. A. Baddeley, P. Gregori, J. Mateu, R. Stoica, and D. Stoyan, pp. 85–97, Springer-Verlag, New York.

Berthelsen, K. K. and Møller, J. (2008), "Non-Parametric Bayesian Inference for Inhomogeneous Markov Point Processes," *Australian & New Zealand Journal of Statistics*, 50, 257–272.

Besag, J. (1977), "Some methods of Statistical Analysis for Spatial Data," *Bulletin of the International Statistical Institute*, 47, 77–92.

Best, N. G., Ickstadt, K., and Wolpert, R. L. (2000), "Spatial Poisson Regression for Health and Exposure Data Measured at Disparate Resolutions," *Journal of the American Statistical Association*, 95, 1076–1088.

Bowman, A. W. (1984), "An Alternative Method of Cross-Validation for the Smoothing of Density Estimates," *Biometrika*, 71, 353.

Brier, G. E. (1950), "Verification of Forecasts Expressed in Terms of Probability," *Monthly Weather Review*, 78, 1–3.

Christensen, O. F., Roberts, G. O., and Rosenthal, J. S. (2005), "Scaling limits for the transient phase of local Metropolis-Hastings algorithms," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 67, 253–268.

Cox, D. R. (1955), "Some statistical methods connected with series of events," *Journal of the Royal Statistical Society. Series B (Statistical Methodology)*, 17, 129–164.

Cressie, N. A. (1993), *Statistics for spatial data, revised edition*, Wiley, New York.

Daley, D. J. and Vere-Jones, D. (1998), *An Introduction to the Theory of Point Processes*, Springer-Verlag, New York.

Diggle, P. and Marron, J. S. (1988), "Equivalence of Smoothing Parameter Selectors in Density and Intensity Estimation," *Journal of the American Statistical Association*, 83, 793–800.

Diggle, P. J. (1983), *Statistical analysis of spatial point patterns*, Academic Press, London.

Dixon, P. M. (2002), "Ripleys K function," in *Encyclopedia of Environmetrics*, vol. 3, pp. 1796–1803, Wiley.

Epstein, E. S. (1969), "A scoring system for probability forecasts of ranked categories," *Journal of Applied Meteorology*, 8, 985–987.

Gelfand, A. E., Diggle, P. J., Guttorp, P., and Fuentes, M. (eds.) (2010), *Handbook of spatial statistics*, Chapman & Hall/CRC Press, London.

Gelman, A. and Meng, X.-L. (1998), "Simulating normalizing constants: from importance sampling to bridge sampling to path sampling," *Statistical Science*, 13, 163–185.

Gelman, A. and Rubin, D. B. (1992), "Inference from Iterative Simulation Using Multiple Sequences," *Statistical Science*, 7, 457–511.

Gelman, A. and Shalizi, C. R. (2013), "Philosophy and the practice of Bayesian statistics." *The British journal of mathematical and statistical psychology*, 66, 8–38.

Gelman, A., Meng, X.-l., and Stern, H. (1996), "Posterior Predictive Assessment of Model Fitness via Realized Discrepancies," *Statistica Sinica*, 6, 733–807.

Gelman, A., Carlin, J. B., Stern, H. S., Dunson, D. B., Vehtari, A., and Rubin, D. B. (2013), *Bayesian Data Analysis*, Chapman and Hall/CRC, Boca Raton, FL, 3 edn.

Georgii, H.-O. (1976), "Canonical and grand canonical Gibbs states for continuum systems," *Communications in Mathematical Physics*, 48, 31–51.

Girolami, M. and Calderhead, B. (2011), "Riemann manifold Langevin and Hamiltonian Monte Carlo methods," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 73, 123–214.

Gneiting, T. and Raftery, A. E. (2007), "Strictly Proper Scoring Rules, Prediction, and Estimation," *Journal of the American Statistical Association*, 102, 359–378.

Green, P. J. (1995), "Reversible Jump Markov Chain Monte Carlo Computation and Bayesian Model Determination," *Biometrika*, 82, 711–732.

Guttorp, P. and Thorarinsdottir, T. L. (2012), "Advances and Challenges in Space-time Modelling of Natural Events," *Advances and Challenges in Space-time Modelling of Natural Events*, 207, 79–102.

Harkness, R. D. and Isham, V. (1983), "A Bivariate Spatial Point Pattern of Ants' Nests," *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 32, 293.

Högmander, H. and Särkkä, A. (1999), "Multitype spatial point patterns with hierarchical interactions." *Biometrics*, 55, 1051–1058.

Huang, F. and Ogata, Y. (1999), "Improvements of the maximum pseudo-likelihood estimators in various spatial statistical models," *Journal of Computational and Graphical Statistics*, 8, 510–530.

Ickstadt, K., Wolpert, R. L., and Lu, X. (1998), "Modeling travel demand in Portland, Oregon," in *Practical Nonparametric and Semiparametric Bayesian Statistics*, pp. 305–322, Springer.

Illian, J., Penttinen, A., Stoyan, H., and Stoyan, D. (2008), *Statistical analysis and modelling of spatial point patterns*, Wiley-Interscience.

Illian, J. B. and Hendrichsen, D. K. (2010), "Gibbs point process models with mixed effects," *Environmetrics*, 21, 341–353.

Illian, J. B., Møller, J., and Waagepetersen, R. P. (2009), "Hierarchical spatial point process analysis for a plant community with high biodiversity," *Environmental and Ecological Statistics*, 16, 389–405.

Ji, C., Merl, D., Kepler, T. B., and West, M. (2009), "Spatial mixture modelling for unobserved point processes: examples in immunofluorescence histology," *Bayesian Analysis*, 4, 297–315.

King, R., Illian, J. B., King, S. E., Nightingale, G. F., and Hendrichsen, D. K. (2012), "A Bayesian Approach to Fitting Gibbs Processes with Temporal Random Effects," *Journal of Agricultural, Biological, and Environmental Statistics*.

Kottas, A. (2006), "Dirichlet Process Mixtures of Beta Distributions, with Applications to Density and Intensity Estimation," in *Workshop on Learning with Nonparametric Bayesian Methods, 23rd International Conference on Machine Learning (ICML)*, Pittsburgh, PA.

Lahiri, S. (1999), "Asymptotic distribution of the empirical spatial cumulative distribution function predictor and prediction bands based on a subsampling method," *Probability Theory and Related Fields*, 114, 55–84.

Lahiri, S. N. (2003), "Central limit theorems for weighted sums of a spatial process under a class of stochastic and fixed designs," *Sankhyā: The Indian Journal of Statistics*, 65, 356–388.

Lewis, P. A. W. and Shedler, G. S. (1979), "Simulation of nonhomogeneous Poisson processes by thinning," *Naval Research Logistics Quarterly*, 26, 403–413.

Møller, J. (2003), "Shot noise Cox processes," *Advances in Applied Probability*, 35, 614–640.

Møller, J. and Waagepetersen, R. P. (2003), *Statistical Inference and Simulation for Spatial Point Processes*, Chapman and Hall/CRC, Florida.

Møller, J. and Waagepetersen, R. P. (2007), "Modern Statistics for Spatial Point Processes," *Scandinavian Journal of Statistics*, 34, 643–684.

Møller, J., Syversveen, A. R., and Waagepetersen, R. P. (1998), "Log Gaussian Cox Processes," *Scandinavian Journal of Statistics*, 25, 451–482.

Møller, J., Pettitt, A. N., Reeves, R., and Berthelsen, K. K. (2006), "An efficient Markov chain Monte Carlo method for distributions with intractable normalising constants," *Biometrika*, 93, 451–458.

Murray, I. and Adams, R. P. (2010), "Slice sampling covariance hyperparameters of latent Gaussian models," in *Advances in Neural Information Processing Systems 23*, eds. J. Lafferty, C. K. I. Williams, J. Shawe-Taylor, R. Zemel, and A. Culotta, pp. 1723–1731.

Murray, I., Adams, R. P., and Mackay, D. J. C. (2010), "Elliptical slice sampling," *JMLR: W&CP*, 9, 541–548.

Neyman, J. and Scott, E. (1958), "Statistical approach to problems of cosmology," *Journal of the Royal Statistical Society. Series B (Methodological)*, 20, 1–43.

Nguyen, X. X. and Zessin, H. (1979), "Integral and differential characterizations of the Gibbs process," *Mathematische Nachrichten*, 88, 105–115.

Numata, M. (1961), "Forest vegetation in the vicinity of Choshi. Coastal flora and vegetation at Choshi, Chiba Prefecture IV (in Japanese)," *Bulletin of the Choshi Marine Laboratory, Chiba University*, 3, 28–48.

Ogata, Y. and Tanemura, M. (1981), "Estimation of Interaction Potentials of Spatial Point Patterns Through the Maximum Likelihood Procedure," *Annals of the Institute of Statistical Mathematics*, 33, 315–338.

Raftery, A. E. and Akman, V. E. (1986), "Bayesian analysis of a Poisson process with a change-point," *Biometrika*, 73, 85–89.

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. Series B (Statistical Methodology)*, 39, 172–212.

Ripley, B. D. (1981), *Spatial Statistics*, John Wiley & Sons, New York.

Sherman, M. and Carlstein, E. (1994), "Nonparametric Estimation of the Moments of a General Statistics Computed from Spatial Data," *Journal of the American Statistical Association*, 89, 496–500.

Simpson, D., Illian, J., Lindgren, F., Sørbye, S. H., and Rue, H. (2011), "Going off grid: Computationally efficient inference for log-Gaussian Cox processes," *arXiv*, pp. 1–19.

Stoyan, D. and Grabarnik, P. (1991), "Second-order Characteristics for Stochastic Structures Connected with Gibbs Point Procosses," *Mathematische Nachrichten*, 151, 95–100.

Strauss, D. J. (1975), "A Model for Clustering," *Biometrika*, 62, 467–475.

Taddy, M. a. (2010), "Autoregressive Mixture Models for Dynamic Spatial Poisson Processes: Application to Tracking Intensity of Violent Crime," *Journal of the American Statistical Association*, 105, 1403–1417.

Taddy, M. A. and Kottas, A. (2012), "Mixture Modeling for Marked Poisson Processes," *Bayesian Analysis*, 7, 335–362.

Tokdar, S. T. and Ghosh, J. K. (2007), "Posterior consistency of logistic Gaussian process priors in density estimation," *Journal of Statistical Planning and Inference*, 137, 34–42.

van Lieshout, M. N. M. (2011), "A J-function for inhomogeneous point processes," *Statistica Neerlandica*, 65, 183–201.

Waagepetersen, R. (2004), "Convergence of posteriors for discretized log Gaussian Cox processes," *Statistics & Probability Letters*, 66, 229–235.

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

Widom, B. and Rowlinson, J. (1970), "New model for the study of liquid-vapor phase transitions," *Journal of Chemical Physics*, 52, 1670–1684.

Wolpert, R. and Ickstadt, K. (1998), "Poisson/gamma random field models for spatial statistics," *Biometrika*, 85, 251–267.

Wolpert, R. L., Clyde, M. a., and Tu, C. (2011), "Stochastic expansions using continuous dictionaries: Lévy adaptive regression kernels," *The Annals of Statistics*, 39, 1916–1962.

Zhang, H. (2004), "Inconsistent Estimation and Asymptotically Equal Interpolations in Model-Based Geostatistics," *Journal of the American Statistical Association*, 99, 250–261.

# Biography

Thomas Jeffrey Leininger was born in Oakland, California, on November 28, 1984, and was raised in Ogden, Utah. He completed an integrated M.S./B.S. degree in Statistics from Brigham Young University in August 2010. He received an M.S. in Statistical Science from Duke University in May 2013 and plans to graduate with his Ph.D. in Statistical Science from Duke University in May 2014. After graduating, he will be working for Capital One in Plano, Texas.