Imperial College London

Imperial College London

Department of Mathematics

Replicating bivariate point patterns that display both aggregation and segregation

Author:
Joshua Corneck-Willcox

Supervisor:
Dr James Martin

September 2, 2021

CID: 01897838

Submitted in partial fulfilment of the requirements for the MSc in Statistics of Imperial College London

Anti-Plagarism Statement

The work contained in this thesis is my own work unless otherwise stated.
Joshua Corneck-Willcox

Abstract

In the analysis of natural tree distributions, it is sometimes observed that two or more species cluster together due to a preference for similar environmental conditions, but then, within these clusters, inhibition is seen between the species, often resulting from competition for resources. In spatial point processes, popular models exist for modelling the effect of aggregation and segregation separately, yet this effect of large-scale aggregation and small-scale inhibition within the same process is currently in need of development, particularly in the context of multivariate processes.

In this thesis, we present a novel extension to the univariate model of Vihrs et al. (2020), looking to capture this multiscale phenomenon in a bivariate setting. This model is not expressible in closed form, and a method for simulating realisations is discussed. Furthermore, we examine the effectiveness of an ABC inference procedure for fitting the model in the context of simulated and real data. The model is shown to provide an improvement on current, popular processes for the Japanese pines data set (Numata 1964). All important code used in this thesis can be found in the GitHub repository SPPs.

Acknowledgements

First and foremost, I would like to extend my deepest thanks to Dr James Martin for his time, expertise and availability over the past few months, without which this thesis would never have come to fruition. I would also like to thank Dr Marina Evangelou for directing a thoroughly enjoyable MSc, even in the midst of such tricky circumstances.

Furthermore, I would like to thank Professors Adrian Baddeley, Rolf Turner and Ege Rubak for their ethereal book *Spatial Point Patterns: Methodology and Applications with R* (Baddeley, Rubak & Turner 2015) and the accompanying `R` package `spatstat`, without which much of this thesis would not have been possible in the given time frame.

Finally, and on a personal note, I would like to thank my family for their continual and unwavering support, even in the face of my incessant and (arguably...) boring mathematical anecdotes.

Contents

1 Introduction **3**

2 Fundamentals of Point Processes **5**
2.1 Precise definition of point processes and their features 5
2.2 Poisson point processes 8
2.3 Density functions of point processes 10

3 Summary Statistics for Point Processes **12**
3.1 General moment measures 12
3.2 First-order characteristics 13
3.3 Second-order characteristics 15
3.4 Estimation of second-order characteristics 18
3.5 Interpretation of second-order characteristics 21

4 Cox and Markov Point Processes **22**
4.1 Random fields and covariance functions 22
4.2 Cox processes 24
4.3 Markov point processes 26

5 Approximate Bayesian Computation and the Birth-Death Metropolis-Hastings Algorithm **30**
5.1 Principle of ABC 30
5.2 ABC rejection sampling 31
5.3 Birth-death Metropolis Hastings for spatial point processes 32

6 LGCP-Strauss Process **34**
6.1 LGCP-Strauss process 34
6.2 Simulation of a univariate LGCP-Strauss process 34
6.3 Bivariate LGCP Strauss model 37
6.4 ABC inference for the bivariate LGCP-Strauss model 43

7 Application of the bivariate LGCP-Strauss model to real data **51**
7.1 Japanese black pines data set 51
7.2 Model fitting and model checking 52

8 Conclusion and recommendations for future work **54**
8.1 Conclusion 54
8.2 Recommendations for future work 54

A Supplementary plots for inference in the general model **58**
A.1 Scaling the A-matrix 58
A.2 Including more random fields 59

Abbreviations and notation

We will make use of a number of abbreviations and notation in this thesis:

- i.i.d - independently and identically distributed
- $\mathbb{N}^+$ - the set of positive integers ($\mathbb{N}$ then denotes the set of non-negative integers)
- $\mathcal{U}(a, b)$ - a uniform distribution on $(a, b) \subseteq \mathbb{R}$
- BDMH - birth-death Metropolis-Hastings
- ABC - approximate Bayesian computation
- LGCP - log-Gaussian Cox process
- $\Gamma(x)$ - the gamma function, defined as $\Gamma(x) = \int_0^\infty t^{x-1} e^{-t} \mathrm{d}t$ for $\mathfrak{R}(x) > 0$.

Chapter 1

Introduction

In a world where climate change is an ever more dire issue, effective replanting of drained and depleted forests is of growing importance. Unfortunately, however, one cannot simply throw seeds into the air and hope that the next Amazon rainforest sprouts from the ground. Many thousands of years of natural selection lead to the distributions of species in forests that we observe today, and thus an understanding of the structure underpinning the interactions between rivaling species, and the preferences of one species for a certain area over another is necessary for efficacious replanting. In mathematics, we usually model the distribution of trees by a *spatial point process*, and it is through the use of these processes that we look to capture the aforementioned interactions.

Heuristically, spatial point processes are considered to be models of the random patterns produced by the spatial locations of "things" of interest. These "things" can be very general: they could be anything from the car crashes in a city shown in Figure 1.1 (McSwiggan 2019) to the distribution of spines on a dendrite in Figure 1.2; however, in this research, the things of interest will be trees. In essence, one looks to construct a *point process model* to describe the spatial distributions of events in some space of our choosing. We call the data set consisting of the spatial distributions of these events a *spatial point pattern*.

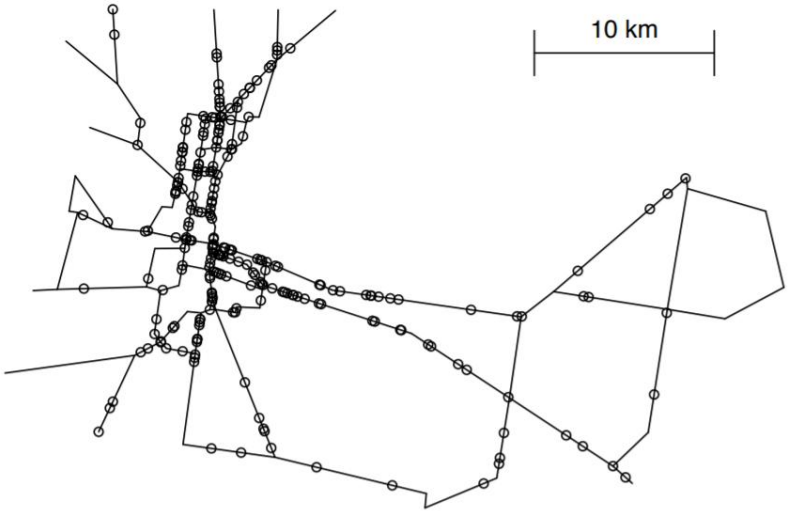


Figure 1.1: A point pattern of the severe car crash incidents in Geelong, Australia between 2009 and 2011 (McSwiggan 2019)

The methodology employed in the field of spatial point process is analogous to that ubiquitous in the vast majority of other statistical disciplines: one hypothesises a (parametric) model to describe a phenomenon, obtains data, uses this data to infer the parameter values in the model and then performs goodness-of-fit tests for the subsequent model. In our framework, we propose a parametric spatial point process model and then use the point pattern and some appropriate methodology to infer the parameters of our model.

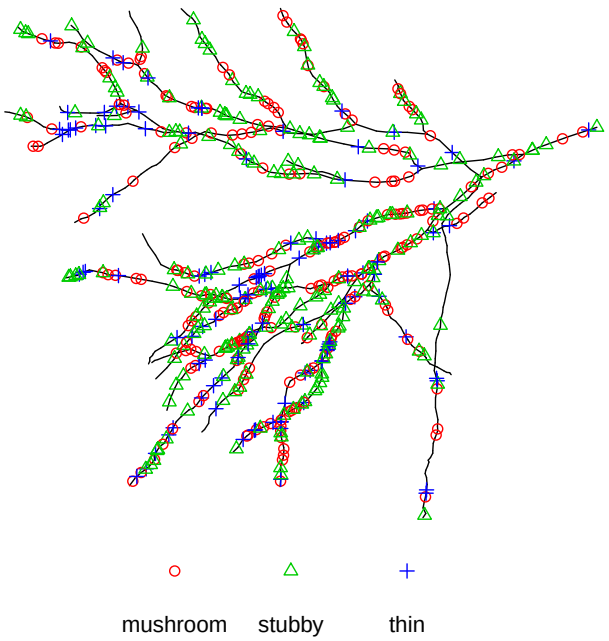


Figure 1.2: The locations of 566 spines observed on one branch of the dendritic tree of a rat neuron. This data set was sourced from the `spatstat` package.

In this thesis, we will place our attention on modelling the phenomenon of *multi-scale* interaction between trees. Specifically, we look to construct a spatial point process model to describe a particular interaction between differing species of trees whereby they are seen to *cluster together on the large-scale*, perhaps due to a preference for similar environmental conditions, but then *segregate on the small-scale* due to competition. The terms large-scale and small-scale are ambiguous, and we don't look to preemptively define these scales, but rather seek to construct models than can operate based on any length-scale definition.

The remainder of this thesis is structured as follows. Chapters 2, 3, 4 and 5 provide the necessary foundations in spatial point processes, simulation methods and inference techniques. Chapter 6 then presents our novel model and analyses the effectiveness of *approximate Bayesian computation* to recover the parameters of this model on simulated data. Chapter 7 then considers an application of this model to a real data set. Finally, Chapter 8 gives a concluding overview and some recommendations of possible future work.

Chapter 2

Fundamentals of Point Processes

In this chapter, we will first give the precise definition of a point process in some generality before moving on to discuss their density functions and the all-important Poisson process. For a more detailed discussion of the material presented here, the reader is encouraged to refer to Møller & Waagpetersen (2004), on which this chapter is structured.

2.1 Precise definition of point processes and their features

2.1.1 Definition of a point process

A *spatial point process* can be defined as a random countable subset of a space S, where S can be arbitrary (Møller & Waagpetersen 2004). For all applications in this thesis we will consider $S = \mathbb{R}^2$. However, to keep the feel of generality, we will make most definitions on $S \subseteq \mathbb{R}^d$ for $d \in \mathbb{N}^+$ to emphasise how the presented material can be generalised. S can in fact be taken to be far more general than $\mathbb{R}^d$, but a discussion of more abstract measure spaces is not inline with the spirit of this thesis. In practice, observations are made in some finite window $W \subseteq S$. For example, in the modelling of car crashes in a city, we may only have data from a particular borough, which would then define our observation window, W. We construct models to simulate spatial point patterns on W.

Much like most of statistics, we can describe a point process via a set of events and corresponding probabilities (Baddeley 2006). In line with standard statistical notation, we will use capital letters to denote a spatial point process and their lower-case counterparts to denote a spatial point pattern, meaning that $x = \{x_1, x_2, \cdots\}$ will represent a spatial point pattern produced by our spatial point process X, with each x_i denoting a spatial location. This is analogous to standard statistical theory where, for example, we may take $X \sim \mathcal{N}(0,1)$ to be a random variable (the point process) and $x = 1.96$ to be an observation of the variable (the point pattern). Furthermore, we will drop the term spatial in spatial point process and spatial point pattern from here onward for conciseness.

We will write $N(x)$ to be the cardinality of a countable subset $x \subseteq S$ and, in the case that we are interested in the number of points of x lying in some bounded Borel set $B \subseteq S$, we will write $N(B)$ as a shorthand for $N(x \cap B)$. The restriction of X to B shall be written as X_B. Throughout this exploration, we will take $S, B, ...$ to be Borel sets, unless specified otherwise.

We will refer to our spatial point process as *locally finite* if $N(B) < \infty$ with probability 1 for any bounded $B \subseteq S$. Furthermore, we will also assume that X satisfies $N(\{b\}) \in \{0,1\}$, where b is some spatial location in B, so that there can be no coincident points in our process. This property shall be referred to as the process being *simple*. In this way, X will take values

in a space N^{lf}, defined by

$$N^{lf} = \{x \subseteq S : N(B) < \infty \text{ for any bounded } B \subseteq S\}$$

This is the family of all *locally finite* patterns. Further, we can equip N^{lf} with the σ-algebra $\mathcal{N}$ defined to be the smallest σ-algebra generated by sets of the form

$$\{\{x \in N^{lf} : N(B) = k\} : \text{ for bounded } B,\ k \in \mathbb{N}\}$$

Using the definitions of N^{lf} and $\mathcal{N}$, we can properly define a spatial point process X on S.

Definition 2.1: Spatial Point Process

A *spatial point process* is a measurable mapping defined on some probability space $(\Omega, \mathcal{F}, \mathbb{P})$ taking values in $(N^{lf}, \mathcal{N})$.

This, of course, is all rather formal, and it is sufficient for this exploration for us to simply think of a locally finite point process in a heuristic sense as a *random mechanism whose output is a finite collection of points in some finite space*. It is worth noting that since we consider only simple point processes, we may just as well define a point process here to be a random set of points.

2.1.2 Marked and multivariate point processes

It is common for point processes to come equipped with extra information, and not simply just the spatial locations of the points. For example, spatial locations of trees may come equipped with a measurement of their diameter at breast height, or a car crash with the number of fatalities. As such, we need a way to extend our definition to encompass these processes.

We will refer to this extra information as a *mark* which takes values in some *mark space* M (depending on the notion of the accompanying information, this mark space can take many forms). There are then two similar yet distinct ways we can think of extending our point process to include these marks. These two extensions will be termed as *multivariate* and *marked* processes, and we start by defining the latter.

Let Y be a point process on $T \subseteq \mathbb{R}^d$ and let our mark space be M, where M can be discrete or continuous. In a *marked point process*, we consider generating Y on T, and then to each $\xi \in Y$ we append an $m_\xi \in M$ according to some generating procedure to obtain our marked process X on $S = T \times M$

$$X = \{(\xi, m_\xi) : \xi \in Y\}$$

On the other hand, in a *multivariate point process*, we consider $n \in \mathbb{N}^+ \cup \infty$ separate point processes, $X_1, X_2, \cdots, X_n$ each on $S \subseteq \mathbb{R}^d$, being produced according *distinct but possibly dependent* generating processes. In this way, we form an n-tuple $X = (X_1, \cdots, X_n)$ on $S^n := S \times \cdots \times S$, with n terms in the product. This framework would clearly not be appropriate in the case that M is continuous. We use the term *subprocess* to refer to any of the X_i individually.

The appropriate choice of one of these two processes will depend on the nature of the model. In our setting, where the marks correspond to species of tree, it would be more appropriate to adopt a multivariate framework, as we can allow n to equal the number of species we consider and then allow the generating procedure of each species to depend on those of the

others.

The left and right panels in Figure 2.1 show point patterns from a multivariate and marked point process, respectively. The tree diameters in the right-panel are clearly continuous quantities, which is why a marked framework is most appropriate.

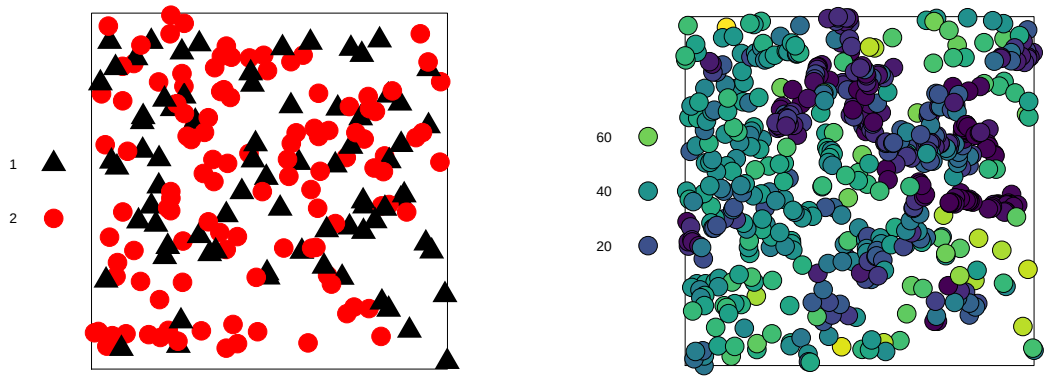


Figure 2.1: The left panel shows an example of a multivariate process, where each point colour corresponds to a different discrete property. The right panel is the longleaf dataset (Platt et al. 1988) procured from the `spatstat` package, and is an example of a marked process where the shade corresponds to the diameter of the trees.

2.1.3 Intensity functions

Heuristically, an *intensity function* can be considered as a description of the density of points per unit area in a point process. An intensity function need not always exist, and it is in fact defined as the derivative of the *intensity measure* (and so clearly differentiability of this measure is a necessary condition for existence).

Definition 2.2: Intensity measure and intensity function

The *intensity measure* μ of a process X on S is defined as

$$\mu(B) = \mathbb{E}(N(B)), \quad B \subseteq S$$

If μ satisfies

$$\mu(B) = \int_B \lambda(\xi)\mathrm{d}\xi$$

for some (non-negative) function λ, we call λ the *intensity function* of X. If λ is constant, then X is called *homogeneous*, otherwise it is called *inhomogeneous*.

We can think of $\lambda(\xi)\mathrm{d}\xi$ as describing the probability of finding a point in a ball of radius $\mathrm{d}\xi$ centred at ξ. As such, to say that a process is homogeneous is to say that the expected number of points in two equally sized areas is the same.

An intensity measure is an example of a *first-order characteristic* of a point process, and we

will take a look at second-order characteristics shortly. In Figure 2.2, we see a realisation of an inhomogeneous point process along with a kernel estimate of the intensity function of the process. This will be described in much more detail in due course, but for now one can think of the colour at a point as relating to the probability of seeing a point there. Notice how we have spatially varying probabilities, with a clear East to West trend, implying that we most likely have an inhomogeneous process.

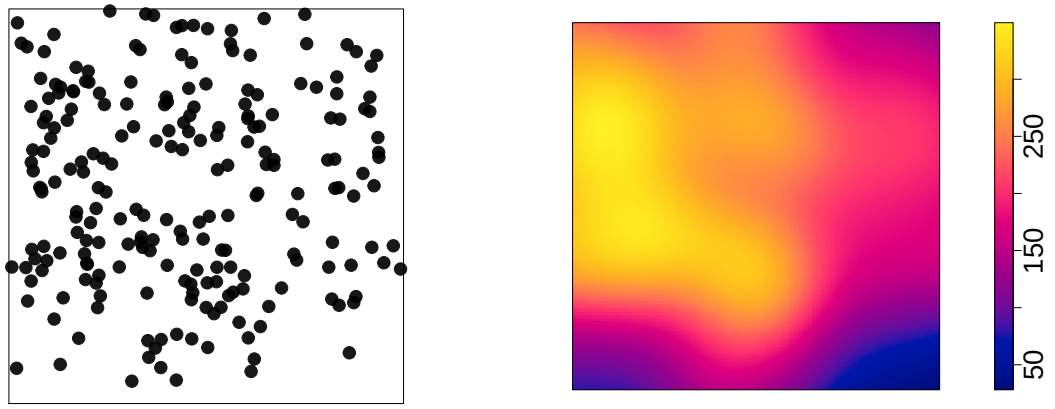


Figure 2.2: The left panel shows a realisation of an inhomogeneous Poisson process and the right panel shows the corresponding intensity function (using the likelihood cross-validation method).

2.1.4 Stationarity and isotropy

A point process X on S is called *stationary* if its statistical properties are invariant under a translation. In particular, if x is a point pattern of cardinality n on $\mathbb{R}^d$, then $x + v = \{x_i + v : i = 1, \cdots, n\}$ is statistically equivalent to x.

It is interesting to note that stationarity implies homogeneity. To see this, let B be a region in $\mathbb{R}^d$ and X a stationary point process on the same space. Then, for any $v \in \mathbb{R}^d$, $N((X + v) \cap B) = N(X \cap (B - v))$, and since $X + v$ has the same statistical properties as X, this implies that $N(X \cap (B - v))$ has the same distribution as $N(X \cap B)$. As such, the expected number of points falling in any region is seen to be the same.

A point process X on S is called *isotropic* if its statistical properties are invariant under a rotation.

2.2 Poisson point processes

A *homogeneous Poisson point process* is the most basic point process we will consider and, as such, it will often serve as a null model. Not only are they the first point of call in building a model for a process, they also serve as a reference point for many summary statistics that we will describe in due course.

The versatility of the Poisson model as a base line lies in its assumption of *complete spatial*

randomness. Specifically, this refers to the property of points being independently and uniformly distributed in the given space. This is rarely a reasonable property to assume, but many models can be built from alterations of this process, and assuming these properties allows us to make inference on the interactions and distributions of points.

The definition of a Poisson process that we will use relies upon the definition of another process called a *binomial point process*. Consider a point process X on S and let f be a density function on $B \subseteq S$. If X has density f and consists of $n \in \mathbb{N}^+$ i.i.d. points on B, we call X a binomial point process of n points in B with density f and write $X \sim \text{binomial}(B, n, f)$. The important point to note here is that the number of points in the process is *fixed* at some positive integer n.

Definition 2.3: Poisson point process

We call a point process X on S a *Poisson point process* with intensity function $\lambda(\xi)$ if each of the following two conditions hold:

1. For any $B \subseteq S$ with $\mu(B) < \infty$, we have $N(B) \sim \text{Poisson}(\mu(B))$, where we recall
$$\mu(B) := \int_B \lambda(\xi) \mathrm{d}\xi$$
2. For any $n \in \mathbb{N}$ and $B \subseteq S$ with non-zero and finite measure, we have
$$X_B | N(B) = n \sim \text{binomial}(B, n, \lambda(\xi)/\mu(B))$$

and write $X \sim \text{Poisson}(S, \lambda)$. If λ is constant, the process is called *homogeneous*, otherwise it is *inhomogeneous*.

An important example of such a process is when $\lambda \equiv 1$, and we call this the *standard/unit Poisson process*. This process is vital for defining the density functions for other less standard point processes.

Figure 2.3 shows two realisations of a homogeneous Poisson process on $W = [0,1]^2$ using the same intensity function. Of course, the subsequent pattern is not uniform, but there is no clear spatial trend, which is precisely what we would expect. An inhomogeneous Poisson process with a clear spatial trend is seen in Figure 2.2.

2.2.1 Expansion of a Poisson process

Let $X \sim \text{Poisson}(S, \lambda(\xi))$ and take $A \subseteq N^{\ell f}$. By the law of total probability, we can write

$$\mathbb{P}(X_B \in A) = \sum_{n=0}^{\infty} \mathbb{P}(X_B \in A | N(B) = n) \mathbb{P}(N(B) = n)$$

for any $B \subseteq S$. By the definition of a Poisson point process, we know each term in the sum is the probability of a binomial point process belonging to A, namely

$$X_B | N(B) = n \sim \text{binomial}(B, n, \lambda(\xi)/\mu(B))$$

and so

$$\mathbb{P}(X_B \in A | N(B) = n) = \int_B \cdots \int_B \mathbb{I}[\{x_1, \cdots, x_n\} \in A] \prod_{i=1}^{n} \frac{\lambda(x_i)}{\mu(B)} \mathrm{d}x_1 \cdots \mathrm{d}x_n$$

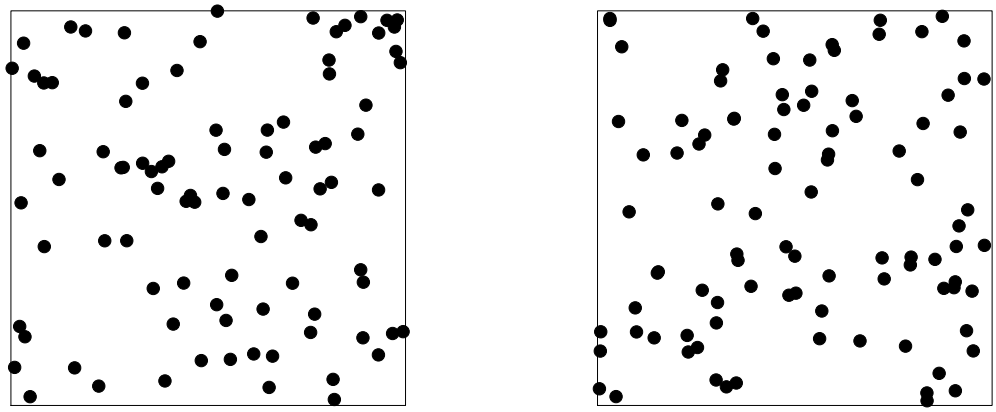


Figure 2.3: Both panels show separate realisations of a homogeneous Poisson process on the unit square with the same intensity.

Further, we know

$$\mathbb{P}(N(B) = n) = \frac{\exp(-\mu(B))\mu(B)^n}{n!}$$

and thus we deduce that

$$\mathbb{P}(X_B \in A) = \sum_{n=0}^{\infty} \frac{\exp(-\mu(B))}{n!} \int_B \cdots \int_B \mathbb{I}[\{x_1, \cdots, x_n\} \in A] \prod_{i=1}^{n} \lambda(x_i) \mathrm{d}x_1 \cdots \mathrm{d}x_n \tag{2.1}$$

It is not hard to show further that for any $h : N_{\mathrm{lf}} \to [0, \infty)$ and $B \subseteq S$ with $|B| < \infty$

$$\mathbb{E}[h(X_B)] = \sum_{n=0}^{\infty} \frac{\exp(-\mu(B))}{n!} \int_B \cdots \int_B h(\{x_1, \cdots, x_n\} \in A) \prod_{i=1}^{n} \lambda(x_i) \mathrm{d}x_1 \cdots \mathrm{d}x_n \tag{2.2}$$

2.3 Density functions of point processes

2.3.1 A general definition

We say that a point process X_1 is *absolutely continuous* with respect to another process X_2 defined on the same space S if and only if $\mathbb{P}(X_2 \in F) = 0$ implies that $\mathbb{P}(X_1 \in F) = 0$ for any $F \subseteq N^{\ell f}$. The Radon-Nikodym theorem tells thus that this is entirely equivalent to there existing a function $f : N^{\ell f} \to [0, \infty]$ such that

$$\mathbb{P}(X_1 \in F) = \mathbb{E}[\mathbb{I}[X_2 \in F] f(X_2)]$$

and we call f a density of X_1 *with respect to* X_2.

It is very often the case that we define the densities of point processes with respect to the unit Poisson process.

2.3.2 Density of a Poisson process

The first density we will derive is that of an arbitrary Poisson process with respect to the standard Poisson process. The density of a Poisson process need not exist with respect to another

Poisson process, but it is guaranteed to exist with respect to the unit process, provided that $|S| < \infty$ (Møller & Waagpetersen 2004).

Theorem 2.1: Lemma 3.8 (Møller & Waagpetersen 2004)

For $i = 1, 2$, let $\lambda_i : S \to [0, \infty)$ so that $\mu_i(S) = \int_S \lambda_i(\xi) \mathrm{d}\xi < \infty$ and $\lambda_2(u) > 0$ whenever $\lambda_1(u) > 0$. Then $X_1 \sim \text{Poisson}(S, \lambda_1)$ is absolutely continuous with respect to $X_2 \sim \text{Poisson}(S, \lambda_2)$, with density

$$f(x) = \exp\left(\mu_2(S) - \mu_1(S)\right) \prod_{\xi \in x} \frac{\lambda_1(\xi)}{\lambda_2(\xi)}$$

for finite point configurations $x \subset S$.

Proof. Using Equation (2.1), we see that we can write

$$\begin{aligned}
\mathbb{P}(X_1 \in A) &= \sum_{n=0}^{\infty} \frac{\exp(-\mu_1(S))}{n!} \int_S \cdots \int_S \mathbb{I}[\{x_1, \cdots, x_n\} \in A] \prod_{i=1}^{n} \lambda_1(x_i) \mathrm{d}x_1 \cdots \mathrm{d}x_n \\
&= \sum_{n=0}^{\infty} \frac{\exp(-\mu_2(S))}{n!} \int_S \cdots \int_S \mathbb{I}[\{x_1, \cdots, x_n\} \in A] e^{\mu_2(S) - \mu_1(S)} \prod_{i=1}^{n} \frac{\lambda_1(x_i)}{\lambda_2(x_i)} \prod_{i=1}^{n} \lambda_2(x_i) \mathrm{d}x_1 \cdots \mathrm{d}x_n \\
&= \mathbb{E}\left(\mathbb{I}[X_2 \in A] f(X_2)\right)
\end{aligned}$$

where

$$f(x) = e^{\mu_2(S) - \mu_1(S)} \prod_{i=1}^{n} \frac{\lambda_1(x_i)}{\lambda_2(x_i)}$$

It then follows that X_1 is absolutely continuous with respect to X_2 with density f. □

Taking X_2 to be the standard Poisson process, we obtain the density $f(x)$ of $X \sim \text{Poisson}(S, \lambda)$ with respect to $\text{Poisson}(S, 1)$ as

$$f(x) = \exp\left(\int_S (1 - \lambda(\xi)) \mathrm{d}\xi\right) \prod_{\eta \in x} \lambda(\eta)$$

2.3.3 Density of an arbitrary process

Let X be a point process on $S \subseteq \mathbb{R}^d$ with $|S| < \infty$. We require $|S| < \infty$ for a well-defined density to exist. Furthermore, let X have density f with respect to the unit Poisson process on S. This density is then concentrated onto the set N_{f} of *finite point configurations*

$$N_{\mathrm{f}} = \{x \subset S : N(x) < \infty\}$$

Write $Y \sim \text{Poisson}(S, 1)$ and $A \subseteq N_{\mathrm{f}}$. By definition, we know

$$\mathbb{P}(X \in A) = \mathbb{E}[\mathbb{I}[Y \in A] f(Y)]$$

By Equation (2.2) we know

$$\mathbb{E}[\mathbb{I}[Y \in A] f(Y)] = \sum_{n=0}^{\infty} \frac{\exp(-|S|)}{n!} \int_S \cdots \int_S \mathbb{I}[\{x_1, \cdots, x_n\} \in A] f(\{x_1, \cdots, x_n\}) \mathrm{d}x_1 \cdots \mathrm{d}x_n$$

and thus

$$\mathbb{P}(X \in A) = \sum_{n=0}^{\infty} \frac{\exp(-|S|)}{n!} \int_S \cdots \int_S \mathbb{I}[\{x_1, \cdots, x_n\} \in A] f(\{x_1, \cdots, x_n\}) \mathrm{d}x_1 \cdots \mathrm{d}x_n \tag{2.3}$$

which gives us an expression for the density of an arbitrary process on a finite space with respect to the unit Poisson process.

Chapter 3

Summary Statistics for Point Processes

Here we introduce the key summary statistics describing the first and second-order characteristics of a point process as well as methods for estimating them. We will introduce more formally the intensity function, along with the important Ripley's K-function and the corresponding L-function. We will begin by introducing them in the case of a homogeneous intensity function before generalising them to inhomogeneous and multivariate processes.

In all instances of this chapter, we will take our space S to be $\mathbb{R}^d$, although the presented definitions extend readily to more general spaces.

3.1 General moment measures

Let us begin by introducing moment measures in generality. Consider again the situation of a point process X on S. For each $n \in \mathbb{N}$, we define the nth order moment measure $\mu^{(n)}$ to be

$$\mu^{(n)} = \mathbb{E}\left\{ \sum_{u_1,\cdots,u_n \in X} \mathbb{I}\left[(u_1,\cdots,u_n) \in B\right] \right\}, \quad B \subseteq S^n$$

where $S^n = S \times \cdots \times S$, with n terms in the product. Notice that if we write $B \subseteq S^n$ as

$$B = B_1 \times \cdots \times B_n, \quad B_i \subseteq S, \text{ for } i = 1,\cdots,n$$

then we can say

$$\sum_{u_1,\cdots,u_n \in X} \mathbb{I}\left[(u_1,\cdots,u_n) \in B\right] = \prod_{i=1}^{n} N(B_i)$$

in terms of the counts variables $N(B_1),\cdots,N(B_n)$. This follows immediately from the observation that $N(B_1)N(B_2)$ is the number of ordered pairs (u,u') of points in X, and extending the reasoning to the product of n counting variables. As such, we see that $\mu^{(n)}$ determines the nth moments of these variables

$$\mu^{(n)}(B_1 \times \cdots \times B_n) = \mathbb{E}\left\{ \prod_{i=1}^{n} N(B_i) \right\}, \quad B_i \subseteq S$$

In this thesis, we will only be focusing on first and second-order characteristics as a means to describe our point processes. The reason for this restriction is that they are sufficient to capture the multi-scale behaviour that we look to describe.

3.2 First-order characteristics

3.2.1 Intensity functions

We have already encountered the first-order moment measure $\mu^{(1)} = \mu$ of a process under the name of an *intensity measure*. Recall the concept of an *intensity function*, λ, defined as

$$\mu(B) = \int_B \lambda(\xi)\mathrm{d}\xi$$

and that if λ is constant, the process is called *homogeneous*.

We now look at ways to estimate the intensity function of a process in both homogeneous, inhomogeneous and multivariate settings.

3.2.2 Estimation of homogeneous intensity functions

In the case of homogeneity, estimation of the intensity function is trivial. We take inspiration from the sample mean in non-spatial statistics by defining an estimator $\bar{\lambda}$ of λ by

$$\bar{\lambda} = \frac{N(W)}{|W|}$$

for an observation window W. The unbiasedness of this estimator is immediate:

$$\mathbb{E}(\bar{\lambda}) = \frac{\mathbb{E}(N(W))}{|W|} = \frac{\lambda|W|}{|W|} = \lambda$$

where we have used that

$$\mathbb{E}(N(W)) = \mu(W) = \int_W \lambda \mathrm{d}\xi = \lambda|W|$$

since λ is constant.

3.2.3 Estimation of inhomogeneous intensity functions

The estimation of an inhomogeneous intensity function is more complicated, and we will focus on a nonparametric method using *kernel estimation*.

Kernel based estimates provide a smoothed estimate of the intensity function, as opposed to other methods not outlined here. To understand a kernel intensity estimate, we use an analogy given by Baddeley et al. (2015): think of placing a square of chocolate on each data point and using a hair dryer to slightly melt the squares. The resulting surface formed represents the undulating intensity function, with higher, less melted parts being regions of high intensity and vice versa.

Define a kernel estimator $\tilde{\lambda}_b$ of the intensity function λ of a point process X by

$$\tilde{\lambda}_b(u) = \sum_{\eta \in x} \frac{\kappa_b(u-\eta)}{e_b(u)} \tag{3.1}$$

for any u in our observation window W. Here $\kappa_b(u)$ is a *kernel function* with *bandwidth* b and $e_b(u)$ is an *edge correction*

$$e_b(u) = \int_W \kappa_b(u-v)\mathrm{d}v$$

We will discuss the choice of kernel function, bandwith and edge correction seperately as each plays an important role in the quality of the subsequent intensity estimate.

Kernel functions

Although there are numerous choices for a kernel function, we will use the *isotropic Gaussian kernel* with bandwith b throughout this thesis. The reason for this decision is that it falls outside the scope of this investigation to dwell on the choice of kernel function in the context of intensity estimation and this kernel is the default option for the `spatstat` package. The isotropic kernel function takes the form

$$\kappa_b(u) = \frac{1}{\sqrt{2\pi}b} \exp\left(-\frac{u^2}{2b^2}\right)$$

Bandwidth

Bandwidth is simply another term for the standard deviation of our kernel and appropriate selection of this value is vital in obtaining a good estimate of the intensity (Baddeley et al. 2015). The bandwidth can be thought of as controlling the *smoothing* of an estimate. As such, selecting too small a bandwidth will produce a "sharp" and overly rough estimate, whereas a value that is too large will blur regions of high intensity with those that are lower. This effect is clearly seen in Figure 3.1, where the increasing bandwidth results in an increasingly smooth estimate of the intensity of the presented point pattern.

There are a number of methods that can be used to select an appropriate bandwidth. Two common choices are Diggle and Berman's mean squared error cross-validation method (Diggle 1985) and the likelihood cross-validation method (Baddeley & Nair 2012). To decide which method is appropriate, it is important to analyse plots of the subsequent intensity estimates to assess whether the output is reasonable.

It is worth noting that kernel estimators are in general slightly biased (Baddeley et al. 2015). To demonstrate this, we need to introduce Campbell's theorem. For a simple proof, see, for example, Baddeley (2006).

Theorem 3.1: Campbell's theorem

Let X be a point process on a space S with intensity measure μ and $f : S \to \mathbb{R}$ be a measurable function. Then the random sum

$$T = \sum_{x \in X} f(x)$$

is a random variable with expectation given by

$$\mathbb{E}\left\{\sum_{x \in X} f(x)\right\} = \int_S f(x)\mu(\mathrm{d}x)$$

In the case that X is a point process in $S = \mathbb{R}^d$ with intensity function λ, we obtain

$$\mathbb{E}\left\{\sum_{x \in X} f(x)\right\} = \int_S f(x)\lambda(x)\mathrm{d}x$$

Note that our estimator $\tilde{\lambda}_b$ is precisely of this form, thus Campbell's theorem gives

$$\mathbb{E}(\tilde{\lambda}_b(u)) = \int_S \frac{\kappa_b(u-\eta)}{e_b(u)}\lambda(\eta)\mathrm{d}\eta$$

This, in general, will not be equal to λ, thus demonstrating the unbiasedness.

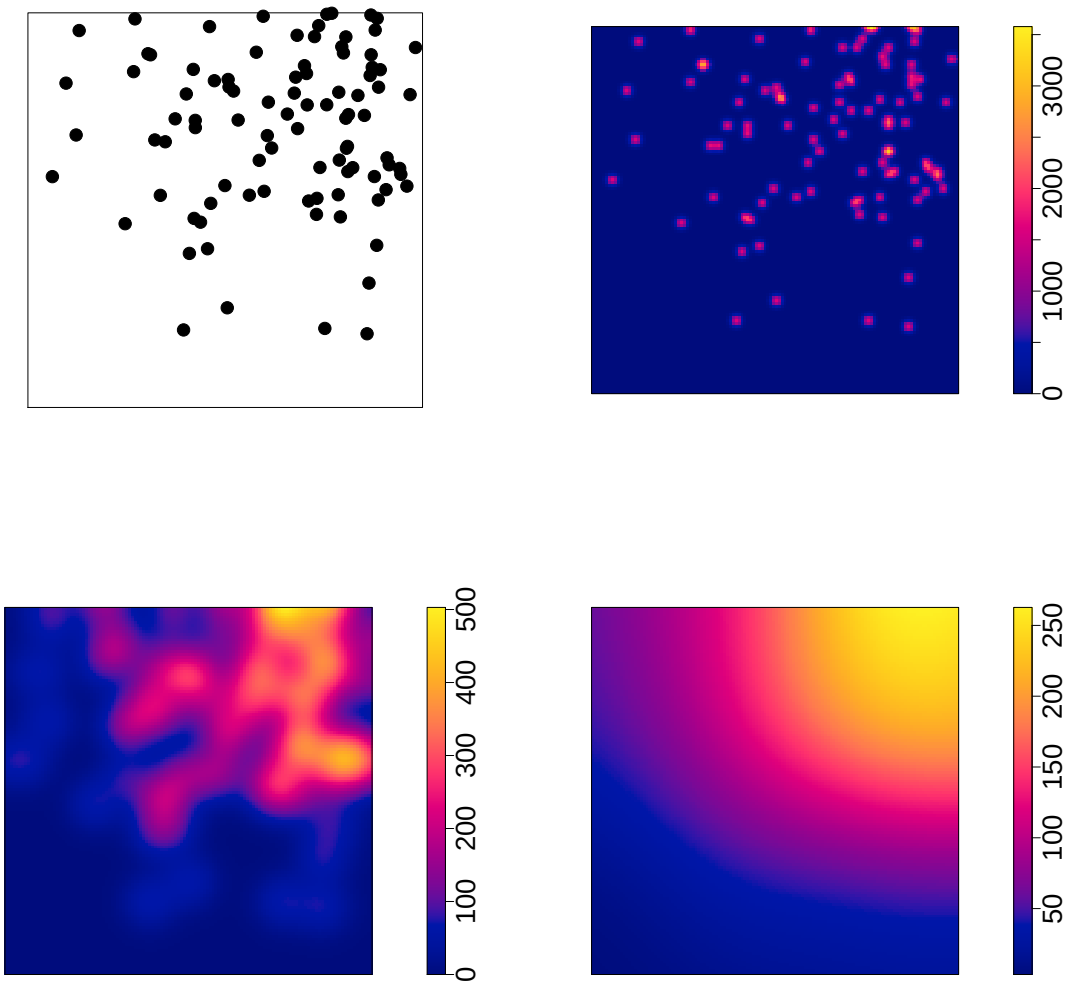


Figure 3.1: The top left plot shows an example plot of an inhomogeneous Poisson process, and then the top right, bottom left and bottom right plots show the intensity estimates using the isotropic Gaussian kernel and bandwidths of 0.01, 0.05 and 0.2, respectively.

Edge corrections

An edge correction has been introduced into the estimator in (3.1) to combat the problem of *edge effects*. Edge effects can be thought of as issues associated with our observation window being finite. In particular, points outside of our window will not contribute to the sum in the computation of our estimator $\tilde{\lambda}_b(u)$, and so we need to adjust so that estimates of the intensity at points u that are close to the boundary receive appropriate values.

3.3 Second-order characteristics

The first-order moment measure characterises the expected value of our counting measure $N(B)$ for some $B \subseteq S$. We are now interested in the variance of the count

$$\text{var}\{N(B)\} = \mathbb{E}\left\{N(B)^2\right\} - \mathbb{E}\left\{N(B)\right\}^2$$

and the covariance of two counts

$$\text{cov}\{N(B_1), N(B_2)\} = \mathbb{E}\left\{N(B_1)N(B_2)\right\} - \mathbb{E}\{N(B_1)\}\mathbb{E}\{N(B_2)\}$$

for $B_1, B_2 \subseteq S$.

We begin by exploring the second-order characteristics of point processes that are univariate and homogeneous, as this will allow us to gain heuristic insight into the definitions of key summary statistics that will then be extended into the multivariate and inhomogeneous setting. When unspecified, a process should be assumed to be univariate and homogeneous for the remainder of this chapter.

3.3.1 Ripley's K-function

Consider centring a disk of radius r at a point within our pattern and counting the number of points that fall within it. A summary statistic of such an operation would be useful in so far as it would convey key information on the pairwise distances of points within the pattern. Ripley (1976) was the first to develop such a statistic, and it is named in his honour as *Ripley's K-function*.

Definition 3.1: Ripley's K-function

Let X be a stationary point process on S with intensity $\lambda > 0$ and write $b(x, r)$ to denote the ball in $\mathbb{R}^d$ of radius r centred at a point x in the pattern. Then, for any non-negative r, given some observation window $W \subset S$, *Ripley's K-function* is defined as

$$K(r) = \frac{\mathbb{E}\left\{\sum_{x \in X \cap W} N\left(b(x, r) \setminus x\right)\right\}}{\lambda \mathbb{E}\{N(W)\}} \tag{3.2}$$

We see that $N(b(x, r) \setminus x)$ counts the number of points that lie within a distance r of x, excluding the actual point x. The expectation of this value summed over all possible x in the observation window is then normalised by the expected total count in the window multiplied by the intensity. This normalisation allows for comparison of values between datasets (Baddeley et al. 2015). Heuristically, $\lambda K(r)$ gives the expected number of points lying less than a distance r apart (conditional on the first point falling in W) divided by the expected number of points in W.

3.3.2 L-function

A minor alteration of Ripley's K-function that stabilises its variance as a function was soon proposed. This variant is known as the *L-function* (Besag 1977). Let ω_d denote the volume of the unit ball in $\mathbb{R}^d$ with ω_d

$$\omega_d = \frac{\pi^{d/2}}{\Gamma\left(1 + d/2\right)}$$

The *L-function* for a point process X on $\mathbb{R}^d$ is given by

$$L(r) = \left(\frac{K(r)}{\omega_d}\right)^{1/d}$$

3.3.3 K and L-functions in the Poisson case

As we have alluded to already, the homogeneous Poisson process is often taken as a reference point for these processes. The main reason for this is that a homogeneous Poisson process is

one of the few instances in which an exact formula for $K(r)$ (and hence $L(r)$) can be written down.

In the case of a homogeneous Poisson process in $\mathbb{R}^d$, we have that $\mathbb{E}\{N(b(x,r) \setminus x)\}$ will simply be the intensity function multiplied by the Lebesgue measure of the ball (minus the zero-measure point x), namely $\lambda\omega_d r^d$. Furthermore, since all points are considered independent and identically distributed,

$$\begin{aligned}\mathbb{E}\left\{\sum_{x\in X\cap W} N(b(x,r)\setminus r)\right\} &= \mathbb{E}\left\{\mathbb{E}\left[\sum_{j=1}^{N(W)} N(b(x_j,r)\setminus x_j)|N(W)\right]\right\}\\ &= \mathbb{E}\left\{N(W)\lambda\omega_d r^d\right\}\\ &= \lambda^2|W|\omega_d r^d\end{aligned}$$

It then follows that

$$K(r) = \omega_d r^d$$

for a homogeneous Poisson process in $\mathbb{R}^d$. We can then deduce easily that the L-function will be $L(r) = r$, which will of course produce a straight-line through the origin in a plot of $L(r)$ against r. It is for this reason that we will be mostly dealing with the L-function: ease of graphical interpretation.

3.3.4 Inhomogeneous K and L-functions

Stationarity is assumed in the derivation of the K-function, and we would like to develop an analogue for the case that the we are unable to assume that the process is stationary.

Baddeley, Møller & Waagepetersen (2000) developed such an analogue, provided certain assumptions are placed on the process. Let X be a point process on $S \subseteq \mathbb{R}^d$ with intensity function λ and $\mathcal{B}_0$ be the class of bounded Borel sets in $\mathbb{R}^d$. For any $A, B \in \mathcal{B}_0$ define

$$M(A,B) = \mathbb{E}\left\{\sum_{\xi\in X\cap A}\sum_{\eta\in X\cap B}\frac{1}{\lambda(\eta)\lambda(\xi)}\right\}$$

We call X *second-order intensity-reweighted stationary* if $M(A,B) = M(A+y, B+y)$ for all $y \in \mathbb{R}^d$ (Baddeley et al. 2000).

Let X now no longer be necessarily homogeneous, but instead be second-order intensity-reweighted stationary. We can then define an *inhomogeneous K-function* for X, the idea behind which is to weight each point x_i in our pattern by $1/\lambda(x_i)$, with the weight being thought of as accounting for the spatial inhomogeneity. We define the inhomogeneous K-function for X, K_{inhom}, as

$$K_{\text{inhom}}(r) = \mathbb{E}\left\{\sum_{x_j\in X}\frac{1}{\lambda(x_j)}\mathbb{I}[0 < \|u - x_j\| \le r]\,\middle|\, u \in X\right\}$$

The assumption that the process is second-order intensity-reweighted stationary says that this is independent of u, and so the interpretation is the same as in the homogeneous case. In fact, note that if $\lambda(u)$ is constant, then this reduces to the usual K-function. Furthermore, in the case that X is an inhomogeneous Poisson process, we have $K_{\text{inhom}}(r) = \omega_d r^d$.

The *inhomogeneous L-function*, L_{inhom}, is defined analogously to the homogeneous case as

$$L_{\text{inhom}}(r) = \left(\frac{K_{\text{inhom}}(r)}{\omega_d}\right)^{1/d}$$

3.3.5 Multivariate K and L-functions

Lotwick & Silverman (1982) proposed a natural extension to the K-function for the case of multivariate point patterns. We will only present the statistics here in the homogeneous case of a bivariate process (i.e. two homogeneous point processes that may or may not be dependent), but there are analogous forms for the case of inhomogeneity and processes that are trivariate or higher (see Baddeley et al. (2015), for example). We don't discuss the forms of these statistics as it would result in an overly-lengthy and unnecessary discussion. The homogeneous, bivariate case is only presented to provide intuition for how a multivariate estimator would work.

Let $X = (X_1, X_2)$ be a bivariate, homogeneous point process. The *bivariate homogeneous K-function*, K_{ij}, gives the expected number of points of subprocess j that lie within a distance r of points of subprocess i, standardised by the intensity of points of type j. In particular, we have

$$K_{ij}(r) = \frac{1}{\lambda_j}\mathbb{E}\left\{\sum_{k=1}^{N(X_j)} \mathbb{I}[0 \leq \|u - x_k\| \leq r] | u \in X_i\right\}$$

where X_i and X_j are the subprocesses corresponding the points of type i and j, respectively. $K_{ii}(r)$ has precisely the same interpretation as the standard K-function and, in particular, it is consistent with $K_{ii}(r) = \pi r^2$ for a Poisson processes in two-dimensions (Baddeley et al. 2015).

From here, we can easily define the *bivariate homogeneous L-function*, L_{ij}, as

$$L_{ij}(r) = \left(\frac{K_{ij}(r)}{\omega_d}\right)^{1/d}$$

We will refer to these functions either as the bivariate K-function and bivariate L-function, respectively. As has been said above, the inhomogeneous and general multivariate forms are natural extensions.

3.4 Estimation of second-order characteristics

Estimation of the K and L-functions differs between the homogeneous, inhomogeneous and multivariate cases, but we will only present the estimates in the case of the homogeneous, univariate setting. The reason for this restriction is that the estimates for the other settings are simple analogues of the estimators presented, and a detailed discussion would draw away from the main focus of this thesis.

3.4.1 Estimation in the homogeneous case

Equation (3.2) gave us an expression for the value of Ripley's K-function. However, we of course need a way to estimate this, as it is commonly intractable. To do so, we simply replace the numerator and denominator by unbiased estimates to form (Baddeley et al. 2015)

$$\hat{K}(r) = \frac{\sum_{x_i \in x \cap W} N(b(x_i, r) \setminus x_i)}{\bar{\lambda} N(x \cap W)} \tag{3.3}$$

where $\bar{\lambda} = N(x \cap W)/|W|$. Note that in essence we counting the number of points that lie within a distance r of $x_j \in x \cap W$, summing this all $x_j \in x \cap W$, and then standardising. Such a scenario is sketched in Figure 3.2, where red points indicted those that fall within a distance r of the white point under consideration.

The immediate problem that presents itself with this estimate is how idealised the situation in Figure 3.2 is. Specifically, we very rarely have access to points outside of the observation window, as data is only usually collected within W. As such, we are going to be undercounting in some sense by not having access to data outside of W. This is what is known as an *edge effect*. We attempt to solve this problem with a technique aptly named an *edge correction*. There are several possible choices of edge correction, but two popular choices that we will consider in this thesis are the *border correction* and the *isotropic correction*.

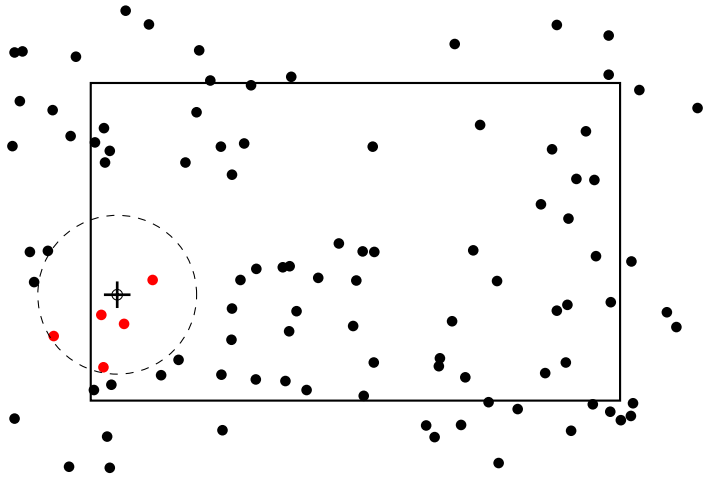


Figure 3.2: An example plot of an indealised situation in which the observation window (the black rectangle) is only a subset of the space on which we have collected data. The points in red fall within a radius r of the relevant point x_i drawn in white.

Border correction

In the border correction, we restrict our attention to those points who lie more than a distance of r from the boundary. In particular, if we denote the boundary of the observation window W as ∂W and write $d(x_i, \partial W)$ to be the shortest distance from a point $x_i \in W$ to ∂W, we consider only those points that lie in $W_{\ominus r}$, where

$$W_{\ominus r} := \{u \in W : d(u, \partial W) \geq r\}$$

Such a set-up is shown in Figure 3.3, where, for a distance r, we would only consider those points within the grey rectangle, which corresponds to $W_{\ominus r}$. For these points, we have $b(x_i, r) \cap W = b(x_i, r)$ as any point within a distance r of x_i must lie within W by the definition of such points. Note that we still count points that lie outside the grey rectangle when considering points within a distance r of x_i. As such, we compute the estimate

$$\hat{K}_{\text{bord}}(r) = \frac{\sum_{i=1}^{N(x)} \mathbb{I}\{d(x_i, \partial W) \geq r\} N(b(x_i, r) \setminus x_i)}{\bar{\lambda} \sum_{i=1}^{n} \{d(x_i, \partial W) \geq r\}} \tag{3.4}$$

It is intuitive that the quality of this estimate increases with the number of points except for in peculiar data sets where, for example, all of the points are concentrated around the perimeter of W. This is reasonable because for a point pattern with relatively few points, we will be throwing away a large proportion of the data when we restrict to the smaller observation window used by the method. For example, consider when $W = [0,1] \times [0,1]$ and we take $r = 0.2$. This leaves us with the window $[0.2, 0.8] \times [0.2, 0.8]$, which removes over 60% of the available data. For a point pattern with few points, this will lead to a poor estimate. As such, it is only sensible to use this estimate when the number of data points in the pattern is large and the data isn't concentrated near ∂W (Baddeley et al. 2015).

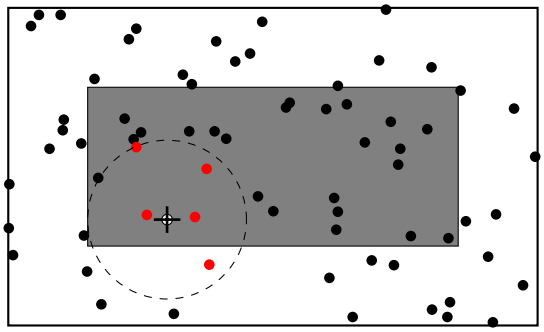


Figure 3.3: A visual display of the border edge correction. The grey rectangle corresponds to the locus of all points a distance of at least r from ∂W (namely, $W_{\ominus r}$) and the dotted circle contains all those points a distance r from the white point under consideration.

Isotropic correction

Consider two points x_i, x_j in a given pattern. If the circle $b(x_i, \|x_i - x_j\|)$ isn't fully contained within W, then it is possible that there are points within the circle that won't be observed. Allow $|\cdot|$ to denote a length so that, in particular, we have $|\partial b(x_i, \|x_i - x_j\|) \cap W|$ as being the length of the perimeter of $b(x_i, \|x_i - x_j\|)$ contained within W. To see this visually, consider Figure 3.4 where the red section of the dashed circle's circumference would correspond to $\partial b(x_i, \|x_i - x_j\|) \cap W$.

Under the assumption of isoptropy, we can argue that any x_j is equally likely to fall anywhere along $\partial b(x_i, \|x_i - x_j\|)$. As such, it seems reasonable to weight our estimate by the probability that x_j lies in W. Specifically, by the probability

$$p(x_i, x_j) = \frac{|\partial b(x_i, \|x_i - x_j\|) \cap W|}{|\partial b(x_i, \|x_i - x_j\|)|}$$

In this way, we form Ripley's isotropic correction estimator (Ripley 1976)

$$\hat{K}_{\text{iso}}(r) = \frac{1}{\bar{\lambda} N(x)} \sum_{i=1}^{N(x)} \sum_{j=1, j\neq i}^{N(x)} \mathbb{I}\{\|x_i - x_j\| \leq r\} \frac{1}{p(x_i, \|x_i - x_j\|)} \tag{3.5}$$

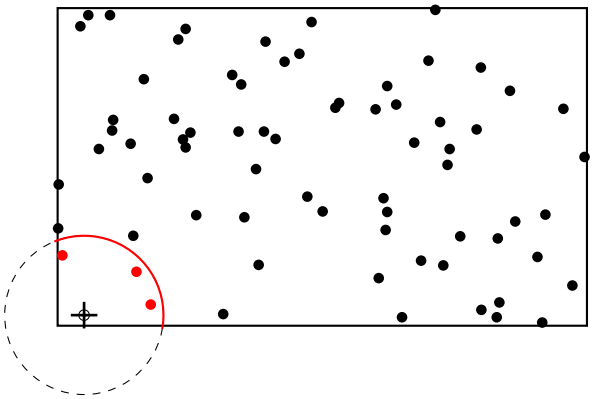


Figure 3.4: A visualisation of applying the isotropic edge correction. The dotted circle is contains all those points a distance r from the white point under consideration and the red section of the circle corresponds to the length that intersects the observation window.

3.5 Interpretation of second-order characteristics

We will be placing much emphasis on the use of these two summary statistics to analyse the small and large-scale behaviour of point patterns. As such, it is vital that we understand how to interpret plots of these statistics.

We will be considering the L-function primarily due to its stabilised variance in comparison with the K-function. As has been discussed, in all three settings we have presented, the Poisson process has $L(r) = r$. Since a homogeneous Poisson process corresponds to complete spatial randomness, we can thus use deviations in a plot of $L(r)$ against r (or, more commonly, of $L(r) - r$ against r) as indications of divergence from total randomness. Specifically, observing $L(r) < r$ ($L(r) > r$) says that we observe less (more) points than expected lying within the circle of radius r centred at the point in question which in turn implies between-point inhibition (aggregation) at that scale.

It is important, however, to understand that the homogeneous statistics assume *stationarity*, and thus it is possible to observe what we will call *first-order clustering*, where points are positively associated, but only due to variations in the density of points, perhaps caused by some unobserved covariate, and not due to second order interactions. As an example, consider the plot in Figure 3.5, where the right-panel plot of $L(r) - r$ (using the isotropic correction) would lead us to believe that there is a positive second-order dependence between the points. However, inspection of the pattern in the left-panel shows clearly that this positive association is caused by a region of high density in the bottom-left corner, perhaps caused by elevation or moisture concentration.

The takeaway here is that the K and L-functions are not always enough by themselves to detect second-order dependence. However, in the models that will be developed in due course, we will only consider stationary processes, and thus any apparent clustering will be due to second-order dependence between the points in the process and it will be sufficient to only consider the K and L-functions. For a more detailed discussion of this, the reader is referred to Baddeley et al. (2015).

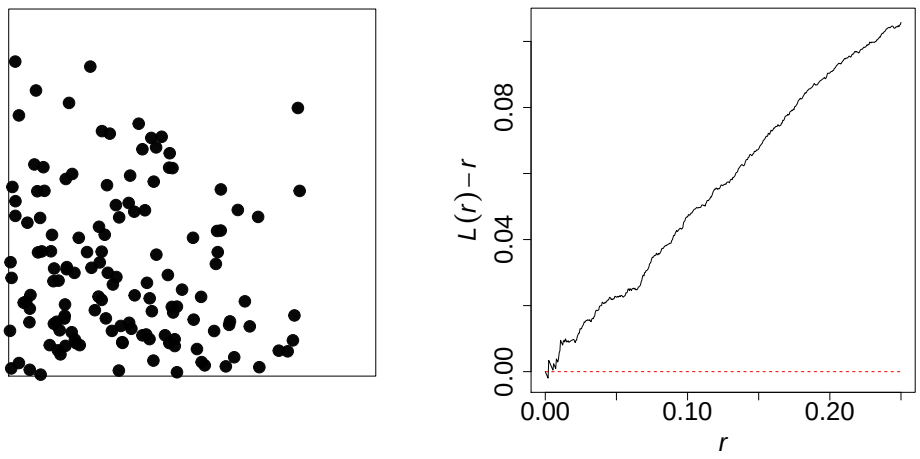


Figure 3.5: An example of when an L-function plot would imply a second-order dependence structure that causes clustering, but it is in fact due to density variations caused by an underlying covariate.

Chapter 4

Cox and Markov Point Processes

In this chapter, we introduce two fundamental types of spatial point process called *Cox* process and *Markov* processes. These processes underpin the novel model we introduce later in this thesis, and thus a thorough understanding is key.

4.1 Random fields and covariance functions

Cox processes are dependent upon a *random field* that defines its correlation structure. As such, before discussing Cox processes, it is vital that we investigate random fields.

Imagine that you are looking at the surface of the ocean and trying to predict the height of the water above an idealised flat ocean floor at different locations. We could feasibly model the height of the surface at each point in some given window by a random variable that oscillates about some midpoint that corresponds to an "average resting height" when there are no waves. We could not, of course, assume that these random variables would be independent: we would expect the height at a given point to be closely related to the height at its neighbouring locations. This dependence, or correlation, would presumably decrease with radial distance, and we could specify this to be the case in our model. This is the effect that a *random field* tries to capture.

A random field can be thought of as a set of correlated random variables, with each random variable corresponding to a different spatial location. We further specify a *covariance function*, which describes how the random variables at different spatial locations relate to one another.

Definition 4.1: Random field

Given a probability space $(\Omega, \mathcal{F}, \mathbb{P})$, we can define an *m-variate random field* $Z(x)$ as a collection of m-dimensional random vectors indexed by $x \in \mathbb{R}^d$ mapping from the space $(\Omega, \mathcal{F}, \mathbb{P})$ to $\mathbb{R}^m$. In other words, it is the set

$$\left\{ Z(x;\omega) = (Z_1(x;\omega), \cdots, Z_m(x;\omega))^T \in \mathbb{R}^m : x \in \mathbb{R}^d, \omega \in \Omega \right\}$$

We will often suppress the ω notation, and simply write $Z(x)$ to be the m-dimensional random vector at the spatial location $x \in \mathbb{R}^d$. We will also refer to the random field itself as Z, and the difference in interpretation will be made obvious.

A random field has an *expectation function* $\mu(x_1) = \mathbb{E}(Z(x_1))$ and *covariance function* $C(x_1, x_2) =$

$\text{cov}(Z(x_1), Z(x_2))$, for $x_1, x_2 \in \mathbb{R}^d$. These, of course, are functions of spatial location as we will have different random variables at each point. In the case of a *Gaussian random field*, which will be discussed in detail shortly, having these two functions will completely describe the random field in much the same way that the mean and variance of a normal random variable completely determines that distribution.

Construction of the covariance function $C(x_1, x_2)$ is a non-trivial task as it must be built subject to certain constraints in order to be valid. As has been mentioned, we can think of a covariance function (or *kernel*) as capturing the similarity between two points in our space; that is, it provides a measure of how the field changes between two points. For any $x_1, x_2 \in \mathbb{R}^d$, the covariance function $C(x_1, x_2)$ will be a *matrix-valued function* taking the form $C(x_1, x_2) = [C_{ij}(x_1, x_2)]_{i,j=1}^m$, where

$$C_{ij}(x_1, x_2) = \text{cov}(Z_i(x_1), Z_j(x_2))$$

We call the covariance function *stationary* if it is a function of $x_1 - x_2$ only and *istropic* if it is a function of $\|x_1 - x_2\|$, for any $x_1, x_2 \in \mathbb{R}^d$. Specifically, the function is stationary if

$$C_{ij}(x, x + h) = C_{ij}(0, h) =: C_{ij}(h), \quad x, h \in \mathbb{R}^d$$

and isotropic if

$$C_{ij}(h_1) = C_{ij}(h_2), \quad h_1, h_2 \in \mathbb{R}^d$$

whenever $\|h_1\| = \|h_2\|$, so that the function is rotationally invariant.

In order for $C(x_1, x_2)$ to be valid, we require $C : \mathbb{R}^d \times \mathbb{R}^d \to M_{m\times m}$ to be *non-negative definite*, where $M_{m\times m}$ is the space of $m \times m$ real-valued matrices. Non-negative definiteness is a requirement on quadratic forms. In particular, given any set of points $x_1, \cdots, x_n \in \mathbb{R}^d$ and any $v \in \mathbb{R}^{mn}$, we require that $v^T \Sigma v \geq 0$, where Σ is the covariance matrix of the random vector $(Z(x_1), \cdots, Z(x_n))^T \in \mathbb{R}^{mn}$ given explicitly as

$$\Sigma = \begin{pmatrix} C(x_1, x_1) & C(x_1, x_2) & \cdots & C(x_1, x_n) \\ C(x_2, x_1) & C(x_2, x_2) & \cdots & C(x_2, x_n) \\ \vdots & \vdots & \ddots & \vdots \\ C(x_n, x_1) & C(x_n, x_2) & \cdots & C(x_n, x_n) \end{pmatrix}$$

The non-negative definiteness of C will ensure the same of Σ. There are a number of common covariance functions that are worth considering, and we will give three popular choices. All the functions we will consider are isotropic, and thus take $h = \|x_1 - x_2\|$ in the following:

1. The *exponential* covariance function defines the covariance between two points as

   $$\text{cov}(x_1, x_2) = \sigma^2 \exp(-h/s)$$

   We call s the *length-scale* of the kernel. Note that this function is infinitely differentiable, and thus it is very smooth. It has been argued by Stein that this smoothness is unrealistic for many physical scenarios (Stein 2012).

2. The *Gaussian* covariance function defines

   $$\text{cov}(x_1, x_2) = \sigma^2 \exp(-h^2/s^2)$$

   Note that it is very similar to the exponential covariance function, except that now we square the terms in the exponential.

3. The *Matérn* covariance function sets

$$\text{cov}(x_1, x_2) = \frac{2^{1-\nu}}{\Gamma(\nu)} \left(\frac{\sqrt{2\nu}h}{\ell} \right)^{\nu} K_\nu \left(\frac{\sqrt{2\nu}h}{\ell} \right)$$

for $\nu, \ell > 0$ and K_ν being a modified Bessel function. This covariance function is very flexible and thus popular in the literature. It is interesting to note that in the case $\nu = p + 1/2$ for $p \in \mathbb{N}$, we have

$$C(x_1, x_2) = \exp\left(-\frac{\sqrt{2p+1}h}{\ell} \right) \frac{\Gamma(p+1)}{\Gamma(2p+1)} \sum_{i=0}^{p} \frac{(p+i)!}{i!(p-i)!} \left(\frac{\sqrt{8p+4}h}{\ell} \right)^{p-i}$$

which is a very interesting case for machine learning (Rasmussen & Williams 2005).

A very important case is that of a *m-variate Gaussian random field*, which is defined simply as a m-variate random field $Z(x)$ indexed by $x \in \mathbb{R}^d$ for which $Z(x_1), \cdots, Z(x_n)$ is multivariate normal for every $(x_1, \cdots, x_n)^T \in \mathbb{R}^{nd}$. This field will be of particular importance in due course.

4.2 Cox processes

A Cox process is a generalisation of a Poisson process where we allow the intensity function to be a realisation of a random field (Cox 1955). These processes, originally coined as *doubly stochastic Poisson processes* by Cox, are models for aggregated or clustered processes where the aggregation or clustering is caused by some external factor (Møller & Waagpetersen 2004).

Taking the growth of trees in a rainforest as an example, we may believe that the concentration of a certain species is heavily dependent on the soil quality at each location. As such, we would say that the covariate "soil quality" controls the intensity function at a location, and thus the density of points in small regions around that point. If we had access to the soil quality data, we could use it to construct an inhomogeneous Poisson process. However, often we don't have access to this data, and we need some way of allowing the intensity function to be influenced by this unknown distribution. Cox processes offer a solution to this problem by allowing the intensity to vary randomly, where the random field over which it varies can be thought of as encoding the unknown influence of the covariate.

4.2.1 Definition of a Cox process

Definition 4.2: Cox process

Let $Z = \{(Z_1(x), \cdots, Z_m(x))^T : x \in \mathbb{R}^d\}$ be an m-variate random field where the mapping $\xi \in \mathbb{R}^d \to Z_i(\xi) \in \mathbb{R}$ is locally integrable (with probability 1) for $i = 1, \cdots, m$. If, conditional on a realisation $Z = (Z_1, \cdots, Z_m)^T$, $X_1, \cdots, X_m$ are independent Poisson processes with intensity functions $Z_1, \cdots, Z_m$, then $X = (X_1, \cdots, X_m)^T$ is an *m-variate Cox process driven by* Z on $\mathbb{R}^d$ (Møller & Waagpetersen 2004).

We will be very interested in the special case that $Z_i = \{Z_i(x) = \exp(Y_i(x)) : x \in \mathbb{R}^d\}$ with $Y = \{(Y_1(x), \cdots, Y_m(x))^T : x \in \mathbb{R}^d\}$ being a Gaussian random field. In this case, we call the subsequent Cox process a *log-Gaussian Cox process* for reasons that are hopefully clear.

In all settings that we will consider in application, we take $m = 1, d = 2$ or $m = d = 2$, so that we are considering either a univariate or bivariate log-Gaussian Cox process on $\mathbb{R}^2$. Figure

4.1 shows a realisation of a log-Gaussian Cox process plotted alongside it's corresponding L-function estimate computed using the isotropic correction. We see from both plots how the points cluster together on all scales (apart from a small region near $r = 0$ which is due to the estimation procedure).

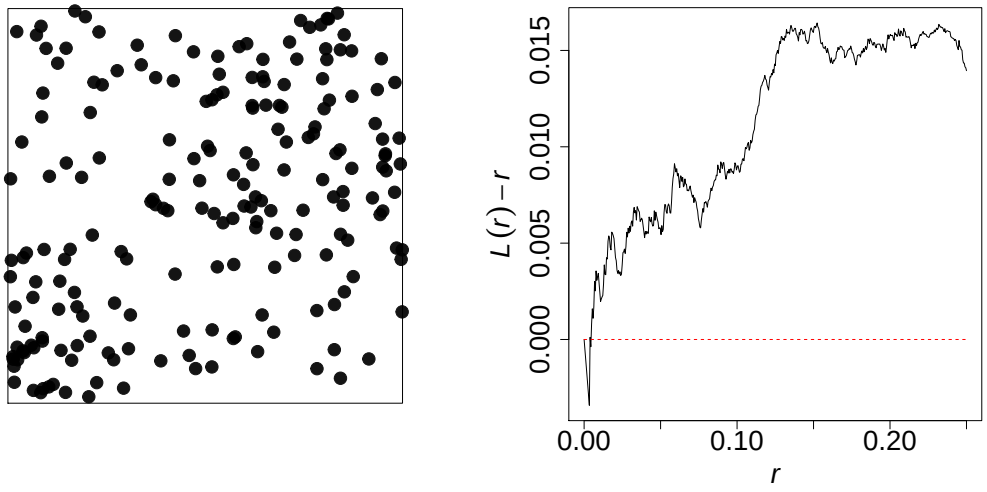


Figure 4.1: Here we show a realisation of a log-Gaussian Cox process on $[0,1]^2$ alongside a plot of $\hat{L}(r) - r$ using the isotropic correction. The field was generated using $\mu = 5$ and an exponential covariance function having $\sigma^2 = 2$ and $s = 0.3$. The bottom row is analogous for a Strauss process with $\gamma = 0.3, R = 0.03$ and $\beta = 100$.

4.2.2 Density function of a Cox process

Recall the definition of the intensity measure μ of a point process X on a space S

$$\mu(B) = \mathbb{E}\left\{N(B)\right\}, \quad B \subseteq S$$

As we know, there are no complications in defining the intensity of a multivariate point process, as we simply consider the intensity for each subprocess separately. As such, let X be univariate in the following discussion.

In the case that X is a Cox process with driving intensity Z, we can obtain this measure by conditioning on a realisation of Z. Specifically, we have

$$\mu(B) = \mathbb{E}\{N(B)\} = \mathbb{E}\left\{\mathbb{E}\left[N(B)|Z\right]\right\} = \mathbb{E}\left\{\int_B Z(\xi)\mathrm{d}\xi\right\} = \int_B \mathbb{E}\left\{Z(\xi)\right\}\mathrm{d}\xi$$

and so $\mathbb{E}(Z)$ is seen to be the intensity function of X. In a very similar way, we can derive the density $f(x)$ with respect to the unit Poisson process

$$\begin{aligned}
\mathbb{P}(X \in A) &= \mathbb{E}\left\{\mathbb{P}(X \in A|Z)\right\} \\
&= \mathbb{E}\left\{\sum_{n=0}^{\infty} \frac{\exp(-|S|)}{n!} \int_S \cdots \int_S \mathbb{I}[\{x_1, \cdots, x_n\} \in A] \exp\left(|S| - \int_S Z(\xi)\mathrm{d}\xi\right) \prod_{i=1}^{n} Z(x_i)\mathrm{d}x_1 \cdots \mathrm{d}x_n\right\} \\
&= \sum_{n=0}^{\infty} \frac{\exp(-|S|)}{n!} \int_S \cdots \int_S \mathbb{I}[\{x_1, \cdots, x_n\} \in A] \mathbb{E}\left\{\exp\left(|S| - \int_S Z(\xi)\mathrm{d}\xi\right) \prod_{i=1}^{n} Z(x_i)\right\} \mathrm{d}x_1 \cdots \mathrm{d}x_n
\end{aligned}$$

from which it follows that the density is given by

$$f(x) = \mathbb{E}\left\{\exp\left(|S| - \int_S Z(\xi)\mathrm{d}\xi\right)\prod_{\xi\in x} Z(\xi)\right\} \tag{4.1}$$

where the expectation is taken with respect to the random field. To extend to the multivariate setting, all we require is to take the product of the densities for each subprocess in the multivariate process, and simply normalise the result to obtain a density.

4.2.3 Validity of a multivariate Cox process

It is a non-trivial exercise to write down a valid multivariate Cox process. The reason for this is the requirement of having a non-negative definite covariance matrix in the underlying field. There are a number of ways to create a valid construction, but the method that we will use is that of the *linear model of coregionalisation*, which uses linear combinations of univariate covariance functions (Bourgault & Marcotte 1991). If we take $Z_1, \cdots, Z_k$ to be independent univariate Gaussian random fields with zero mean, unit variance and stationary correlation function $r_i(\cdot)$, then taking

$$Y_j(s) = \sum_{\ell=1}^{k} A_{j\ell} Z_\ell(s) + \mu_j(s)$$

with $A = [A_{ij}]_{i,j=1}^{m,k}$ being full-rank and $\mu_1, \cdots, \mu_m$ deterministic functions makes $Y_1, \cdots, Y_m$ well-defined random fields in the sense that they have a valid cross-covariance structure. In particular, we have

$$C_{ij}(s) = \sum_{\ell=1}^{k} A_{i\ell} A_{j\ell} r_\ell(s), \quad i,j = 1, \cdots, m, \quad s \in \mathbb{R}^d$$

In the bivariate case with which we are concerned, we have

$$C(s) = \begin{pmatrix} \sum_{\ell=1}^{k} A_{1\ell}^2 r_\ell(s) & \sum_{\ell=1}^{k} A_{1\ell}A_{2\ell} r_\ell(s) \\ \sum_{\ell=1}^{k} A_{1\ell}A_{2\ell} r_\ell(s) & \sum_{\ell=1}^{k} A_{2\ell}^2 r_\ell(s) \end{pmatrix} \tag{4.2}$$

Notice how the sign of $\sum_{\ell=1}^{k} A_{1\ell}A_{2\ell} r_\ell(s)$ determines whether the two fields are positively or negatively correlated.

4.3 Markov point processes

For a more detailed description of the material presented here, see Van Lieshout (2000).

A Markov point process is defined in much the same way that a Markov chain in time is defined, which will no doubt be a concept more familiar to the reader. Recall that for a discrete Markov chain $X_1, \cdots, X_n$, we have

$$\mathbb{P}(X_i = x_i | X_0 = x_0, \cdots, X_{i-1} = x_{i-1}) = \mathbb{P}(X_i = x_i | X_{i-1} = x_{i-1})$$

so that the "future" of the chain is independent of the "past" given the "present". There is no clear analogue of time's natural ordering in a spatial process, and so we state a symmetric analogue:

$$\mathbb{P}(X_i = x_i | \cdots, X_{i-1} = x_{i-1}, X_{i+1} = x_{i+1}, \cdots) = \mathbb{P}(X_i = x_i | X_{i-1} = x_{i-1}, X_{i+1} = x_{i+1})$$

Using the intuition of this analogue, a Markov point process is defined heuristically to be a point process that satisfies this property, whereby each point is influenced only by its neighbours (a concept to be made more precise shortly).

4.3.1 Papangelou conditional intensity

Recall the definition of the space $N^{\ell f}$ on which our spatial point process X on S takes values:

$$N^{\ell f} = \{x \subseteq S : N(B) < \infty \text{ for bounded } B \subseteq S\}$$

The *Papangelou conditional intensity* for a point process X with a density function f is given by

$$\lambda(u; x) = \frac{f(x \cup u)}{f(x)}$$

for $x \in N^{\ell f}$ and $u \in S \setminus x$. Note the minor abuse of notation here: we take $x \cup u$ to mean $x \cup \{u\}$, namely $x \cup u$ is the point pattern x with the addition of a point $u \in S$. Heuristically, we can think of $\lambda(u; x)$ as defining the conditional distribution of finding a point in an infinitesimal region centred at u, with the conditioning being on the rest of the process.

4.3.2 Markov point processes

Let us now formalise the concept of two points in a space being neighbours. Consider a point process X on a space S. Let $\sim$ be a symmetric and reflexive relation on S. An example of such a relation $\sim$ would be

$$u \sim v \quad \iff \quad \|u - v\| \leq r$$

for some distance r, where we are saying that $u \sim v$ if and only if they lie within a (Euclidean) distance r of one another. We define the points $u, v \in S$ to be *neighbours* if $u \sim v$. We can then define the *neighbourhood* $\partial(A)$ of a set $A \subseteq S$ to be

$$\partial(A) = \{u \in S : u \sim a \text{ for some } a \in A\}$$

Essentially, a neighbour relation $\sim$ defines a condition for whether points in a process will interact with one another. For example, in a *hardcore process*, points are not allowed to fall within a distance R of one another, and thus we would have that points u and v influence one another if and only if $\|u - v\| \leq R$, which is the example of a relation $\sim$ given above.

We call a finite point process X with density function f *hereditary* if

$$f(x) > 0 \text{ and } y \subseteq x \quad \Rightarrow \quad f(y) > 0$$

That is to say that a subset of any possible configuration must itself be possible. This concept works both ways in that $f(x) = 0$ implies that $f(y) = 0$ for $y \subseteq x$, so any subset of an impossible point pattern is itself impossible.

Definition 4.3: Markov point process

Let X be a point process on S which is specified by a density function $f(x)$ with respect to the distribution of the unit Poisson process on S. We call X a Markov point process with respect to the symmetric, reflexive relation $\sim$ on S if, for any $x \in N^{\ell f}$ such that $f(x) > 0$, we have

1. X is hereditary
2. for all $u \in S$, $\lambda(x; u)$, the Papangelou conditional intensity at u, depends only on u and $\partial(\{u\}) \cap x$.

The second condition here is a clear analogue of the defining property of Markov chains. Specifically, we see that the intensity at a point is dependent only on its immediate neighbours in x, where the neighbours are defined by $\sim$.

4.3.3 Pairwise interaction processes

Consider again the situation where X is a point process on a space S with $|S| < \infty$ with a density function f with respect to the standard Poisson process. In such a setting, the density function will take the form

$$f(x) \propto \prod_{\xi \in x} \phi(\xi) \prod_{\{\xi,\eta\} \subseteq x} \phi(\{\xi, \eta\})$$

(Møller & Waagpetersen 2004). For $u \notin x$, $f(x \cup u)$ will be $f(x)$ multiplied by every term of the form $\phi(u)\phi(\{u, \eta\})$ where $\eta \in x$. As such, the Papangelou conditional intensity at u is

$$\lambda(x; u) = \phi(u) \prod_{\eta \in x} \phi(\{u, \eta\})$$

Such an interaction process is termed *homogeneous* if $\phi(\xi)$ is constant and $\phi(\{\xi, \eta\}) = \phi_2(\|\xi - \eta\|)$ for $\phi_2 : (0, \infty) \to [0, \infty)$ (Møller & Waagpetersen 2004).

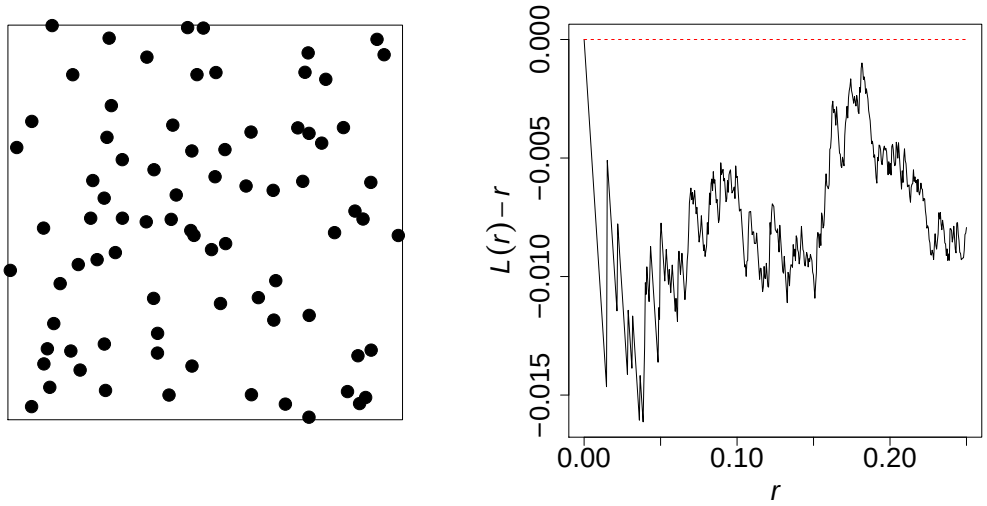


Figure 4.2: Here we show a realisation of a Strauss process on $[0, 1]^2$ with $\gamma = 0.3, R = 0.03$ and $\beta = 100$ alongside a plot of $\hat{L}(r) - r$ using the isotropic correction.

Arguably the most commonly used example of a pairwise interaction process is a *Strauss process* (Strauss 1975). A *Strauss process* takes

$$\phi_2(\|\xi - \eta\|) = \gamma^{\mathbb{I}[\|\xi - \eta\| \leq R]}$$

for $0 \leq \gamma \leq 1$ and $R > 0$, where we set $0^0 = 1$. This gives a density of the form

$$f(x) \propto \beta^{N(x)} \prod_{\{\xi,\eta\} \subseteq x} \gamma^{\mathbb{I}[\|\xi - \eta\| \leq R]}$$

where $\phi(\xi) = \beta > 0$ for all $\xi \in x$. We can see that

$$\lambda(x; u) = \beta \gamma^{N(x \cap b(u,R))}$$

Such a process forces inhibition of a strength controlled by γ on a scale controlled by R. Essentially, decreasing γ decreases the chances of there being a point within a radius R of any other points. One can think of this as modelling competition between species of tree, for

instance. There are plenty of other examples of pairwise interaction processes, but they are not pertinent to this thesis and, as such, will not be explored.

In Figure 4.2, we see a realisation of a Strauss process with $\beta = 100$, $\gamma = 0.3$ and $R = 0.03$ alongside its L-function estimate using the isotropic correction. From both plots we see the segregation on all scales, which is precisely the interaction between points that a Strauss process forces.

Chapter 5

Approximate Bayesian Computation and the Birth-Death Metropolis-Hastings Algorithm

We now look to discuss methods for simulating realisations from models of spatial point processes and, subsequently, techniques for fitting such models to real data. Fitting these models to data rests on the ability to make inference on the parameters within the model we look to fit, and thus we need to develop a framework for this estimation. We will adopt a Bayesian approach, whereby we place a prior on each parameter and seek to obtain its posterior distribution.

For the vast majority of models, the normalising constant in the corresponding density function makes the likelihood intractable, and thus sampling from the posterior of the parameters cannot be achieved exactly. As such, we turn our attention to *approximate Bayesian computation (ABC)*, which is a framework that allows for approximate sampling from a parameter's posterior.

5.1 Principle of ABC

Before describing the principle of ABC, we must first introduce some notation. Let the (possibly vector) parameter of interest be θ and write x_{obs} for the observed point pattern. We will be simulating observations from our model for a given value of $\theta = \theta_s$, and so let x_s be the simulated pattern for such a θ_s.

In standard Bayesian inference, given some data x_{obs} and parameter θ with prior $\pi(\theta)$, we look to compute the *posterior distribution* of θ. Specifically, we try to evaluate

$$p(\theta|x_{\text{obs}}) \propto p(x_{\text{obs}}|\theta)\pi(\theta)$$

where $p(x_{\text{obs}}|\theta)$ is the *likelihood* of the data. In practice, it is commonly the case that this likelihood is unavailable in closed form or requires a very computationally intensive procedure to evaluate, and the question arises of how to sample effectively from the posterior in this case. The solution proposed by ABC is to utilise modern computing techniques to replace the computation of the likelihood with the computation of a *suitable comparison* between simulated and true data in an effort to form an *approximate posterior distribution*.

The key insight that underpins ABC is the realisation that

$$p(\theta|x_{\text{obs}}) = \lim_{\varepsilon \to 0} p(\theta|d(T(x_{\text{obs}}), T(x_s)) < \varepsilon)$$

where $d(\cdot,\cdot)$ is an appropriately chosen *discrepancy measure* that characterises the distance of the sampled data from the true data in some appropriate sense and $T(x) = (T_1(x), \cdots, T_m(x))$ is a vector of summary statistics. In other words, this says that in the limit that the distance between the summary statistics of our sampled and simulated patterns approaches zero, the statistical distribution of the true posterior and our *approximate posterior* become the same.

There is no explicit need to compute a discrepancy measure based on summary statistics as opposed to simply the data itself. The main reason we choose to base our measure on summary statistics is that it increases the chance of accepting a sample without having to increase ε when running an ABC procedure.

It is important to carefully choose T. We can say with near certainty that a sufficient statistic won't be available in the settings where ABC is to be applied, and thus it is important to investigate suitable alternative choices. Fearnhead & Prangle (2011) argue that the optimal choice of $T(x)$ is $\mathbb{E}(\theta|x)$, and discuss a linear regression approach to construct an estimate for it through a *pilot run*. Essentially, one generates k_{pilot} samples $\left\{\left(\theta^{\text{pilot},i}, x^{\text{pilot},i}\right)\right\}_{i=1}^{k_{\text{pilot}}}$ and then uses them to fit the following linear model

$$\mathbb{E}[\theta_j|x] \approx \theta_j(x) := \alpha^j + \beta^{j^T}(T(x) - T(x_{\text{obs}}))$$

We can then base the discrepancy measure on this linear model and the subsequent fitted values, $\hat{\alpha}^j$ and $\hat{\beta}^j$. The approach we will pursue will be discussed in later chapters.

Clearly, the choice of ε and the discrepancy measure will determine the accuracy of this approximation. A smaller value of ε improves the accuracy of the approximation, but it may take a very long time to obtain a sample that achieves the required tolerance. As such, there is a trade-off between accuracy and computational time that must be made.

5.2 ABC rejection sampling

The most simple method for implementing ABC is *ABC rejection sampling* (Pritchard et al. 2000). The method is easy to understand, and is presented in Algorithm 1. Essentially, one continually proposes a value of θ from its prior; generates a realisation of the model conditional on this proposed θ and only accepts θ as a sample from the target distribution if the value of the discrepancy measure between this realisation and the true value falls below the required tolerance. If we repeat this N times, we obtain N approximate samples from the posterior.

Algorithm 1 ABC rejection sampling

Supply tolerance $\varepsilon > 0$, observed data x_{obs}, density function $f(x|\theta)$ and discrepancy measure $d(\cdot,\cdot)$.
for $i = 1, \cdots, N$ **do**
  repeat
    Sample $\theta^s \sim \pi(\theta)$
    Simulate $x_s \sim f(x|\theta^s)$
  until $d(x_{\text{obs}}, x_s) < \varepsilon$
  Assign $\theta_i \leftarrow \theta^s$
end for

Algorithm 2 Birth-death Metropolis-Hastings

For $j = 0, 1, 2, \cdots$, given $X_j = x^j$ generate X_{j+1} using the following procedure:
Generate $r_1, r_2 \sim \mathcal{U}[0, 1]$
if $r_1 \leq 1/2$ **then**
  Generate $u \sim q_b(x, \cdot)$
  Set

$$X_{j+1} = \begin{cases} x^j \cup u & \text{if } r_2 \leq A(x^j \mapsto x^j \cup u) \\ x^j & \text{otherwise} \end{cases}$$

else
  if $x^j = \emptyset$ **then**
    Set $X_{j+1} = x^j$
  else
    Generate $x_i \sim q_d(x^j, \cdot)$
    Set

$$X_{j+1} = \begin{cases} x^j \setminus x_i & \text{if } r_2 \leq A(x^j \mapsto x^j \setminus x_i) \\ x^j & \text{otherwise} \end{cases}$$

  end if
end if

Clearly, this algorithm requires that we have a method by which we can actually sample from our proposed model. This is not a small ask, and it can be an arduous task to complete. The method we will adopt is that of *birth-death Metropolis-Hastings* (Geyer & Møller 1994).

5.3 Birth-death Metropolis Hastings for spatial point processes

Geyer & Møller (1994) proposed a way to apply a Metropolis-Hastings algorithm to spatial point processes, and it has since received much interest. The process is called a birth-death Metropolis-Hastings algorithm (BDMH) and is outlined in pseudocode in Algorithm 2.

The principle is elegant. Suppose the we are at step j in the algorithm so that the simulated pattern is x^j. To proceed, one starts by deciding whether to propose a birth or death with probabilities p and $1 - p$, respectively. If a birth is chosen, a new point is sampled from the *birth density*, $q_b(x^j, \cdot)$, and in the case of the latter we select a point randomly to die from the *death density* $q_d(x^j, \cdot)$. In both cases, an *acceptance probability* $A(\cdot)$ is computed and, based on this probability, a decision is made as to whether the proposition should be accepted.

In the case of the "birth" of a point u, $A(\cdot)$ takes the form

$$A(x^j \mapsto x^j \cup u) = \min \left\{ 1, \frac{f(x^j \cup u)}{f(x^j)} \frac{(1-p)}{p} \frac{q_d(x^j \cup u, u)}{q_b(x^j, u)} \right\}$$

and for the "death" a point $x_i \in x^j$, it becomes

$$A(x^j \mapsto x^j \setminus x_i) = \min \left\{ 1, \frac{f(x^j \setminus x_i)}{f(x^j)} \frac{p}{(1-p)} \frac{q_b(x^j \setminus x_i, x_i)}{q_d(x^j, x_i)} \right\}$$

Algorithm 2 can be extended easily to a more general case in which f is no longer a density with respect to the standard Poisson process, but rather with respect to Poisson(S, λ), where $\int_S \lambda(\xi) \mathrm{d}\xi < \infty$. To do so, we simply need to replace $q_b(x, \xi)$ with $q_b(x, \xi)/\rho(\xi)$ in the ratios

above. This is very useful for application of the algorithm to a multivariate process (Møller & Waagpetersen 2004).

To apply the algorithm to a multivariate process, it is easiest to first consider a marked process on $T \times M$, with $M \subseteq \mathbb{R}^p$ being the mark space and $T \subseteq \mathbb{R}^d$. Equip this process with an unnormalised density f with respect to Poisson$(T \times M, p_M(m))$, where $p_M(m)$ is a mass function on M quantifying the probability of a point having mark $m \in M$. We can then apply the BDMH algorithm by replacing $q_b(x, \cdot)$ with $q_b(x, \cdot)/p_M(\cdot)$. In the multivariate framework, we consider M to be the set $\{1, \cdots, n\}$, with each $m \in \{1, \cdots, n\}$ corresponding to an index of a particular subprocess of the n-tuple $X = (X_1, \cdots, X_n)$. In this setting, we make the same transformation of the birth density, except now we can think of the mass function $p_M(m)$ as accounting for the probabilities of selecting one of the n subprocess in the n-tuple $X = (X_1, \cdots, X_n)$. For our purposes, we will give equal mass to each subprocess.

Chapter 6

LGCP-Strauss Process

The work presented here (for the univariate case) follows directly the work of Vihrs et al. (2020). In this paper, the authors devise a model built from a log Gaussian Cox process and a Strauss process that looks to model the phenomenon of large-scale aggregation but small-scale segregation in a univariate process. We will present and investigate this model before proposing a novel extension to the multivariate case. Vihrs et al. (2020) refer to their model as an *LGCP-Strauss process*, and in this spirit we will refer to our extension as a *multivariate LGCP-Strauss process* (forgiving the tautology).

6.1 LGCP-Strauss process

Let X be a point process on a bounded window $W \subset \mathbb{R}^2$ and let the density be written in the form

$$f(x|\psi, \varphi) \propto \prod_{i=1}^{n} \psi(x_i) \prod_{i<j} \varphi(\|x_i - x_j\|)$$

where $x = \{x_1, \cdots, x_n\}$ with $0 \leq n < \infty$ (for $n = 0$ we take $x = \emptyset$). The functions $\psi : W \to [0, \infty)$ and $\varphi : [0, \infty) \to [0, \infty)$ determine the first-order and second-order interactions, respectively. Note that φ is a function of distance only, and thus the process is seen to be isotropic. Vihrs et al. (2020) set ψ equal to $\Psi(u) = \exp(Z(u))$, with $Z := \{Z(u) : u \in W\}$ being a Gaussian random field. In this way, Ψ is the random intensity function of a log Gaussian Cox process. The purpose of this construction is to encourage the points in the process to cluster on the large scale. Furthermore, they take the Gaussian random field to have a constant mean $\mu \in \mathbb{R}$ (enforcing stationarity) and an exponential covariance function:

$$c(u, v) = \sigma^2 \exp\left(-\|u - v\| / s\right), \quad u, v \in W$$

To encourage small-scale segregation, the authors take φ to be the interaction term of a Strauss process, so that the density becomes

$$f(x|\theta) \propto \mathbb{E}\left\{ \prod_{i=1}^{n} \exp(Z(x_i)) \prod_{i<j} \gamma^{\mathbb{I}[\|x_i - x_j\| \leq R]} \right\} \tag{6.1}$$

with $\theta = (\mu, \sigma^2, s, \gamma, R)$ and $\mathbb{E}$ is taken with respect to the underlying Gaussian random field.

6.2 Simulation of a univariate LGCP-Strauss process

In line with the work of Vihrs et al. (2020), we will use the birth-death Metropolis-Hastings algorithm desccribed in Algorithm 2 to simulate a realisation of this process, simulating X

subject to $Z = z$. Conditioning on this realisation, we find

$$\frac{f(x \cup u)}{f(x)} = \exp(z(u)) \prod_{i=1}^{n} \gamma^{\mathbb{I}[\|x_i - u\| \leq R]} \quad \text{and} \quad \frac{f(x \setminus u)}{f(x)} = \exp(-z(u)) \prod_{j=1}^{n} \gamma^{-\mathbb{I}[\|u - x_j\| \leq R]}$$

where the expectations in the density (6.1) vanish due to the conditioning. Vihrs et al. took $q_b(x, \cdot) \propto \exp(z)$, but it was found here to be computationally faster to take $q_b(x, \cdot) = 1/|W|$, so that points are generated uniformly on the window. We will take $q_d(x, \cdot) = 1/N(x)$, so that the proposed point to die is also selected uniformly from the existing points, and we will further set $p = 1/2$ so that we select a birth or a death with equal probability. In this way, we obtain

$$A(x \mapsto x \cup u) = \exp(z(u)) \prod_{i=1}^{n} \gamma^{\mathbb{I}[\|x_i - u\| \leq R]} \frac{|W|}{(N(x) + 1)}$$

$$A(x \mapsto x \setminus x_i) = \left(\exp(z(x_i)) \prod_{j=1}^{n} \gamma^{\mathbb{I}[\|x_i - x_j\| \leq R]} \right)^{-1} \frac{N(x)}{|W|}$$

which can be substituted into the pseudocode of Algorithm 2 to generate realisations of this process.

Discretisation of W

It is necessary for us to discretise the observation window W in the simulation of the random field since we clearly cannot produce a numerical realisation of a continuous space. This is an imporant subtlety, and the choice of discretisation must be carefully considered as we need to ensure that we will not be creating identifiability issues with the interaction radius, R.

Specifically, let the points $\{x_1, \cdots, x_m\}$ and $\{y_1, \cdots, y_m\}$ be partitions of the x and y axes, respectively, where $x_i - x_{i-1} = y_i - y_{i-1} = r$, for all $i = 2, \cdots, m$. Consider drawing circles of radius R centred at a point (x_i, y_j) in the subsequent grid. This corresponds to the interaction region of the Strauss potential in the model. We will then have a series of annuli created by those values of R that result in counting the same number of points and are therefore equivalent in the context of the model. For example, any R falling in $[0, r)$ will be equivalent. The same holds for those values in $[r, \sqrt{2}r)$ or $[\sqrt{2}r, 2r)$ etc. In Figure 6.1, we shade some example annuli.

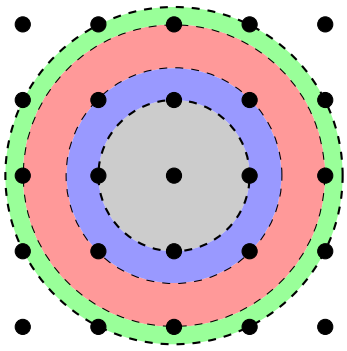
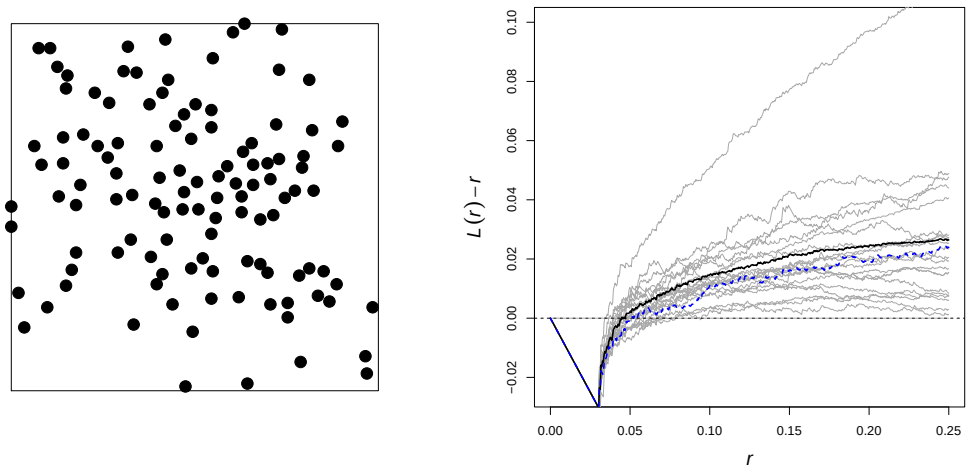


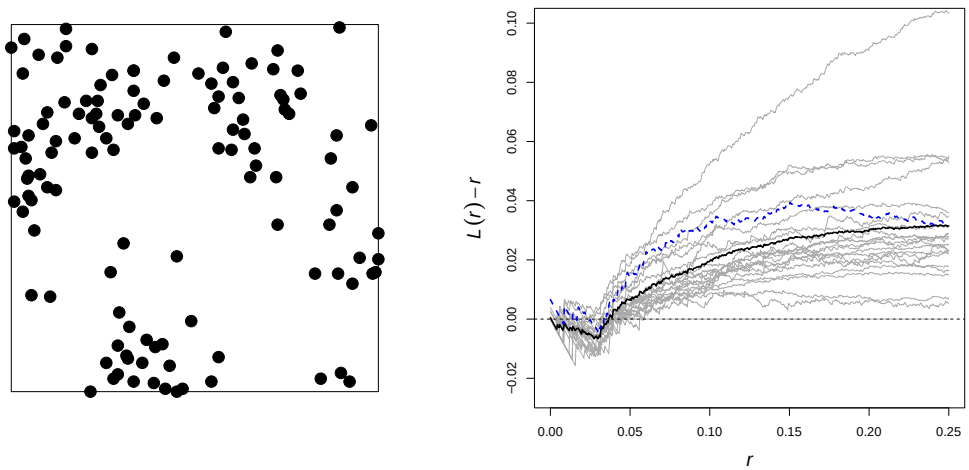
Figure 6.1: Each black circle corresponds to a point in the partition of W. Any value of R that would produce a circle whose perimeter lies within the same annulus as that of R' would be equivalent to R' in the model.

After some experimentation, a choice of 2^8 was found to be suitable as it created annuli of sufficiently small width (on average) to allow for differentiation between R values, whilst maintaining a feasible computation time in the simulation of the random field.

$\gamma = 0$



$\gamma = 0.3$



$\gamma = 0.6$

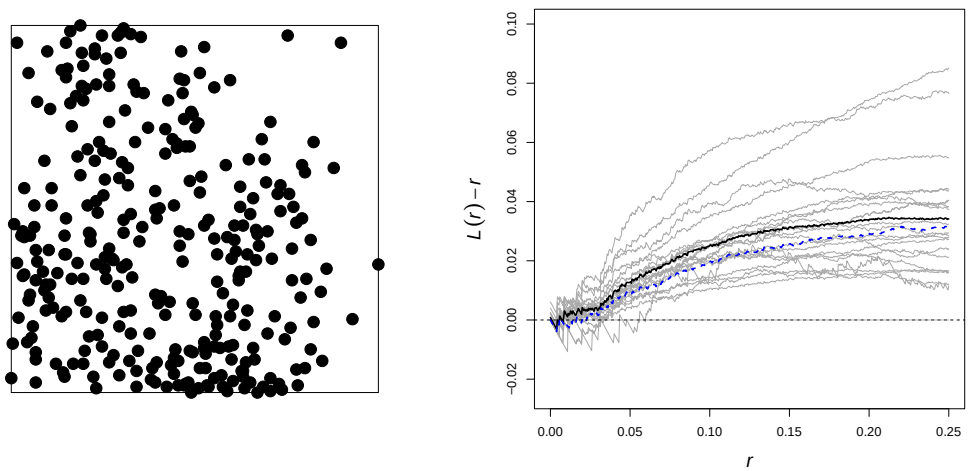


Figure 6.2: Each row of the first column shows a realisation of the LGCP-Strauss process with the given γ value and $\mu = 5, R = 0.03, s = 0.3$ and $\sigma^2 = 2$ in the simulation of the underlying field. The second column shows 20 plots of L-function with the given pattern shown in blue. The grey lines show 19 other simulations under the same conditions, and the black line is their mean.

Simulations and discussion

In Figure 6.2, we see three realisations of the LGCP-Strauss process using $\sigma^2 = 2$, $s = 0.3$, $R = 0.03$ and $\mu = 5$, while allowing γ to vary over the set $\{0, 0.3, 0.6\}$. These parameter values were chosen to emulate the work of Vihrs et al. (2020) so as to check that our code is running as expected.

The L-function plots (using the isotropic edge correction) show that the degree of small-scale inhibition is very dependent on the γ value, with clear segregation for $\gamma = 0$ and $\gamma = 0.3$, but less apparent separation for $\gamma = 0.6$. This dependence on γ is what was expected from the work of Vihrs et al.. The choice of the isotropic edge correction is justified as the model is constructed to be both isotropic and stationary.

In all three cases, we do see large-scale aggregation, as was expected. Interestingly, the degree of large-scale aggregation doesn't seem to be dependent on γ. Of course, the value of γ has no bearing on the underlying random field, but one might have expected that the amount of segregation on the small-scale might have an impact on the degree of clustering. The case of $\gamma = 0$ corresponds to a *hardcore* process within the interaction term of the density, and thus we would expect to see no points lying within R of one another. This is precisely the behaviour we see in all the realisations.

For more realisations and an in-depth discussion of this model, the reader is encouraged to read Vihrs et al. (2020).

6.3 Bivariate LGCP Strauss model

We will now present a novel extension to the work of Vihrs et al. (2020), that aims to model large-scale aggregation and small-scale segregation in a multivariate process. For simplicity, we will only present and examine our multivariate model in the bivariate setting, but the model can be trivially extended to higher dimensions.

Let $X = (X_1, X_2)$ be a bivariate pairwise interaction process with density f on $S \subseteq \mathbb{R}^2$. Each X_i is to be driven by the random intensity function $\Lambda_i = \{\exp(Y_i(s)) : s \in \mathbb{R}^2\}$ for $i = 1, 2$, in much the same way as a bivariate Cox process, where $Y(s) = \{(Y_1(s), Y_2(s)) : s \in \mathbb{R}^2\}$ is a bivariate stationary Gaussian random field with (possibly spatially varying) mean $\mu = (\mu_1, \mu_2)$ and covariance functions $c_{ij}(h) = \operatorname{cov}(Y_i(s_1), Y_j(s_2))$, with $h = \|s_1 - s_2\|$, $s_1, s_2 \in \mathbb{R}^2$. In this context, X_1 and X_2 are to be thought of as representing the point process of species A and B, respectively. The distinction between species A and the matrix A defined below will always be made obvious. Note that we will always take the exponential covariance function in application, and so σ and s denote the usual parameters (see Section 4.1). When there are k fields, we write σ_i and s_i for the parameters of the ith covariance function.

6.3.1 The general model

In our model, we will allow both the segregation and aggregation to depend on the subprocess in question. In particular, we will allow both the underlying random fields that drive the clustering behaviour and the Strauss processes causing the between-species inhibition to be subprocess-dependent. In the following, take Z_ℓ, $\ell = 1, \cdots, k$, to be zero-mean univariate Gaussian random fields with a covariance function r_ℓ of the users choice and let μ_1, μ_2 be deterministic functions and $A = [A_{i,j}]_{i,j=1}^{2,k}$ be a $2 \times k$ matrix. Recall the linear model of coregionalisation, which ensures that any $Y_j(s)$ constructed as $Y_j(s) = \sum_{\ell=1}^{k} A_{j\ell} Z_\ell(s) + \mu_j(s)$

with $A = [A_{ij}]_{i,j=1}^{m,k}$ being full-rank and $\mu_1, \cdots, \mu_m$ deterministic functions is a well-defined random field. Our model will be constructed using random fields of precisely this form so as to ensure the validity of its covariance structure. Furthermore, let m_{x_j} be 1 or 2, depending on the subprocess $x_j \in X$ belongs to. We can now explicitly define our process.

Definition 6.1: Bivariate LGCP-Strauss Process

Taking the parameter definitions as above, we define a bivariate process $X = (X_1, X_2)$ on $S \subseteq \mathbb{R}^2$ to be a *bivariate LGCP-Strauss process* if it has a density f on S of the form

$$f(x) \propto \mathbb{E}\left\{ \prod_{i=1}^{N(x)} \exp\left(\sum_{\ell=1}^{k} A_{m_{x_i}\ell} Z_\ell(x_i) + \mu_{m_{x_i}}(x_i) \right) \prod_{i<j} \Gamma(x_i, x_j)^{\mathbb{I}[\|x_i - x_j\| \leq R(x_i, x_j)]} \right\} \tag{6.2}$$

where

$$\Gamma(x_i, x_j) = \begin{cases} \gamma_1 & \text{if } m_{x_i} = m_{x_j} = 1 \\ \gamma_2 & \text{if } m_{x_i} = m_{x_j} = 2 \\ \gamma & \text{if } m_{x_i} \neq m_{x_j} \end{cases} \quad \text{and} \quad R(x_i, x_j) = \begin{cases} R_1 & \text{if } m_{x_i} = m_{x_j} = 1 \\ R_2 & \text{if } m_{x_i} = m_{x_j} = 2 \\ R & \text{if } m_{x_i} \neq m_{x_j} \end{cases}$$

with $\gamma_1, \gamma_2, \gamma \in [0, 1]$.

The first product in the expectation corresponds to the terms driving the aggregation. We can adjust the entries of A to alter the large-scale aggregation of the points. The second term corresponds to the segregation, which is designed in such a way that intraspecies inhibition is species-dependent. By this we mean that points within the same subprocess interact with one another (as well as with those of the other subprocess), and the strength of this interaction is dependent on the subprocess in question. Again, we have taken a Strauss model to cause the small-scale segregation.

Some special cases include when $\mu_1 = \mu_2 = 0$ and A is the zero matrix. For these values, we simply recover the bivariate Strauss process. Similarly, allowing $\Gamma \equiv 1$, we obtain a bivariate Cox process.

6.3.2 The simple model

A particularly interesting case is when $k = 1$, $A = (1, 1)^T$, $\gamma_1 = \gamma_2 = 1$ and $\mu_{m_i} = \mu$, $R_i = R$ for all i, with μ and R being fixed constants. In this set-up, we force the points to aggregate in a species-independent way, which will cause between-species aggregation, but to segregate in a species-dependent way. We will refer to this model as the *simple model*, with a density of the form

$$f(x|\theta) \propto \mathbb{E}\left\{ \prod_{i=1}^{N(x)} \exp\left(Z(x_i)\right) \prod_{i<j} \gamma^{\mathbb{I}[\|x_i - x_j\| \leq R, m_{x_i} \neq m_{x_j}]} \right\} \tag{6.3}$$

This is not a particularly flexible model as both species aggregate accoring to the same linear scaling of the same random field. However, as will soon be discussed, it enjoys a large reduction in the number of parameters, and thus is far better suited to the inference techniques employed by Vihrs et al. (2020) for the LGCP-Strauss model. Furthermore, the use of only a single random field (i.e. setting $k = 1$) makes the behaviour of the model arguably more intuitive.

6.3.3 Simulation

The bivariate LGCP-Strauss process can be simulated using Algorithm 2, with the adaptations for a bivariate setting that have been discussed. We will consider simulating the model in the case of an exponential covariance function in the underlying random field. To gain some intuition, let us first consider some realisations of the simple model.

Simulation of the simple model

In Figures 6.3, 6.4 and 6.5, we have produced simulations of the simple model using the BDMH algorithm. Each figure contains an example point pattern along with bivariate L-function estimates (both between and within species) for 20 simulations with the same parameter values. Note that the notation $L_{A,B}(r)$ is equivalent to $L_{12}(r)$. For the simulations in each of the figures we have taken $\mu = 5$, $R = 0.03$ and then $s = 0.3$ and $\sigma^2 = 2$ in the exponential covariance function, but have allowed γ to vary. Specifically, we set $\gamma = 0$, 0.3 and 0.6 in Figures 6.3, 6.4 and 6.5, respectively.

For the choice of $\gamma = 0$ and $\gamma = 0.3$, we see that the model behaves as desired: the bivariate L-function plots (with the isotropic correction) show small-scale inhibition and large-scale aggregation in almost all the plots and the individual L-function plots for each species show aggregation only, which is to be expected as we took $\gamma_1 = \gamma_2 = 1$. However, for $\gamma = 0.6$, despite each subpattern clearly aggregating, the small-scale inhibition is less apparent as we see that the mean line doesn't fall below the 0 mark. This indicates that the model isn't overly effective for weaker inhibition parameters (i.e. for γ closer to 1 than 0). This is precisely what was observed for the LGCP-Strauss model in the simulations shown in Figure 6.2. This perhaps suggests that observing only aggregation in the case of γ values closer to 1 than 0 is an inherent relic of a model based on combining a Strauss and Cox process.

Simulations of the general model

As has been described, the general model is very flexible and can cause aggregation and segregation over different length scales and with different strengths. In particular, it can create within-species inhibition, unlike the simple model. We present only two sets of realisations of the general model as the examples chosen are sufficient to demonstrate its flexibility.

In Figure 6.6, we again produce 20 realisations (but this time of the general model) and plot both the within-species L-function estimates and between-species bivariate L-function estimates, along with the mean of the runs and that of the shown pattern. In these realisations, we made parameter choices with the intent of showing how the model can show small-scale behaviour that varies between subprocess whilst displaying the desired multiscale behaviour between the subprocesses. The parameters are listed in the figure caption, but importantly we took $\gamma_1 = 0.3$, $\gamma_2 = 1$ and $\gamma = 0$. We see the hard-core behaviour for the plots of $L_{AB}(r) - r$, as expected. For the plot of $L(r) - r$ for subprocess X_2, we see aggregation on all scales, which again agrees with a choice of $\gamma_2 = 1$. For X_1, however, we see an L-function plot that does not exhibit the standard plots of segregation or aggregation. A choice of $\gamma_1 = 0.3$ should induce some within-species segregation, however it is apparent that the between-species inhibition is interfering and causing the points of X_1 to cluster together more than might have been expected. One can imagine a set of magnetised balls repelling one another, while another external magnet tries to force them together. Nonetheless, we see the flexibility of the model and its ability to cause asymmetric subprocess behaviour.

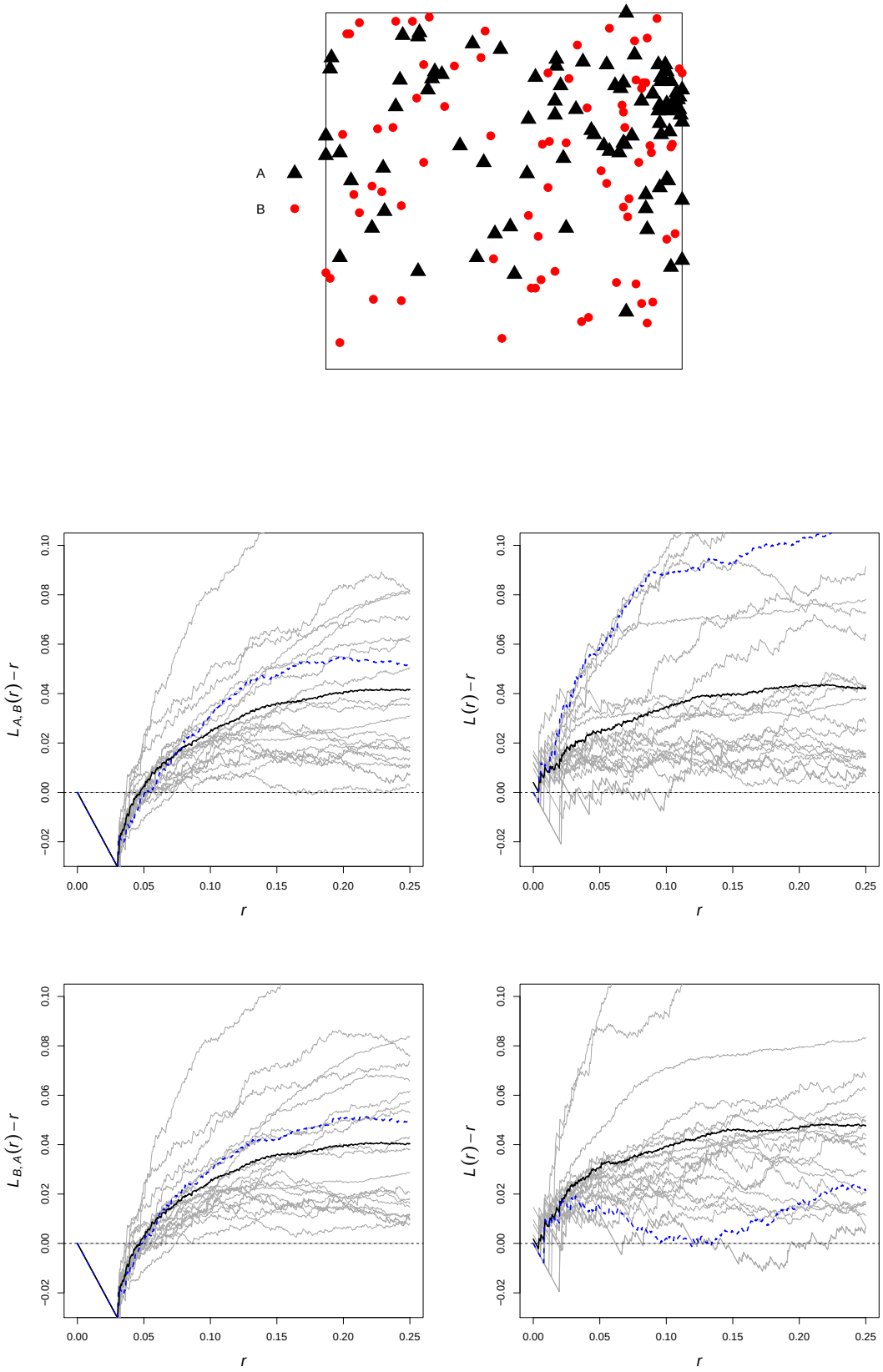


Figure 6.3: (Simple model) The first row shows a realisation of a simple bivariate LGCP-Strauss process with $\gamma = 0, \mu = 5, R = 0.03, s = 0.3$ and $\sigma^2 = 2$. The second row shows plots of (estimates of) $L_{AB}(r) - r$ and $L(r) - r$ (for X_1) using the istropic correction. The third row shows (estimates of) $L_{BA}(r) - r$ and $L(r) - r$ (for X_2) with the same correction. The colours are as in Figure 6.2.

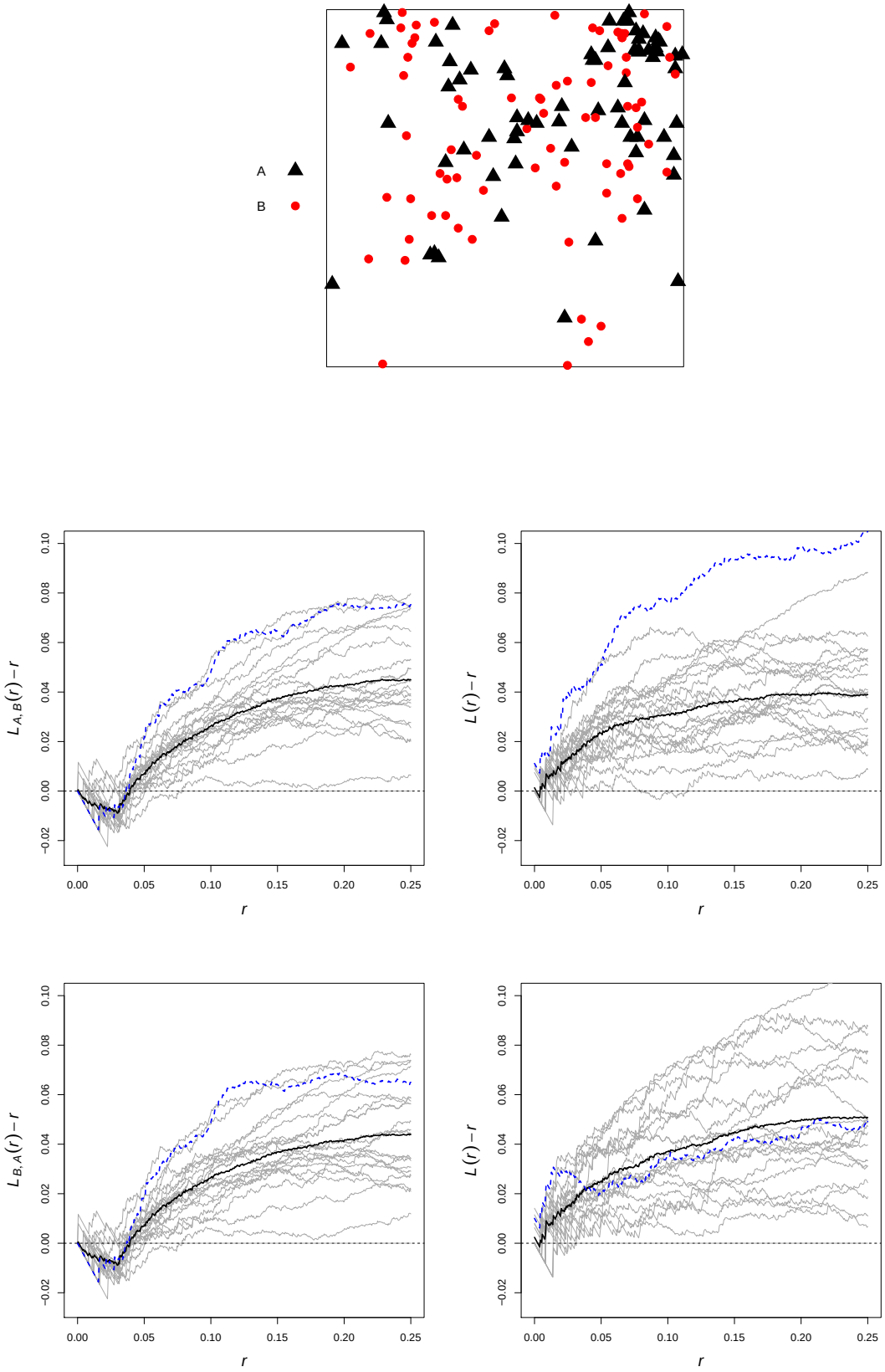


Figure 6.4: (Simple model) The plots are analogous to Figure 6.3 but with $\gamma = 0.3, \mu = 5, R = 0.03, s = 0.3$ and $\sigma^2 = 2$.

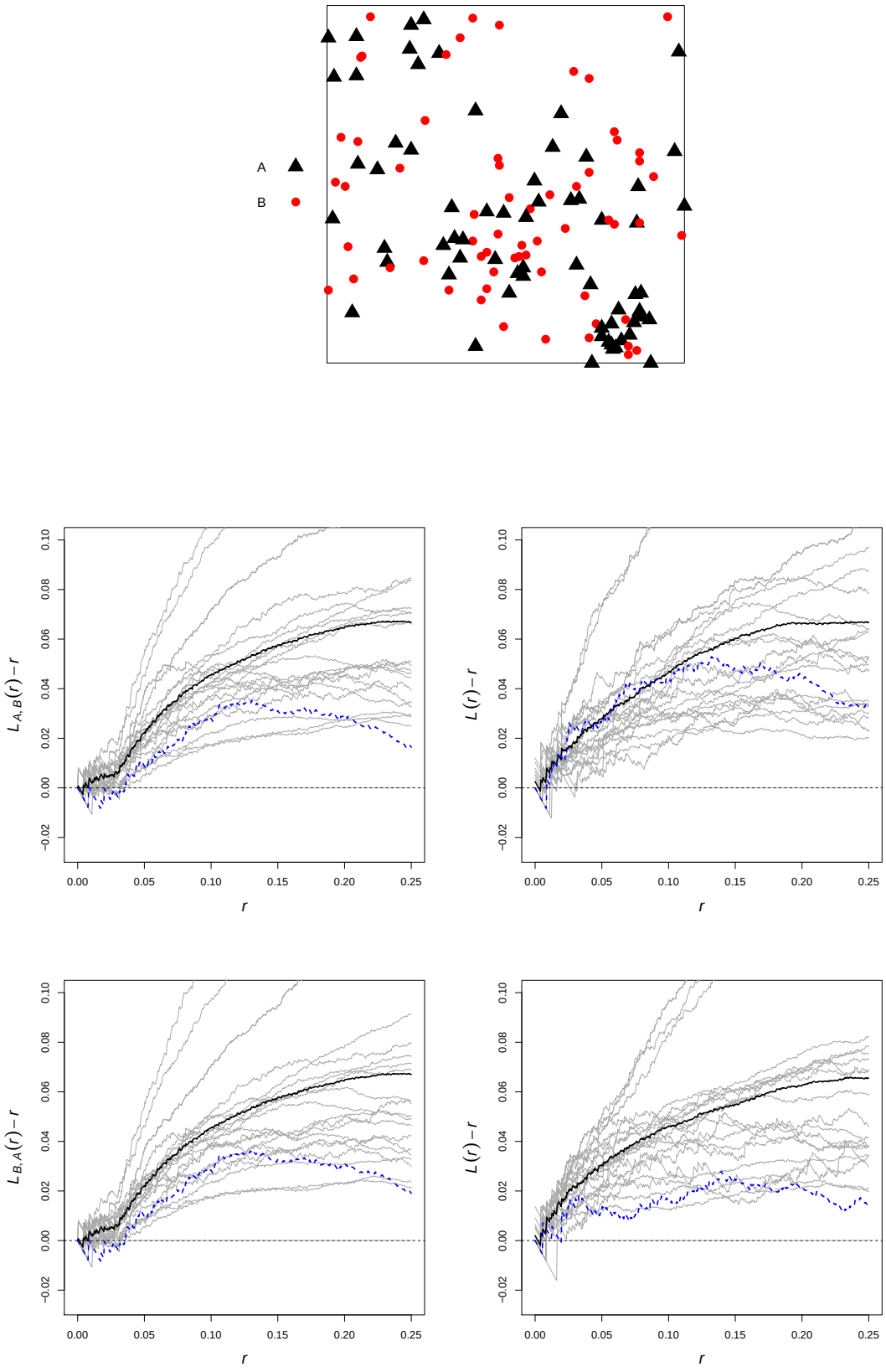


Figure 6.5: (Simple model) The plots are analogous to Figure 6.3 but with $\gamma = 0.6, \mu = 5, R = 0.03, s = 0.3$ and $\sigma^2 = 2$.

Precisely the same plots are shown in Figure 6.7, but in this case the parameters were chosen to be identical between the subprocesses. In particular, we took $\gamma_1 = \gamma_2 = \gamma = 0.3$. From the plots, we see symmetry in the subprocess L-functions, as would be expected from the parameter choices. It is interesting to note that neither of the subprocesses nor the overall process display any significant segregation at any scale. The explanation for this is analgous to before, wherein the between-species inhibition interfere with the within-species repulsion and results only in limiting the small-scale aggregation rather than causing any small-scale segregation. This is worth noting as it will be relevant in future discussions on inference for these processes.

6.4 ABC inference for the bivariate LGCP-Strauss model

We now look to examine the effectiveness of an ABC procedure at estimating the parameters of a realisation of a bivariate LGCP-Strauss process. We will run the algorithm of Vihrs et al. (2020), which we give in Algorithm 3.

As a first point of call, we will look at the effectiveness of the ABC procedure in Algorithm 3 in recovering parameters of the simple model. From here, we will progress to examine inference in the full model.

6.4.1 Inference for the simple model

As has been discussed, the simple model has only a limited number of parameters, each of which interfere only mildly with one another, and thus we would expect the ABC procedure to be relatively effective. We will use an exponential covariance function again, and thus have the parameters s and σ^2 to infer, as well as those explicitly stated in the model. It has been shown that some parameters in a Matérn model (of which the exponential covariance function is a subset) may not be consistently identified, but that the ratio σ^2/s can be (Vihrs et al. 2020, Zhang 2004). This is worth considering, and will discussed once the posteriors for the parameters are obtained.

We need to select appropriate summary statistics to construct our discrepancy measure. As discussed in Vihrs et al. (2020), there are infinite choices of summary statistic, but, of course, we want to select those that most aptly capture the structure of the data. We will take inspiration from the statistics selected by the aforementioned authors. For a detailed description of the relevance of the following summary statistics, the reader is referred to Vihrs et al. (2020), for here we will only give a brief summary.

A key summary statistic to use is the bivariate L-function, $L_{12}(r)$, as it will help to describe the scale of interaction of the subprocesses. Note that we only consider $L_{12}(r)$ and not $L_{21}(r)$ because Figures 6.3, 6.4 and 6.5 show that there is only a marginal difference between the two functions, and so we can safely justify selecting only one for our summary statistics. We will also compute the counts of points in different regions (and variants thereof) as these should help to describe the spread of points and thus to describe the underlying field that drives this spread. All counts will be taken independent of type because all the parameters are mark-independent in the case of the simple model, and there is thus no need for us to differentiate between points of different types when counting.

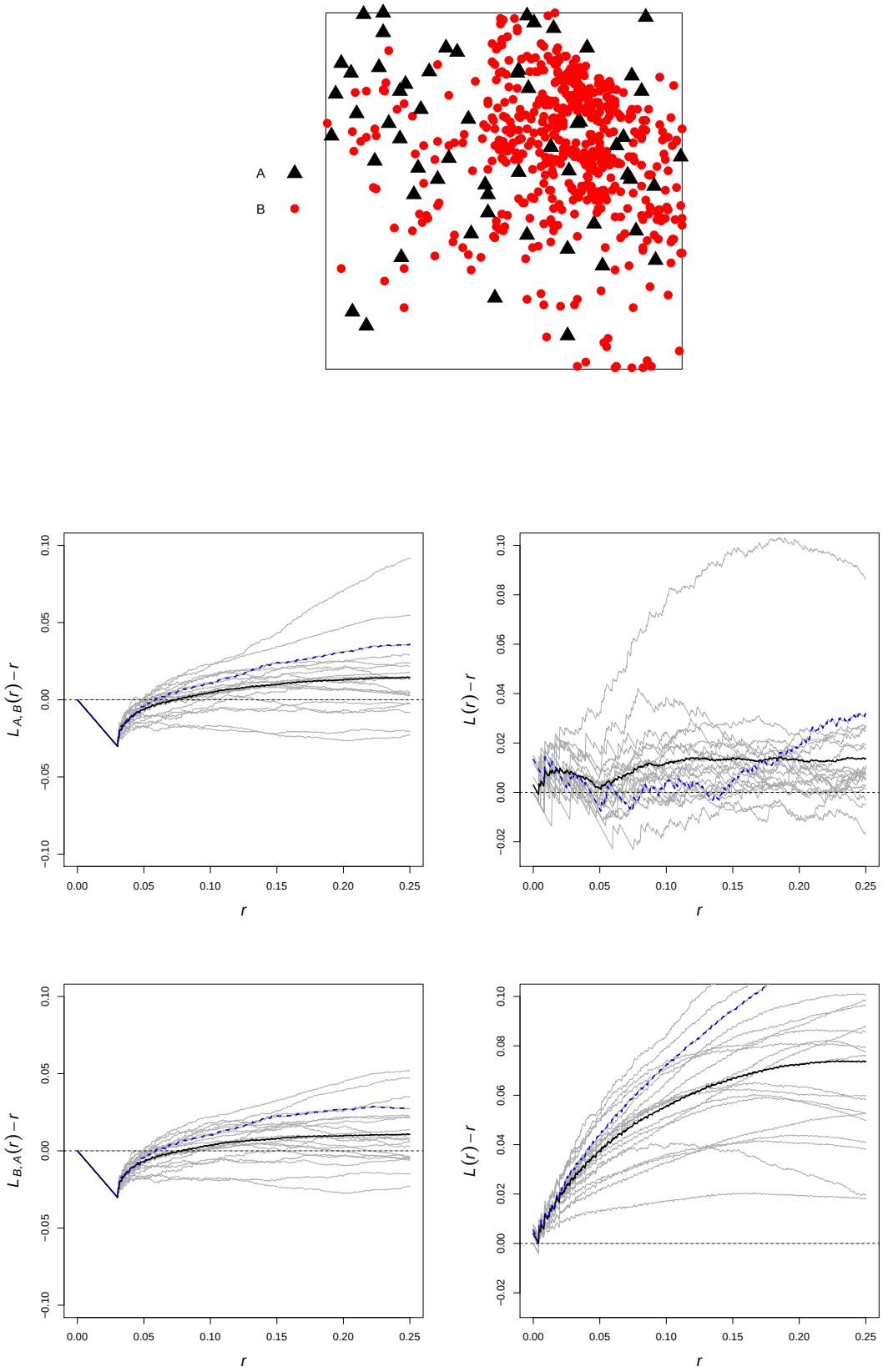


Figure 6.6: (General model) The plots are analogous to Figure 6.3. We take $\mu_1 = 5$, $\mu_2 = 6$, $s_1 = 0.2$, $s_2 = 0.3$, $\sigma_1 = \sigma_2 = 1$, $R_1 = 0.05$, $R_2 = 0$, $R = 0.03$, $\gamma_1 = 0.3$, $\gamma_2 = 1, \gamma = 0$ and $A = \begin{pmatrix} 2 & 1 \\ 1 & 3 \end{pmatrix}$

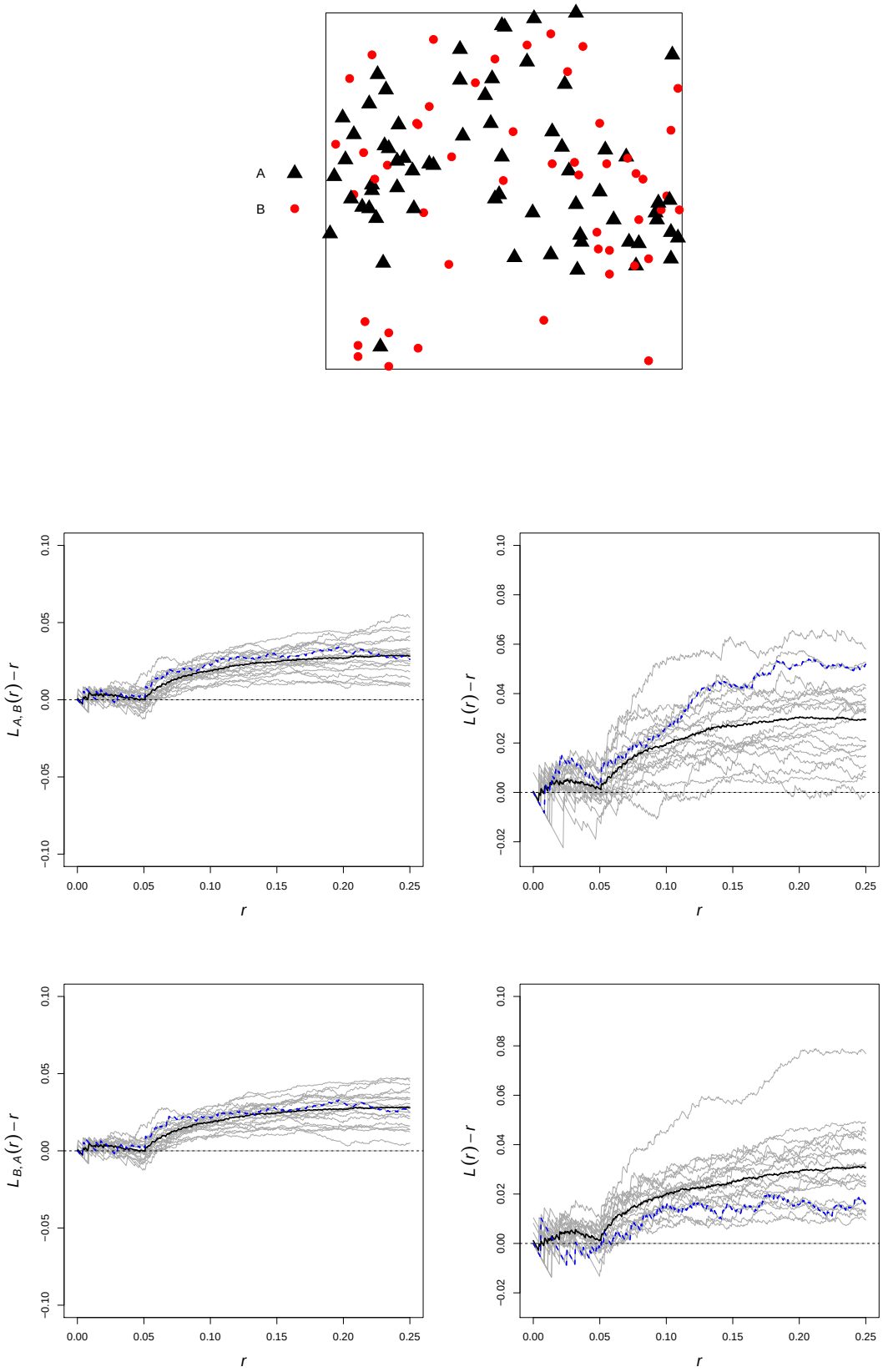


Figure 6.7: (General model) The plots are analogous to Figure 6.3. We take $\mu_1 = \mu_2 = 6$, $s_1 = s_2 = 0.3$, $\sigma_2 = \sigma_1 = 1$, $R_1 = R_2 = R = 0.03$, $\gamma_1 = \gamma_2 = \gamma = 0.3$ and $A = \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix}$

The summary statistics we will use are as follows:

1. $n_{\log} := \log(N(x))$
2. $L_{\max} := \max(\hat{L}_{12}(r) - r)$
3. $L_{\min} := \min(\hat{L}_{12}(r) - r)$
4. $L_{\text{argmin}} := \text{argmin}(\hat{L}_{12}(r) - r)$
   where $\hat{L}_{12}(r)$ denotes a nonparametric estimate of $L_{12}(r)$
5. $L_1, \cdots, L_m$, which denote $L_{12}(r) - r$ evaluated at m equally spaced values between 0 and $0.2h$, where the observation window W is assumed a square (for simplicity) of side length h.

   Divide W into q^2 identical smaller squares, $W_{i,j}$.
6. $C_{\max} := \max_{i,j=1,\cdots,q} \{N(W_{i,j})/N(x)\}$
7. $C_{\min} := \min_{i,j=1,\cdots,q} \{N(W_{i,j})/N(x)\}$
8. $C_{\log \text{var}} := \log\left(\hat{\text{var}}\left(\{N(W_{i,j})/N(x)\}_{i,j=1}^{q}\right)\right)$ where vâr is, of course, the empirical variance.

In summary, statistics 1 through 5 look to capture information on γ and R by looking at the interaction between points of different types. Vihrs et al. (2020) discuss how, in the univariate case, the minimum of a $L(r) - r$ estimate often occurs close to the true value of R, and so it seems sensible to include an analogous statistic here. On the other hand, statistics 6, 7 and 8 are aimed at inferring the parameters of the underlying field. Heuristically, the reasoning here is that the parameters of the field only have an influence over the clustering behaviour, and thus examining the behaviour of the counts in different parts of the observation window should help to describe them.

Of course, priors must be selected to run the ABC procedure in Algorithm 3. For the selection of appropriate priors, we will simply take uniform priors for all parameters:

$$\mu \sim \mathcal{U}(4,6), \quad s \sim \mathcal{U}(0.01, 0.5), \quad \sigma^2 \sim \mathcal{U}(0,4), \quad R \sim \mathcal{U}(0, 0.05), \quad \gamma \sim \mathcal{U}(0,1)$$

The only modification we have made from Vihrs et al. (2020) is to alter the prior of μ to be uniform on $(4,6)$ rather than $(3,6)$. This change was made because simulations using $\mu \in (3,4]$ took far too long to reach the required acceptance threshold of 40 points of each species. It would have been interesting to experiment with different priors, particularly those that allowed us to make use of preliminary data analysis and place more mass near where we expect the true values to lie (for example, a gamma prior). However, time constraints did not permit us to investigate this.

To test the inference procedure, we generated two observed patterns (again, more would have been preferable but time was a limiting factor) using parameter choices: $\mu = 5$, $s = 0.3$, $R = 0.03$, $\sigma^2 = 2$ and $\gamma = 0.3$ in the first run and $\mu = 5.5$, $s = 0.2$, $R = 0.04$, $\sigma^2 = 2$ and $\gamma = 0.2$. Note that the parameter choices in each run are quite similar. The reasoning here is that the first run inadvertently used values at or close to the mean of the priors, and so a second was done with shifts of some of the true values to investigate whether the accuracy of the first run was chance. We then ran Algorithm 3 using $k_{\text{pilot}} = 10,000$, $k_{\text{ABC}} = 0.3$ and $\varepsilon = 10\%$. Ideally, a smaller value of ε would have been taken so as to sample from a distribution closer resembling the true posterior, but the ABC part of Algorithm 3 was found to take

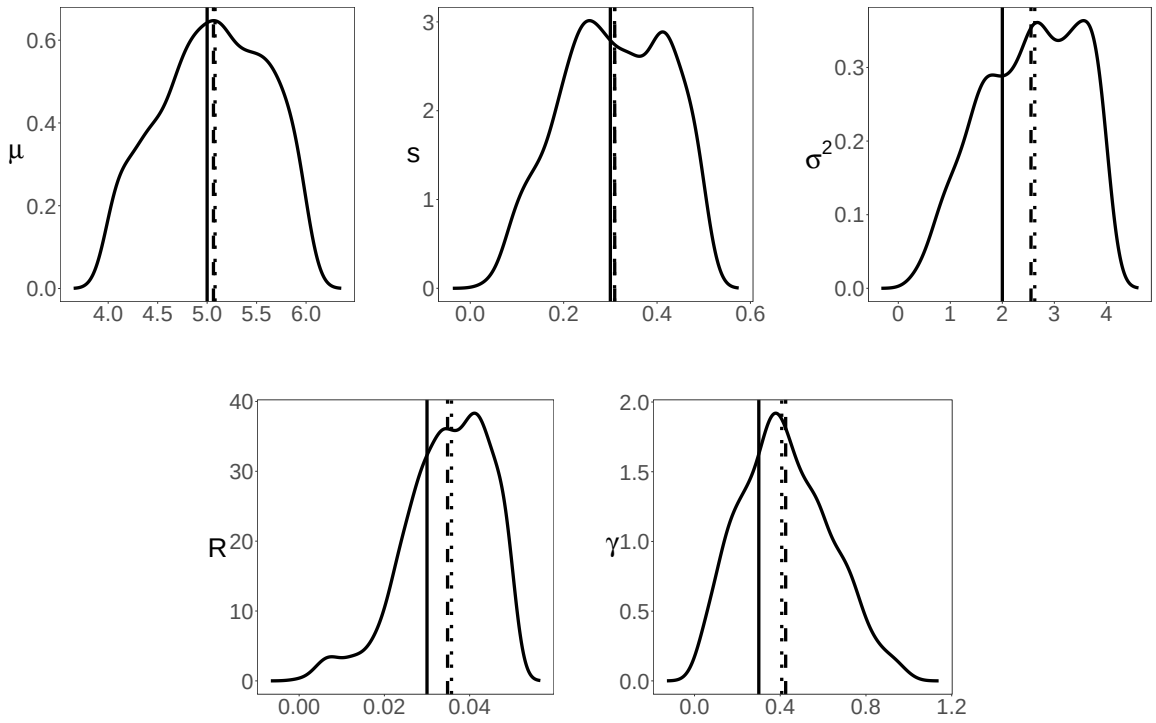


Figure 6.8: Plots of the posterior distributions of each parameter in the simple model using 1000 iterations. The true values are $\mu = 5$, $s = 0.3$, $\sigma^2 = 2$, $R = 0.03$ and $\gamma = 0.3$. The thick vertical line is the true value, and the dashed and dotted lines are the mean and median of the posterior, respectively.

over 50 hours (on 20 separate cores) even at this choice of ε, and so a smaller value wouldn't have been feasible in the time constraints of this thesis. Code for this procedure can be found in the GitHub repository SPPs.

The posterior distributions of each parameter on the different runs are plotted in Figures 6.8 and 6.9. In both figures, we see how the posteriors of each parameter are quite different from their priors. In both cases, we see that the values of μ, R and γ are well-recovered in the sense of the means and medians being close to the true value in the simulated "observed" data. It is worth considering that this is at a 10% level, which is quite a high choice of ε, and so with less time-constraints one could feasibly expect very accurate recovery.

Although the shape of the posterior distribution for s does differ from that of its prior, the means and medians of the posteriors in both inference procedures is the same as that of the prior, suggesting that the accuracy of the estimate in the run of Figure 6.8 was simply a relic of selecting s to be the mean of the prior.

The value of σ^2 was the same in both runs, and although the shape of the distributions do differ slightly (with the density of Figure 6.9 being more left-skewed), the means and medians are approximately the same. This is encouraging, as it is likely that the discrepancy between the true value and posterior estimates is a result of the large choice of ε.

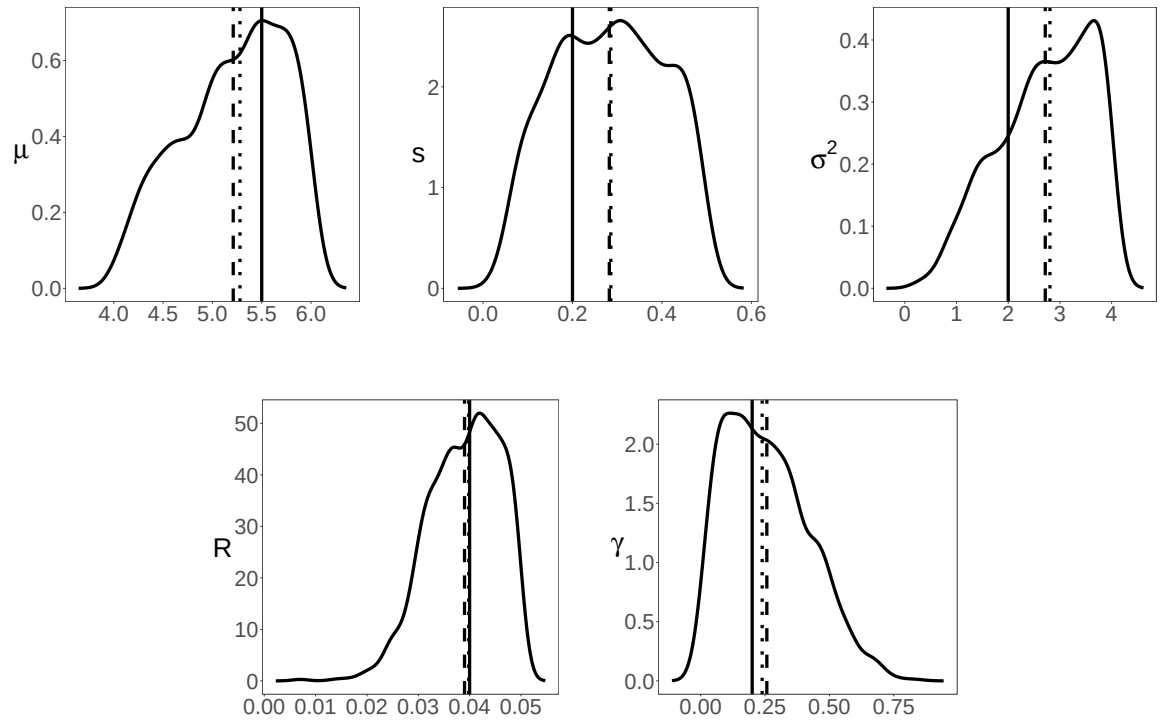


Figure 6.9: Plots of the posterior distributions of each parameter in the simple model using 1000 iterations. The true values are $\mu = 5.5$, $s = 0.2$, $\sigma^2 = 2$, $R = 0.04$ and $\gamma = 0.2$. The thick vertical line is the true value, and the dashed and dotted lines are the mean and median of the posterior, respectively.

6.4.2 Inference for the full model

The same procedure was run on the full model with identical choices of k_{pilot} and k_{ABC}. The priors for the parameters in the general model that weren't in the simple model were chosen to be the same as those stated previously (i.e. we took $\mu_1, \mu_2 \sim \mathcal{U}(4,6)$ etc.) Similarly, the summary statistics were extended in the natural way. By this we mean that we included analogous summaries of $L_1(r)$, $L_2(r)$ (subscript corresponding to the subprocess on which the L-function was estimated) additionally to those of $L_{12}(r)$, and extended the counts from just the overall process to include those of each subprocess. We do not list them in full for brevity. Further to this, the number of random fields, k, that we took in the linear combination was fixed at $k = 2$. The reason here being that additional fields were found to have a negligible impact on the behaviour of the process. Demonstrative plots from this investigative work are seen in Appendix A.2.

After running the ABC procedure, the obtained samples from the approximate posteriors of the parameters were found to be very similar to those of the priors, indicating that very little information was gained by running the procedure. The plots are not included as they offer no additional insight. The failure of the ABC procedure here is disappointing, though not entirely unexpected, as we will now discuss.

In the full model, there are $2k$ entries in A, $2k$ parameters for the k random fields (for the choice of exponential covariance function: variance and length scale), 2 mean values, 3 radii and 3 interaction parameters. This brings the total to $4k + 8$ parameters that need to be inferred. Even for a small choice of $k = 2$, we will have 16 parameters to infer. This by itself is not a limiting issue, as ABC procedures can be effective even for large numbers of parame-

Algorithm 3 Algorithm for ABC (Vihrs et al. 2020)

Input x_{obs}, a prior $\pi(\theta)$ for $\theta = (\theta_1, \cdots, \theta_p)$, a summary statistic $T(x) = (T_1(x), \cdots, T_n(x))$, positive integers k_{pilot} and k_{ABC} and nonnegative integer m.

Pilot run:

for $i = 1, \cdots, k_{\text{pilot}}$ **do**

  repeat

    Sample $\theta^{\text{pilot},i} \sim \pi(\theta)$ and simulate $x^{\text{pilot},i}$ using Algorithm 2 with parameter values $\theta^{\text{pilot},i}$

  until $N(x^{\text{pilot,i}}) > m$

end for

for $j = 1, \cdots, p$ **do**

  Use the sample $\left\{\left(\theta^{\text{pilot},i}, x^{\text{pilot},i}\right)\right\}_{i=1}^{k_{\text{pilot}}}$ to fit the following linear model

$$\mathbb{E}[\theta_j | x] \approx \theta_j(x) := \alpha^j + \beta^{j^T}(T(x) - T(x_{\text{obs}}))$$

  where $\alpha^j \in \mathbb{R}$ and $\beta^j \in \mathbb{R}^n$. Writing $\hat{\alpha}^j$ and $\hat{\beta}^j$ for the fitted values, let $\hat{\theta}_j(x)$ be the corresponding estimate of $\theta_j(x)$.

end for

Define the distance measure

$$\chi\left(\hat{\theta}(x), \hat{\theta}(x_{\text{obs}})\right) = \sum_{j=1}^{p} \frac{(\hat{\theta}_j(x) - \hat{\alpha}^j)^2}{\hat{\text{var}}(\hat{\theta}_j)}$$

where $\hat{\theta}(x) = (\hat{\theta}_1(x), \cdots, \hat{\theta}_p(x))$ and $\hat{\text{var}}(\hat{\theta}_j)$ is the sampling variance of $\left\{\hat{\theta}_j(x^{\text{pilot},i})\right\}_{i=1}^{k_{\text{pilot}}}$.

Let ε be empirical 1% percentile of $\left\{\chi(x^{\text{pilot},i}, x_{\text{obs}})\right\}_{i=1}^{k_{\text{pilot}}}$

ABC rejection sampling:

for $i = 1, \cdots, k_{\text{ABC}}$ **do**

  repeat

    repeat

      Sample $\theta^{\text{ABC},i} \sim \pi(\theta)$ and simulate $x^{\text{ABC,i}}$ using Algorithm 2 with parameter values $\theta^{\text{ABC},i}$

    until $N(x^{\text{ABC,i}}) > m$

  until $\chi\left(\hat{\theta}(x^{\text{ABC},i}), \hat{\theta}(x_{\text{obs}})\right)$

end for

ters. Instead, the overarching problem we face is that of parameter identifiability. Specifically, many of the observed effects of each parameter overlap with one another which makes it hard to distinguish them in the inference procedure.

Consider first the role of the matrix A, and σ^2 from the covariance function. Equation (4.2) shows how the entries of A will multiply with σ^2 in the formation of the covariance matrix of the process. In this way, for any one product of σ^2 and the i, jth entry of A, A_{ij}, there will be uncountably many choices that lead to the same result since $\sigma^2 A_{ij} = C \in \mathbb{R}$ has an (uncountably) infinite number of solutions over any finite interval. As such, it will be very challenging to effectively infer either of their values. A proposed solution is to set $\sigma^2 = 1$, and to only allow A to vary. This is a reasonable simplification to make as it has no impact on the flexibility of the model.

Another interaction is apparent from Figure 6.7, where $\gamma_1 = \gamma_2 = \gamma = 0.3$, yet no small-scale inhibition was observed. As has been discussed, the reason for this is an effective cancellation of the effect of the within and between-species inhibition. Extending this reasoning further, we see that it will be very challenging to adequately ascertain the value of each γ_i or of γ when each interferes with the others.

A similar interaction is observed between the entries of A and γ. In Figures A.1, A.2 and A.3 we have three sets of L-function plots from 20 simulations with each figure corresponding to a process with A being scaled by a factor larger than 1 from the previous (in numerical order). In other words, the A matrix of the processes in each figure are proportional. In each of the processes, a value of $\gamma = 0.3$ was taken, which is known from previous plots to produce significant small-scale inhibition. As we move through the figures, we see the small-scale inhibition decreasing as the entries of A are scaled with increasing magnitude. From this we infer that selecting entries of A with large (positive) magnitude can mask the impact of γ in the model, which again will again cause identifiability issues. A potential solution here is to restrict the entries of A to a small interval, say $[0.01, 2]$, so that the effects of A and γ can be distinguished. One could even take an interval allowing negative values, but this would have the potential to lead to a negative covariance between the species and thus no large-scale aggregation.

These, of course, are only some of the parameter interactions, and we do not include a full discussion of every possible interaction for fear of labouring the point. An advisable first step for fitting this model in the future would be to use preliminary data analysis to restrict priors for each parameter, and even to set certain parameter values preemptively. In this way, the ABC procedure might enjoy more success. Suggestions for this further work are delayed until the end of the thesis.

Overall, the interplay between the parameters has lead to the ABC procedure struggling to correctly identify the important summary statistics when fitting the LASSO and linear model in Algorithm 3. Consequently, the process was unsuccessful for the full model. That being said, considerable success was enjoyed for the simple model. As such, we now look to fit the simple model to a real data set.

Chapter 7

Application of the bivariate LGCP-Strauss model to real data

7.1 Japanese black pines data set

We now look to apply the simple model to a real data set. The point pattern we consider is the natural stands of seedlings and saplings of the Japanese black pine consisting of $N = 204$ points on a 10m $\times$ 10m area (Numata 1964). Ogata & Tanemura (1985) found that those pines of height 20cm or less exhibited the desired multiscale interaction with those of height greater than 20cm.

In Figure 7.1, we see that there is clear small-scale segragation and some large-scale aggregation between the two subprocesses. The large-scale aggregation is arguably not as clear, yet it appears sufficient for the fitting of our simple model.

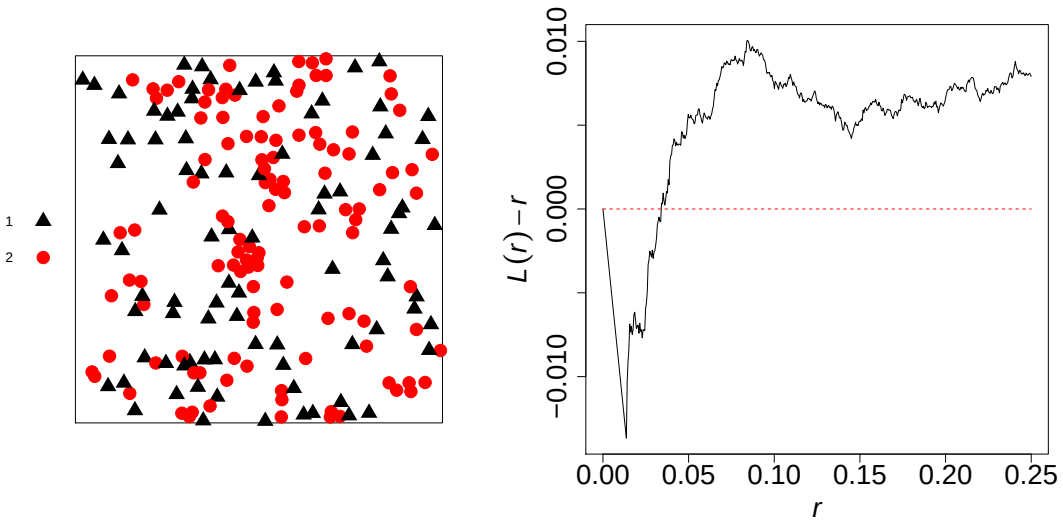


Figure 7.1: Plots of the pattern and estimated L-function for the Japanese pines data set (Numata 1964). The data itself was procured from the `spatstat` package. Points with mark 1 are those of height greater than 20cm.

7.2 Model fitting and model checking

To fit our simple model to the Japanese pines data set, we first re-scale the data to the unit square and then implement Algorithm 3 using the usual uniform priors and $\varepsilon = 10\%$. The resulting posteriors are shown in Figure 7.2. Note how different the posteriors are to their uniform priors. The posterior of R is approximately normal, while those of σ^2 and γ are right skew and that of μ is left skew. The posterior for s has a strange shape, which is possibly the result of again taking $\varepsilon = 10\%$ which places a limitations on the accuracy of our posterior sampling.

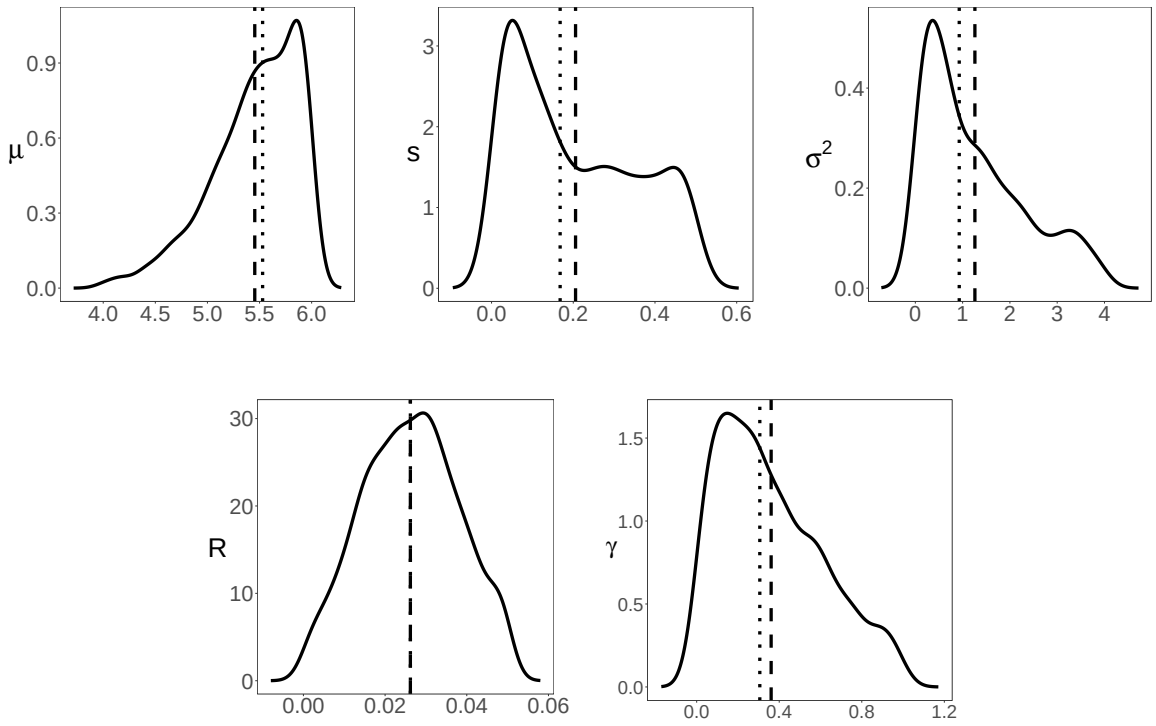


Figure 7.2: Posterior distributions computed for the Japanese pines data set using the ABC procedure of Algorithm 3. The dashed and dotted lines in each panel indicate the means and medians of the posteriors, respectively.

After fitting the model, we are interested in whether the fit is better (in an appropriate sense) than that of a bivariate Strauss process to the same data. We do not concern ourselves with the fit of a log-Gaussian Cox process due to the degree of small-scale inhibition. We make global envelope tests based on posterior predictions in the way outlined by (Vihrs et al. 2020). Namely, we start by running Algorithm 3 for a bivariate Strauss process to obtain posterior samples. Then, for each ABC realisation of θ, we simulate a pattern x subject to the process that generated θ and then compute an estimate of the bivariate L-function. Using these functional summary statistic estimates, we can construct global envelopes and, subsequently, tests based on these (Myllymäki et al. 2017, Mrkvička et al. 2020). To construct these envelopes, we used the `R` package `GET` (Myllymäki et al. 2017).

We used 95% global envelopes based on the L-function to compare the fit of a bivariate Strauss process to that of a simple bivariate LGCP-Strauss model. It is not worth discussing the construction of these envelopes as it falls outside the scope of this thesis, but, in essence, one can say that the probability that the empirical curve lies within the interval is approximately 95%. Such intervals for the bivariate Strauss and the simple bivariate LGCP-Strauss

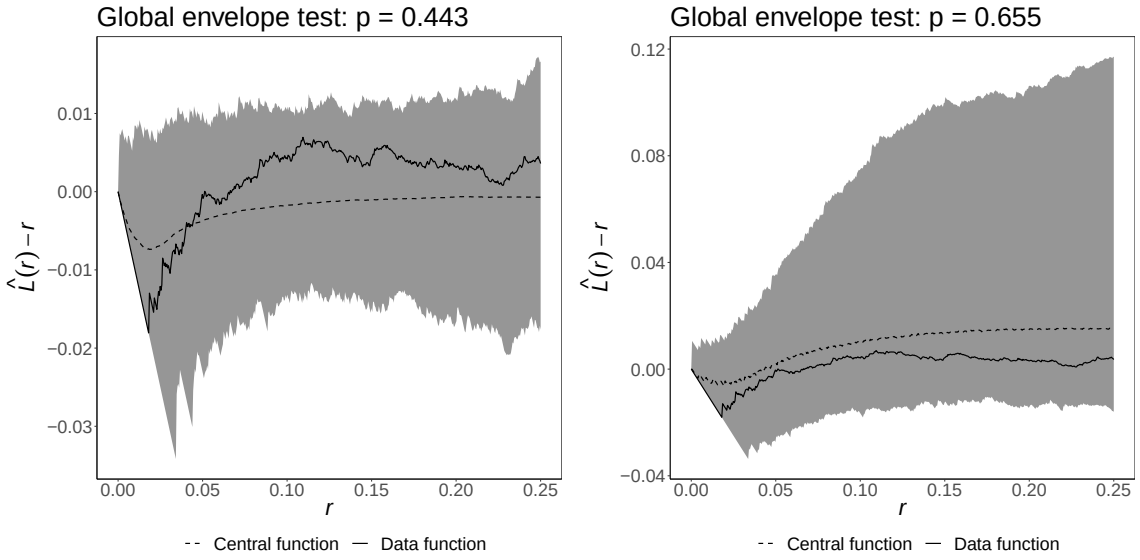


Figure 7.3: Global envelopes based on the bivariate L-function for the Strauss process (left-panel) and the simple LGCP-Strauss model (right-panel). The solid black line corresponds to the summary statistic estimate for the Japanese pines data set and the dotted line to the mean of the 1000 posterior predictions. On top of each plot is the p-value of the corresponding global envelope test.

model can be seen in Figure 7.3. The `GET` package computes p-values for the tests, which are shown in Figure 7.3. We see that the p-value for the new model is larger than that of the Strauss process, indicating that it is a better fit to the data. It is a significant result in the context of this thesis that the simple LGCP-Strauss model gave a better fit to this data set than a bivariate Strauss process as the Japanese pines set does not show much aggregation on the large scale, and is thus not an ideal candidate for our model. This goes to show the flexibility of our new model.

Chapter 8

Conclusion and recommendations for future work

8.1 Conclusion

In this thesis, we have presented and analysed an extension to the model of Vihrs et al. (2020) that seeks to capture the phenomenon of large-scale aggregation and small-scale inhibition within the same process. In the first part, we provided the foundations of spatial point processes necessary for an understanding of our model. We then proceeded to introduce the model in its full form and produced realisations to demonstrate its flexibility and effective capturing of the desired multi-scale interactions. From here, an ABC procedure was implemented to try to recover the parameters of the full and simple models on a simulated point pattern. Time constraints placed limitations of the precision of this procedure, as the tolerance threshold ε had to be taken as 10% over an ideal 1%, or lower, to allow for a manageable computation time. Even with this choice of ε an entire pilot and ABC run took, on average, around 5 days. Unfortunately, little success was had with the full model due to heavy interactions between parameters. If time had allowed, alternative summary statistics and ways to restrict parameters would have been explored.

Despite the large tolerance level, the ABC procedure was found to effectively recover the parameters of the simulated patterns in the case of the simple model. Encouraged by this result, the same procedure was applied to the Japanese black pines data set (Numata 1964). The recovered approximate posteriors varied greatly from their uniform priors, which gave us confidence that the method worked as desired. To check this, we used a global envelope test to check the fit, and found that the resulting p-value demonstrated that the model was fitted well to the data, and that it captured the behaviour better than a bivariate Strauss process could. This clearly shows the effectiveness of our new model, and its potential for future application.

8.2 Recommendations for future work

The ABC proceudre of Vihrs et al. (2020) that was used effectively on the simple model did not appear to work on the full bivariate LGCP-Strauss process. Future work may focus on examining alternative summary statistics that are able to distinguish between the effect of each of the many parameters. Alternatively, a different ABC procedure, or even a non-Bayesian approach, could be adopted and applied to this model.

Another way to extend the work given here would be to consider spatial inhomogeneities.

Specifically, one could look to incorporate covariate information into the mean function of the underlying linear combination of Gaussian random fields that drives the large-scale aggregation in our model.

Bibliography

Baddeley, A. (2006), 'Spatial point processes and their applications', *Stochastic Geometry: Lectures given at the C.I.M.E. 2004, Lecture Notes in Mathematics 1892* .

Baddeley, A. J., Møller, J. & Waagepetersen, R. (2000), 'Non- and semi-parametric estimation of interaction in inhomogeneous point patterns', *Statistica Neerlandica* **54**(3), 329–350.
URL: *https://onlinelibrary.wiley.com/doi/abs/10.1111/1467-9574.00144*

Baddeley, A. & Nair, G. (2012), 'Fast approximation of the intensity of Gibbs point processes', *Electronic Journal of Statistics* **6**, 1155 – 1169.
URL: *https://doi.org/10.1214/12-EJS707*

Baddeley, A., Rubak, E. & Turner, R. (2015), *Spatial Point Patterns: Methodology and Applications with R*, Chapman and Hall/CRC Press, London.
URL: *https://www.routledge.com/Spatial-Point-Patterns-Methodology-and-Applications-with-R/Baddeley-Rubak-Turner/9781482210200/*

Besag, J. (1977), 'Contribution to the discussion on Dr. Ripley's paper', *Journal of the Royal Statistical Society: Series B (Methodological)* **B39**(2), 193–195.
URL: *https://rss.onlinelibrary.wiley.com/doi/abs/10.1111/j.2517-6161.1977.tb01616.x*

Bourgault, G. & Marcotte, D. (1991), 'Multivariable variogram and its application to the linear model of coregionalization', *Mathematical Geology* **23**, 899–928.

Cox, D. (1955), 'Some statistical methods connected with series of events', *Journal of the royal statistical society series b-methodological* **17**, 129–157.

Diggle, P. (1985), 'A kernel method for smoothing point process data', *Journal of the Royal Statistical Society. Series C (Applied Statistics)* **34**(2), 138–147.
URL: *http://www.jstor.org/stable/2347366*

Fearnhead, P. & Prangle, D. (2011), 'Constructing summary statistics for approximate bayesian computation: Semi-automatic abc'.

Geyer, C. J. & Møller, J. (1994), 'Simulation procedures and likelihood inference for spatial point processes', *Scandinavian Journal of Statistics* **21**(4), 359–373.
URL: *http://www.jstor.org/stable/4616323*

Lotwick, H. W. & Silverman, B. W. (1982), 'Methods for analysing spatial processes of several types of points', *Journal of the Royal Statistical Society. Series B (Methodological)* **44**(3), 406–413.
URL: *http://www.jstor.org/stable/2345499*

McSwiggan, G. (2019), Spatial point process methods for linear networks with applications to road accident analysis, PhD thesis, The University of Western Australia.

Møller, J. & Waagpetersen, R. (2004), *Statistical inference and simulation for spatial point processes*, Chapman and Hall/CRC.
URL: *https://doi.org/10.1201/9780203496930*

Mrkvička, T., Myllymäki, M., Jílek, M. & Hahn, U. (2020), 'A one-way anova test for functional data with graphical interpretation', *Kybernetika* p. 432–458.
URL: *http://dx.doi.org/10.14736/kyb-2020-3-0432*

Myllymäki, M., Mrkvička, T., Grabarnik, P., Seijo, H. & Hahn, U. (2017), 'Global envelope tests for spatial processes', *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* **79**, 381–404.
URL: *https://dx.doi.org/10.1111/rssb.12172*

Numata (1964), 'Forest vegetation, particularly pine stands in the vicinity of choshi-flora and vegetation at choshi, chiba prefecture, vi (in japanese)', *Bulletin of the Choshi Marine Laboratory, No 6* pp. 27–37.

Ogata, Y. & Tanemura, M. (1985), 'Estimation of interaction potentials of marked spatial point patterns through the maximum likelihood method', *Biometrics* **41**(2), 421–433.
URL: *http://www.jstor.org/stable/2530867*

Platt, W. J., Evans, G. W. & Rathbun, S. L. (1988), 'The population dynamics of a long-lived conifer (pinus palustris)', *The American Naturalist* **131**(4), 491–525.
URL: *https://doi.org/10.1086/284803*

Pritchard, J., Seielstad, M., Perez-Lezaun, A. & Feldman, M. (2000), 'Population growth of human y chromosomes: A study of y chromosome microsatellites', *Molecular biology and evolution* **16**, 1791–8.

Rasmussen, C. E. & Williams, C. K. I. (2005), *Gaussian Processes for Machine Learning (Adaptive Computation and Machine Learning)*, The MIT Press.

Ripley, B. D. (1976), 'The second-order analysis of stationary point processes', *Journal of Applied Probability* **13**(2), 255–266.

Stein, M. (2012), *Interpolation of Spatial Data: Some Theory for Kriging*, Springer Series in Statistics, Springer New York.
URL: *https://books.google.co.uk/books?id=aZXwBwAAQBAJ*

Strauss, D. J. (1975), 'A model for clustering', *Biometrika* **62**(2), 467–475.
URL: *https://doi.org/10.1093/biomet/62.2.467*

Van Lieshout, M. N. M. (2000), *Markov point processes and their applications*, Imperial College Press, London.

Vihrs, N., Møller, J. & Gelfand, A. E. (2020), 'Approximate bayesian inference for a spatial point process model exhibiting regularity and random aggregation'.

Zhang, H. (2004), 'Inconsistent estimation and asymptotically equal interpolations in model-based geostatistics', *Journal of the American Statistical Association* **99**(465), 250–261.
URL: *https://doi.org/10.1198/016214504000000241*

Appendix A

Supplementary plots for inference in the general model

In this appendix, we give some supplementary plots for the discussion on inference in the general model.

A.1 Scaling the A-matrix

The plots in this section are of realisations in which we scale the matrix in successive figures. The purpose of this exercise being to demonstrate the effect that increasing the magnitude of the entries of the matrix A (whilst keeping them in proportion) has on the observed pattern.

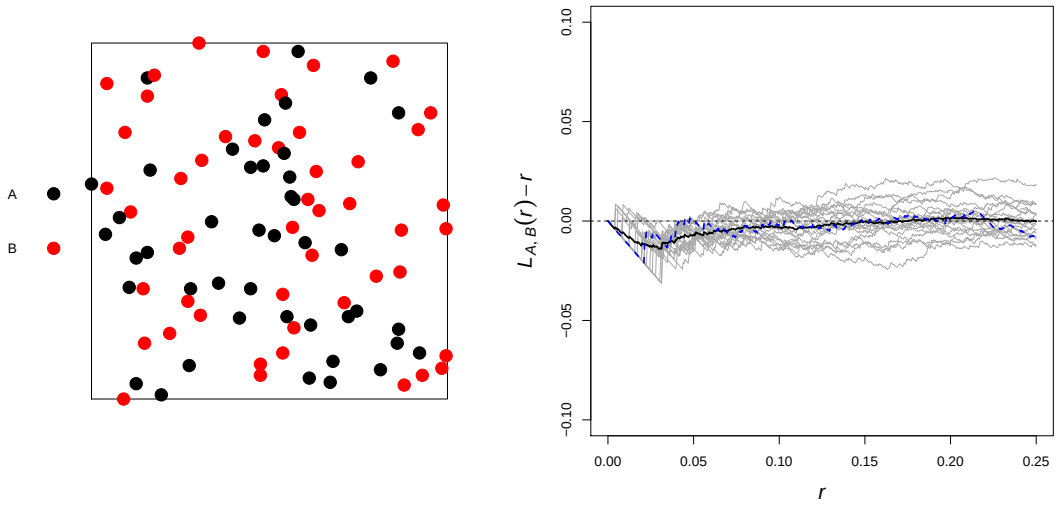


Figure A.1: The right-panel shows bivariate L-function estimates for 20 realisations of the bivariate LGCP-Strauss model in grey, with their mean in black and the observed pattern in blue, which is shown on the left. The realisations all take $\mu_1 = \mu_2 = 5$, $s_1 = 0.2$, $s_2 = 0.5$, $\sigma_1 = \sigma_2 = 1$, $R_1 = 0.05$, $R_2 = 0.02$, $R = 0.03$, $\gamma_1 = 0.7$, $\gamma_2 = 0.8$, $\gamma = 0.3$ and $A = A_0 := \begin{pmatrix} 0.2 & 0.1 \\ 0.1 & 0.1 \end{pmatrix}$.

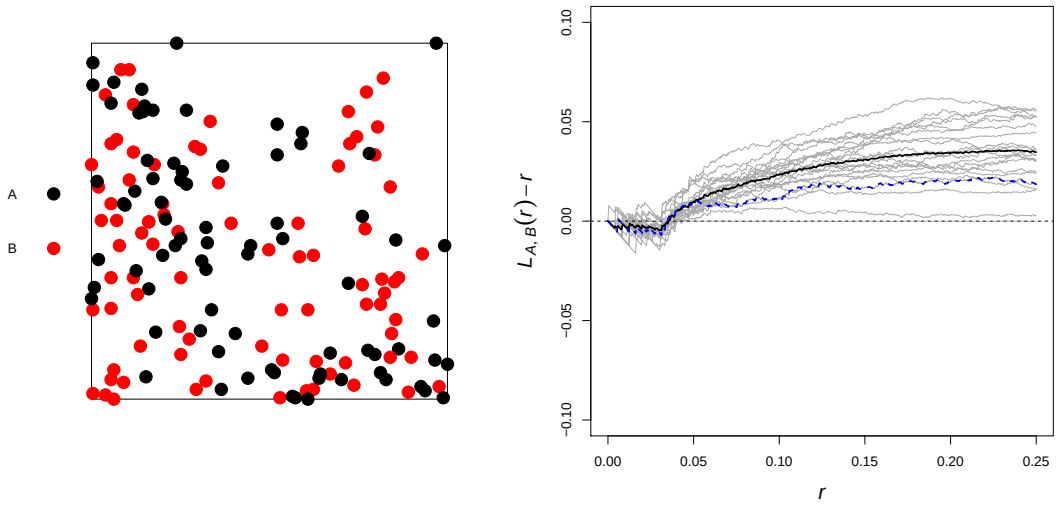


Figure A.2: The plots are identical to those of Figure A.1, except that the matrix $A = 10A_0$ in these realisations.

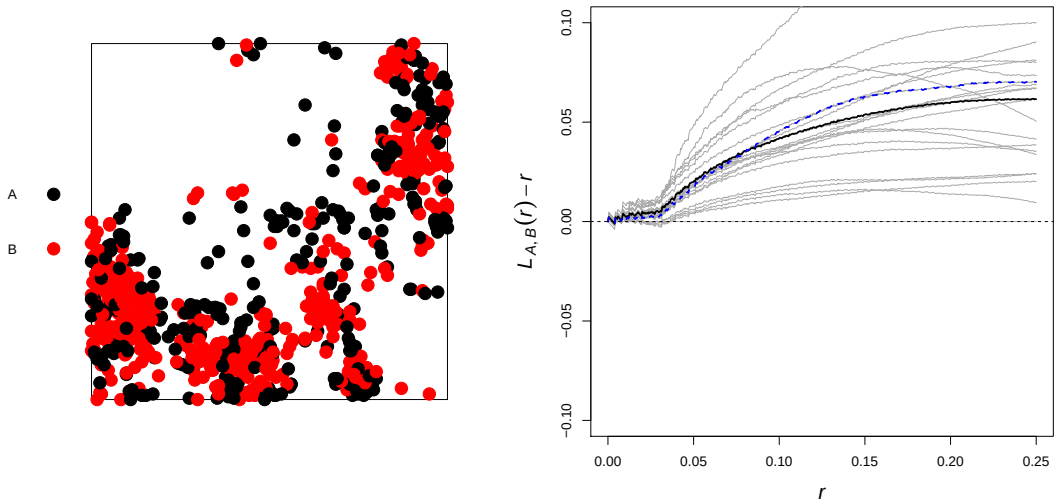


Figure A.3: The plots are identical to those of Figure A.1, except that the matrix $A = 50A_0$ in these realisations.

A.2 Including more random fields

In this section, each plot shows two sets of L-function estimates. The left panel of each plot corresponds to realisations using linear combinations of $k = 2$ random fields. The right panels show realisations based on $k = 3$ random fields, where the first 2 have identical parameters to those in the left-panel. Essentially, these plots look to show the negligible impact of including more than 2 underlying random fields in the bivariate LGCP-Strauss model. The figures are presented on the following pages.

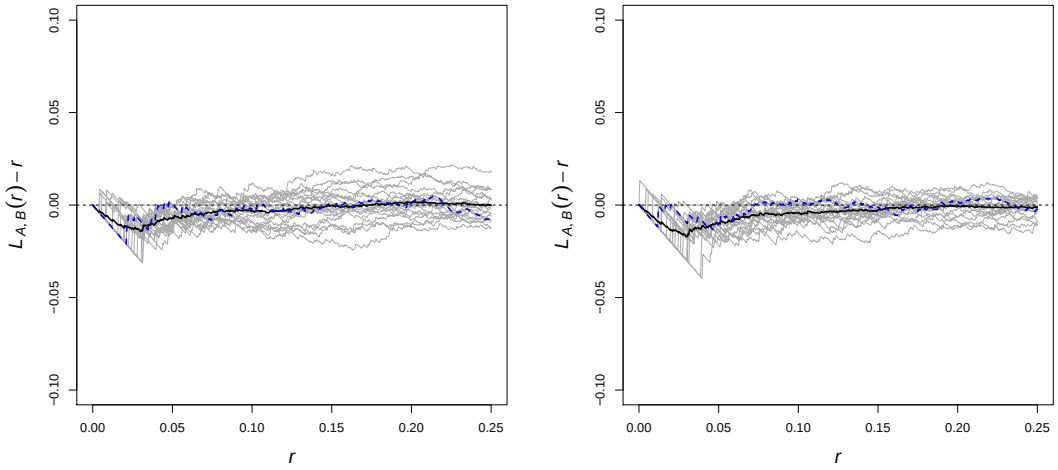


Figure A.4: Each plot shows bivariate L-function estimates for 20 realisations of the bivariate LGCP-Strauss model in grey, with their mean in black and the observed pattern in blue (observed patterns not plotted). Both panels take $\mu_1 = \mu_2 = 5$, $s_1 = 0.2$, $s_2 = 0.5$, $\sigma_1 = \sigma_2 = 1$, $R_1 = 0.05$, $R_2 = 0.02$, $R = 0.03$, $\gamma_1 = 0.7$, $\gamma_2 = 0.8$, $\gamma = 0.3$. The plots on the left use $A = A_1 := \begin{pmatrix} 0.2 & 0.1 \\ 0.1 & 0.1 \end{pmatrix}$ and those on the right use $A = A_1 := \begin{pmatrix} 0.2 & 0.1 & 0.3 \\ 0.1 & 0.1 & 0.2 \end{pmatrix}$ so that they are identical apart from the right-panel realisations using an additional random field.

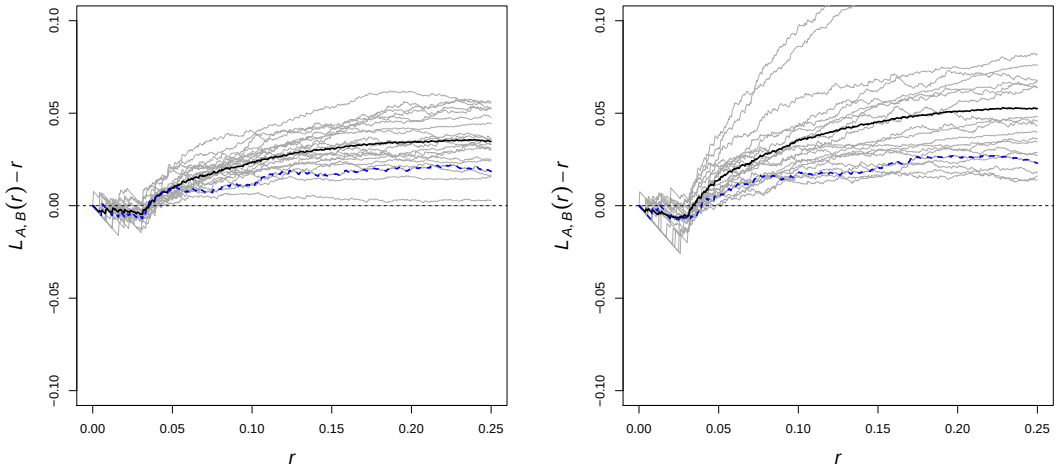


Figure A.5: The plots are identical to those in Figure A.4, other than using $A = 10A_1$ in the left-panel and $A = 10A_2$ on the right.

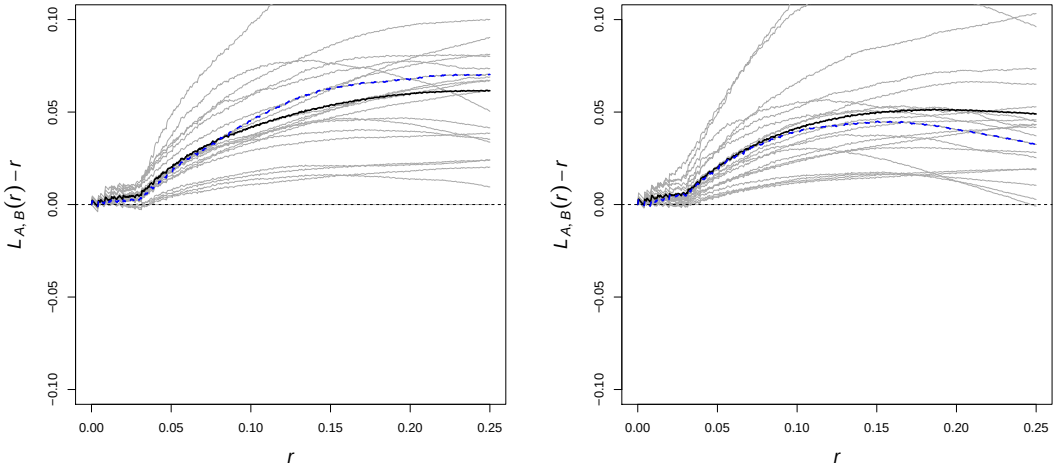


Figure A.6: The plots are identical to those in Figure A.4, other than using $A = 50A_1$ in the left-panel and $A = 50A_2$ on the right.