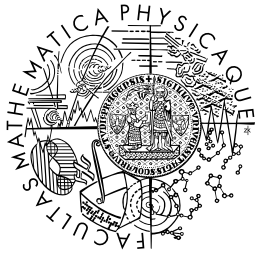

FACULTY
OF MATHEMATICS
AND PHYSICS
Charles University

MASTER THESIS

Diana Pavlovičová

Bayesian inference for anisotropic cluster point processes

Department of Probability and Mathematical Statistics

Supervisor of the master thesis: RNDr. Jiří Dvořák, Ph.D.

Study programme: Probability, Mathematical Statistics and Econometry

Prague 2024

I declare that I carried out this master thesis on my own, and only with the cited sources, literature and other professional sources. I understand that my work relates to the rights and obligations under the Act No. 121/2000 Sb., the Copyright Act, as amended, in particular the fact that the Charles University has the right to conclude a license agreement on the use of this work as a school work pursuant to Section 60 subsection 1 of the Copyright Act.

In ..... date ....

Author's signature

I would like to thank my supervisor, RNDr. Jiří Dvořák, Ph.D., for his tremendous help, support, patience and for the words of encouragement. I would also like to thank my family, friends and my boyfriend for their support and understanding, and most importantly for always believing in me, even when I did not believe in myself. I would not have done this without you all. Thank you.

Title: Bayesian inference for anisotropic cluster point processes

Author: Diana Pavlovičová

Department: Department of Probability and Mathematical Statistics

Supervisor: RNDr. Jiří Dvořák, Ph.D., Department of Probability and Mathematical Statistics

Abstract: Point processes are stochastic models widely used in biology, forestry, or astronomy. In this thesis, we are going to deal mainly with anisotropic cluster point processes. We present a new method for estimating parameters of such models. The basis of this method is the use of Bayesian statistics combined with Markov Chain Monte Carlo algorithms, which are a useful way to estimate parameters which are difficult or impossible to estimate using traditional methods. We describe the method in detail and present several examples of its application to simulated and real-life datasets and discuss the difficulties associated with it. Finally, we prove theoretical results about the convergence of the corresponding Markov chain under specific assumptions on the model and discuss the difficulties we encounter when examining these properties.

Keywords: anisotropy, point process, covariates, parameter estimation, Bayesian statistics, Markov Chain Monte Carlo, cluster point process

Název práce: Bayesovská analýza anizotropních shlukových bodových procesů

Autor: Diana Pavlovičová

Katedra: Katedra pravděpodobnosti a matematické statistiky

Vedoucí bakalářské práce: RNDr. Jiří Dvořák, Ph.D., Katedra pravděpodobnosti a matematické statistiky

Abstrakt: Bodové procesy jsou stochastické modely, které se hojně využívají v biologii, lesnictví, nebo například astronomii. V této diplomové práci se budeme zabývat především anisotropními shlukovými bodovými procesy. Představíme novou metodu, kterou lze využít k odhadům parametrů těchto modelů. Základem této metody je využití bayesovské statistiky spolu s metodami Monte Carlo, které jsou užitečným nástrojem pro odhadování parametrů, jež není možné odhadnout pomocí tradičních metod, nebo jejichž odhadování je obtížné. Metodu detailně popíšeme a ukážeme její funkčnost. Dále provedeme několik analýz simulovaných a reálných dat a budeme diskutovat obtíže s touto metodou spojené. Nakonec odvodíme teoretické závěry o konvergenci odpovídajícího markovského řetězce za specifických předpokladů kladených na použitý model a diskutujeme problémy, jež vyvstávají při zkoumání jeho konvergenčních vlastností.

Klíčová slova: anisotropie, bodový proces, kovariáty, odhady parametrů, bayesovská statistika, Markov Chain Monte Carlo, shlukový bodový proces

Contents

Introduction

Spatial point processes are useful models when we want to capture the randomness of the positions of the observed units. They are extensively used in geography, epidemiology, astronomy, forestry or computational neuroscience and biology. By using point processes, we can model the spreading of a disease, stars in galaxies, the distribution of plant species in varied terrain or even the distribution of cells of a patricular type in the human or animal body. Since clustering is a natural phenomenon in all of the aforementioned, cluster point processes are still widely studied today.

Let us introduce some examples of the use of point processes in real life examples. The dataset used in Figure 1 (left) is from Møller and Toftaker (2014). We can see the locations of 110 Welsh chapels rescaled to the unit square. As we can see in the figure, the chapels are clustered in the smaller scale and also show signs of dependence on direction in the larger scale. While the clustering may be a sign that there is a larger town or a city nearby, the dependence on the direction could be caused by the terrain or water bodies. Another example of the usage of point processes is illustrated in Figure 1 (right), where there are crimes recorded over the course of approximately two weeks near the University in Chicago. These crimes can also be divided according to the type of the offence into several smaller groups, and one could ask for example whether most of the serious offences took place near some specific locations.

We are particulary interested in point processes that show signs of dependence on direction, which is called anisotropy. Realization of two cluster point processes can be seen in Figure 2. Here, the left picture shows a cluster point process without any directional dependence. On the other hand, the right picture shows a strong directional dependence in the horizontal direction and one can also see that the more on the right in the picture we are, the more points there are in the clusters. This is a property that we can try to explain using available covariates.

In the analysis we do, Bayesian statistics plays a significant role. It is a relatively new way on how to view statistical methods. While the Bayes' Theorem comes from the 18^{th} century, the Bayesian view on probability did not come to life up until the 20^{th} century. It is an interesting way how to interpret unknown parameters that is widely used in business and stock markets.

We use the Bayesian approach to show how one can explore the model parameters. We first give a brief description of Bayesian methods and then describe how we use them in our setting. In the next part, we talk about the Markov Chain Monte Carlo methods, which are a popular extension to the Bayesian statistics. One of the most popular techniques used by statisticians up until today is the *Metropolis-Hastings algorithm*, which can be traced back to Metropolis et al. (1953) and Hastings (1970). We use this particular algorithm as a tool for our analysis, combined with the *Gibbs sampler*, the foundations of which were also laid in the aforementioned articles.

In this thesis, we focus on exploring a model that generalizes the model introduced

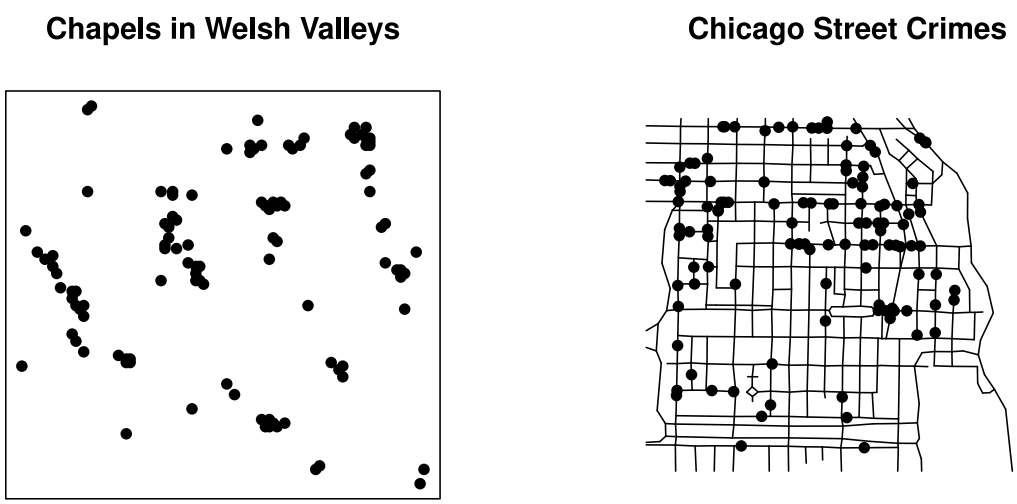

Figure 1 Locations of Welsh chapels in rescaled to the unit square $[0,1]^2$ (left) and crimes reported in the period of circa two weeks near the University of Chicago (right).

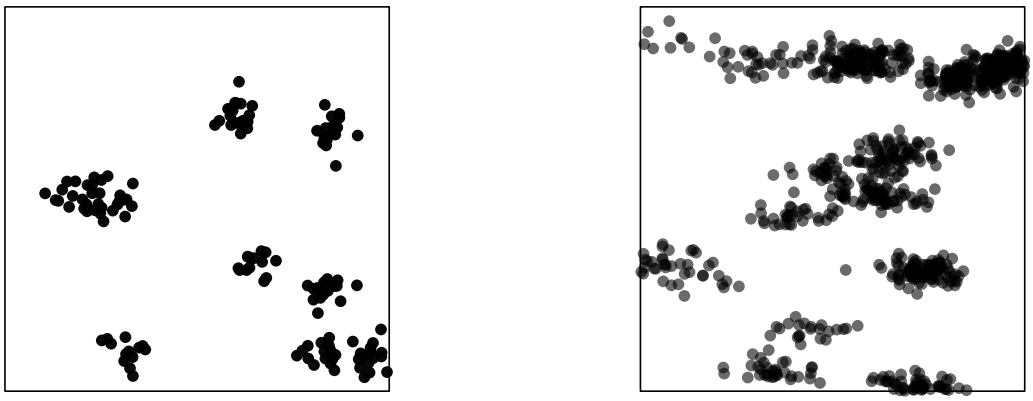

Figure 2 Two realizations of cluster point processes in the unit square window $[0,1]^2$. An isotropic cluster point process (left) and an anisotropic cluster point process (right).

in Dvořák et al. (2022). In this article, the main focus is on introducing complex inhomogeneities into cluster point process models and explaining how the package binspp for the software R can be used. We focus on adding possible anisotropy into this model and explore how one can extend the methods described therein onto anisotropic point processes.

One of the objectives of this thesis is to create a model from which one could simulate realizations of spatial anisotropic cluster point processes, which are possibly dependent on covariates. Using these simulations, we then examine the properties of the Bayesian inference and MCMC methods in detail. We show the results from several simulated datasets, as well as a result from real-life data. We also introduce the dependence on a covariate and show how the algorithm could be used for testing hypotheses about the significance of the dependencies.

1. Point Processes in $\mathbb{R}^d$

In this chapter we recall the definition of the point processes in $\mathbb{R}^d$. We then describe the particular model that we are going to be studying in this thesis, the anisotropic cluster point process. We then give some examples of point processes closely related to the topic of this thesis. Lastly, we explain the model we are going to work with and introduce parametrizations that will be subsequently used. Both Section 1.3 and 1.4 are based on the article Dvořák et al. (2022) as they generalize the model described therein.

1.1 Definition of anisotropic cluster point process

The aim of this section is to properly define an *anisotropic cluster point process*. Most of the definitions in this section are taken from Rataj (2006). To be able to define such object, we first have to introduce sets of measures with specific properites. By $\mathbb{R}^d$, we denote the d-dimensional Euclidean space, $d \in \mathbb{N}$. In what follows, we use the notation on the space $\mathbb{R}^d$:

$\mathcal{B}(\mathbb{R}^d)$ Borel σ-algebra on $\mathbb{R}^d$

$\mathcal{B}_0(\mathbb{R}^d)$ bounded Borel subsets of $\mathbb{R}^d$

$\mathcal{K}(\mathbb{R}^d)$ compact sets on $\mathbb{R}^d$.

We shall abbreviate these, for example we write $\mathcal{B}$ instead of $\mathcal{B}(\mathbb{R}^d)$.

Definition 1. *A measure μ (i.e. a nonnegative σ-additive set function) on $(\mathbb{R}^d, \mathcal{B})$ is said to be locally finite if it is finite on $\mathcal{B}_0$.*

By $\mathcal{M} \equiv \mathcal{M}(\mathbb{R}^d)$ we denote the space of all locally finite measures on $(\mathbb{R}^d, \mathcal{B})$ and by

$$\mathcal{M}_f \equiv \mathcal{M}_f(\mathbb{R}^d) = \left\{\mu \in \mathcal{M} : \mu(\mathbb{R}^d) < \infty\right\},$$

we denote the space of all finite measures on $(\mathbb{R}^d, \mathcal{B})$. Moreover, we denote

$$\mathcal{N} \equiv \mathcal{N}(\mathbb{R}^d) = \{\mu \in \mathcal{M} : \mu(B) \in \mathbb{N} \cup \{0,\ \infty\}\ \forall B \in \mathcal{B}\}$$

the space of all locally finite counting measures on $(\mathbb{R}^d, \mathcal{B})$ and $\mathcal{N}_f = \mathcal{M}_f \cap \mathcal{N}$ the space of all finite counting measures.

Notation. By $\sigma\{\mu \mapsto h(\mu)\ measurable,\ h \in H\}$ we denote the smallest σ-algebra on $\mathcal{M}$ with respect to which all the functions h from the system of functions H on $\mathcal{M}$ are measurable.

Definition 2. *Let us introduce the following σ-algebras on $\mathcal{M}$ and $\mathcal{N}$:*

$$\mathfrak{M} = \sigma\{\mu \mapsto \mu(B)\ measurable,\ B \in \mathcal{B}\},$$
$$\mathfrak{N} = \{M \cap \mathcal{N} :\ M \in \mathfrak{M}\}.$$

Definition 3. *Let $(\Omega, \mathcal{A}, \mathbb{P})$ be a probability space. A measurable mapping*

$$\Psi : (\Omega, \mathcal{A}, \mathbb{P}) \to (\mathcal{M}, \mathfrak{M})$$

is called a random measure on $\mathbb{R}^d$*. The probability measure* $Q = \mathbb{P}\,\Psi^{-1}$ *is the distribution of the random measure* Ψ *and the measure* $\Lambda(\cdot) = \mathbb{E}\,\Psi(\cdot)$ *is the intensity measure of the random measure* Ψ*.*

Remark. The probability measure Q can be alternatively defined as

$$Q(\mathcal{U}) = \mathbb{P}\left(\{\omega \in \Omega : \Psi(\omega) \in \mathcal{U}\}\right), \quad \mathcal{U} \in \mathfrak{M}.$$

Now we have all the ingredients to define a *point process.*

Definition 4. *A point process on* $\mathbb{R}^d$ *is a measurable mapping*

$$\Phi : (\Omega, \mathcal{A}, \mathbb{P}) \to (\mathcal{N}, \mathfrak{N}).$$

A point process is simple if $\mathbb{P}\left[\Phi \in \mathcal{N}^*\right] = 1$*, where*

$$\mathcal{N}^* = \left\{\nu \in \mathcal{N} : \ \nu(\{x\}) \leq 1 \text{ for every } x \in \mathbb{R}^d\right\}.$$

Remark. Correctness of the definition of a simple point process is guaranteed by Lemma 4.2 in Rataj (2006), which states that $\mathcal{N}^*$ is measurable in $\mathfrak{N}$.

Remark. A point process is a special case of a random measure.

Remark. A simple point process could be understood as a measurable mapping $\Phi : (\Omega, \mathcal{A}, \mathbb{P}) \to (\mathcal{N}^*, \mathfrak{N}^*)$, where $\mathfrak{N}^* = \{\mathcal{U} \cap \mathcal{N}^* : \mathcal{U} \in \mathfrak{N}\}$ is the trace of the σ-algebra $\mathfrak{N}$ on $\mathcal{N}^*$.

Definition 5. *Let* $\Lambda \in \mathcal{M}$ *and* Φ *be a point process on* $\mathbb{R}^d$ *such that for all* $n \in \mathbb{N}$ *and* $B_1, \ldots, B_n \in \mathcal{B}_0$ *pairwise disjoint it holds that*

i. random variables $\Phi(B_1), \ldots, \Phi(B_n)$ *are independent,*

ii. $\Phi(B_i)$ *has a Poisson distribution with parameter* $\Lambda(B_i)$*.*

Then Φ *is called a Poisson point process with intensity measure* Λ*.*

Remark. The existence and uniqueness of random variables from Definition 5 is given by Theorem 3.2 and Corollary 3.1 in Rataj (2006).

Definition 6. *For* $z \in \mathbb{R}^d$ *we denote* t_z *the shift operator on* $\mathcal{M}$*, i.e.*

$$(t_z\mu)(A) = \mu(A - z), \quad \mu \in \mathcal{M},\ A \in \mathcal{B}.$$

We say that a random measure Φ *on* $\mathbb{R}^d$ *is stationary if* $t_z\Phi \overset{d}{=} \Phi$ *for all* $z \in \mathbb{R}^d$*, i.e., if its distribution* Q *satisfies* $Qt_z^{-1} = Q$ *for all* $z \in \mathbb{R}^d$*. We say that* Q *is translation invariant.*

These next two definitions are taken from Pawlas (2023).

Definition 7. *Denote by* $\mathcal{N}_f^* = \mathcal{N}_f \cap \mathcal{N}^*$ *the space of all simple finite counting measures on* $\mathbb{R}^d$*.*

Definition 8. *Let* $\tilde{\Phi}$ *be a simple point process on* $\mathbb{R}^d \times \mathcal{N}_f^*$*. Assume that the point process* $\Phi_p(\cdot) = \tilde{\Phi}(\cdot \times \mathcal{N}_f^*)$ *is a simple point process (so-called parent point process). Define*

$$\Phi(B) = \sum_{(X,\zeta) \in supp\tilde{\Phi}} (t_X\zeta)(B), \quad B \in \mathcal{B},$$

and assume that $\Phi(K) < \infty$ *for all* $K \in \mathcal{K}$ *with probability* 1. *Then* Φ *is a point process a.s. and it is called a cluster point process. For* $(X, \zeta) \in$ *supp* $\tilde{\Phi}$, *we refer to* $t_X\zeta$ *as the daughter point process associated with the parent point* X.

Definition 9. *Let* Ψ *be a random measure with intensity measure* Λ. *If there exists a density* λ *of* Λ *w.r.t. the Lebesgue measure, i.e.,*

$$\Lambda(B) = \int_B \lambda(x)\, dx, \quad B \in \mathcal{B},$$

then λ *is called the intensity function. If it is a constant, we call it intensity.*

Lastly, we define anisotropy.

Definition 10. *For a rotation* $\mathcal{O}$ *around the origin, we denote by* $R_{\mathcal{O}}$ *the rotation operator on* $\mathcal{M}$ *as*

$$(R_{\mathcal{O}}\mu)(A) = \mu(\mathcal{O}^{-1}A), \quad \mu \in \mathcal{M}, A \in \mathcal{B}.$$

A random measure Ψ *on* $\mathbb{R}^d$ *is called isotropic if* $R_{\mathcal{O}}\Psi$ *and* Ψ *have the same distribution for any rotation* $\mathcal{O}$, *i.e., the distribution of the random measure is rotation-invariant. Otherwise, the measure* Ψ *is said to be anisotropic.*

1.2 Examples of point processes

In this section, we show some interesting examples of point processes and briefly mention their usage in modelling real life situations. Most of the examples in this section are taken from Møller and Waagepetersen (2004).

One of the most widely used point process models is the *Poisson point process* from Definition 5. It is a useful model widely used in geography, as it is the most natural model when considering *complete spatial randomness*. This essentialy means that if we have an observation window, then the number of points in every bounded subwindow also follows a Poisson distribution and is independent of the number of points in all the other subwindows, which are all mutually disjoint. It is a very natural setting in situations where we do not observe any dependence of our observations. With this process, we can model for example the incidence of non-contagious diseases, but it is mostly used as a theoretical tool, for reference values, since we mostly observe some kind of dependence in practical use. Three different realizations of Poisson point process can be seen in Figure 1.1.

Because of the fact that the points in the Poisson point process are independent, it is not of use in situations where dependencies can be detected. To model these situations, cluster point processes may be of great use.

We are now going to introduce a widely used type of a cluster point process from Definition 8, which is the Neyman-Scott process first introduced in Neyman and Scott (1958). A realization of this process can be seen in Figure 1.2 on the left.

Definition 11. *Let* C *be a stationary Poisson process on* $\mathbb{R}^d$ *with intensity* $\kappa > 0$. *Conditional on* C, *let* X_c, $c \in C$, *be independent processes on* $\mathbb{R}^d$ *where* X_c *has*

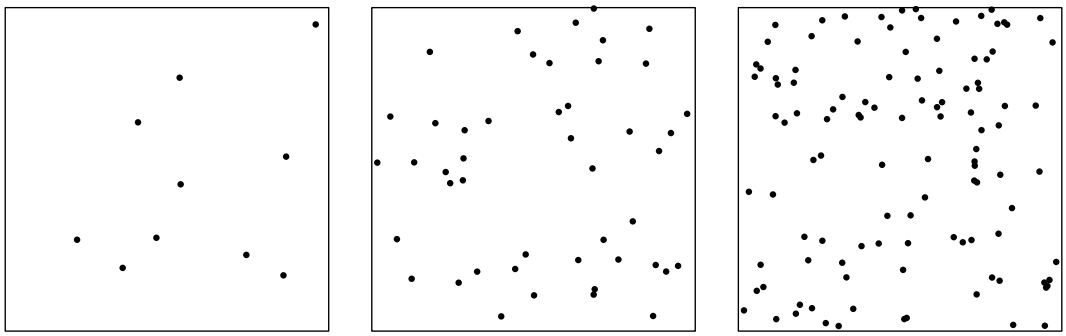

Figure 1.1 Three different realizations of a homogeneous Poisson point process in the unit window $W = [0,1]^2$ with intensities $\lambda = 10,\ 50,\ 100$ respectively. The true numbers of observed points are 10, 46 and 110.

intensity function

$$\rho_c(\xi) = \alpha \cdot k(\xi - c), \quad \xi \in \mathbb{R}^d$$

where $\alpha > 0$ is a parameter and k is a kernel, i.e., for all $c \in \mathbb{R}^d$, $\xi \mapsto k(\xi - c)$ is a probability density function. Then $X = \bigcup_{c \in C} X_c$ is a Neyman-Scott process with cluster centres C and clustres X_c, $c \in C$.

This is the model that we are going to be working with later on. In this model, the parent point process is not observed and we observe the union of the daughter point processes. This model was suggested as a way to model the clustering of galaxies but can also be used in medicine for modelling the spread of contagious diseases or in biology for modelling populations of trees or plants. A special example of this family of point processes is the Thomas point process firstly introduced in Thomas (1949). Two different realizations of Thomas process are depicted in Figure 1.2 (middle and right).

Definition 12. *The Thomas process is a Neyman-Scott point process where*

$$k(\xi) = exp\left\{-\|\xi\|^2/(2\omega^2)\right\}/(2\pi\omega^2)^{d/2}, \quad \xi \in \mathbb{R}^d,$$

is the density for $N_d(0, \omega^2 I_d)$, where $\omega > 0$ is the standard deviation of random displacement of a daughter point from its parent point.

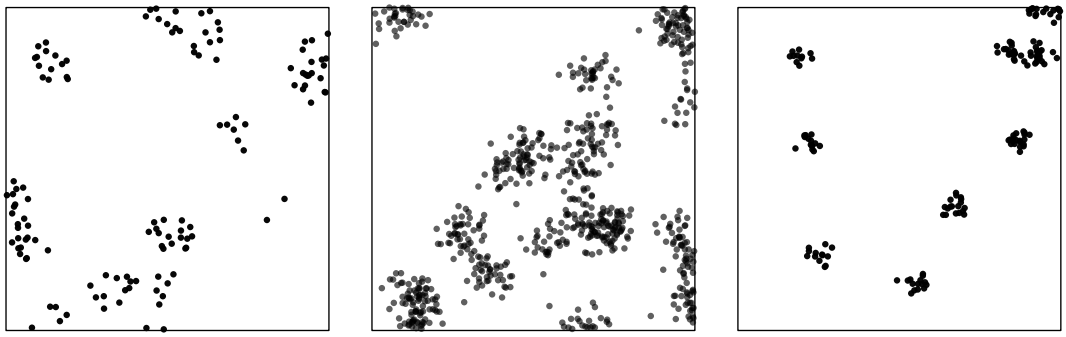

Figure 1.2 Realizations of cluster point processes in the unit window $W = [0,1]^2$. A Neymann-Scott process with daughter points distributed uniformly in a circle around each parent point (left) with $\kappa = 10$, $\alpha = 10$ and radius of the circle is 0.07. Two realizations of Thomas point process with $\kappa = 20$, $\omega = 0.04$ and $\alpha = 40$ (middle) and $\kappa = 10$, $\omega = 0.02$ and $\alpha = 15$ (right).

1.3 Anisotropic Thomas point process

In this section, we are going to introduce the anisotropic cluster point process model. We do this in the general way, introducing possible dependence on covariates for each of the considered model components.

The model can be understood as a so-called doubly stochastic model: firstly, we construct a Poisson point process which represents the parent point process and secondly, given the positions of the parent points, we construct independent clusters around each parent point. We will denote X the observed point process of daughter points and we will write $X = \bigcup_{c \in C} X_c$, where C is the parent point process being Poisson point process with intensity function $\kappa \cdot l(u;\beta)$, $u \in \mathbb{R}^2$, where $\kappa > 0$. The daughter point processes X_c, $c \in C$, are mutually independent Poisson point processes with intensity function $\alpha \cdot k(u - c; \sigma_1, \sigma_2, \rho)$, $\alpha > 0$, $u \in \mathbb{R}^2$, where k is the probability density function determining the displacement of the daughter points around its parent point and α stands for the expected number of daughter points per cluster. In our case, function k represents bivariate normal distribution centered at 0 with standard deviations $\sigma_1 > 0$, $\sigma_2 > 0$ in the directions of the x-axis and y-axis, respectively, and the correlation coefficient $\rho \in (-1, 1)$. Let us denote (c_1, c_2) coordinates of the parent point c, then we can write

$$k\left(\begin{pmatrix} x - c_1 \\ y - c_2 \end{pmatrix}; \sigma_1, \sigma_2, \rho\right) = \frac{exp\left\{-\frac{1}{2(1-\rho)^2}\left[\left(\frac{x-c_1}{\sigma_1}\right)^2 - 2\rho\left(\frac{x-c_1}{\sigma_1}\right)\left(\frac{y-c_2}{\sigma_2}\right) + \left(\frac{y-c_2}{\sigma_2}\right)^2\right]\right\}}{2\pi\sigma_1\sigma_2\sqrt{1-\rho^2}},$$

where $(x, y)^T \in \mathbb{R}^2$.

Remark. The daughter point process can be viewed as a *Cox process*, see Definition 5.1 in Møller and Waagepetersen (2004), because we view it, conditionally on the

parent point process, as an inhomogeneous Poisson point process with intensity function $\alpha \cdot k(\cdot - c; \sigma_1, \sigma_2, \rho)$.

As we previously mentioned, we are particularly interested in anisotropic cluster point processes. That means that we are interested in point processes that show signs of dependence on direction. In our study, we are not trying to investigate whether the cluster point process exhibits stationarity, i.e. shift-invariance, or not; although we will asume stationarity of the parent point process.

To introduce anisotropy into our model, we have several options how to do so: to have different standard deviations σ_1, σ_2, to have $\rho \neq 0$ or to introduce the anisotropy via covariates.

1.4 Parametrization of model components

We want to express the possible dependence of the model components on some spatial covariates, that could conceivably cause the process to be anisotropic. In real-life situations, this could be for example the influence of wind, altitude, terrain slope or distance from a water source.

Firstly, let us introduce the generalized model. Let us have the process of cluster centres C being the Poisson point process with its intensity function equal to $\kappa \cdot l(u; \beta)$, where $\kappa > 0$, $\beta \in \mathbb{R}^k$, $u \in \mathbb{R}^2$. For every cluster center $c \in C$, we have a cluster X_c that is the Poisson point process with intensity function $\alpha(c; \mu) \cdot k(u - c; \sigma_1(\nu_1, c), \sigma_2(\nu_2, c), \rho(\xi, c))$, $u \in \mathbb{R}^2$. Here, $\alpha(c; \mu)$ represents the expected number of points per cluster around parent point c and $\mu \in \mathbb{R}^{l+1}$ is a parameter. The daughter point processes follow a bivariate normal distribution centered around the parent point and the covariance matrix is formed with parameters $\sigma_1(\nu_1, c), \sigma_2(\nu_2, c), \rho(\xi, c)$. Here, $\nu_1 \in \mathbb{R}^{m+1}$, $\nu_2 \in \mathbb{R}^{n+1}$, $\xi \in \mathbb{R}^{o+1}$ are parameters. Therefore, we will introduce the following parametrization of the intensity function of the parent point process, α, σ_1, σ_2 and ρ. For the intensity function, α, σ_1 and σ_2, we will use exponentials which is a natural choice when we want the parameters to be strictly positive. For ρ, we will use parametrization via hyperbolic tangent, since we want this parameter to take values in the interval $(-1, 1)$. For the intensity function of the parent point process, we will write:

$$\kappa \cdot l(u; \beta) = \kappa \cdot exp\left\{\beta_1 z_1(u) + ... + \beta_p z_p(u)\right\},$$

where $p \in \mathbb{N}_0$, $\beta_1, ..., \beta_p \in \mathbb{R}$ are regression coefficients and $z_1, ..., z_p$ are spatial covariates influencing the parent point process. In the same manner, we define the following three parametrizations:

$$\begin{aligned} \alpha(c; \mu) &= exp\left\{\beta_0^\alpha + \beta_1^\alpha z_1^\alpha(c) + ... + \beta_q^\alpha z_q^\alpha(c)\right\} \\ \sigma_1(c; \nu_1) &= exp\left\{\beta_0^{\sigma_1} + \beta_1^{\sigma_1} z_1^{\sigma_1}(c) + ... + \beta_m^{\sigma_1} z_m^{\sigma_1}(c)\right\} \qquad (1.1) \\ \sigma_2(c; \nu_2) &= exp\left\{\beta_0^{\sigma_2} + \beta_1^{\sigma_2} z_1^{\sigma_2}(c) + ... + \beta_n^{\sigma_2} z_n^{\sigma_2}(c)\right\}, \qquad (1.2) \end{aligned}$$

where $q,\ m,\ n \in \mathbb{N}_0$, $\mu = \left(\beta_0^\alpha, ..., \beta_q^\alpha\right) \in \mathbb{R}^{q+1}$, $\nu_1 = \left(\beta_0^{\sigma_1}, ..., \beta_m^{\sigma_1}\right) \in \mathbb{R}^{m+1}$ and $\nu_2 = \left(\beta_0^{\sigma_2}, ..., \beta_n^{\sigma_2}\right) \in \mathbb{R}^{n+1}$ are parameters, $z_1^\alpha, ..., z_q^\alpha$ are spatial covariates influencing the cluster size, $z_1^{\sigma_1}, ..., z_m^{\sigma_1}$ are spatial covariates influencing the

standard deviation in the direction of the x-axis and $z_1^{\sigma_2}, \dots, z_n^{\sigma_2}$ are spatial covariates influencing the standard deviation in the direction of the y-axis.

For the correlation coefficient we define the parametrization

$$\rho(c;\xi) = tanh\left\{\beta_0^\rho + \beta_1^\rho z_1^\rho(c) + \dots + \beta_o^\rho z_o^\rho(c)\right\},$$

where $o \in \mathbb{N}_0$, $\xi = (\beta_0^\rho, \dots, \beta_o^\rho) \in \mathbb{R}^{o+1}$ is a parameter and $z_1^\rho, \dots, z_o^\rho$ are spatial covariates influencing the correlation coefficient.

Remark. We want to emphasise here that no intercept in the parametrization of the function $l(u;\beta)$ is needed since its role is taken by the additional parameter κ.

We would like to mention here that we are able to simulate realizations of point processes for every choice of $p,\ q,\ m,\ n,\ o$. If all of them are equal to 0, and moreover $\sigma_1 = \sigma_2$, we end up with the stationary Neymann-Scott point process. If $\sigma_1 \neq \sigma_2$, the possible anisotropy is introduced purely by the relative distribution of daughter points in each cluster, as they form the so-called 'Gaussian ellipses'. By adding covariates that could possibly also introduce different anisotropic behaviour, one could create a rather complex structure and construct anisotropic cluster point processes in the most general way.

We are going to restrict ourselves to the model where the parent point process is a homogeneous Poisson point process. We will denote its intensity by $\kappa > 0$. We also consider α to be a real one-dimensional parameter, i.e., independent of any covariates. We assume that the correlation coefficient ρ is equal to zero. We have chosen this model as it is the simplest anisotropic model to be constructed for which the functionality of the Bayesian inference could be explored. This also leads to the conclusion that only the standard deviations can be modelled using spatial covariates in our analysis.

1.4.1 Identifiability issues

As mentioned in Dvořák et al. (2022), there may be some issues when exploring the dependence on covariates. The problem is that the intensity function of the daughter points is very closely related to the intensity function of the parent points. If there were a covariate, say z_1, that would possibly influence both the mean value of daughter points per cluster and the intensity function of the parent point process, we would not be able to compute the corresponding regression parameters β_1 and β_1^α correctly. That is because we would have to estimate the vector β first, and that would lead to β_1 being completely different from what it should be, as it would cover also the influence of this covariate in the clusters. If the true value of β_1 were 0, β_1^α were a non-zero value, we would then estimate β_1 with approximately β_1^α, which could lead to false conclusions when testing whether a covariate influences the components or not.

2. Bayesian Inference

In this chapter, we are going to discuss the Bayesian approach to obtain some information about the model parameters. To begin with, we are going to introduce the basics of Bayesian statistics compared to frequentist statistics. Then we describe in detail how Bayesian inference is used in the model we are working with. Finally, we discuss how we can sample from the joint distribution of the parameters of interest.

2.1 Bayesian statistics

In this section, we state the basics of Bayesian statistics. Most of the terminology and principles are taken from Chapter 1 of the book Watanabe (2018).

Let us have a random element $\mathcal{X}$ on $(\mathbb{R}^d, \mathcal{B})$ and its one realization x. Let us assume that the random element depends on a vector of parameters, say $\theta \in \Theta \subseteq \mathbb{R}^d$. In Bayesian statistics, we assume that the vector θ is random, i.e., it has a probability density function w.r.t. some σ-finite measure υ. In such a case, the distinction between parameter estimation and prediction is blurred. On the other hand, frequentist approach views parameters as fixed (but still possibly uknown). In other words, Bayesian approach views parameters as random variables (or random vectors), whereas frequentist approach views them as non-random values (or vectors). The advantage of Bayesian approach is that it can model uncertainty about the parameters to be estimated.

The conditional probability density function $f(x|\theta)$ of $\mathcal{X}$ is called the *statistical model*. The probability density function $p(\theta)$ is then called *prior*. The aim of the Bayesian inference is to estimate the unknown *true distribution* of the parameters using the observations we have.

The Bayes' Theorem for conditional probability density functions says that if $\int_\Theta f(x|\theta)p(\theta)\upsilon(d\theta) > 0$, then we can write

$$p(\theta|x) = \frac{f(x|\theta)p(\theta)}{\int_\Theta f(x|\theta)p(\theta)\upsilon(d\theta)}, \quad \theta \in \Theta,\ x \in \mathbb{R}^d. \tag{2.1}$$

Here, $f(x|\theta)$ corresponds to the probability density function of the observed data, and $p(\theta)$ is the *prior probability density function* of parameter θ. In Bayesian inference, we are interested in $p(\theta|x)$, i.e., the probability density function of the unknown parameters conditionally on the observed realization x of $\mathcal{X}$. Since we cannot observe this distribution directly, we can use the Bayes' Theorem and rewrite the relation (2.1) as

$$p(\theta|x) \propto f(x|\theta)p(\theta),$$

where $\propto$ denotes equality up to a multiplicative constant. We refer to the left hand side of the expression (2.1) as the *posterior distribution*, $p(\theta)$ is referred to as *prior distribution* and $f(x|\theta)$ is called the *likelihood*. The prior distribution essentially represents our knowledge of the parameters before we see the data,

whereas the posterior distribution represents our knowledge of the parameters after we see the data.

2.2 Inference for anisotropic Thomas point process

We are going to apply the Bayesian approach to the model we are studying, see Section 1.3. Let us recall that C is the parent point process, which is assumed to be a homogeneous Poisson point process with intensity κ. We work on an observation window $W \subseteq \mathbb{R}^2$.

We are interested in the joint distribution of the parent point process C and the parameters, given the observed point process X of daughter points. The parameters of interest are:

κ intensity of the parent point process C

α the expected number of daughter points per cluster

ν_1 parameter of the standard deviation σ_1 in the bivariate normal distribution determining the relative displacement of daughter points around its parent point, see relation (1.1)

ν_2 parameter of the standard deviation σ_2 in the bivariate normal distribution determining the relative displacement of daughter points around its parent point, see relation (1.2).

Remark. In Bayesian statistics, the parent point process is usually taken as a nuisance parameter as it is not of particular interest in the inference about the model.

We use the so-called *modified Bayesian method*, which was proposed in Kopecký and Mrkvička (2016), where we do not take κ as a standard parameter for which we set prior distribution and then compute the joint distribution of all the parameters including κ, but rather 'fix' it in every step and recompute it from the other estimated parameters. Let us denote M the number of daughter points. For its expectation $\mathbb{E}\, M$ it holds that

$$\mathbb{E}\, M = \kappa \cdot \alpha \cdot |W|. \tag{2.2}$$

This means that everytime we get a new α, we can estimate

$$\kappa \approx \frac{\mathfrak{m}}{\alpha \cdot |W|}, \tag{2.3}$$

where $\mathfrak{m}$ denotes the observed number of daughter points. This method does not include κ in any probability density function from the Bayes' Theorem, as it does not have the same properties as other parameters.

We will denote $\theta = (\alpha, \nu_1, \nu_2)$. Let us assume that (C, θ) has a probability density function w.r.t. some σ-finite measure ι. From Bayes' Theorem, we have for the anisotropic Thomas point process that

$$p(C, \theta|X) = \frac{f(X|C,\theta)p(C|\theta)p(\theta)}{\int_{W\times\Theta} f(X|C,\theta)p(C|\theta)p(\theta)\iota(d(C,\theta))}. \tag{2.4}$$

The model parameters α, ν_1 and ν_2 are taken a priori mutually independent, and so we can break down the vector θ and rewrite (2.4) up to a multiplicative constant as

$$p(C, \alpha, \nu_1, \nu_2|X) \propto f(X|C, \alpha, \nu_1, \nu_2)p(C|\alpha)p(\alpha)p(\nu_1)p(\nu_2). \tag{2.5}$$

On the right hand side of (2.5), $p(C|\alpha)$ stands for the probability density function of the parent point process C given the parameter α w.r.t. the homogeneous unit Poisson point process, that is a Poisson point process with $\lambda = 1$, see Definitions 5 and 9. It is in fact a probability density function parametrized by κ, but since we have the relation (2.2), we can view this probability density function as parametrized by α. The functions $p(\alpha)$, $p(\nu_1)$ and $p(\nu_2)$ are prior probability density functions of the respective parameters.

From the assumptions of the model, we have that in (2.5), $f(X|C, \alpha, \nu_1, \nu_2)$ is the probability density function of the observed point process, $p(C|\alpha)$ is the probability density function of a homogeneous Poisson point process with intensity κ, which can be recomputed from α (see below) and $p(\alpha)$, $p(\nu_1)$ and $p(\nu_2)$ are prior densities of the respective parameters.

2.3 Simulations from the joint posterior distribution

We want to sample from the joint posterior distribution $p(C, \mu, \nu_1, \nu_2|X)$ since its normalizing constant is analytically intractable. That is because to be able to compute the normalizing constant, one would need to integrate over the space of locally finite point configurations. Therefore, the posterior distribution is approximated by the samples from a Markov chain, whose stationary distribution and limiting distribution are both equal to the posterior distribution we want to study. To construct the Markov chain, we are going to use Markov Chain Monte Carlo (hereafter referred to as MCMC) methods. These methods are widely used in the Bayesian inference as they give us a tool that can satisfactorily solve this issue.

We use iterative procedures for updating both the parent point process and the parameters altogether, which we are going to describe in detail in the next chapter.

3. MCMC methods

In this chapter, we discuss the MCMC algorithms that we used in our method in detail. If not stated otherwise, all the concepts and characteristics of MCMC algorithms in this chapter are taken from Møller and Waagepetersen (2004). Core of Algorithm 1 is also taken from Møller and Waagepetersen (2004), Algorithm 2 is taken from Robert (2016) and Algorithm 3 is taken from van Ravenzwaaij et al. (2016).

Let us have $B \subset S \subset \mathbb{R}^d$ such that $0 < |B| < \infty$. Let Y denote a point process on B with an unnormalised density h w.r.t. a unit Poisson point process on B. The process Y may also be defined on S and be additionally conditioned on B to obtain h, see e.g. Proposition 6.1 and Remark 6.6 in Section 6.3.3. in Møller and Waagepetersen (2004).

An appropriately set up and built MCMC algorithm generates a *Markov chain* $Z_1, Z_2, ...$ with a specified limiting distribution, which in our case is the posterior distribution. For two states Z_i and Z_{i+j}, we want their possible dependence to be eliminated as j increases. In such a case, the Markov chain is said to be *well mixing*.

In our situation, Y correspond to the joint posterior distribution $p(C, \mu, \nu_1, \nu_2|X)$ and we iteratively construct a Markov chain with the state space $\mathcal{N}_f^*(W) \times \Theta$, where $\mathcal{N}_f^*(W)$ is the space of all simple finite counting measures on W, which is the observation window we are working on. Each Z_i, $i = 1, 2, ...$ then corresponds to the joint distribution of a point process C_i and a vector parameter θ_i, conditionally on the observed point process X.

The next definition is taken from Meulen and Schauer (2022).

Definition 13. *Let $S = (E, \mathfrak{B})$, $S' = (E', \mathfrak{B}')$ be Borel measurable spaces. A Markov kernel between S and S' is denoted by $K : S \to S'$, where S is the 'source' and S' the 'target'. That is, $K : E \times \mathfrak{B}' \to [0, 1]$, where*

i. the map $x \mapsto K(x, B)$ is $\mathfrak{B}$-measurable for every $B \in \mathfrak{B}'$ and

ii. the map $B \mapsto K(x, B)$ is a probability measure on S' for every $x \in E$.

Let Q be a Markov kernel, $Q(x, dy) = q(x, y)\upsilon(dy)$ for some density function $q(x, \cdot)$ defined on B. We call $q(x, \cdot)$ the *proposal density*.

The following two algorithms in Sections 3.1 and 3.2 below consist of two main steps. Firstly, they make a proposal (of either an update of the population of parent points or an update of the vector parameter θ) and secondly, they accept the proposal with some probability, computed from the current and proposed state. This probability is then called the *acceptance probability* of the proposal.

3.1 Birth-Death-Move step

We first introduce the algorithm that is used to update the population of the parent points. In every step, the algorithm randomly proposes one of the three possible

updates, first of which is birth, that adds a new point to the current population of parent points (so-called *birth update*), the second is then death, which removes an existing parent point (so-called *death update*) and finally move, which means changing the location of an existing parent point (so-called *move update*). Let $n \in \mathbb{N}$. Let $x = \{x_1, x_2, \dots, x_n\}$ be the current state of the parent point process. We denote $p_{birth}(x)$ the probability of proposing birth, if the current state of the algorithm is x. In a similar way, we define $p_{move}(x)$ and $p_{death}(x)$. Now we are ready to introduce the *Birth-Death-Move Step* in Algorithm 1.

In Algorithm 1, R_m and ζ_m or η_m or x_i are mutually conditionally independent given the random variables used for the generation of $Y_0, \dots, Y_m$ and I_m. Also I_m and R_m are mutually independent. The matrix Σ in the move step determines the possible shift of the chosen point x_i. A natural choice is a diagonal matrix

$$\begin{pmatrix} \sigma_\circ^2 & 0 \\ 0 & \sigma_\bullet^2 \end{pmatrix}$$

with hyperparameters $\sigma_\circ,\ \sigma_\bullet > 0$ that we a priori specify.

Let $\bar{x}$ be the current state of the parent point process and $\bar{y}$ the proposed state of this process. The acceptance probabilities of the move, birth or death update are all of the form

$$\gamma(\bar{x}, \bar{y}) = \begin{cases} min\left\{\frac{h(\bar{y})q(\bar{y},\bar{x})}{h(\bar{x})q(\bar{x},\bar{y})},\ 1\right\}, & \text{if } h(\bar{x})q(\bar{x},\bar{y}) > 0 \\ 1, & \text{otherwise,} \end{cases} \tag{3.1}$$

where $q(\bar{x}, \bar{y})$ is the probability that we go from the current state $\bar{x}$ to the new state $\bar{y}$ and $h(\bar{x})$, $h(\bar{y})$ are joint conditional probability density functions of the parameters and the parent point process in the states $\bar{x}$ and $\bar{y}$, which we compute as in (2.5), since we assume h to be unnormalised. We work under the assumption that all the parent points lie in the observation window W. Let us illustrate the densities $q(\bar{x}, \bar{y})$, resp. $q(\bar{y}, \bar{x})$ on each of the possible proposals in the m^{th} step. We denote the number of parent points in the state $\bar{z}$ as $n(\bar{z})$. Let $n(\bar{x}) = \mathfrak{n}$.

birth proposal $\bar{x} \mapsto \bar{x} \cup \eta$, $n(\bar{x} \cup \eta) = (\mathfrak{n} + 1)$, the proposal densities are $q(\bar{x}, \bar{y}) = p_{birth}(\bar{x}) \cdot \frac{1}{|W|}$, $q(\bar{y}, \bar{x}) = p_{death}(\bar{x}) \cdot \frac{1}{\mathfrak{n}+1}$

death proposal $\bar{x} \mapsto \bar{x} \setminus \xi$, $n(\bar{x} \setminus \xi) = \mathfrak{n} - 1$, the proposal densities are $q(\bar{x}, \bar{y}) = p_{death}(\bar{y}) \cdot \frac{1}{\mathfrak{n}}$, $q(\bar{y}, \bar{x}) = p_{birth}(\bar{x}) \cdot \frac{1}{|W|}$

move proposal $\bar{x} \mapsto \bar{x} \setminus x_i \cup \zeta$, $n(\bar{x} \setminus x_i \cup \zeta) = \mathfrak{n}$, the proposal densities are $q(\bar{x}, \bar{y}) = p_{move}(\bar{x}) \cdot \frac{1}{\mathfrak{n}} \cdot f_{move}(\bar{x})$, $q(\bar{y}, \bar{x}) = p_{move}(\bar{x}) \cdot \frac{1}{n} \cdot f_{move}(\bar{y})$,

where $f_{move}(\bar{x})$ and $f_{move}(\bar{y})$ are the densities for the move update in the current state $\bar{x}$ and proposed state $\bar{y}$, respectively, which we choose to be bivariate normal distribution. Since we work under the condition that all the parent points lie inside the observation window W, the prior density for the parent point process is equal to 0 outside of W, and so $h(\bar{y})$ in equation (3.1) is equal to 0 if the proposed move update moves the chosen point out of the observation window, i.e., the proposed state has a point outside the observation window, and the acceptance probability in this case is 0.

Algorithm 1: Birth-Death-Move Step

Input: Point pattern $Y_m = x = \{x_1, \dots, x_n\}$
Output: Point pattern Y_{m+1}
Set $0 \leq p_{move}(x) < 1$, $0 \leq p_{birth}(x) < 1$, $0 \leq p_{death}(x) < 1$ such that $p_{move}(x) + p_{birth}(x) + p_{death}(x) = 1$;
Draw $R_m \sim Unif([0,1])$;
if $R_m \leq p_{move}(x)$ **then**
    Draw $I_m \sim Unif(\{1, \dots, n\})$ and further assume $I_m = i$;
    Generate $\zeta_m \sim N(x_i, \Sigma)$;
    Calculate the acceptance probability $\gamma_{move}(x, \zeta_m)$;
    Generate $U \sim Unif([0,1])$;
    if $U < \gamma_{move}(x, \zeta_m)$ **then**
        Set $Y_{m+1} = \{x_1, \dots, x_{i-1}, \zeta_m, x_{i+1}, \dots, x_n\}$;
    else
        Set $Y_{m+1} = x$;
    end
else
    if $R_m \leq p_{move}(x) + p_{birth}(x)$ **then**
        Generate uniformly at random η_m in the observation window W;
        Calculate the acceptance probability $\gamma_{birth}(x, \eta_m)$;
        Generate $U \sim Unif([0,1])$;
        if $U < \gamma_{birth}(x, \eta_m)$ **then**
            Set $Y_{m+1} = \{x_1, \dots, x_n, \eta_m\}$;
        else
            Set $Y_{m+1} = x$;
        end
    else
        Draw $I_m \sim Unif(\{1, \dots, n\})$ and further assume $I_m = i$;
        Calculate the acceptance probability $\gamma_{death}(x)$;
        Generate $U \sim Unif([0,1])$;
        if $U < \gamma_{death}(x)$ **then**
            Set $Y_{m+1} = \{x_1, \dots, x_{i-1}, x_{i+1}, \dots, x_n\}$;
        else
            $Y_{m+1} = x$;
        end
    end
end

Let us have $C = \bar{x}$ the current parent point pattern and $C' = \bar{y}$ the proposed one. For $h(\bar{x})$ and $h(\bar{y})$, we have that

$$h(\bar{x}) = f(X|C, \alpha^m, \nu_1^m, \nu_2^m)p(C|\alpha^m)p(\alpha^m)p(\nu_1^m)p(\nu_2^m), \tag{3.2}$$

and

$$h(\bar{y}) = f(X|C', \alpha^m, \nu_1^m, \nu_2^m)p(C'|\alpha^m)p(\alpha^m)p(\nu_1^m)p(\nu_2^m). \tag{3.3}$$

Here, $f(X|C, \alpha^m, \nu_1^m, \nu_2^m)$ is the density of the observed point process X given the current state of the parent point process and the current state of the parameters α^m, ν_1^m, ν_2^m. Then $p(C|\alpha^m)$ denotes the probability density function for the homogeneous Poisson point process in the observation window W given α^m and $p(\alpha^m)$, $p(\nu_1^m)$, $p(\nu_2^m)$ are the prior probability density functions evaluated in the respective parameters. Further $f(X|C', \alpha^m, \nu_1^m, \nu_2^m)$ is the density of the observed point process X given the proposed state of the parent point process and the current state of the parameters. Since we do not propose any change to the parameters all the factors $p(\alpha^m)$, $p(\nu_1^m)$ and $p(\nu_2^m)$ in (3.2) cancel out with the corresponding factors in (3.3) when computing the acceptance probability in (3.1) and we are left with the ratio

$$\frac{f(X|C', \alpha^m, \nu_1^m, \nu_2^m)p(C'|\alpha^m)q(\bar{y}, \bar{x})}{f(X|C, \alpha^m, \nu_1^m, \nu_2^m)p(C|\alpha^m)q(\bar{x}, \bar{y})}.$$

Let us now derive the formula for $f(X|C, \alpha^m, \nu_1^m, \nu_2^m)$, from which we can anologously derive the formula for $f(X|C', \alpha^m, \nu_1^m, \nu_2^m)$. Let $\mathfrak{m}$ denote the number of observed daughter points in the observation window W. Conditionally on the parent point process C, we view the daughter point process as an inhomogeneous Poisson point process with the intensity function given by the current values of α^m, ν_1^m and ν_2^m as

$$\lambda^m(x) = \sum_{c \in C} \alpha^m \cdot k(x - c; \nu_1^m, \nu_2^m), \tag{3.4}$$

where $k(x - c; \nu_1^m, \nu_2^m)$ is the probability density function corresponding to the relative displacement of the daughter points in the clusters, given the values of parameters in the current step, and α^m represents the mean number of daughter points per cluster. Then we can write

$$f(X|C, \alpha^m, \nu_1^m, \nu_2^m) = \exp\left\{|W| - \int_W \lambda^m(x)dx\right\} \cdot \prod_{x \in X} \lambda^m(x). \tag{3.5}$$

After the substitution of the intensity function (3.4) into (3.5), the right hand side can be rewritten as

$$\exp\left\{|W| - \alpha^m \cdot \int_W \sum_{c \in C} k(x - c; \nu_1^m, \nu_2^m)dx\right\} \cdot \prod_{x \in X} \left(\alpha^m \cdot \sum_{c \in C} k(x - c; \nu_1^m, \nu_2^m)\right).$$

Using the relations

$$\sigma_1^m = \exp\{\nu_1^m\}, \quad \sigma_2^m = \exp\{\nu_2^m\},$$

see Section 1.3, the fact that $n(X) = \mathfrak{m}$, properties of the exponential and the definition of the function k, we obtain

$$f(X|C, \alpha^m, \nu_1^m, \nu_2^m) =$$
$$\exp\left\{|W| - \frac{\alpha^m}{2\pi\sigma_1^m\sigma_2^m} \cdot \int_W \sum_{c\in C} \exp\left\{-\frac{1}{2}\left[\frac{(x_1-c_1)^2}{\sigma_1^m} + \frac{(x_2-c_2)^2}{\sigma_2^m}\right]\right\} dx\right\}$$
$$\cdot \left(\frac{\alpha^m}{2\pi\sigma_1^m\sigma_2^m}\right)^{\mathfrak{m}} \cdot \prod_{x\in X}\left(\sum_{c\in C} \exp\left\{-\frac{1}{2}\left[\frac{(x_1-c_1)^2}{\sigma_1^m} + \frac{(x_2-c_2)^2}{\sigma_2^m}\right]\right\}\right). \quad (3.6)$$

For the ratio $\frac{p(C'|\alpha^m)q(\bar{y},\bar{x})}{p(C|\alpha^m)q(\bar{x},\bar{y})}$, we have to distinguish between the possible proposed updates. If we propose the move update, we obtain 1, since all the points lie inside the observation window. For the birth update, the ratio simplifies to

$$\frac{p(C'|\alpha^m)q(\bar{y},\bar{x})}{p(C|\alpha^m)q(\bar{x},\bar{y})} = \frac{\exp\{|W|(1-\kappa^m)\}\,(\kappa^m)^{\mathfrak{n}+1}\; p_{death}(\bar{y})\frac{1}{\mathfrak{n}+1}}{\exp\{|W|(1-\kappa^m)\}\,(\kappa^m)^{\mathfrak{n}}\; p_{birth}(\bar{x})\frac{1}{|W|}}$$
$$= \frac{\kappa^m|W|}{\mathfrak{n}+1} \cdot \frac{p_{death}(\bar{y})}{p_{birth}(\bar{x})},$$

and for the death update, we obtain

$$\frac{p(C'|\alpha^m)q(\bar{y},\bar{x})}{p(C|\alpha^m)q(\bar{x},\bar{y})} = \frac{\exp\{|W|(1-\kappa^m)\}\,(\kappa^m)^{\mathfrak{n}-1}\; p_{birth}(\bar{y})\frac{1}{|W|}}{\exp\{|W|(1-\kappa^m)\}\,(\kappa^m)^{\mathfrak{n}}\; p_{death}(\bar{x})\frac{1}{\mathfrak{n}}}$$
$$= \frac{\mathfrak{n}}{\kappa^m|W|} \cdot \frac{p_{birth}(\bar{y})}{p_{death}(\bar{x})},$$

where we used the fact that the parent point process is assumed to be a homogeneous Poisson point process on W, i.e., its probability density function for the value of κ being κ^m is

$$(\kappa^m)^{\mathfrak{n}} e^{(1-\kappa)|W|}. \quad (3.7)$$

3.2 Metropolis-Hastings step

In this section, we introduce the algorithm used for updating scalar parameters. As we mentioned in Section 2.2, we update κ differently than other parameters, since we recompute it in each step from α and do not generate it from any distribution.

For the Algorithm 2, the acceptance probabilities are generally of the form

$$\gamma(\theta, \vartheta) = \begin{cases} min\left\{\frac{f(\vartheta)q(\vartheta,\theta)}{f(\theta)q(\theta,\vartheta)},\ 1\right\}, & \text{if } f(\theta)q(\theta,\vartheta) > 0 \\ 1, & \text{otherwise,} \end{cases}$$

where $f(\theta)$ is the joint conditional probability density function of the parent point proces with the parameters in the current state with the components of θ being equal to the current values of the parameters and $f(\vartheta)$ is the probability density function with the components of ϑ being equal to the proposed values of the parameters. The function $q(\theta, \vartheta)$ stands for the probability density function of

Algorithm 2: Metropolis-Hastings Step With Kappa Update

Input: $\theta_m = (\alpha^m, \nu_1^m, \nu_2^m)\,,\ \kappa^m$
Output: $\theta_{m+1} = \left(\alpha^{m+1}, \nu_1^{m+1}, \nu_2^{m+1}\right),\ \kappa^{m+1}$
Generate $\vartheta = (\alpha^{prop}, \nu_1^{prop}, \nu_2^{prop})$ from $Q(\theta_m, \cdot)$;
Compute

$$\kappa^{prop} = \frac{\mathfrak{m}}{\alpha^{prop} \cdot |W|}$$

;
Calculate the acceptance probability $\gamma\left(\theta_m, \kappa^m, \vartheta, \kappa^{prop}\right)$;
Generate $U \sim Unif([0,1])$;
if $U < \gamma\left(\theta_m, \kappa^m, \vartheta, \kappa^{prop}\right)$ **then**
  $\theta_{m+1} = \vartheta_m,\ \kappa^{m+1} = \kappa^{prop}$;
else
  $\theta_{m+1} = \theta_m,\ \kappa^{m+1} = \kappa^m$;
end

the update from θ to ϑ and analogously $q(\vartheta, \theta)$ is the probability density function of the update back from ϑ to θ.

Let us assume that the current number of parent points is $\mathfrak{n}$ and that we observe $\mathfrak{m}$ daughter points. For the vector $\theta = (\alpha, \nu_1, \nu_2)$ and parameter κ, we have that

$$\begin{aligned}\frac{f(\vartheta)\, q(\vartheta, \theta)}{f(\theta)\, q(\theta, \vartheta)} &= \frac{f(X|C, \vartheta)\, p(C|\alpha^{prop})\, p(\alpha^{prop})\, p(\nu_1^{prop})\, p(\nu_2^{prop})}{f(X|C, \theta)\, p(C|\alpha^m)\, p(\alpha^m)\, p(\nu_1^m)\, p(\nu_2^m)} \\ &= \frac{f(X|C, \vartheta)\, (\kappa^{prop})^{\mathfrak{n}}\, e^{(|W| - \kappa^{prop}|W|)}\, p(\alpha^{prop})\, p(\nu_1^{prop})\, p(\nu_2^{prop})}{f(X|C, \theta)\, (\kappa^m)^{\mathfrak{n}}\, e^{(|W| - \kappa^m|W|)}\, p(\alpha^m)\, p(\nu_1^m)\, p(\nu_2^m)},\end{aligned}$$

where we used the fact that the parent point process is assumed to be a homogeneous Poisson point process on W, i.e., its probability density function for the value of κ being κ^m is given by the relation (3.7).

In the nominator, $f(X|C, \vartheta)$ is the probability density function of the observed point process X given C, the current state of the parent point proces, and the proposed values of the parameters α^{prop}, ν_1^{prop} and ν_2^{prop}, $p(C|\alpha^{prop})$ is the probability density function for the parent point process given the proposal α^{prop}, which is a homogeneous Poisson process with intensity equal to $\mathfrak{m}/\alpha^{prop}$. Lastly, $p(\alpha^{prop})$, $p(\nu_1^{prop})$ and $p(\nu_2^{prop})$ are the prior distribution functions for the parameters evaluated in the proposed values. For the denominator, $f(X|C, \theta)$ is the probability density function of the observed point process given C and the current value of the parameters, $p(C|\alpha^m)$ is the probability density function of the parent point process given the current value α^m, i.e., homogeneous Poisson point process with intensity $\mathfrak{m}/\alpha^m$ and $p(\alpha^m)$, $p(\nu_1^m)$ and $p(\nu_2^m)$ are the values of the prior density function in the current values of parameters.

Analogously as in the previous section, we can derive that $f(X|C, \theta)$ is equal to

$$\exp\left\{|W| - \frac{\alpha^m}{2\pi\sigma_1^m\sigma_2^m}\cdot\int_W \sum_{c\in C}\exp\left\{-\frac{1}{2}\left[\frac{(x_1-c_1)^2}{\sigma_1^m}+\frac{(x_2-c_2)^2}{\sigma_2^m}\right]\right\}dx\right\}$$
$$\cdot\left(\frac{\alpha^m}{2\pi\sigma_1^m\sigma_2^m}\right)^{\mathfrak{m}}\cdot\prod_{x\in X}\left(\sum_{c\in C}\exp\left\{-\frac{1}{2}\left[\frac{(x_1-c_1)^2}{\sigma_1^m}+\frac{(x_2-c_2)^2}{\sigma_2^m}\right]\right\}\right),$$

and we could also analogously derive $f(X|C,\vartheta)$.

We generate the updates of each component of θ independently of other components, and we use normal distribution centered in the current value of the component with some standard deviation. More precisely, $q(\theta_m,\cdot)$ is a trivariate normal distribution $N_3(\theta_m,\Sigma)$, where

$$\Sigma = \begin{pmatrix}\sigma_\alpha^2 & 0 & 0\\ 0 & \sigma_{\nu_1}^2 & 0\\ 0 & 0 & \sigma_{\nu_2}^2\end{pmatrix}.$$

Here, σ_α, σ_{ν_1}, $\sigma_{\nu_2}^2 > 0$ are hyperparameters which we a priori specify. We set the prior for α to be some uniform distribution on an interval on the positive semi-axis of $\mathbb{R}$, and so if there is a proposal for a negative value of α, $p(\alpha^{prop})$ is equal to 0, therefore the acceptance probability is 0 in this case. In Chapter 4, we update directly σ_1 and σ_2 instead of ν_1 and ν_2, since we assume there that ν_1 and ν_2 are scalar, i.e., there are no covariates influencing σ_1 and σ_2. Then we set their priors to be uniform on some intervals on the positive semi-axis of $\mathbb{R}$, hence if the proposal for one of these parameters is negative, then either $p(\sigma_1^m)$ or $p(\sigma_2^m)$ is also equal to 0, and so the acceptance probability of such cases becomes 0.

3.3 Metropolis-within-Gibbs algorithm

Now, we are going to combine the two update steps into the *Metropolis-within-Gibbs* algorithm, given heuristically in Algorithm 3. This algorithm alternates between updating the parameters and updating the parent point process. As stated in Chapter 2, the parent point process is taken as a nuisance parameter and we are interested only in the inference about the model parameters.

Algorithm 3: Metropolis-within-Gibbs Algorithm

Input: Vector of initial values θ_0, initial parent point pattern C_0, observed point process X consisting of $\mathfrak{m}$ points

Output: samples from the posterior distributions of the components of θ, corresponding values of κ and locations of the parent points

for $n = 0, 1, \ldots, N$ **do**

  STEP 1: update of scalar parameters;

  with θ_n, κ_n do Metropolis-Hastings step with kappa update to obtain θ_{n+1}, κ_{n+1};

  if $n \mod 100 = 0$ **then** save the values θ_{n+1}, κ_{n+1};

  STEP 2: update of the parent point process;

  with C_n do Birth-Death-Move step

end

After the algorithm has finished, we obtain samples from the posterior distribution after the convergence of the chain to the limiting distribution. We are interested in the posterior distributions of each parameter. To obtain those, we plot histograms of the aposterior values. To obtain a point estimate, we use median of the aposterior values.

Remark. We want the neighbouring obtained values of parameters to be independent (or at least less dependent), and that is why we save and use only the information from every 10^{th} iteration. This procedure is called *thinning of the Markov chain*.

3.3.1 Burn-in

To obtain reasonable values for the aposterior distributions, one has to take the convergence into consideration. The *burn-in* is a number of the steps that are deleted from the beginning of the chain to obtain only the values from the algorithm that are close to the coveted values. To be able to set the burn-in, one has to estimate it from the traceplots of parameters describing the states of the Markov chain and determine the step from which the trace seems to stabilize, i.e, does not show any major deviations. The burn-in depends on the complexity of the model, as well as the prior distributions, and therefore there is no universal recommendation.

3.3.2 Mean vs. median

It is both natural and reasonable to consider both mean and median as a possible point estimate. We have performed a few simulations to see whether these two would differ significantly. We introduced the prior distributions on bounded intervals and as we can see from the posterior distributions in Chapter 4, these distributions are approximately symmetrical. Hence the two estimates, mean and median of the posterior sampled values, are very close and do not differ significantly. Therefore we opted for median as it is a more robust estimate. Because of these reasons, we do not include the mean in our tables and further calculations.

3.3.3 Recommended choice of initial values

In the first step, the parent point process is created using no information about the location of the observed daughter points. This can lead to the logarithm of the acceptance probability being very small which runs into the problem of machine precision when simulating in R. In such cases, an error ocurrs, because the inequality $\log(U) < \log(\gamma_.)$ from Algorithm 1 cannot be properly evaluated (that is because $\log(\gamma_.)$ in this case is computed by R as $-\infty$). To avoid such errors, we recommend using a lower initial value for α, which causes the initial number of parent points to be higher than anticipated. This causes the distances from the observed points to the closest parent point to be smaller and we can therefore avoid this problem.

4. Simulation study for anisotropic Thomas cluster point processes without covariate dependencies

In this chapter, we are going to discuss the results of the simulations in the case where all the model parameters are one-dimensional, i.e., no covariates were considered to influence the models. For the daughter point processes, we are interested in their standard deviations σ_1 and σ_2 which determine the relative displacement of these points around the parent point, their mean number in each cluster α and the intensity of the parent point process κ, as we described in Section 1.3.

We would like to mention that we use the software R (version 4.3.2) for all the calculations. We use packages spatstat (version $3.0-7$) and mvtnorm (version $1.2-4$). The source codes are available in the electronic attachments of this thesis.

In Section 4.1, we verify that the algorithm works well and gives reasonable values. Next, in Section 4.2, we illustrate the method on simulated data, and give the numerical and graphical outputs that can be obtained from the analysis. In the next Section 4.3, we show the results of the analysis on real data and the obtained numerical and graphical outputs. Finally, in Section 4.4, we discuss the issue of parameter identifiability.

4.1 Performance of the MCMC algorithm

In this section, we show how well the algoritm works. To do so, we performed 10 independent runs of the algorithm with 100 000 steps and 50 000 step burn-in. 3 realizations out of these 10 are shown in the Figure 4.1.

In each run, we generated an anisotropic Thomas point process using $\kappa = 25$, $\alpha = 10$, $\sigma_1 = 0.04$ and $\sigma_2 = 0.02$, independent of all the other generated point processes. We used uniform priors $Unif([0.003, 30])$ for α and $Unif([0.001, 0.2])$ for both σ_1 and σ_2 and set the initial values of α to be 5, both σ_1 and σ_2 to be 0.01. From the initial value of alpha, we computed the initial value of κ from the total number of daughter points. The proposal distributions for α, σ_1 and σ_2 were normal distributions centered in the current value of the parameter with the standard deviations equal to $\sigma_\alpha = 0.25$, $\sigma_{\sigma_1} = 0.005$ and $\sigma_{\sigma_2} = 0.005$, respectively, see Section 3.2.

The initial parent point patterns were created as Poisson point processes in the observation window W with the intensity equal to the initial value of κ. The probabilities of suggesting move, birth or death were all equal to 1/3. For the move proposal, we used the bivariate normal distribution centered in the point

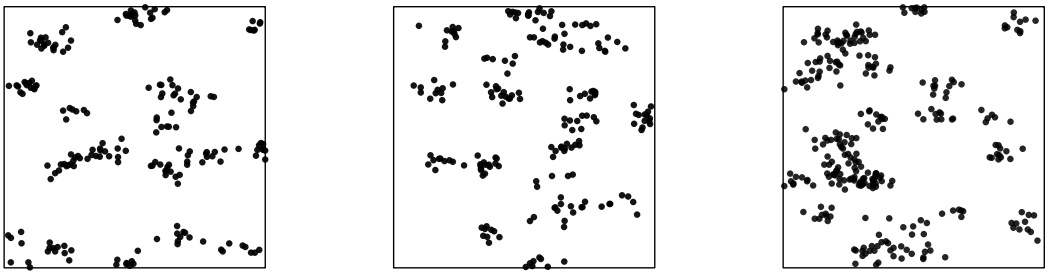

Figure 4.1 3 realizations out of a total of 10 used for testing the performance of the algorithm.

that we wanted to move, with $\sigma_\circ = \sigma_\bullet = 0.01$, see Section 3.1.

To evaluate how well the algorithm performs, we computed the point estimates of the parameters, the relative mean squared error and relative bias. We also computed the 95% credibility intervals. Let us first give the definition of the relative mean squared error and the relative bias.

Definition 14. *Let $\hat{\theta}$ be an estimator of parameter θ. The mean squared error (MSE) is defined as*

$$MSE(\hat{\theta}) = \mathbb{E}\left[\left(\theta - \hat{\theta}\right)^2\right].$$

The relative mean squared error (rMSE) is then defined as

$$rMSE(\hat{\theta}) = \frac{MSE(\hat{\theta})}{\theta^2}.$$

Moreover, we define the bias as

$$bias(\hat{\theta}) = \mathbb{E}\left[\hat{\theta} - \theta\right],$$

and the relative bias as

$$r.\,bias(\hat{\theta}) = \frac{bias(\hat{\theta})}{\theta}.$$

Credible intervals are a Bayesian counterpart of the confidence intervals. A 95% credibility interval of a scalar parameter is obtained as an interval between 0.025 and 0.975 quantiles of the posterior distribution of this parameter.

The results for $rMSE$ and the relative bias for each parameter are presented in Table 4.1. As we can see, we did not observe any large $rMSE$s or large bias in any of the model parameters. The values of $rMSE$ are very small and the relative bias is also satisfactory. Therefore we can conclude that the method used gives sensible results.

For the analysis using credible intervals, we plotted a 95% credibility interval for each parameter combined with the point estimate obtained as the posterior median. For reference, we also plotted the parameter value used for simulation of the point patterns considered here. We can examine the results in Figure 4.2. From there we can see that the real value of the parameter α lies inside 9 out of the 10 credible intervals. For σ_1 and σ_2, all of the credible intervals cover the real

parameter	$rMSE$	relative bias
κ	0.033	-0.103
α	0.010	0.039
σ_1	0.002	0.031
σ_2	0.003	0.008

Table 4.1 $rMSE$ and relative bias for each parameter computed from 10 independent runs of the algorithm rounded to 3 decimals.

value of the parameters. From this, we can conclude that the algoritm works well and gives satisfactory results.

For parameter κ, 6 intervals cover the real value of the parameter, as we can see in Figure 4.3. This is a much smaller value than anticipated, as we wanted 95% credibility intervals and this gives us only 60%, but we assume that by changing the prior distribution for α, one could obtain wider credibility intervals with possibly better coverage if necessary. That is because since we set the prior for α to be $Unif([0.003, 30])$, the prior for κ can be determined from this using the *Monotone transformation theorem*. The prior probability density function for κ is then equal to

$$f_\kappa(x) = \frac{12\,500}{999} \cdot \frac{1}{x^2} \mathbb{1}_{(x)} \left[\frac{25}{2}, 12\,500\right], \quad x \in \mathbb{R},$$

which we can also observe in Figure 4.4. As we can see, larger values are therefore suppressed and we believe that this is the reason why the credibility intervals which do not contain the used value of κ are under this value.

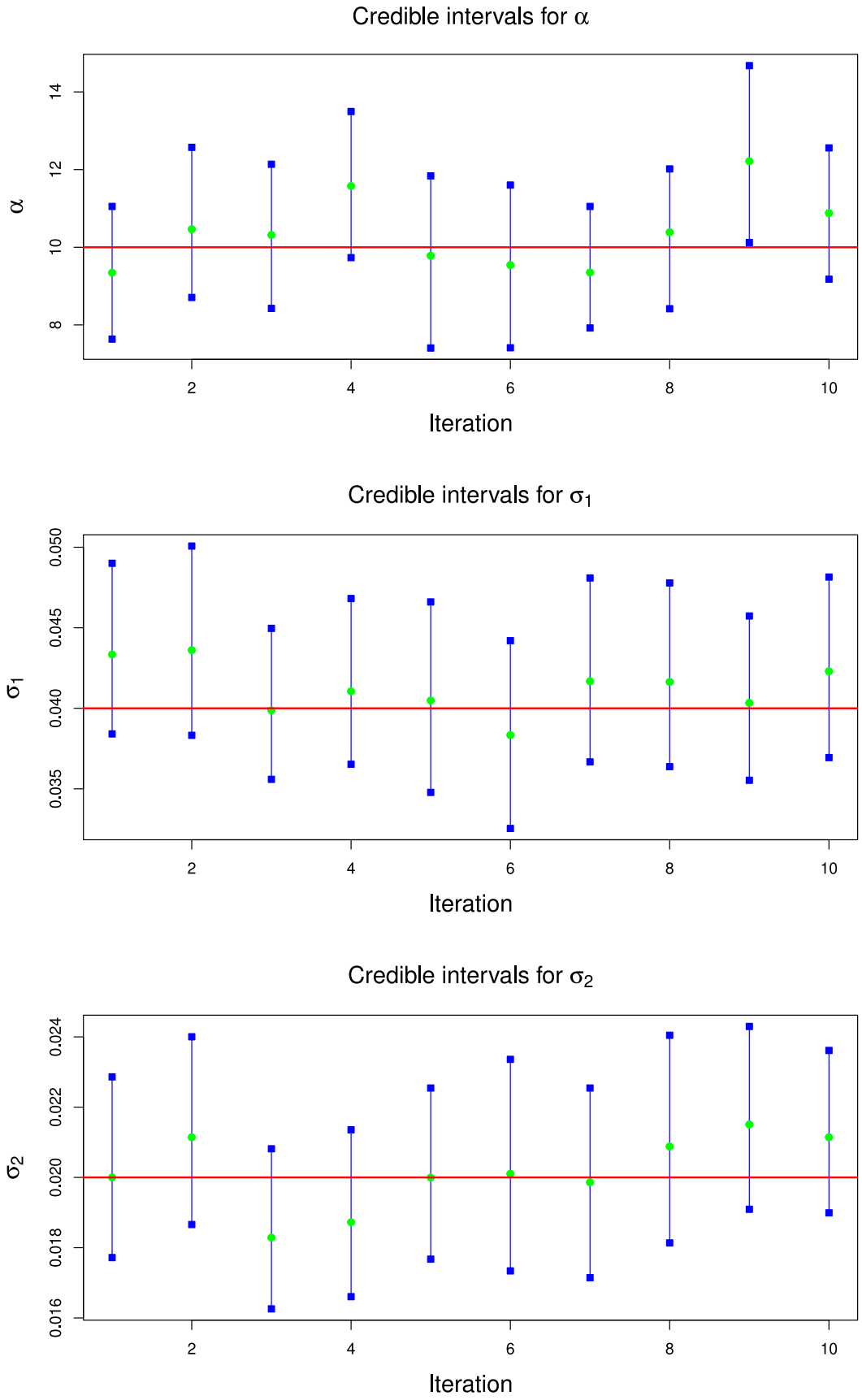

Figure 4.2 95% credibility intervals for parameters α, σ_1 and σ_2. Green squares are the estimated values from each run of the algorithm, red line is the value of the parameters used for the simulations: $\alpha = 10$, $\sigma_1 = 0.04$ and $\sigma_2 = 0.02$.

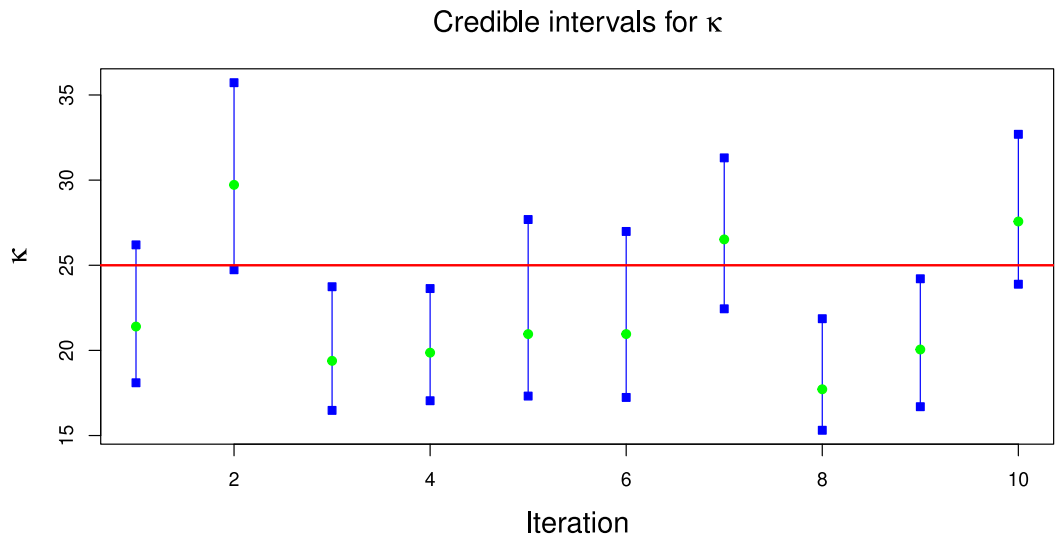

Figure 4.3 95% credibility intervals for κ. Green squares are the estimated values from each run of the algorithm, red line is the value of the parameter used for the simulations $\kappa = 15$.

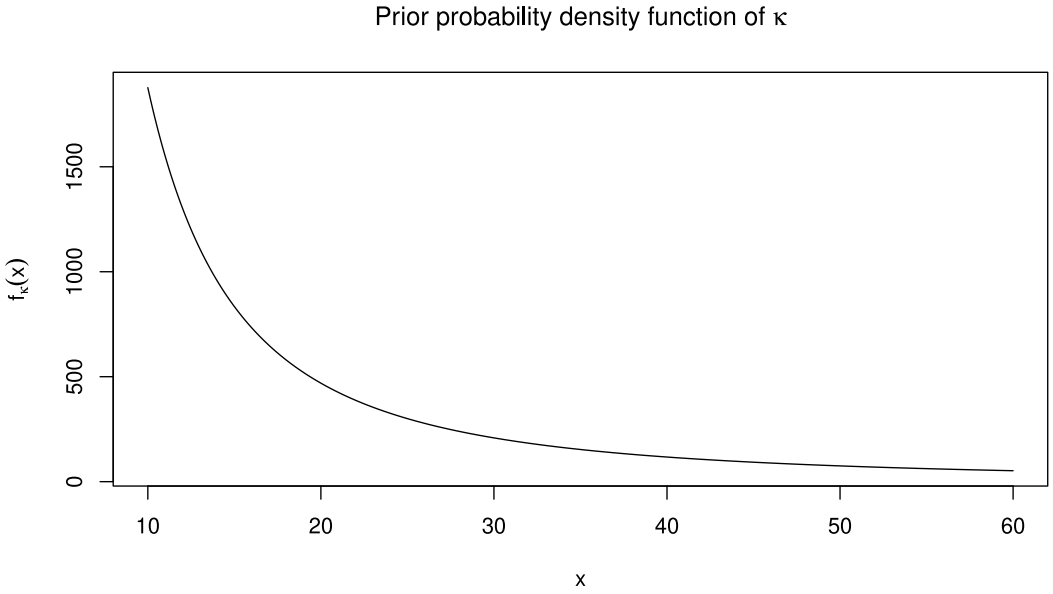

Figure 4.4 Prior probability density function of κ restricted to $x \in \left[\frac{25}{2}, 60\right]$.

4.2 Analysis and outputs for a simulated dataset

In this section, we present the results from one run of the algoritm. We show the results that we can obtain from the run combined with graphic outputs.

First, we simulated a realization of an anisotropic Thomas point process in a unit square window $W = [0,1]^2$ using $\kappa = 15$, $\alpha = 25$, $\sigma_1 = 0.02$ and $\sigma_2 = 0.04$. We can see the realization in the top left picture in Figure 4.5. The initial value of parameter α was 10, for both σ_1 and σ_2 we set the initial value as 0.01. We set the prior distribution for α to be $Unif([0.003, 30])$, for σ_1 and σ_2 we used $Unif([0.001, 0.2])$. For updates, we chose normal distributions centered in the current value of the parameter with standard deviations $\sigma_\alpha = 3$, $\sigma_{\sigma_1} = \sigma_{\sigma_2} = 0.003$.

The choice of the standard deviations for the updates can be based on the examination of shorter pilot runs and the preliminary estimates of credible intervals. We try to set the standard deviation to be of the same order as the length of the credible interval for the parameter. By doing this, we control the fraction of accepted updates and also the correlations in the sequence of the saved samples from the posterior distribution. From the experience of the previous section, we have used larger standard deviation for α in this section. We will discuss the fractions of accepted updates and the correlations between the samples later on in this section.

Since we simulated the data that we analyse, we also have the information about the real number of parent points and their locations, which we can see in Figure 4.5 in the top right picture. We can see that there were 13 parent points in total which were randomly scattered around the observation window W. In the bottom left picture, we can see the initial locations of the 31 parent points, which were created as a homogeneous Poisson point process with intensity $\frac{300}{10} = 30$, which is the estimate from relation (2.3), as we observed 300 daughter points and set the initial value of α to be 10. We set the probabilities for birth, death and move update as 1/3. In the move update, we move the chosen point using normal distribution centered in this chosen point with $\sigma_\circ = \sigma_\bullet = 0.01$. As we can see in the bottom right picture from this figure, there are estimated to be 14 parent points very close to their exact locations in the last step of the algorithm. The fact that there were estimated to be 14 parent points should not be given much weight as it represents only one sample from the posterior distribution of the parent point process. There were also many populations of parent points where there were 15 or 13 parent points. We performed 250 000 steps of the algorithm with a 100 000 step burn-in and saved every 100^{th} value.

Next, we provide the pictures of the intensity function of the daughter point process conditional on the parent point process in Figure 4.6. First, we plotted the conditional intensity function, given the chosen parameters for the simulations, of the daughter point process

$$\lambda(x) = \sum_{c \in C} \alpha \cdot k(x - c; \sigma_1, \sigma_2) \tag{4.1}$$

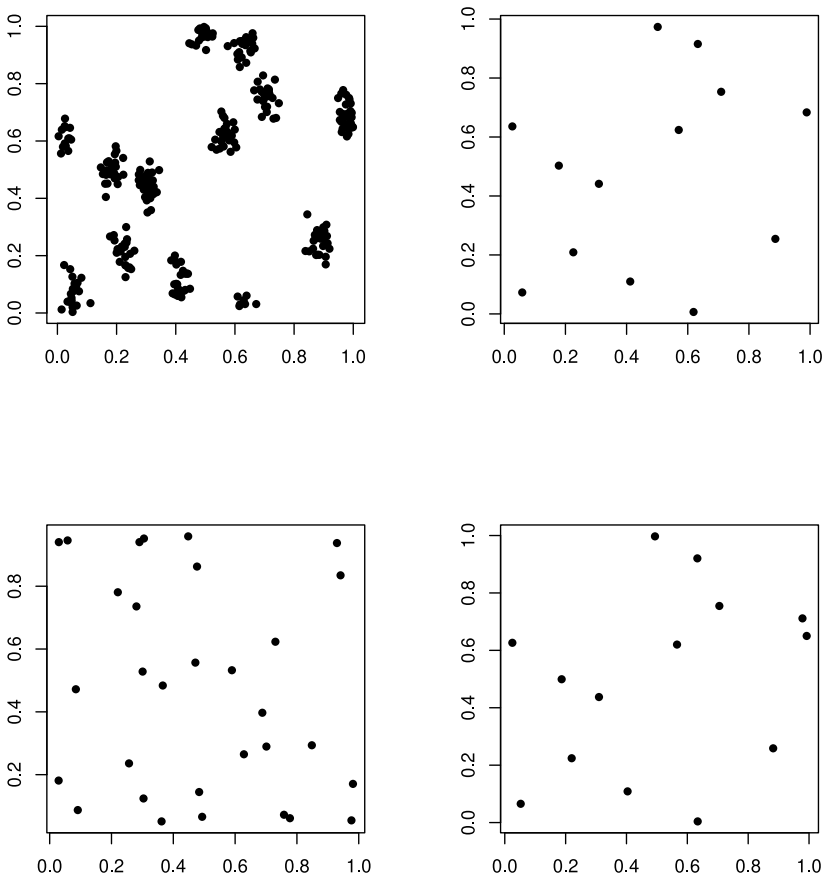

Figure 4.5 Realization of an anisotropic Thomas point process used for analysis (top left). Parent point locations: the real locations (top right), the initial locations from the first step of the algorithm (bottom left) and the final locations from the last step (bottom right).

and then the estimated conditional intensity function of the daughter point process

$$\widehat{\lambda(x)} = \sum_{c \in \widehat{C}} \hat{\alpha} \cdot k(x - c; \widehat{\sigma_1}, \widehat{\sigma_2}), \tag{4.2}$$

where $\widehat{C}$ are the locations of the parent points in the last step of the algorithm and $\hat{\alpha}$, $\widehat{\sigma_1}$ and $\widehat{\sigma_2}$ are the values of the respective parameters from the last step. Although we obtain posterior distributions for the parent point process, as well as the posterior distributions for parameters after the algorithm has finished, we chose the last population of parent points and the last accepted values of parameters purely for illustrative purposes. As we can see, the intensity functions are very similar, and we can see that the estimates are very close to the used values of the parameters and to the original locations of the parent points.

Next, we have a look at the posteriors of parameters. In Figure 4.7, we can see the four histograms of the samples from the posterior distributions, which give us estimates of the posterior probability density functions of the model parameters.

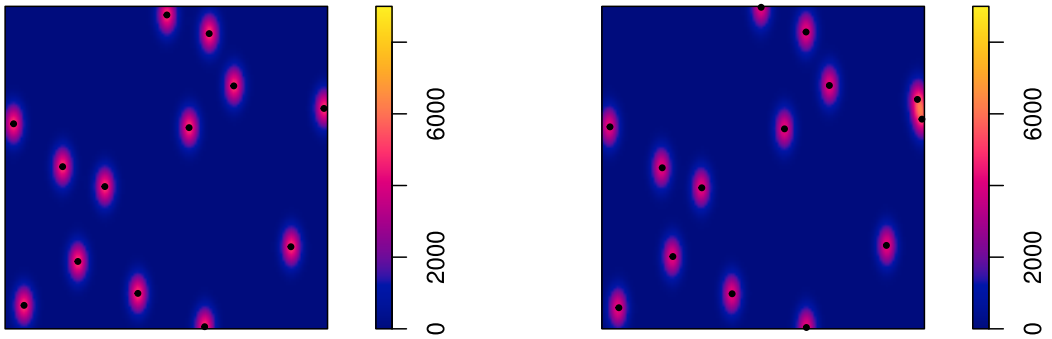

Figure 4.6 Real intensity function of the daughter point processes (left) and the estimated intensity function computed at the end of the algoritm (right).

	κ	α	σ_1	σ_2
used value	15	25	0.020	0.040
estimated value	12.037	24.923	0.020	0.041
lower credibility bound	10.680	21.933	0.018	0.038
upper credibility bound	13.680	28.090	0.022	0.045

Table 4.2 Real and estimated values of the parameters for a simulated dataset.

We also plotted the prior probability density functions (dark blue lines). As we can see in the figure, all the histograms represent reasonable unimodal distributions which are significantly different from the prior distributions. We can see that the distributions are close to being symmetrical, and therefore in connection with the discussion in Subsection 3.3.2, we can conclude that the posterior median is a reasonable point estimate of the parameters. The estimated posterior values combined with corresponding credibility intervals, rounded to three decimals, are shown in Table 4.2. We can see that the obtained values and credibile intervals are reasonable and give good estimates of the used values.

In Figure 4.8 we can see the traceplots describing the current state of the parameter in the MCMC algorithm together with the point estimate (solid red line) and the end points of the credible intervals (dashed red lines). Let us recall that we use every 100^{th} observed value of parameters. As we can see in the plots, most of the values in the chain after the burn-in stay in the credibility intervals and no significant deviation from the intervals were detected. We can also see that the chains have converged to their limiting distributons. We also plotted the traceplot for log-likelihood, which we can see in Figure 4.9. We can examine that there were no significant deviances and that it remains quite stable.

In order to find the *sampling frequency*, i.e., how often we should save the computed value, we first saved every computed value and then had a look at the autocorrelation function of the posterior values. From that, we estimated the frequency so that the neighbouring values would be as little correlated as possible, while still having enough observations to perform the analysis of the output. A reasonable choice is 100, meaning that we save every 100^{th} value. We can examine the corresponding plots of autocorrelation functions of the posterior values thinned like this for α and σ_1 in Figure 4.10.

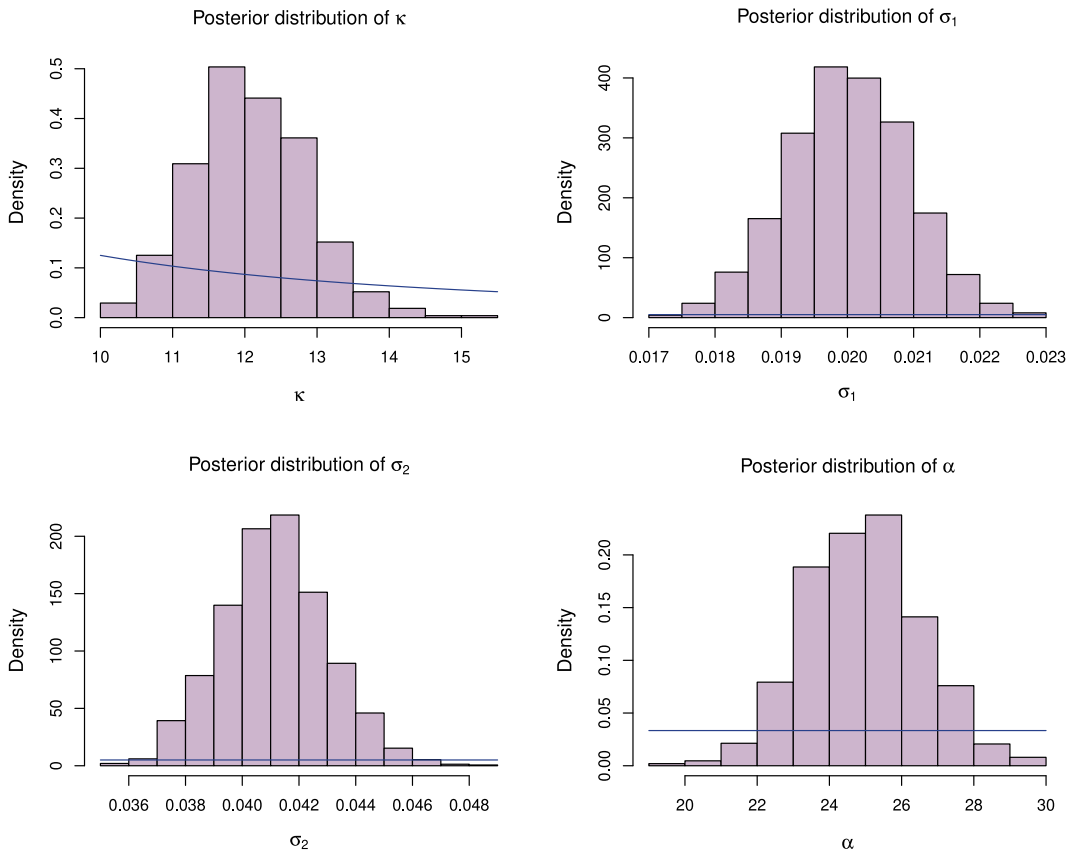

Figure 4.7 Histogram of the samples from the posterior distribution (estimating the posterior probability density function) with prior density lines (dark blue lines) for the simulated data.

Lastly, we had a look at the fractions of accepted updates. To do that, we computed throughout the run of the algorithm the fraction of accepted proposals always in the last 1 000 steps. We can examine the results in Figure 4.11. For better clarity, we ommited larger values from the beginning of the algorithm where more updates were accepted, as the algorithm starts from point in the state space that is rather extreme with respect to the limiting distribution. The optimal acceptance rate should not be too large, as that would indicate that the chain is not in a 'good' state and still changes a lot, on the other hand it should not be too small, as this would lead to the chain remaining on certain values, which would mean that it does not explore the entire state space sufficiently. Values around 20% are generally considered to be acceptable.

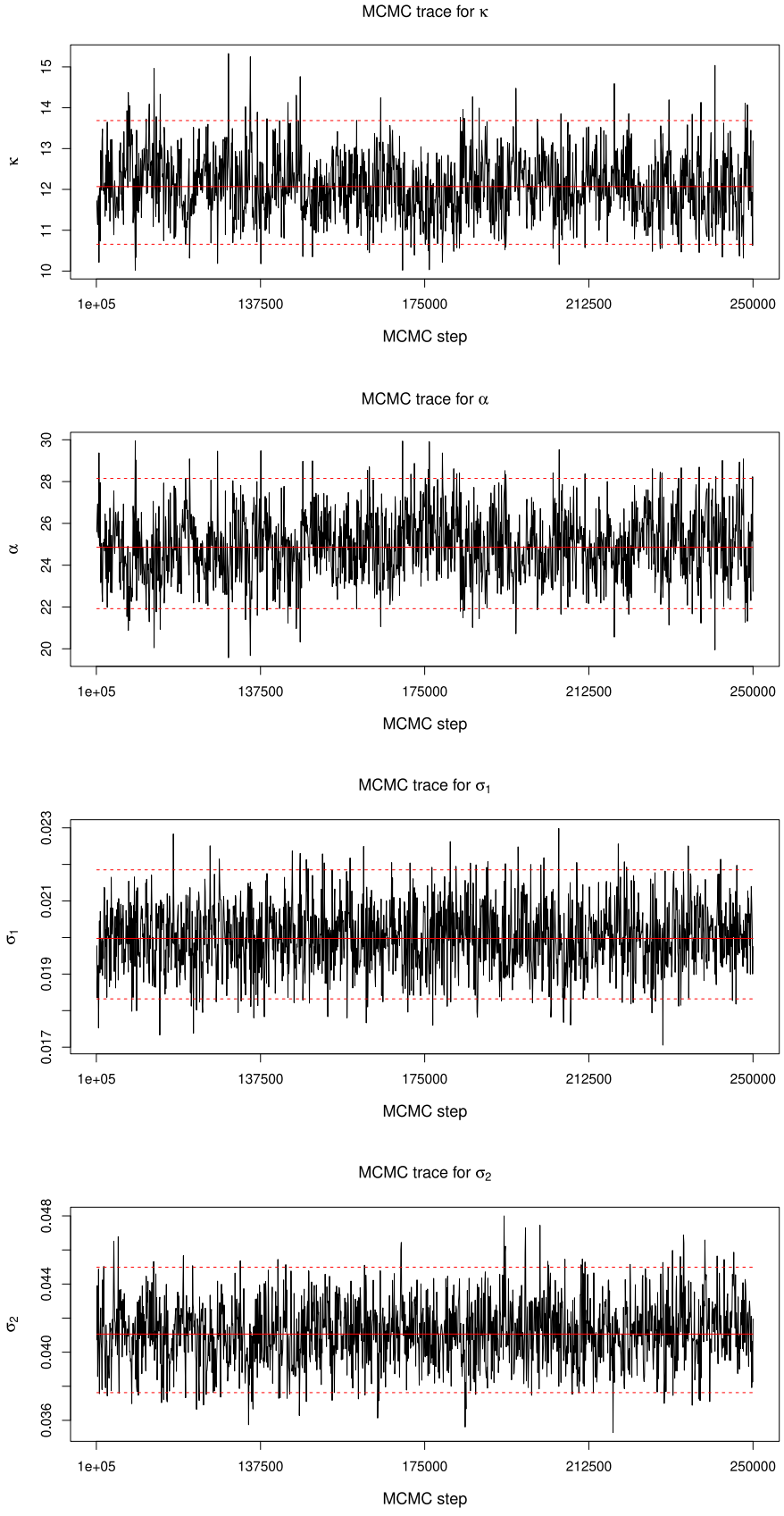

Figure 4.8 Traceplots for model parameters describing the state of the MCMC algorithm for simulated data. We saved the values after the burn-in of 100 000 steps. Only every 100^{th} value of each parameter is recorded.

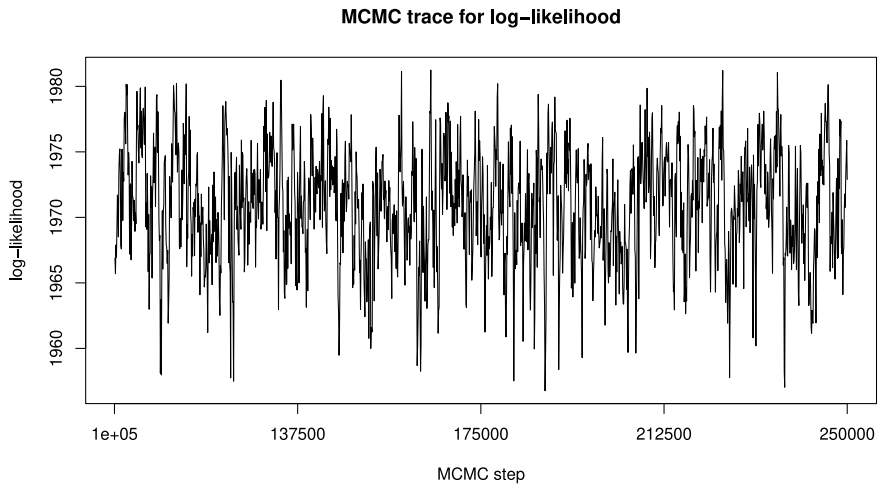

Figure 4.9 Traceplot for log-likelihood during the MCMC algorithm after the burn-in while saving every 100^{th} value for simulated data.

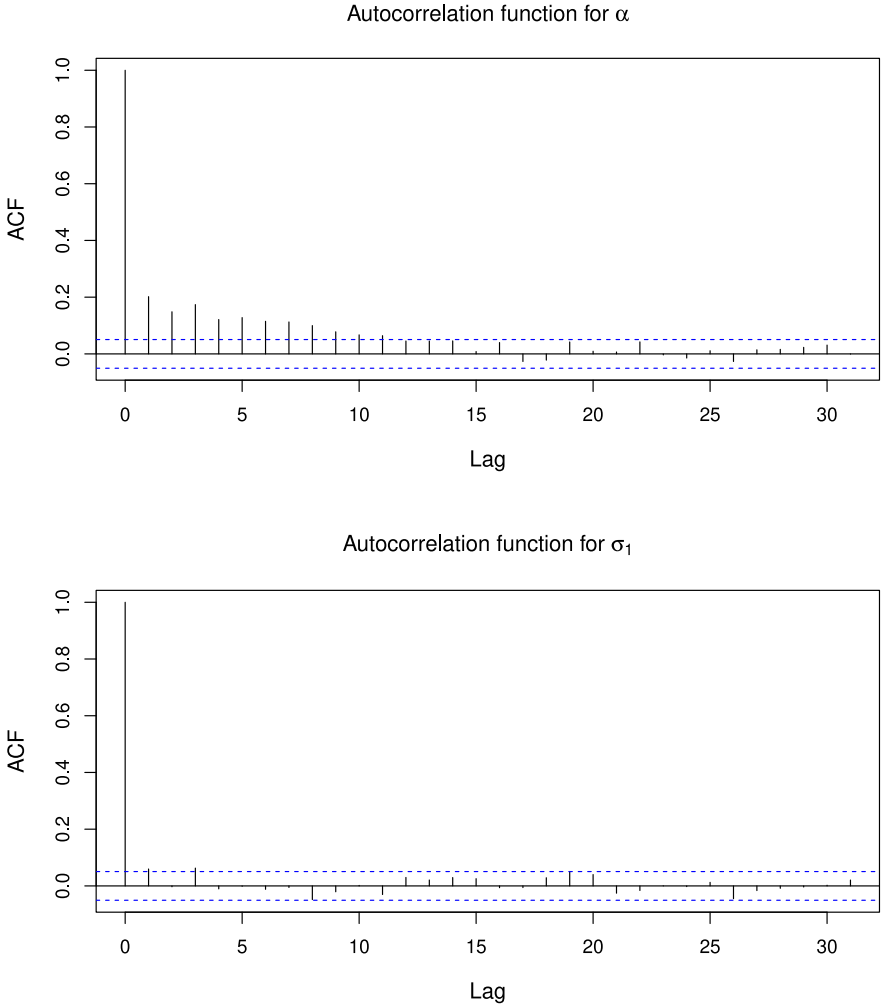

Figure 4.10 Estimated autocorrelations for posterior values of α and σ_1 for simulated data where we save every 100^{th} computed value.

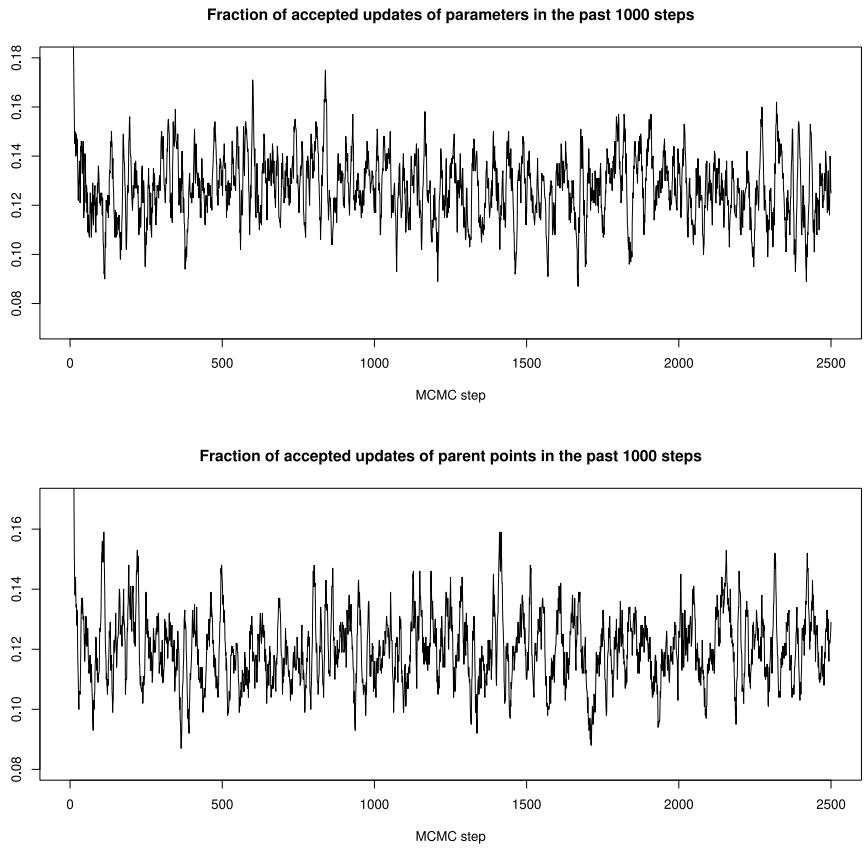

Figure 4.11 Fractions of accepted proposals in the last 1000 steps in the birth-death-move update and the Metropolis-Hastings update for simulated data.

4.3 Analysis for real life data

In this section, we are going to present the results of the algorithm on a real-life data set. For this purpose we chose the dataset redwoodfull available in the R package spatstat. These data contain information about the locations of 195 seedlings and saplings of California Giant Redwood rescaled to the unit square window. We can see the whole data in Figure 4.12 (left). As we can see in this picture, there is not a clustering pattern all over the observation window, but rather in the left upper half of the picture, whereas below the diagonal, there are no visible clusters. Moreover, since we work under the assumption that $\rho = 0$, i.e., the covariance matrix describing the relative displacement of the daughter points in the clusters is diagonal, we cannot properly test the situation where the dependence on the direction is diagonal rather than horizontal or vertical. Because of these two fact, we made some transformations for the dataset used.

First of all, we have to introduce a special type of directional K-function. For more information, see Møller and Waagepetersen (2004).

Definition 15. *For $-\pi/2 \leq \varrho \leq \pi/2$, $\varrho \leq \psi \leq \varrho + \pi$, and $r > 0$, define $K'(\varrho, \psi, r) = K(B(\varrho, \psi, r))$, where K is the K-function and*

$$B(\varrho, \psi, r) = \{t(cos\iota, sin\iota) : 0 \leq t \leq r, \phi \leq \iota \leq \psi \text{ or } \phi + \pi \leq \iota \leq \psi + \pi\}$$

is the union of the two sectors of $b(0, r)$ with the angles of the first sector between ϱ and ψ, and the angles of the second sector between $\varrho + \pi$ and $\psi + \pi$. We call the above defined function the sector K-function.

Thanks to the sector K-function, one can investigate possible anisotropy in the data by investigating several directions in the picture and determining whether or not there is a significantly larger value in some of the directions. Thanks to this, we can also determine the direction where the strongest anisotropy can be detected. We computed the sector K-function for several directions and several different values of r to obtain the direction where the most anisotropy can be detected. We found out that the direction that shows the largest anisotropy is around $5\pi/18$. We then rotated the dataset accordingly and cut out a rectangle of the dimensions 0.29×0.5 where we observed clusters. The final dataset that we used for testing can be seen in Figure 4.12 (middle). This rectangle was cut out of the lower right corner of the original dataset after the rotation. This is for illustrative purposes only; analyzing a dataset with a general direction of anisotropy would require large changes in the computer codes.

Since we worked on a much smaller window than in the previous examples, where we worked on the unit square window, we had to make changes to the hyperparameters of the model to obtain the results which would have satisfying properties, mainly the fraction of accepted updates of both the parent point process and parameters around 20%. We began with the initial values $\alpha = 5$, $\sigma_1 = \sigma_2 = 0.01$ and used the same prior distribution for these parameters as in the previous sections. For the updates, we again used normal distributions centered in the current value with $\sigma_\alpha = 0.91$, $\sigma_{\sigma_1} = \sigma_{\sigma_2} = 0.0065$. For the parent point process, we used $\sigma_\circ = \sigma_\bullet = 0.1$ in the move update and proposed birth, death and move update all with probability 1/3. We did 250 000 steps with 100 000 step burn-in and saved every 100^{th} value.

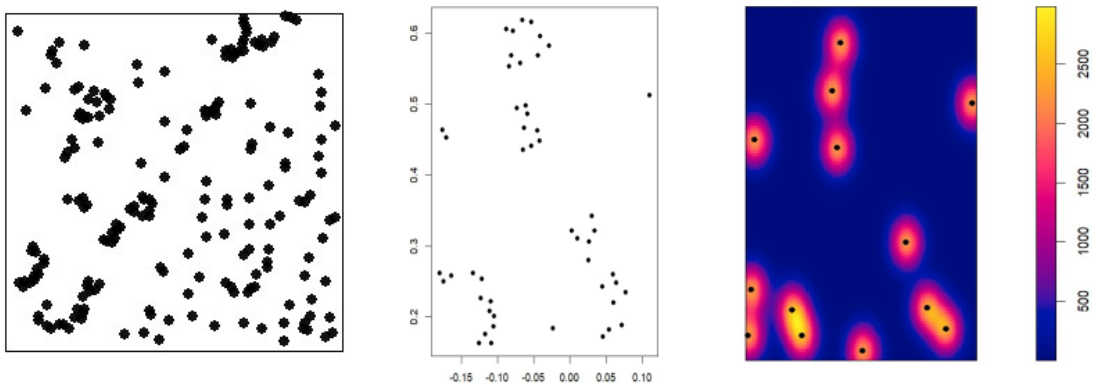

Figure 4.12 Redwoodfull dataset (left), used realization of the dataset (middle), estimated conditional intensity function (right)

	κ	α	σ_1	σ_2
estimated value	87.082	3.881	0.011	0.025
lower credibility bound	55.691	2.137	0.008	0.014
upper credibility bound	158.104	6.068	0.016	0.039

Table 4.3 Table of estimated values and credible intervals for redwood data.

The results of the analysis rounded to 3 decimals are summed up in Table 4.3. It is worth mentioning that the credible intervals are much larger than in the previous simulated examples, since we are working with real data with not many observations on a much smaller window. We also plotted the conditional intensity function of the daughter point processes in Figure 4.12 (right), where we can see the estimated locations of the parent points. Let us recall that we expect all the parent points to lie inside the window.

We can observe the posterior distributions estimating the posterior probability density functions in Figure 4.13. Compared to the histograms obtained in Figure 4.7, we can see that the histograms in this case are more skewed to the right. But nevertheless, the distributions are still unimodal and for α, σ_1 and σ_2, they differ from the uniform priors significantly.

Lastly, we take a look at the traceplots of the Markov chain, after the burn-in with every 100^{th} value of parameter. The results can be seen in Figure 4.14. We also ploted the corresponding credible intervals (dashed red lines) and the estimated value of the parameter (full red line). Compared to the simulated data, we can see that the traces tend to leave the credible intervals to very high values. Nevertheless, the sampler seems to mix well and the chain has converged to its limiting distribution. We can also observe the traceplot for log-likelihood in Figure 4.15. There we also observe somewhat larger dispersion of the values compared to the simulated dataset.

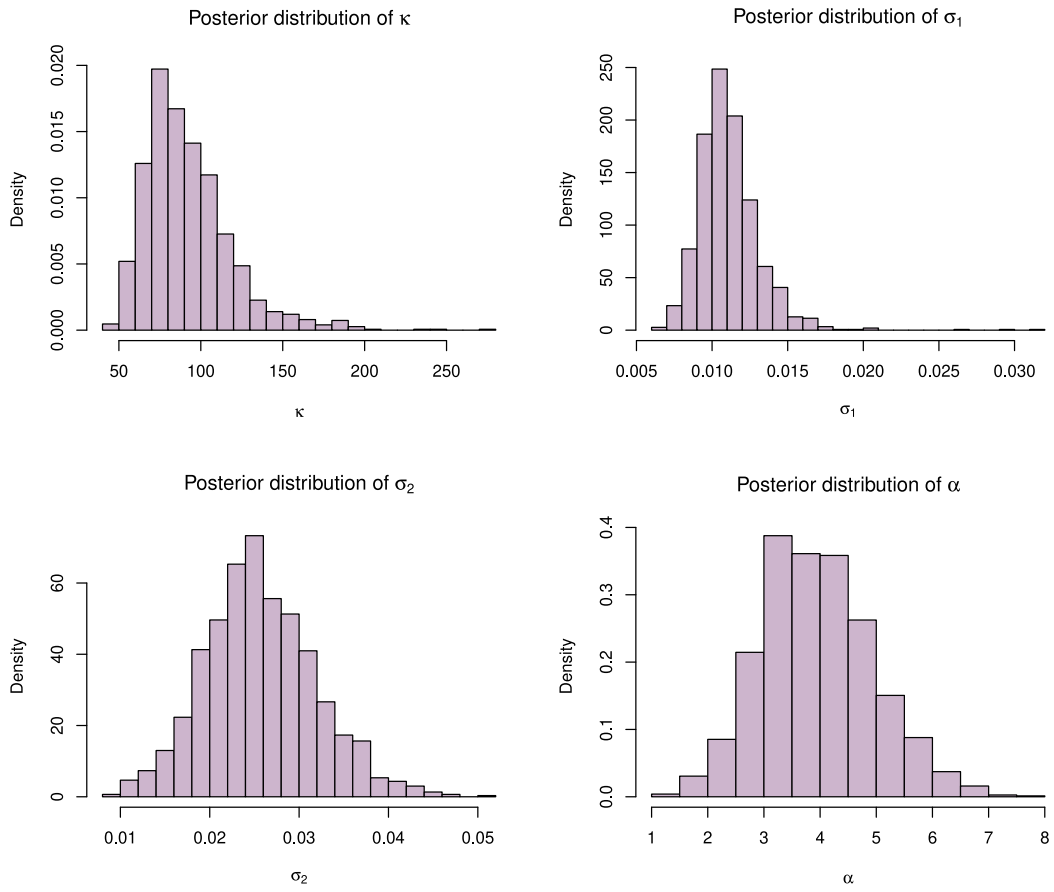

Figure 4.13 Histogram of the samples from the posterior distribution (estimating the posterior probability density function) for redwood dataset.

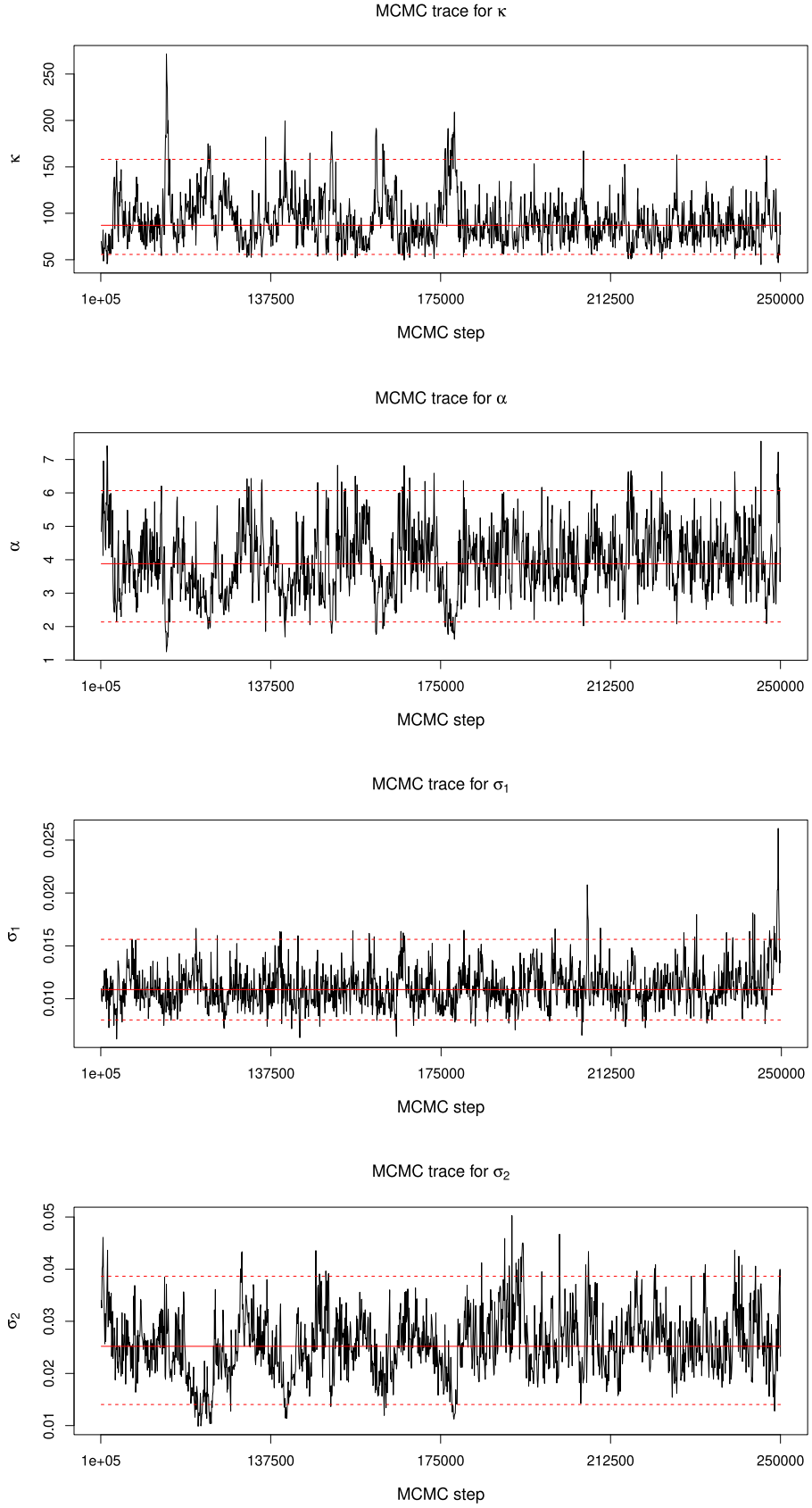

Figure 4.14 Traceplots for model parameters describing the state of the MCMC algorithm for redwood dataset. We saved the values after the burn-in of 100 000 steps. Only every 100^{th} value of each parameter is recorded.

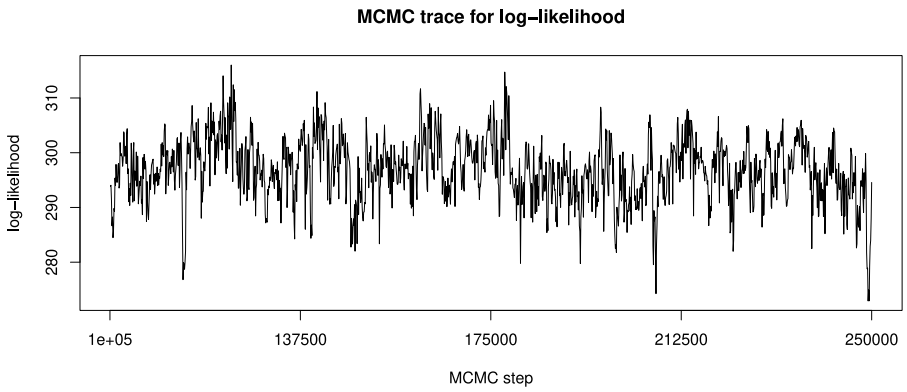

Figure 4.15 Traceplot for log-likelihood during the MCMC algorithm for redwood dataset.

4.4 Identifiability of κ and α

As we mentioned in Section 1.4, there may be some indentifiability issues with covariates influencing the parent point process and the mean number of the daughter points per cluster.

When performing the MCMC algorithm described in Chapter 3, one has to keep in mind that the result may be influenced by the relationship between the parameters κ and α, as we estimate κ using the relation (2.3). To put it simply, the algorithm cannot distinguish between one cluster with many daughter points and more overlapping clusters, each with a smaller number of daughter points.

As an extreme case where the results are far from the real values of the parameters, we computed the point estimators of the parameters σ_1, σ_2, κ and α, where α is 20 times larger than κ. To be precise, we used $\alpha = 100$, $\kappa = 5$, $\sigma_1 = 0.02$ and $\sigma_2 = 0.03$. We used 100 000 steps with a 50 000 steps of burn-in. We computed the real conditional intensity function of the daughter point process and the estimated conditional intensity function of the daughter point process as in relations (4.1) and (4.2), and plotted them as pictures in Figure 4.16. We also ploted the exact locations of the parent points in the picture with the real intensity function and the parent points estimated in the last step of the algorithm in the picture with the estimated intensity of the daughter point process. Although we estimate the locations of the parent points from the run of the MCMC algorithm, we chose their location from the last step of the algorithm for illustrative purposes. As we can see, the intensity functions are very similar, but the number of parent points in the estimated case is larger. The real number of the parent points is 4, but the algorithm estimated the number to be 12 and clustered them.

The used values and their estimates rounded to 3 decimals are presented in Table 4.4. As we can see in the table, the estimates of κ and α are very different from the actual values of the parameters. That is again because of the strong relationship between these two parameters.

As a solution to this problem we would recommend the user to set the number of the parent points after these 100 000 steps to 4, as from the Figure 4.16 we can see that there are 4 'bumps' in the intensity function, and continue with the algorithm with this number of parent points thereafter. To set the 4 new parent points, one could for example find the center of gravity of each cluster of parent points. A more in depth analysis of this issue is beyond the scope of this thesis.

	κ	α	σ_1	σ_2
used value	5	100	0.02	0.03
estimated value	14.604	29.649	0.016	0.026

Table 4.4 Real and estimated values of the parameters in a situation where α is 20 times larger than κ.

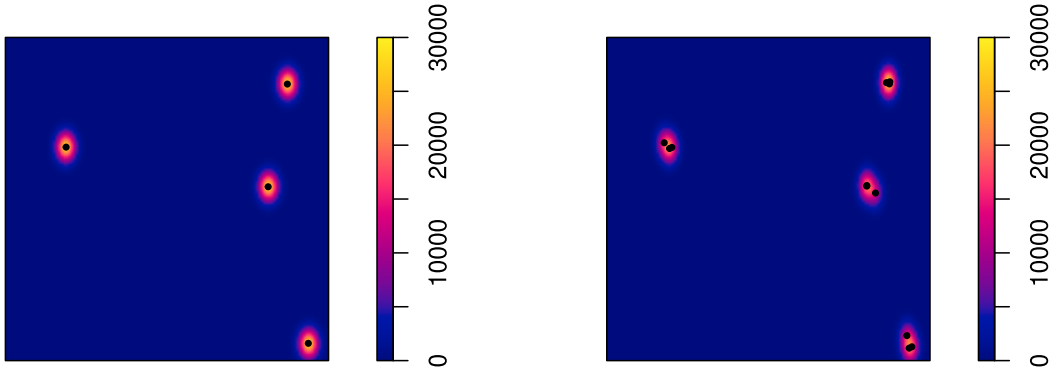

Figure 4.16 The real intensity function of the daughter point processes with the exact locations of the parent points (left). The estimated intensity function of the daughter point processes with the locations of the parent points extracted from the 100000^{th} step of the MCMC algorithm (right).

5. Simulation study for anisotropic Thomas cluster point processes with covariate dependencies

In this chapter, we are going to present the results for the situation where there might be a covariate present. We perfomed two analyses, one for a situation where we simulated a point process dependent on a covariate, and we show how well the algorithm estimates the effect of the covariate, and the second one for a situation where no covariate is present, but set the algorithm as if there were a covariate influencing the process.

In both cases, we assume that the possible dependence on a covariate lies within σ_1, which is one of the standard deviations describing the relative displacement of the daughter point processes around their parent points. That is, we assume that σ_1 is of the form

$$\sigma_1(c;\nu_1) = \exp\left\{\beta_0^{\sigma_1} + \beta_1^{\sigma_1} z_1^{\sigma_1}(c)\right\},$$

where $c = (c_1, c_2)$ denotes a parent point and $\nu_1 = (\beta_0^{\sigma_1}, \beta_1^{\sigma_1}) \in \mathbb{R}^2$ corresponds to the regression coefficients of the parametrization.

5.1 Analysis for covariate dependent process

We are first going to analyse the situation where a covariate is present in the data. The covariate used is a function $z_1^{\sigma_1}(u_1, u_2) = u_1, (u_1, u_2) \in W$. We simulated a Thomas cluster point process with parameters $\kappa = 25$, $\alpha = 10$, $\sigma_2 = 0.015$, $\beta_0^{\sigma_1} = -4.605$ (if no covariate was observed, this would correspond to σ_1 being equal to 0.01, as $\log(0.01) \doteq -4.605$) and $\beta_1^{\sigma_1} = 2$. The used realization can be seen in the top left picture in Figure 5.1. We can see that the clusters on the left side of the window are tighter, whereas the clusters on the right side of the picture are more dispersed.

For the initial values, we chose $\alpha = 5$, $\beta_0^{\sigma_1} = -3$, $\beta_1^{\sigma_1} = 1$, $\sigma_2 = 0.02$. As the prior distributions, we again used uniform priors $Unif([0.03, 30])$ and $Unif([0.001, 0.2])$ for α and σ_2, respectively. Since both $\beta_0^{\sigma_1}$ and $\beta_1^{\sigma_1}$ may reach negative values, we set their priors to be $N(0, 5)$ in both cases. For the update of the parameters, we used normal distributions centered in the current value with standard deviations for α, σ_2, $\beta_0^{\sigma_1}$ and $\beta_1^{\sigma_1}$ to be 0.15, 0.003, 0.003 and 0.15, respectively. For the move update of the parent point process, we used a diagonal matrix with $\sigma_\circ = \sigma_\bullet = 0.01$. We again propose birth, death, and move with probability 1/3 for all the possible updates. We performed 250 000 steps with 100 000 step burn-in.

As there is an additional parameter to be estimated compared to the previous chapter, the convergence of the algorithm may therefore be somewhat slower. In order to start the algorithm from a reasonable part of the state space, we start

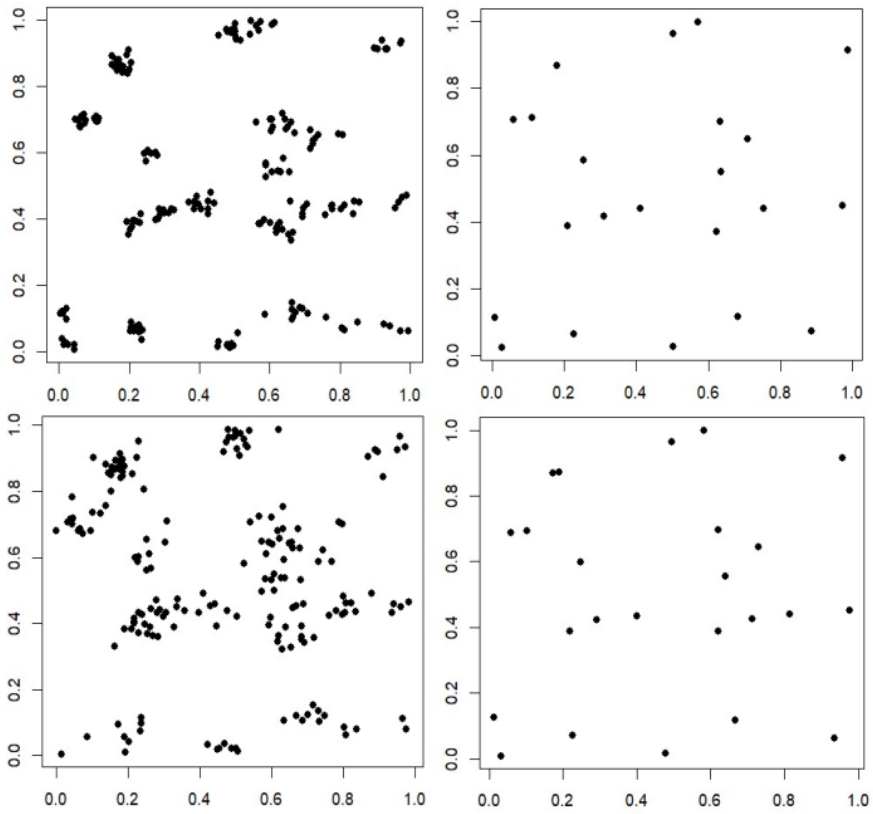

Figure 5.1 Point process dependent on a covariate (top left), corresponding parent point process (top right), initial parent point process (bottom left) and the final point process from the last step of the algorithm (bottom right).

with the parent point pattern generated as a Poisson point process with intensity function equal to the kernel estimate of the intensity function of the daughter point process with Diggle's edge correction. This helps the parent point pattern to converge faster, since the parent points in the first step are generated in realistic positions, rather than completely at random.

In Figure 5.1 we can see the point process observed in the top left picture and the corresponding parent points in the top right picture. The initial parent point process is in the bottom left picture, and we can see that the points are scattered around the positions of the clusters, but there are a lot more parent points than anticipated. The final parent point process from the last step of the algorithm is in the bottom right picture. We can see that the locations of the points are very close to the exact locations.

We are now going to present the outputs of the algorithm. The estimated values of the parameters rounded to 3 decimals with their respective credible intervals are presented in Table 5.1. As we can see, all the credibility intervals contain the used values of the parameters and the estimated values are also close to the used ones. In Figure 5.2, we can examine the histograms of the samples from the posterior distribution, which estimate the posterior probability density functions. As we can see, all of them are reasonable unimodal distributions. The value 0 is not present in the posterior samples of $\beta_0^{\sigma_1}$, and the influence of the covariate is well estimated.

	κ	α	$\beta_0^{\sigma_1}$	$\beta_1^{\sigma_1}$	σ_2
used value	25	10	−4.605	2	0.015
estimated value	21.873	9.281	−4.662	2.117	0.015
lower credibility bound	18.687	7.722	−4.900	1.625	0.014
upper credibility bound	26.287	10.863	−4.413	2.571	0.017

Table 5.1 Real and estimated values of the parameters for a simulated dataset with a covariate.

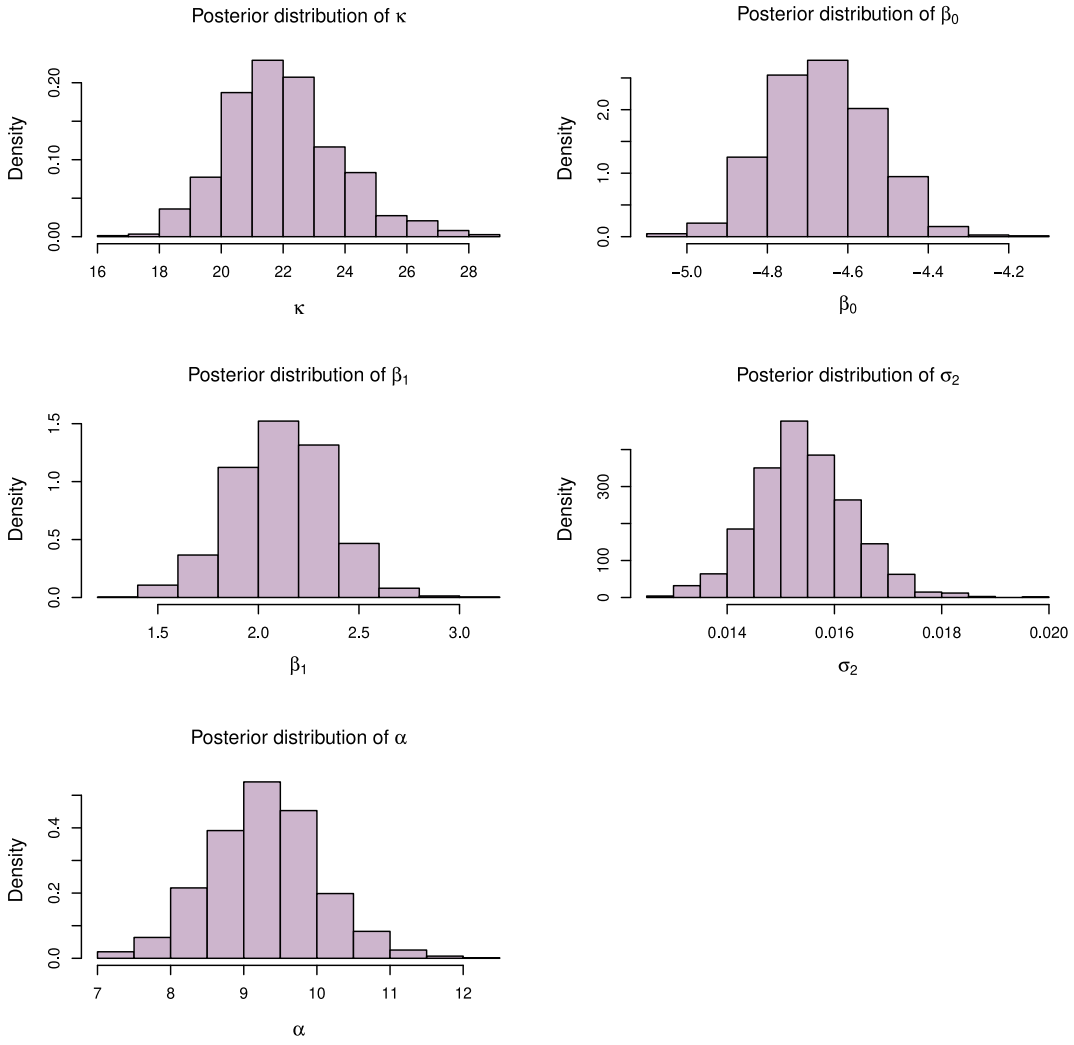

Figure 5.2 Histogram of the samples from the posterior distribution (estimating the posterior probability density function) for simulated data with a covariate present.

Lastly, we take a look at the traceplots of the parameters in Figures 5.3 and 5.4. As we can see, the chains converged to their limiting distribution and the sampler is well mixing. We can also see the traceplot for log-likelihood in Figure 5.5. As we can see, the values do not show significant deviances and are quite stable.

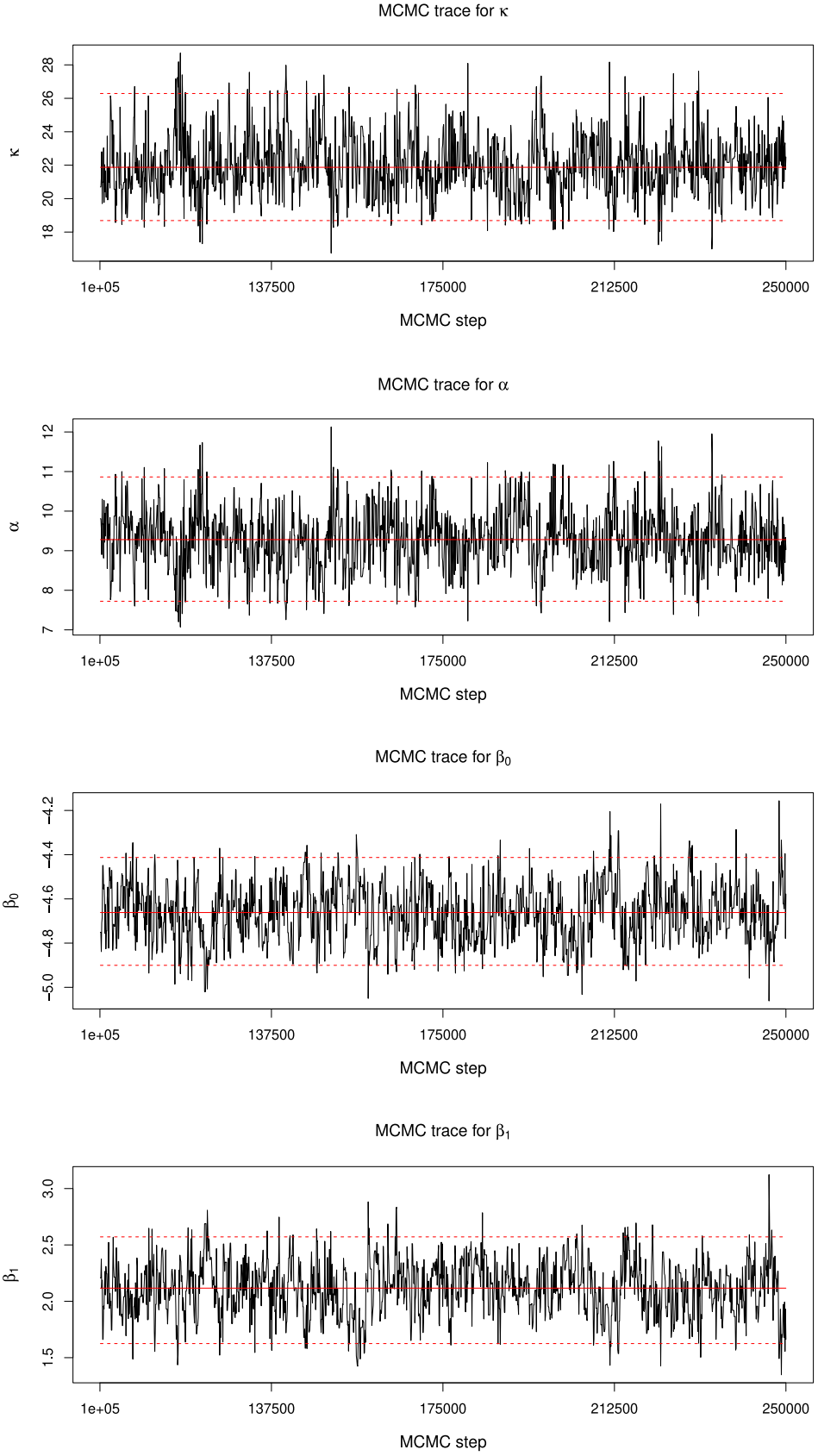

Figure 5.3 Traceplots for model parameters describing the state of the MCMC algorithm for simulated data with a covariate. We saved the values after the burn-in of 100 000 steps and only every 100^{th} computed value is used.

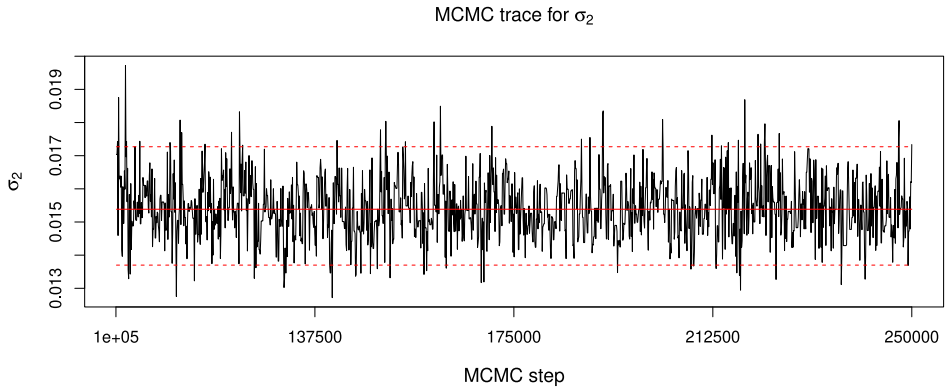

Figure 5.4 Traceplot for σ_2 describing the state of the MCMC algorithm for simulated data with a covariate. We saved the values after the burn-in of 100 000 steps and only every 100^{th} computed value is used.

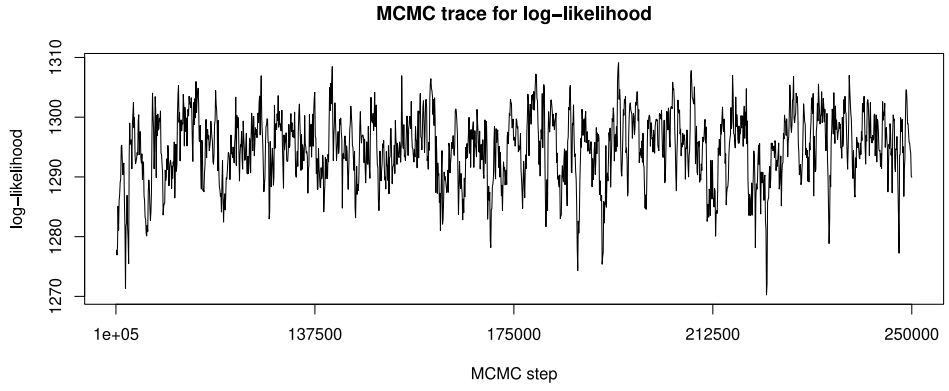

Figure 5.5 Traceplot for log-likelihood during the MCMC algorithm for simulated data with a covariate.

5.2 Analysis for a process not depending on the covariate

In this section, we analyse the situation where we fit the model depending on the covariate from the previous section, but simulate a realization of a process that does not depend on it. If the algorithm works well, the credible interval for $\beta_1^{\sigma_1}$ should contain 0. Thanks to this, we could then test the dependence of the model on covariates, as we would not reject the null hypothesis that the regression parameter $\beta_1^{\sigma_1}$ is equal to zero.

For this example, we generated a Thomas cluster point process with $\kappa = 10$, $\alpha = 20$, $\sigma_2 = 0.06$, $\beta_0^{\sigma_1} = -3.507$ (if no covariate is present, this would correspond to σ_1 being equal to 0.03, as $\log(0.03) \doteq -3.507$), $\beta_1^{\sigma_1} = 0$. We can examine the point pattern generated from the model in the top left picture of Figure 5.6. The corresponding parent point pattern can be seen in the top right picture of this figure.

The initial values of the parameters were $\alpha = 5$, $\sigma_2 = 0.02$, $\beta_0^{\sigma_1} = -3$, $\beta_1^{\sigma_1} = 1$. We used the same prior densities as in the previous section, i.e., $Unif([0.03, 30])$, $Unif([0.001, 0.2])$, $N(0, 5)$ and $N(0, 5)$ for α, σ_2, $\beta_0^{\sigma_1}$ and $\beta_1^{\sigma_1}$, respectively. For the parent point process, we again used the kernel estimate of the intensity function of the daughter point process as the intensity function of the initial parent point process. We can examine the initial parent point pattern in the bottom left picture in Figure 5.6. For the move update, we used standard deviations $\sigma_\circ = \sigma_\bullet = 0.01$, where move, birth and death updates are proposed with probabilities 1/3 each. We performed 250 000 steps with 100 000 step burn-in.

The estimated values of the parameters rounded to 3 decimals can be seen in Table 5.2. As we can see, the credibility interval for $\beta_1^{\sigma_1}$ covers the value 0, and so the algorithm correctly estimated that the dependence on a covariate is not present. For the other parameters, we see that posterior medians are reasonable estimates of the values of the parameters that we used. We would like to remind here that we discussed the lower estimates for κ and the corresponding credible intervals in Section 4.1. We can also see the parent point pattern from the last step of the algorithm in Figure 5.6 in the bottom right picture. We can compare this result with the picture above with the real parent point pattern. We see that they are quite similar, we once again have an extra parent point in the window, but since this is only one sample from the posterior distribution, it should not be given much focus.

In Figure 5.7 we can see the histograms of the samples from the posterior distributions of the parameters. As we can see, all of them are unimodal distributions and although some of them are skewed a little, they are still approximately symmetrical. For $\beta_1^{\sigma_1}$, we see that 0 is a typical value in the posterior distribution.

Finally, we take a look at the traceplots in Figure 5.8 and Figure 5.9. We see that the sampler is well mixing, the values are stabilized and the chains have converged to their limiting distributions. In Figure 5.10, we can see the log-likelihood during the algorithm. As we can see, it also gives nice values that do not change significantly.

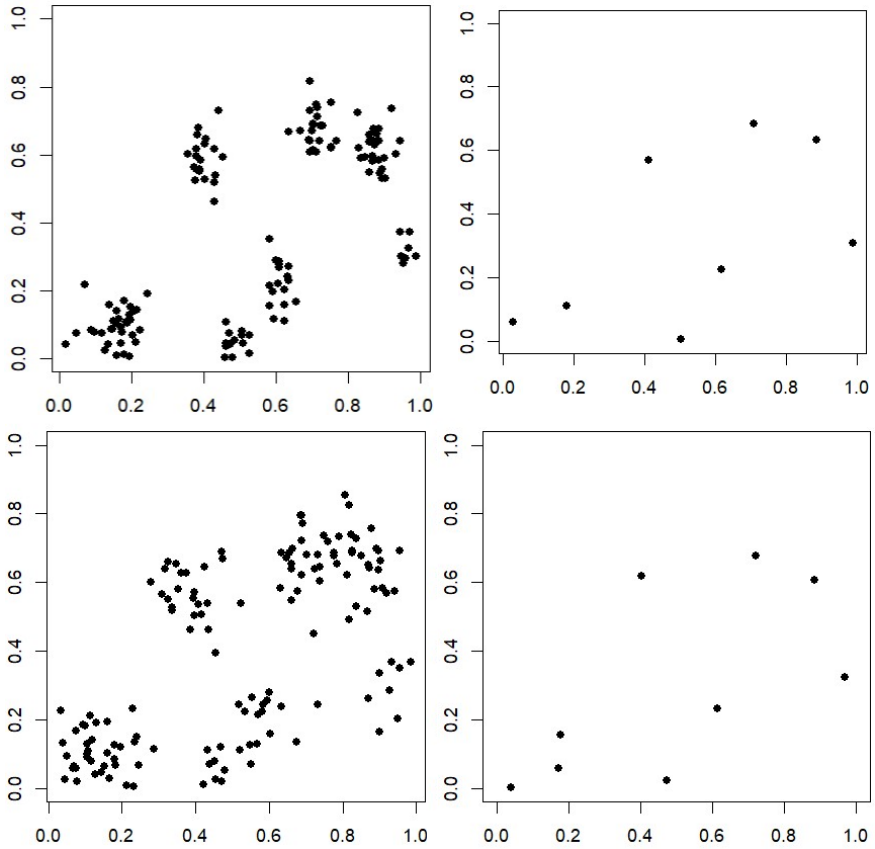

Figure 5.6 Point process not depending on the covariate (top left), corresponding parent point process (top right), initial parent point process (bottom left) and the final point process from the last step of the algorithm (bottom right).

	κ	α	$\beta_0^{\sigma_1}$	$\beta_1^{\sigma_1}$	σ_2
used value	10	20	-3.507	0	0.06
estimated value	7.280	19.093	-3.326	-0.315	0.058
lower credibility bound	6.047	15.287	-3.738	-0.927	0.050
upper credibility bound	9.093	22.987	-2.922	0.282	0.067

Table 5.2 Real and estimated values of the parameters for a simulated dataset without a covariate.

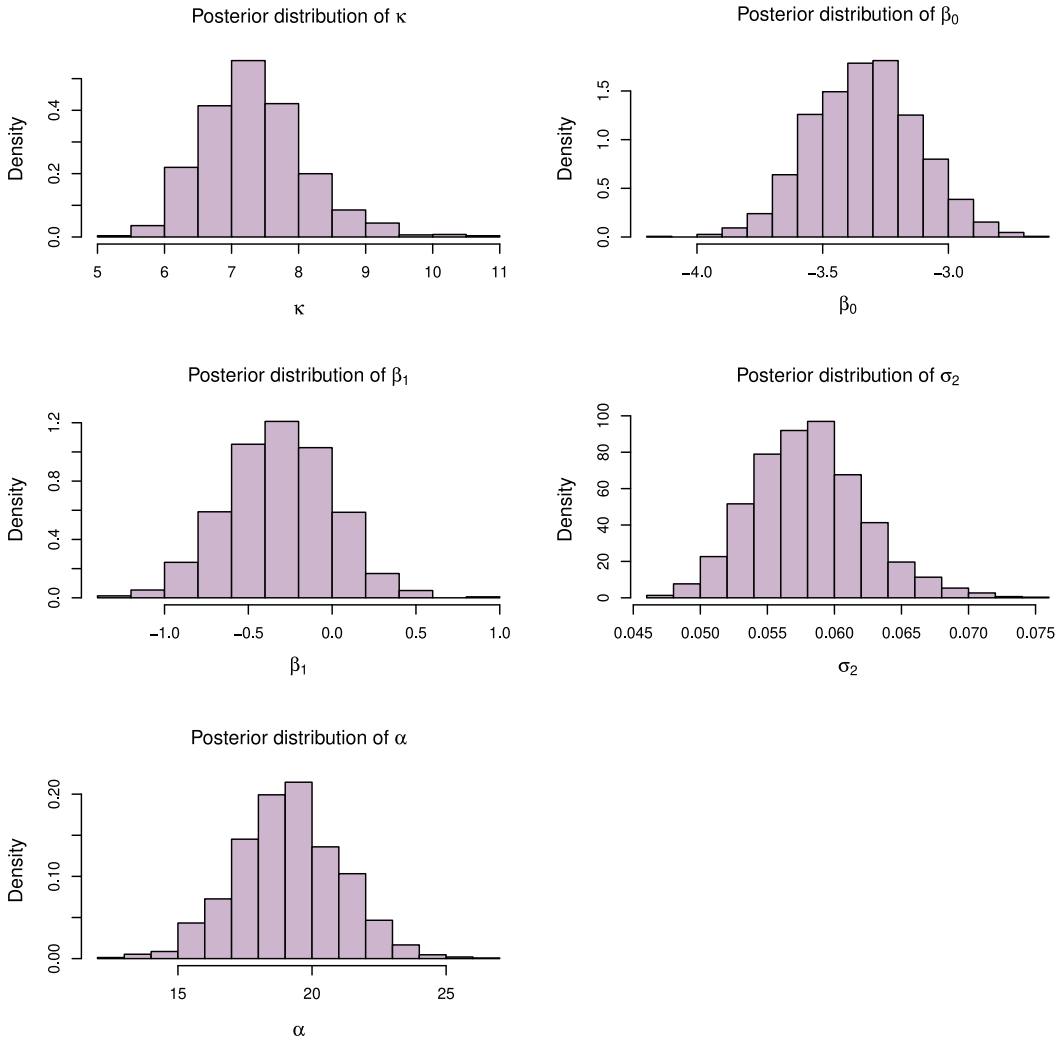

Figure 5.7 Histograms of the samples from the posterior distribution (estimating the posterior probability density function) for a process not depending on a covariate.

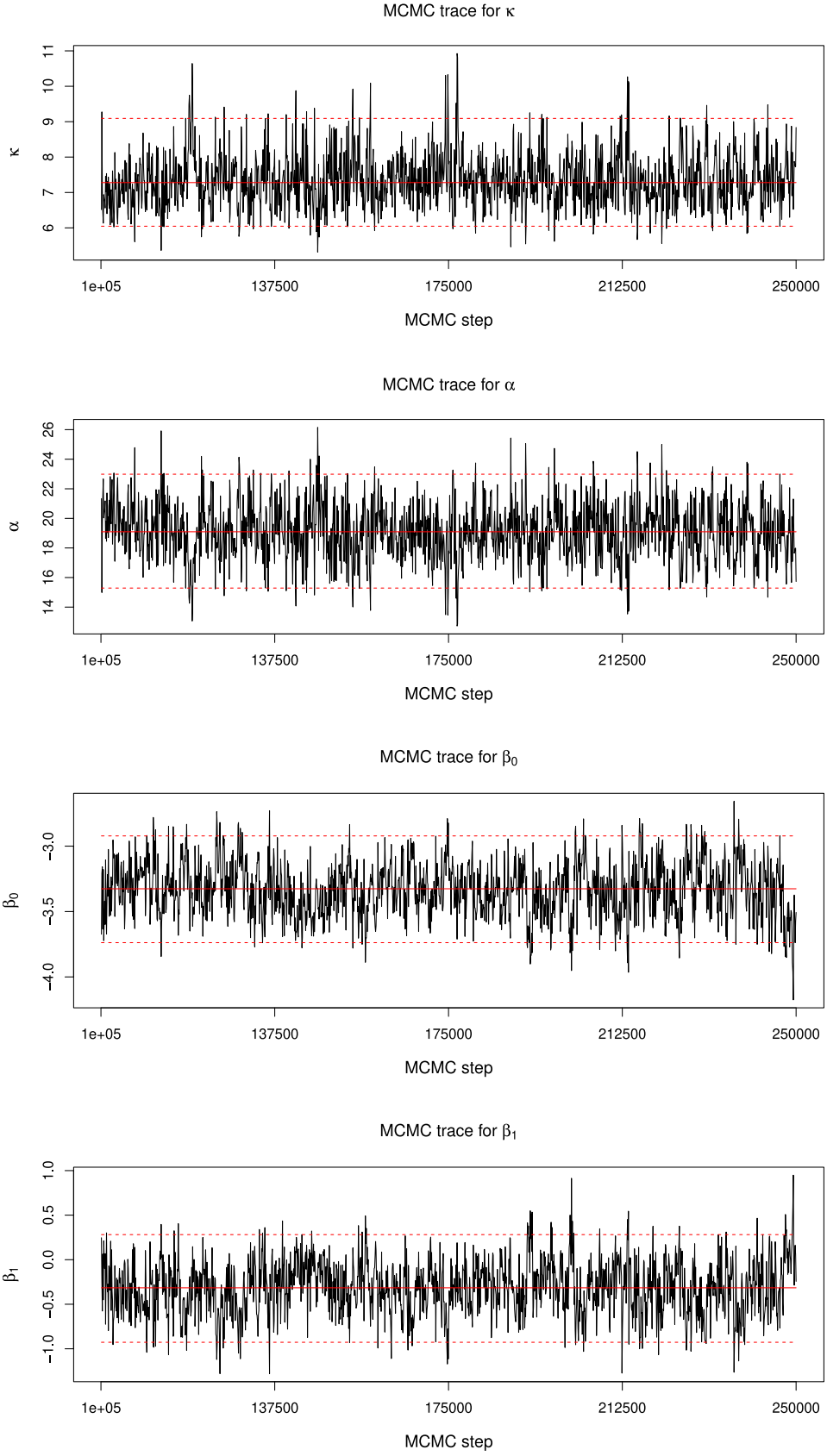

Figure 5.8 Traceplots for model parameters describing the state of the MCMC algorithm for simulated data without a covariate. We saved the values after the burn-in of 100 000 steps and only every 100^{th} computed value is used.

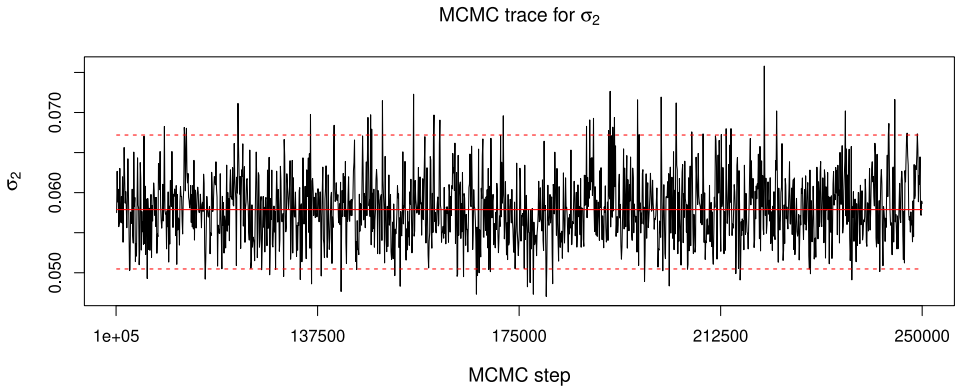

Figure 5.9 Traceplot for σ_2 describing the state of the MCMC algorithm for simulated data without a covariate. We saved the values after the burn-in of 100 000 steps and only every 100^{th} computed value is used.

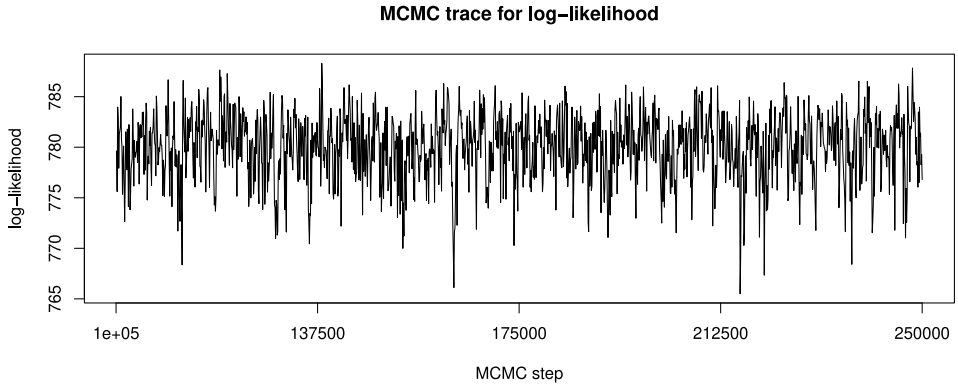

Figure 5.10 Traceplot for log-likelihood during the MCMC algorithm for simulated data without a covariate.

6. Convergence properties

In this chapter, we are going to study the convergence properties of the algorithm described in Chapter 4 in a simplified setting where all the parameters are assumed to be known and the only unknown object of interest is the parent point process. The Metropolis-within-Gibbs algorithm is then simplified and we only do Birth-Death-Move steps. We will use the probability of proposing birth, death or move update equal to 1/3, since we used these values in all of our previous examples. Similarly as in Section 2.2, we can derive that the probability density function of the parent point process is of the form

$$p(C|X) \propto f(X|C)p(C), \tag{6.1}$$

where X denotes the point process of daughter points and C denotes the parent point process.

Since all the parameters are assumed to be known, we do not include them in the densities nor the likelihood. We work on a unit square window W, where we assume that all the parent points are inside the observation window and that we observe at least one daughter point, i.e., $n(X) \geq 1$.

6.1 Definitions and results

To be able to study the properties of the resulting Markov chain, we first need to state several results from the Markov chain theory on a general state space. The definitions and results presented in this section are taken from Roberts and Rosenthal (2004), Durrett (2019), Levin and Peres (2017), Meyn et al. (2009) and Møller and Waagepetersen (2004).

Let $\mathcal{X}$ be a general set and let $\mathcal{B}(\mathcal{X})$ be a countably generated σ-field on $\mathcal{X}$, which we will shortly denote as $\mathcal{B}$. Let us denote $\mathfrak{X} = \prod_{i=1}^{\infty} \mathcal{X}$, where $\prod$ stands for the Cartesian product. Let $\mathfrak{B}$ denote the product σ-algebra $\otimes_{i=1}^{\infty} \mathcal{B}$.

Let us recall the Definition 13 of a Markov kernel. We will use it in a situation where the source and the target are both equal to the space $(\mathcal{X}, \mathcal{B})$, i.e., the kernel is a mapping $P : \mathcal{X} \times \mathcal{B} \to [0, 1]$.

Proposition 1. *For any initial measure μ (i.e. a measure governing the random variable X_0) on $\mathcal{B}$ and any Markov kernel $P = \{P(x, A), x \in \mathcal{X}, A \in \mathcal{B}\}$, there exists a stochastic process $X = \{X_0, X_1, ...\}$ on $\mathfrak{X}$ measurable with respect to $\mathfrak{B}$ and a probability measure $\mathcal{P}_\mu$ on $\mathfrak{B}$ such that $\mathcal{P}_\mu(B)$ is the probability of the event $\{X \in B\}$ for $B \in \mathfrak{B}$; and for $A_i \in \mathcal{B}$ and any $n \in \mathbb{N}_0$*

$$\begin{aligned} &\mathcal{P}_\mu(X_0 \in A_0, X_1 \in A_1, \dots, X_n \in A_n) \\ &\qquad = \int_{A_0} \cdots \int_{A_{n-1}} P(y_{n-1}, A_n) P(y_{n-2}, dy_{n-1}) \cdots P(y_0, dy_1) \mu(dy_0). \end{aligned} \tag{6.2}$$

Proof. See Theorem 3.4.1. in Meyn et al. (2009). □

Definition 16. *A stochastic process $X = (X_0, X_1, ...)$ is called time-homogeneous Markov chain with initial distribution μ and Markov kernel P, if the equation (6.2) is satisfied for finite dimensional distributions of X for every $n \in \mathbb{N}_0$.*

Definition 17. *We define the n-step Markov kernel iteratively. We set*

$$P^0(x, A) = \delta_x(A),$$

where

$$\delta_x(A) = \begin{cases} 1 & x \in A \\ 0 & x \notin A. \end{cases}$$

For $n \geq 1$, we define inductively

$$P^n(x, A) = \int_{\mathcal{X}} P^{n-1}(y, A) P(x, dy), \quad x \in \mathcal{X}, A \in \mathcal{B}.$$

We write P^n for the n-step Markov kernel $\{P^n(x, A), x \in \mathcal{X}, A \in \mathcal{B}\}$.

Definition 18. *A measure π on $(\mathcal{X}, \mathcal{B})$ is said to be a stationary measure of a Markov chain with Markov kernel P if*

$$\int_{\mathcal{X}} P(x, A)\pi(ds) = \pi(A), \quad A \in \mathfrak{B}.$$

If π is a probability measure, we call π a stationary distribution.

Definition 19. *A Markov chain is said to have a limiting distribution, if there exists a probability measure π such that for all $A \in \mathcal{B}$, we have that*

$$\lim_{n\to\infty} P^n(x, A) = \pi(A) \text{ for } \pi\text{-almost all } x \in \mathcal{X}.$$

Definition 20. *A chain is ϕ-irreducible if there exists a non-zero σ-finite measure ϕ on $(\mathcal{X}, \mathcal{B})$ such that for all $A \in \mathcal{B}$ with $\phi(A) > 0$, and for all $x \in \mathcal{X}$, there exists a positive integer $a = a(x, A)$ such that $P^a(x, A) > 0$.*

Definition 21. *A Markov chain with Markov kernel P on the state space $\mathcal{X}$ is reversible with respect to a probability distribution $\pi(\cdot)$ on $\mathcal{X}$, if*

$$\pi(dx)P(x, dy) = \pi(dy)P(y, dx), \quad x, y \in \mathcal{X}.$$

Proposition 2. *If a Markov chain is reversible with respect to $\pi(\cdot)$, then $\pi(\cdot)$ is stationary for the chain.*

Proof. See Proposition 1 in Roberts and Rosenthal (2004). □

Definition 22. *A Markov chain with stationary distribution $\pi(\cdot)$ is aperiodic if there do not exist $d \geq 2$ and disjoint sets $A_1, A_2, ..., A_d \in \mathcal{B}$ with $P(x, A_{i+1}) = 1$ for all $x \in A_i$, $1 \leq i \leq d-1$, and $P(x, A_1) = 1$ for all $x \in A_d$, such that $\pi(A_1) > 0$ (and hence $\pi(A_i) > 0$ for all i). Otherwise, the chain is periodic with period d and periodic decomposition $A_1, ..., A_d$.*

Definition 23. *We define the total variation norm*

$$\|\mu - \nu\|_{TV} = \sup_{A \subseteq \mathcal{X}} |\mu(A) - \nu(A)|$$

for any two probability measures μ and ν defined on $(\mathcal{X}, \mathcal{B})$.

Theorem 1. *If a Markov chain on $(\mathcal{X}, \mathcal{B})$ is ϕ-irreducible and aperiodic, and has a stationary distribution $\pi(\cdot)$, then for π-almost all $x \in \mathcal{X}$,*

$$\lim_{n \to \infty} \|P^n(x, \cdot) - \pi(\cdot)\|_{TV} = 0.$$

In particular, $\lim_{n \to \infty} P^n(x, A) = \pi(A)$ *for all $A \in \mathcal{B}$.*

Proof. See Theorem 4 in Roberts and Rosenthal (2004). □

6.2 Properties of the chain

Let us note that the parent point process is assumed to be a finite point process on $W \subset \mathbb{R}^2$, which we set to be the unit square window $[0, 1]^2$. We can take $d(\xi, \eta) = \|\xi - \eta\|$ the Euclidean distance between the two points in W. By doing this, we ensure that W is a Polish space. Let us equip $\mathcal{N}_f^*(W)$, the space of all simple finite counting measures on W, with the σ-algebra

$$\mathfrak{N}_f^*(W) = \sigma\left(\left\{\bar{x} \in \mathcal{N}_f^* : \bar{x}(B) = m\right\},\ B \in \mathcal{B}_0(W), m \in \mathbb{N}_0\right),$$

where $\mathcal{B}_0(W)$ denotes the class of bounded Borel sets on W. Now thanks to the Proposition $B.1$ in Møller and Waagepetersen (2004), we know that $\mathfrak{N}_f^*(W)$ is countably generated, and hence the theory from Section 6.1 applies here.

We now have everything ready to show some properties of the Markov chain generated by the Birth-Death-Move Algorithm. We denote h the unnormalized density of the parent point process with respect to a unit Poisson process, i.e., it corresponds to $f(X|C)p(C)$ in the relation (6.1).

Proposition 3. *The Markov chain generated by the Birth-Death-Move algorithm is reversible with respect to h.*

Proof. See Proposition 7.15 in Møller and Waagepetersen (2004). □

From Proposition 3, we have that Algorithm 1 is reversible with respect to the (unnormalized) posterior distribution of the Markov chain it generates. Moreover, Proposition 2 guarantees that the (normalized) posterior distribution is a stationary distribution of the chain.

We shall show that the resulting chain is also aperiodic.

Proposition 4. *The Markov chain generated by the Birth-Death-Move algorithm is aperiodic.*

Proof. We will prove this by contradiction. Let π be the stationary distribution of the chain. Let us assume that there exist disjoint sets $A_1, A_2, \dots A_d \in \mathfrak{N}_f^*$, $d \geq 1$, such that $P(x, A_{i+1}) = 1$ for all $x \in A_i$, $1 \leq i \leq d$, and $P(x, A_1) = 1$ for all $x \in A_d$, such that $\pi(A_1) > 0$. Let $\bar{x} = \{x_1\} \in A_k$, for some $k \in \mathbb{N}$, be a state of the Markov chain where there is only one parent point x_1 in W. Then the death update is proposed with probability 1/3. But we know that at least one parent point must be present, since we have that $n(X) \geq 1$, hence the acceptance probability of such update is zero. To show that, let us first denote C the current point pattern, i.e., $C = \bar{x} = \{x_1\}$, and C' the proposed parent point pattern, i.e., $C' = \emptyset$. Then the the acceptance probability of the death update under the assumptions we made is of the form

$$\frac{f(X|C')}{f(X|C)} \cdot \frac{1}{\kappa},$$

and $f(X|C') = 0$ since in the product in (3.6), the factor

$$\prod_{x \in X} \left(\sum_{c \in C'} \exp\left\{ -\frac{1}{2} \left[\frac{(x_1 - c_1)^2}{\sigma_1^m} + \frac{(x_2 - c_2)^2}{\sigma_2^m} \right] \right\} \right)$$

is equal to 0. This means that $P(\bar{x}, A_{k+1}) < 1$ and therefore the chain is aperiodic. □

Now we will discuss the proof of ϕ-irreducibility.

We assume that $n(X) \geq 1$, which means that at least one parent point must be present in the observation window W, otherwise no daughter points could be generated. Since the acceptance probability of the proposal of deleting the last parent point is 0, which we proved in Proposition 4, we know that $P^m(x, \emptyset) = 0$ for all $m \in \mathbb{N}$, $x \in \mathcal{N}_f^*$ such that $n(x) \geq 1$. It is interesting that because of this, we cannot derive the ϕ-irreducibility analogously as in Proposition 7.13 in Møller and Waagepetersen (2004). The main issue is that we are unable to construct a measure ϕ in a similar way as in the aforementioned proposition, since the empty set plays a significant role there and cannot be replaced with any other configuration so that the proof would have come through similarly. Therefore we cannot use the proof technique that we are familiar with.

If we change the Birth-Death-Move Algorithm so that in the birth update, the proposal distribution would be of the form

$$\varepsilon \cdot \delta_{x_0}(\cdot) + (1 - \varepsilon) \cdot \frac{1}{|W|} \mathbb{1}\left[\cdot \in W\right],$$

for some $x_0 \in W$, $1 > \varepsilon > 0$, then we could set

$$\phi(A) = \mathbb{1}\left[\{x_0\} \in A\right], \quad A \in \mathfrak{N}_f^*,$$

and we could proceed with the proof similarly as in the proposition. We would also need to verify that we did not generate the point x_0 if this point is already in the point pattern. We would then need to prove the aperiodicity and reversibility in such a case, because we would not be able to refer to the proofs given in Section 7.3.2. in Møller and Waagepetersen (2004) since they rely on the assumption of the proposal distribution being continuous.

Conclusion

In this thesis we discussed a new method that can be used to estimate model parameters for anisotropic point processes.

In the first chapter, we gave a brief summary of basic definitions concerning point processes on $\mathbb{R}^d$. We also mentioned a few particular examples of point processes and described a general model for anisotropic cluster point processes.

In the second chapter, we gave a description of the basics of Bayesian statistics and then derived the specific situation which we used for the model studied.

In the third chapter, we outlined the algorithm used, giving in depth description of each part and derived all the functions featured there for the studied model. We also did a short discussion of possible changes to the algorithm.

The main contribution of this work lies within two simulation studies, which were described in Chapter 4 and 5. We verified that the presented method provides reasonable estimates and provided a full analysis of a simulated dataset with all the possible outputs from the algorithm that one could have a look at. We also discussed the problems when specific parameters are used and showed the limitations of the algorithm on another example. We also applied the method to a real-life dataset. Afterwards, we presented a situation in which the method can be used for hypothesis testing and did one test ourselves. We therefore demonstrated that the method works well, if the problematic parts are adequately controlled. We illustrated the use of the algorithm on the most straightforward examples and with only one possible covariate present in the parametrization. It is therefore possible to extend this method onto more complex examples and study its properties there.

We also proved some properties of the algorithm when the model is simplified and discussed the issues we encounter when exploring the convergence properties of the Markov chain. No available literature provides suitable theorems or results for such complex problems, but as this area has been studied in the past years intensively, we can expect some general results for such models in the future.

Bibliography

Durrett, Rick, 2019. *Probability: theory and examples*. Vol. 49. Cambridge university press.

Dvořák, Jiří; Remeš, Radim; Beránek, Ladislav; Mrkvička, Tomáš, 2022. *binspp: An R Package for Bayesian Inference for Neyman-Scott Point Processes with Complex Inhomogeneity Structure* [online]. [visited on 2024-04-11]. Available from arXiv: 2205.07946 [stat.ME].

Hastings, Wilfred Keith, 1970. Monte Carlo sampling methods using Markov chains and their applications. *Biometrika*. Vol. 57, no. 1, pp. 97–109.

Kopecký, Jiří; Mrkvička, Tomáš, 2016. On the Bayesian estimation for the stationary Neyman-Scott point processes. *Applications of Mathematics*. Vol. 61, pp. 503–514.

Levin, David A; Peres, Yuval, 2017. *Markov chains and mixing times*. Vol. 107. American Mathematical Soc.

Metropolis, Nicholas; Rosenbluth, Arianna W; Rosenbluth, Marshall N; Teller, Augusta H; Teller, Edward, 1953. Equation of state calculations by fast computing machines. *The journal of chemical physics*. Vol. 21, no. 6, pp. 1087–1092.

Meulen, Frank van der; Schauer, Moritz, 2022. *Automatic Backward Filtering Forward Guiding for Markov processes and graphical models* [online]. [visited on 2024-04-22]. Available from arXiv: 2010.03509 [stat.CO].

Meyn, Sean; Tweedie, Richard L.; Glynn, Peter W., 2009. *Markov Chains and Stochastic Stability*. 2nd ed. Cambridge University Press. Cambridge Mathematical Library.

Møller, Jesper; Toftaker, Håkon, 2014. Geometric Anisotropic Spatial Point Pattern Analysis and Cox Processes. *Scandinavian Journal of Statistics*. Vol. 41, no. 2, pp. 414–435.

Møller, Jesper; Waagepetersen, Rasmus Plenge, 2004. *Statistical inference and simulation for spatial point processes*. CRC press.

Neyman, Jerzy; Scott, Elizabeth L., 1958. Statistical Approach to Problems of Cosmology. *Journal of the Royal Statistical Society. Series B (Methodological)*. Vol. 20, no. 1, pp. 1–43.

Pawlas, Zbyněk, 2023. *Spatial Modelling* [online]. [visited on 2024-04-15]. Available from: https://www.karlin.mff.cuni.cz/~pawlas/2024/MTP438/spatial.pdf.

Rataj, Jan, 2006. *Bodové procesy*. Bodové procesy. 2., opravené vydání. Praha: Karolinum. ISBN 80-246-1182-1.

Robert, Christian P., 2016. *The Metropolis-Hastings algorithm* [online]. [visited on 2024-04-01]. Available from arXiv: 1504.01896 [stat.CO].

Roberts, Gareth O.; Rosenthal, Jeffrey S., 2004. General state space Markov chains and MCMC algorithms. *Probability Surveys*. Vol. 1. Available from DOI: 10.1214/154957804100000024.

Thomas, Marjorie, 1949. A Generalization of Poisson's Binomial Limit For use in Ecology. *Biometrika*. Vol. 36, no. 1/2, pp. 18–25.

van Ravenzwaaij, Don; Cassey, Pete; Brown, Scott D., 2016. A simple introduction to Markov Chain Monte–Carlo sampling. *Psychonomic bulletin & review*. Vol. 25, no. 1, pp. 143–154.

Watanabe, Sumio, 2018. *Mathematical theory of Bayesian statistics*. Mathematical theory of Bayesian statistics. First edition. CRC Press. isbn 1-315-35569-8.

List of Figures

List of Tables

A. Attachments

A.1 Electronic attachments

The electronic version of this thesis contains 3 R scripts with implemented method destribed here. The first one contains the implementation of the function generating anisotropic cluster point processes described in full generality in Chapter 1, Sections 1.3 and 1.4. The second one contains the implementation for the situations from Chapter 4 and the third one contains the implementation for the situations from Chapter 5.