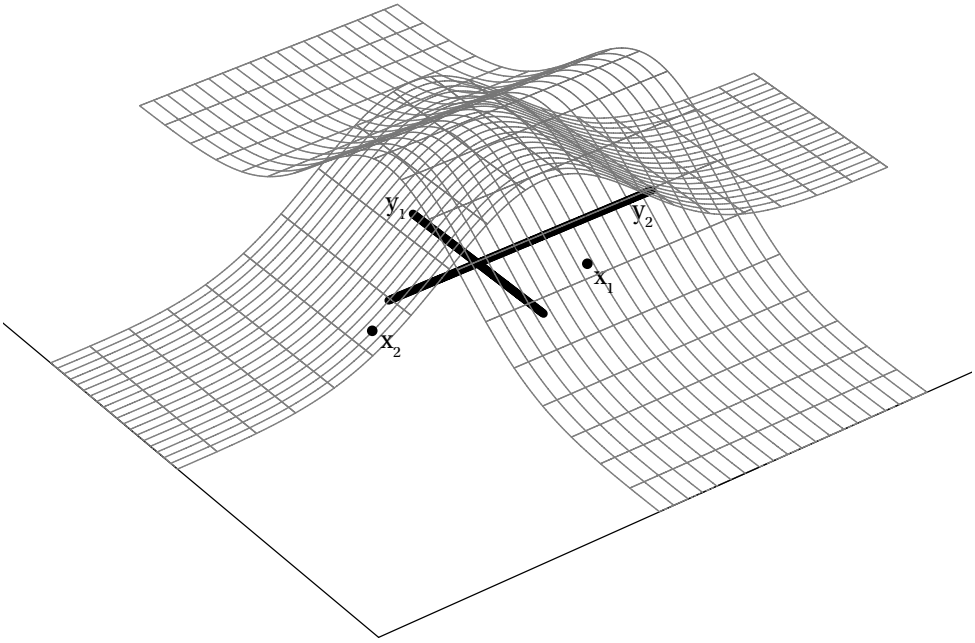
AALBORG UNIVERSITET

Institut for Matematiske Fag – www.math.aau.dk

MODELLERING AF RUMLIGE PUNKTMØNSTRE MED LINEÆRE STRUKTURER



Speciale i matematik af Ronni Post

AALBORG UNIVERSITET

Institut for Matematiske Fag
Fredrik Bajers Vej 7G
9220 Aalborg Øst
Telefon 96 35 88 04
Fax 98 15 81 29
http://www.math.aau.dk

Titel: Modellering af rumlige punktmønstre med lineære strukturer

Periode: 10. semester, efterår 2011

Forfatter:

Ronni Yde Post

Vejleder: Jakob Gulddahl Rasmussen

Oplagstal: 4

Sidetal: 108

Afsluttet: 5. januar 2012

Synopsis:

I specialet opstilles en model til at beskrive todimensionale punktmønstre med lineære strukturer, hvor der tages udgangspunkt i et datasæt om gravhøje. Den opstillede model er en Cox punktproces, som er inspireret af Thomas punktprocessen.

Der gennemgås grundlæggende teori om punktprocesser med fokus på typen Poisson og Cox, og der redegøres for forskellige deskriptive størrelser til bl.a. modelkontrol samt for anvendelsen af MCMC-metoder til at udføre inferens.

I den praktiske del af specialet gives en detaljeret beskrivelse af den foreslåede model, og det vises, hvorledes den kan implementeres, og hvordan der kan udføres inferens for parametrene. Herefter udføres en modelkontrol for at sammenligne modellens overensstemmelse med datasættet, og der afsluttes med en diskussion af muligheder for forbedring af model og inferens.

Abstract

The master's thesis is concerned with the issue of modelling spatial point patterns with linear structures. A dataset of Danish barrows is used as a starting-point for proposing a parametric model able to represent the kind of linear structures that are present in the dataset. For this purpose a combination of Poisson point processes and random independent displacement is used to represent line segments and points respectively. The model is analysed for the purpose of inference regarding the parameters and various approaches to model checking is performed. Simulations of the model, an algorithm for estimating parameters and functions for performing model checking has been implemented in R.

Before introducing the model, Chapter 1 introduces the theory of spatial point processes that is used in later chapters. Basic definitions, summary statistics, Poisson point processes and Cox point processes is treated, and the chapter also contains a section on using MCMC-methods for simulation. The model is presented in Chapter 2 where also various issues regarding implementing the model is discussed, and some examples of point patterns created by simulating the model are shown. The chapter is concluded by determining a density for the model wrt. the standard Poisson point process.

A bayesian approach utilising Metropolis-within-Gibbs is used for inference, so likelihoods, priors, posteriors and Hastings ratios are the topics in the first sections of Chapter 3 after which a detailed description of an estimation algorithm is given. Inference is first performed for simulated data using various methods and secondly for a reduced version of the dataset as the full dataset turns out to be too large to handle within reasonable time. The results are used for model checking in Chapter 4 which starts with a visual comparison of simulated point patterns and the data. Other model checks are performed including looking at the distribution of angles and comparing the summary statistics from Chapter 1. It turns out the model is fairly suitable for representing the linear structures themselves, however it is not performing well in representing clusters in the data, so the final conclusion and discussion is used for evaluating the challenges regarding estimation parameters for the full dataset and improving the model.

Forord

Dette speciale er skrevet som afslutningen af kandidatuddannelsen i matematik med specialiseringsretningen statistik ved Institut for Matematiske Fag ved Aalborg Universitet. Specialet er skrevet indenfor rumlig statistik med fokus på punktprocesser, og det er både af teoretisk og anvendelsesorienteret art.

Der er på tidligere semestre bl.a. arbejdet med teori om bayesiansk statistik og punktprocesser, og i specialets kapitel 1 er der genanvendt dele fra et tidligere projekt om punktprocesser. Materialet er dog blevet tilpasset og omstruktureret, så det bedst muligt passer til den anvendelse, der finder sted i senere kapitler, og det er yderligere oversat fra engelsk.

Ud over selve dette skriftlige indhold af specialet, så er der også skrevet en omfattende mængde kode i `R`, [R Development Core Team, 2011], hvoraf en del er lavet vha. pakkerne `coda`, [Plummer et al., 2006], og `spatstat`, [Baddeley & Turner, 2005]. Det mest centrale kode er inkluderet som bilag sidst i specialet.

Der skal lyde en tak til Jakob G. Rasmussen for vejledning gennem forløbet.

Vedr. henvisninger og notation

Henvisninger følger Harvard-formatet og er på formen [forfatter(e), år [, placering i kilden]]. Der henvises ikke eksplicit til grundlæggende resultater om sandsynlighedsregning og bayesiansk statistik, for mere information herom kan anvendes henholdsvis [Olofsson, 2005] og [Lee, 2004].

Om notationen bør der bemærkes følgende:

- log er den naturlige logaritmefunktion ln.
- Der anvendes fede bogstaver til at betegne punktprocesser og punktmønstre, eksempelvis $\boldsymbol{X}$ og $\boldsymbol{x}$. Vektorer skrives *ikke* med fede bogstaver, det bør fremgå af konteksten, hvorvidt et symbol er en vektor eller ej.
- Middelværdien af X skrives $\mathbf{E}[X]$, variansen som $\mathbf{Var}[X]$, og indikatorfunktioner som $\mathbf{1}[\boldsymbol{X} \in F]$. Pga. mange komplekse udtryk anvendes eksplicitte parenteser ved produkter og summer, eksempelvis $\sum_{i=1}^{n} [l_i]$ og $\prod_{i=1}^{n} \left[\rho(u)\right]$.

Indholdsfortegnelse

Introduktion **1**

Kapitel 1 Punktprocesser og MCMC-metoder **3**
1.1 Grundlæggende teori om punktprocesser 3
1.2 Deskriptive størrelser 8
1.3 Poisson punktprocesser 17
1.4 Cox punktprocesser 36
1.5 Markovkæde Monte Carlo og Gibbs sampling 41

Kapitel 2 Opstilling og implementation af model **49**
2.1 Model-specifikation 49
2.2 Implementation af modellen 51
2.3 Eksempler på genererede punktmønstre 55
2.4 Tætheder for modellen 57

Kapitel 3 Inferens for modellens parametre **59**
3.1 Likelihood og posterior 59
3.2 Opbygning af algoritmen 61
3.3 Implementation af algoritmen 67
3.4 Inferens for simulerede data 70
3.5 Inferens for datasættet om gravhøje 78

Kapitel 4 Modelkontrol **83**
4.1 Visuelle sammenligninger 83
4.2 Fordeling af vinkler 86
4.3 Deskriptive størrelser 88

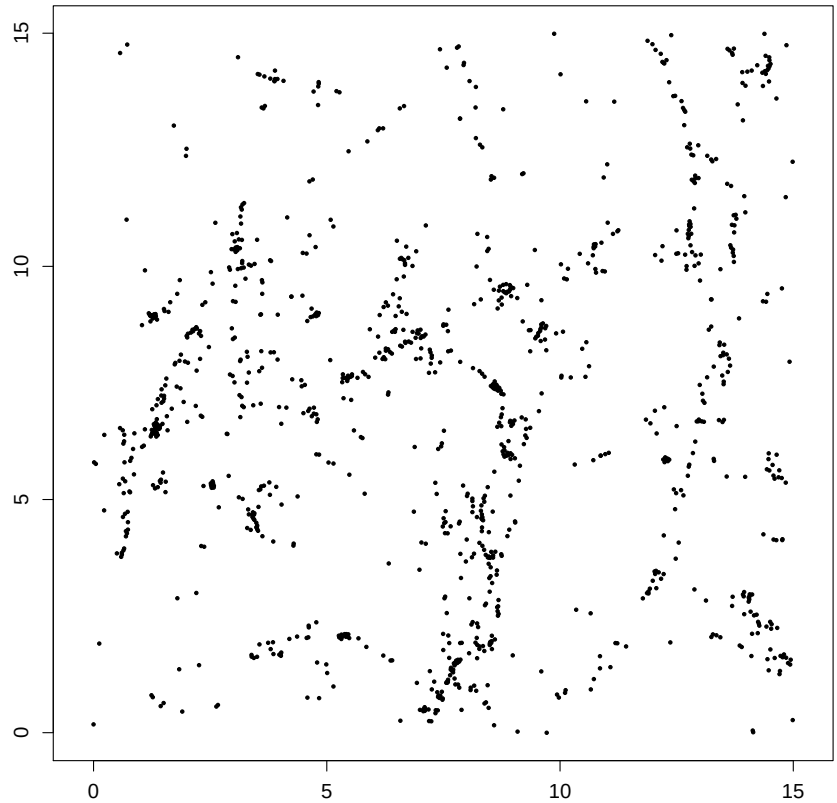
Konklusion og diskussion **93**

Referencer **97**

Bilag A R-kode **101**
A.1 R-kode til simulation af modellen 101
A.2 R-kode til estimation af parametre 103
A.3 R-kode til modelkontrol 107

Introduktion

Mange af virkelighedens datasæt består af punktmønstre, der er opgivet som koordinater i to eller flere dimensioner, og der tales i så fald om rumlige punktmønstre. Et eksempel på et observeret todimensionalt punktmønster ses på figur 0.1, hvor punkterne angiver placeringen af gravhøje i et område på 15×15 km^2 i nærheden af Skjern i Vestjylland. Det viste plot indeholder i alt 1147 gravhøje, og det er et udsnit af et større datasæt med i alt 4171 gravhøje fordelt på et område på ca. 29×45 km^2.



Figur 0.1 *– Plot af 1147 gravhøjes placering i et område på* 15×15 *km*^{2} *ved Skjern i Vestjylland.*

Ved at betragte plottet ses det, at der mange steder er en tendens til, at punkterne udviser en lineær struktur, dvs. at de ligger omtrent langs rette linjestykker, hvilket bl.a. tilskrives, at mange af gravhøjene menes at være opført i nærheden af forhistoriske veje. Noget tilsvarende kan observeres i en række andre datasæt, og det er derfor relevant at forsøge og opstille statistiske modeller, der genererer punktmønstre, som udviser sådanne lineære strukturer.

Således er både dette datasæt og tilsvarende datasæt med lineære strukturer tidligere forsøgt modelleret på forskellig vis. I et tidligere speciale, [Haals & Jensen, 2009], anvendes det samme datasæt om gravhøje, og der bliver anvendt en såkaldt linjeproces til at generere et underliggende antal rette linjer, som repræsenterer veje, gennem hele observationsvinduet. Derefter genereres de egentlige punkter langs disse linjer. Specialet indeholder yderligere en uddybning af forskellige historiske forhold og sammenligninger mellem datasættet og kendte forhistoriske veje baseret på arkæologiske kilder. Derved søges det at styrke formodningen om, at en væsentlig del af gravhøjene vitterligt er opført langs veje, og at det dermed er på et validt grundlag, at der forsøges at modellere en statistisk model for gravhøjenes placering ved brug af modeller, som indeholder lineære strukturer.

En anden tilgang til at modellere både dette datasæt samt et andet om bjergtoppes placering i Pyrenæerne gives i [Møller & Rasmussen, 2010]. Deri introduceres en model, hvor punkterne genereres sekventielt, og hvert punkt bliver klassificeret som enten et baggrundspunkt, der ikke har nogen tilknytning til de lineære strukturer, eller som et uafhængigt eller afhængigt klyngepunkt. De afhængige klyngepunkter afhænger af det forrige punkt i klyngen og placeres tæt herpå, og pga. den valgte tæthedsfunktion får klyngerne karakter af at ligge langs rette linjestykker. Bemærk at der for denne model ikke anvendes en underliggende model af rette linjer, den lineære struktur fremkommer gennem den sekventielle generering af punkter.

I dette speciale er formålet at opstille en ny type model, som også genererer punktmønstre med lineære strukturer, men som adskiller sig fra de to ovennævnte tilgange. I lighed med modellen fra [Haals & Jensen, 2009] vil der blive anvendt en underliggende model af veje, men der vil ikke være tale om linjer, som går gennem hele observationsvinduet men derimod om linjestykker, der er centreret omkring nogle punkter, og som har forskellige længder. På hvert af linjestykkerne vil der blive genereret et antal punkter, som til sidst forskydes i forhold til linjestykket for at få nogle punktmønstre, der har lineære strukturer, men uden at punkterne ligger præcist på linjestykkerne.

Punktprocesser og MCMC-metoder 1

I dette kapitel præsenteres den teori, der ligger til grund for det praktiske arbejde med modellering, inferens og modelkontrol i de efterfølgende kapitler. Der introduceres først i afsnit 1.1 de mest fundamentale definitioner og begreber vedr. punktprocesser, og i afsnit 1.2 redegøres for en række deskriptive størrelser, der kan anvendes til at beskrive karakteristiske egenskaber ved punktprocesser. Den opstillede model bygger bl.a. på Poisson punktprocesser, som indføres i afsnit 1.3 sammen med en mængde teoretiske resultater herom, og i afsnit 1.4 beskrives Cox punktprocesser. Kilden for disse første fire afsnit er med mindre andet er nævnt [Møller & Waagepetersen, 2004, kap. 2–5]. Kapitlet afsluttes med afsnit 1.5, hvor der gennemgås metoder til at simulere komplicerede tætheder.

1.1 Grundlæggende teori om punktprocesser

1.1.1 Målteori

Før de egentlige definitioner vedr. punktprocesser gives, er det relevant at nævne et par aspekter om målteori. Det skyldes, at en del af de teoretiske resultater i dette kapitel hviler på målteoretiske detaljer. Der vil dog kun i sparsom grad blive anvendt sådanne detaljer, og derfor introduceres kun de mest elementære koncepter.

Definition 1.1 (Borelmængde):
En *Borelmængde* $B \subseteq S$ for en arbitrær mængde S er enhver mængde, der kan konstrueres af åbne delmængder af S under følgende betingelser:

(i) Hvis $B \in \mathcal{B}$, så er $S \setminus B \in \mathcal{B}$.
(ii) Hvis $B_i \in \mathcal{B}$ for $i = 1, 2, \ldots$, så er $\bigcup_{i=1}^{\infty} B_i \in \mathcal{B}$,

hvor $\mathcal{B}$ er mængden af alle Borelmængder.

Når der i dette speciale omtales en mængde, så er det underforstået, at der er tale om en Borelmængde. Et andet koncept, som vil blive anvendt, er begrebet *mål*.

Definition 1.2 (Mål):
Et *mål* er en funktion $\nu : \mathcal{B} \to [0\,;\infty]$, som opfylder:

(i) $\nu(\emptyset) = 0$.

(ii) Hvis $B_i \in \mathcal{B}, i = 1,2,\ldots$, er et tælleligt antal disjunkte mængder, så er

$$\nu\left(\bigcup_{i=1}^{\infty} B_i\right) = \sum_{i=1}^{\infty} \Big[\nu(B_i)\Big].$$

1.1.2 Punktprocesser på $\mathbb{R}^d$

Definition 1.3 (Punktproces og punktmønster):
En *punktproces* $\boldsymbol{X}$ er en stokastisk tællelig delmængde af et rum S. En realisation $\boldsymbol{x}$ af en punktproces $\boldsymbol{X}$ kaldes et *punktmønster*.

I almindelighed er $S \subseteq \mathbb{R}^d$. Når det kommer til egentlige observationer, så ses på punkterne i et begrænset *observationsvindue* $W \subseteq S$, der ofte er en d-dimensional kasse, men W kan have enhver form. Hvis kun en del af $\boldsymbol{X}$ eller $\boldsymbol{x}$ skal betragtes, eksempelvis den del, som er indeholdt i en begrænset mængde $B \subseteq S$, så skrives denne restriktion som henholdsvis $\boldsymbol{X}_B := \boldsymbol{X} \cap B$ og $\boldsymbol{x}_B := \boldsymbol{x} \cap B$. Antallet af punkter i $\boldsymbol{X}$ eller $\boldsymbol{x}$ betegnes henholdsvis med $n(\boldsymbol{X})$ og $n(\boldsymbol{x})$, men dette anvendes dog forholdsvis sjældent i praksis, fordi der i de fleste tilfælde fokuseres på en restriktion til $B \subseteq S$. I så fald skrives $n(\boldsymbol{X})$ som $n(\boldsymbol{X}_B)$, og der anvendes oftest *tællefunktionen* $N : S \to \mathbb{N}_0$ til at betegne det $N(B) := n(\boldsymbol{X}_B)$.

Definition 1.4 (Lokalt endelig):
Hvis $n(\boldsymbol{x}_B) < \infty$ for enhver begrænset mængde $B \subseteq S$, så siges $\boldsymbol{x}$ at være *lokalt endelig*. Rummet af alle lokalt endelige $\boldsymbol{x}$ betegnes

$$N_{\text{lf}} := \big\{\boldsymbol{x} \subseteq S \mid n(\boldsymbol{x}_B) < \infty \text{ for enhver begrænset mængde } B \subseteq S\big\}.$$

Elementerne i N_{lf} kaldes *lokalt endelige punktkonfigurationer* eller blot *punktmønstre*.

Definition 1.5 (Fordelingen af en punktproces):
Den endeligtdimensionale fordeling af tællefunktionen for en punktproces $\boldsymbol{X}$ kaldes *fordelingen af punktprocessen.*

Definitionen betyder, at fordelingen af en punktproces $\boldsymbol{X}$ er bestemt af den simultane fordeling af $N(B_1),\ldots,N(B_n)$ for alle begrænsede mængder $B_1,\ldots,B_n$, $n\in\mathbb{N}$, og det er dermed et alternativ at definere en punktproces ud fra en stokastisk tællefunktion. En punktproces siges at være *simpel*, hvis der ingen sammenfaldende punkter er, og der vil i dette speciale kun blive anvendt sådanne.

Et vigtigt koncept for punktprocesser er begrebet *intensitet*, hvor en punktproces betragtes ud fra det forventede antal punkter i en mængde B.

Definition 1.6 (Intensitetsmål):
For en punktproces $\boldsymbol{X}$ er *intensitetsmålet* $\mu(B)$ det forventede antal punkter i en begrænset mængde B, dvs. $\mu(B) := \mathbf{E}[N(B)]$ for begrænset $B\subseteq S$.

Frem for intensitetsmålet μ benyttes ofte intensitetsfunktionen $\rho(\cdot)$.

Definition 1.7 (Intensitetsfunktion):
Hvis intensitetsmålet μ kan skrives som

$$\mu(B) \;=\; \mathbf{E}[N(B)] \;=\; \int_B \rho(u)\,\mathrm{d}u, \qquad \rho\geq 0\,,\, B\subseteq\mathbb{R}^d$$

så kaldes ρ *intensitetsfunktionen* eller blot *intensiteten.*

Idet der kun anvendes simple punktprocesser, så er der en intuitiv fortolkning af intensiteten ρ, hvis der betragtes et infinitesimalt lille område A. Så er

$$\rho(u)|A| \;\approx\; \mu(A) \;=\; \mathbf{E}[N(A)] \;\approx\; P(\boldsymbol{X} \text{ har et punkt i } A),$$

så intensiteten $\rho(u)$ angiver, hvor kompakt punkterne er spredt omkring et arbitrært $u\in S$. Det giver anledning til følgende definition.

Definition 1.8 (Homogenitet):
Hvis ρ er konstant, så siges en punktproces $\boldsymbol{X}\subseteq S$ at være *homogen* på S med rate eller intensitet ρ, ellers siges $\boldsymbol{X}$ at være *inhomogen* på S.

Bemærk at for områder af $A \subseteq S$ med enhedsstørrelse, da følger det af definition 1.7 og 1.8, at når ρ er konstant, så er $\mu(A) = \rho|A|$, dvs. ρ er det gennemsnitlige antal punkter pr. enhedsstørrelse.

Definition 1.9 (Stationaritet og isotropi):
En punktproces $\boldsymbol{X} \subseteq \mathbb{R}^d$ siges at være *stationær*, hvis dens fordeling er invariant under parallelforskydning, dvs. hvis $\boldsymbol{X} \sim \{u + s \,|\, u \in \boldsymbol{X}\}$ for alle $s \in \mathbb{R}^d$.

En punktproces $\boldsymbol{X} \subseteq \mathbb{R}^d$ siges at være *isotropisk*, hvis dens fordeling er invariant under rotation, dvs. hvis $\boldsymbol{X} \sim \{Ru \,|\, u \in \boldsymbol{X}\}$ for R en rotation om origo.

Hvis en punktproces er stationær eller isotropisk (eller begge), så er intensitetsmålet μ og intensitetsfunktionen ρ også invariant under parallelforskydning og/eller rotation. Det skyldes, at fordelingen er invariant pr. definition, og idet den er baseret på tællefunktionen N jf. definition 1.5, så nedarves disse egenskaber, da μ og derigennem ρ er defineret på baggrund af tællefunktionen.

Det kan i visse tilfælde være gavnligt at have mulighed for at sammensætte eller fjerne dele af en punktproces, og sådanne operationer kan defineres som følger.

Definition 1.10 (Superpositionering):
En disjunkt forening $\bigcup_{i=1}^{\infty} \boldsymbol{X}_i$ af punktprocesser kaldes en *superposition*.

Definition 1.11 (Uafhængig udtynding):
Lad $\boldsymbol{X}$ være en punktproces på S, og lad $p : S \to [0,1]$ være en funktion. Punktprocessen $\boldsymbol{X}_{\text{thin}} \subseteq \boldsymbol{X}$, som fås ved at beholde hvert $u \in \boldsymbol{X}$ i $\boldsymbol{X}_{\text{thin}}$ med sandsynlighed $p(u)$, kaldes en *uafhængig udtynding* af $\boldsymbol{X}$ med *bevaringssandsynlighed* $p(u)$, $u \in S$. Hvert punkt bevares eller udelades uafhængigt af øvrige punkter.

En praktisk måde at specificere $\boldsymbol{X}_{\text{thin}}$ på er $\boldsymbol{X}_{\text{thin}} := \{u \in \boldsymbol{X} \,|\, R(u) \leq p(u)\}$, hvor $R(u) \sim \text{Unif}[0,1]$ for alle $u \in S$ og uafhængig af $\boldsymbol{X}$, og det er typisk denne måde, der bliver anvendt ved implementation, og det benyttes også i visse beviser.

Endnu en måde at karakterisere punktprocesser på er ved at betragte de mængder, som ikke indeholder nogen punkter.

Definition 1.12 (Void-sandsynligheder og hændelser):
Void-sandsynlighederne v er sandsynligheden for, at der ikke er nogle punkter i en begrænset mængde $B \subseteq S$, dvs. $v(B) := P(N(B) = 0)$ for B begrænset.

Hændelserne $F_B := \{\boldsymbol{x} \in N_{\text{lf}} | n(\boldsymbol{x}_B) = 0\}$ kaldes *void-hændelser*. Bemærk at $\boldsymbol{X} \in F_B$ hvis og kun hvis $N(B) = 0$.

Void-sandsynlighederne kan anvendes til entydigt at bestemme en punktproces, som det fremgår af den følgende sætning, der angives uden bevis.

Sætning 1.13 (Entydighed ud fra void-sandsynligheder):
En punktproces er entydigt bestemt ud fra sine void-sandsynligheder.

Det kan undertiden være ønskeligt at tilknytte ekstra information til punkterne i et punktmønster. Denne ekstra information kaldes *mærker*, og punktprocesser af denne type kaldes *mærkede punktprocesser*.

Definition 1.14 (Mærket punktproces):
Lad $\boldsymbol{Y}$ være en punktproces på $T \subseteq \mathbb{R}^d$. Givet tilfældige *mærker* $\{m_u | u \in \boldsymbol{Y}\}$ fra en *mærkerum* M, da siges $\boldsymbol{X} = \{(u, m_u) | u \in \boldsymbol{Y}\}$ at være en *mærket punktproces* med punkter i T og mærker i M.

Hvis mærkerne er i.i.d. med fordeling Q, så kaldes Q for *mærkefordelingen*.

En mærket punktproces kan eksempelvis bruges til både at indeholde placeringen af træer samt deres størrelse, eller den kan anvendes til at skelne imellem forskellige typer af punkter. Det er også muligt at have flerdimensionale mærker, således vil den opstillede model i kapitel 2 indeholde en mærket punktproces med todimensionale mærker, som angiver henholdsvis en vinkel og en længde.

1.1.3 Standardbeviset

I visse beviser i de efterfølgende afsnit er der brug for at fastslå ligheder af typen

$$\mathbf{E}\big[f(\boldsymbol{X})\big] \;=\; H(f) \tag{1.1}$$

for et $H : f \to \mathbb{R}$, hvor H er enten et integrale, en sum, en middelværdi eller en kombination af disse. Ved at antage, at f kan skrives som en grænseværdi af

summen af skalerede indikatorfunktioner, så er det tilstrækkeligt at vise, at (1.1) er sand for indikatorfunktioner, dvs. at $\mathbf{E}[\mathbf{1}[\boldsymbol{X} \in F]] = P(\boldsymbol{X} \in F) = H(\mathbf{1}_F)$ for en mængde F. Dette er kendt som *standardbeviset*. Teknisk set skal det vises, at de tre følgende punkter er opfyldte for funktionen H, for at fastslå ligheden i (1.1).

(i) $\mathbf{E}[\mathbf{1}[\boldsymbol{X} \in F]] = P(\boldsymbol{X} \in F) = H(\mathbf{1}_F)$ som allerede omtalt.

(ii) $\mathbf{E}\left[\sum_{i=1}^{n}\left[a_i \mathbf{1}[\boldsymbol{X} \in F_i]\right]\right] = \sum_{i=1}^{n}\left[a_i \mathbf{E}[\mathbf{1}[\boldsymbol{X} \in F_i]]\right] = \sum_{i=1}^{n}\left[a_i H\left(\mathbf{1}_{F_i}\right)\right]$
$= H\left(\sum_{i=1}^{n}\left[a_i \mathbf{1}_{F_i}\right]\right) = H(f)$. Her er a_i'erne konstanter, og f er en sum af skalerede indikatorfunktioner. Ligheden følger af lineariteten af H og $\mathbf{E}[\cdot]$.

(iii) $\mathbf{E}\left[\lim_{n\to\infty} f_n(\boldsymbol{X})\right] = \lim_{n\to\infty} \mathbf{E}\left[f_n(\boldsymbol{X})\right] = \lim_{n\to\infty} H(f_n) = H\left(\lim_{n\to\infty} f_n\right) = H(f)$.
I dette tilfælde er f grænseværdien af en følge af funktioner. Ligheden følger af *Lebesgues monotone konvergenssætning*, [Rudin, 1987, sæt. 1.26].

Hvis funktionen H opfylder disse tre punkter, så medfører det ligheden af $\mathbf{E}\left[f(\boldsymbol{X})\right] = H(f)$. Styrken ved standardbeviset er, at hvis blot ligheden i **(i)** kan vises for en given funktion H, så følger punkt **(ii)** og **(iii)** automatisk.

1.2 Deskriptive størrelser

Der findes en række deskriptive størrelser, som kan anvendes til at kvantificere forskellige egenskaber ved punktprocesser, og de kan eksempelvis indikere, om punkterne i et punktmønster udviser tiltrækning eller frastødning af hinanden. Yderligere kan det også være relevant at undersøge variationen i punktmønstre genereret af samme punktproces.

1.2.1 Andenordens-egenskaber

En udvidelse af begreberne om intensitet findes i følgende definition.

Definition 1.15 (Faktorielt momentmål og produkttæthed):
Det *andenordens faktorielle momentmål* $\alpha^{(2)}$ på $\mathbb{R}^d \times \mathbb{R}^d$ er givet ved

$$\alpha^{(2)}(A \times B) = \mathbf{E}\left[\sum_{u,v \in \boldsymbol{X}}^{\neq}\left[\mathbf{1}[u \in A, v \in B]\right]\right], \text{ hvor } A \times B \subseteq \mathbb{R}^d \times \mathbb{R}^d.$$

Hvis $\alpha^{(2)}$ kan skrives som

$$\alpha^{(2)}(A \times B) \;=\; \int_A \int_B \rho^{(2)}(u,v)\,\mathrm{d}u\mathrm{d}v, \text{ hvor } \rho^{(2)} \geq 0 \text{ og } A \times B \subseteq \mathbb{R}^d \times \mathbb{R}^d,$$

så kaldes $\rho^{(2)}$ for *andenordens produkttætheden.*

Heuristisk er $\rho^{(2)}(u,v)\,\mathrm{d}u\mathrm{d}v$ sandsynligheden for simultant at observere et par af punkter i to infinitesimalt små kugler med centrum i henholdsvis u og v og med volumener $\mathrm{d}u$ og $\mathrm{d}v$. I den følgende sætning introduceres *Campbells formler*, der er et nyttigt resultat i adskillige efterfølgende beviser.

Sætning 1.16 (Campbells formler):
Antag at $\boldsymbol{X}$ har intensitet ρ og andenordens produkttæthed $\rho^{(2)}$. Så gælder for funktioner $h_1 : \mathbb{R}^d \to [0,\infty)$ og $h_2 : \mathbb{R}^d \times \mathbb{R}^d \to [0,\infty)$, at

$$\mathbf{E}\left[\sum_{u \in \boldsymbol{X}} \Big[h_1(u)\Big]\right] \;=\; \int h_1(u)\rho(u)\,\mathrm{d}u \qquad \text{samt at}$$

$$\mathbf{E}\left[\sum_{u,v \in \boldsymbol{X}}^{\neq} \Big[h_2(u,v)\Big]\right] \;=\; \iint h_2(u,v)\rho^{(2)}(u,v)\,\mathrm{d}u\mathrm{d}v.$$

Bevis:
Fra definition 1.7 fås, at

$$\mathbf{E}\left[\sum_{u \in \boldsymbol{X}} \Big[\mathbf{1}[u \in B]\Big]\right] \;=\; \mu(B) \;=\; \int \mathbf{1}[u \in B]\rho(u)\,\mathrm{d}u,$$

og fra definition 1.15 fås, at

$$\mathbf{E}\left[\sum_{u,v \in \boldsymbol{X}}^{\neq} \Big[\mathbf{1}\,[u \in A, v \in B]\,\Big]\right] \;=\; \alpha^{(2)}(A \times B) \;=\; \iint \mathbf{1}\,[u \in A, v \in B]\,\rho^{(2)}(u,v)\,\mathrm{d}u\mathrm{d}v.$$

Heraf ses det, at formlerne gælder for indikatorfunktioner, og det følger derfor af standardbeviset, at de gælder for funktionerne h_1 og h_2. ■

Den følgende sætning viser, hvorledes ρ og $\rho^{(2)}$ for en punktproces $\boldsymbol{X}$ påvirkes ved udførsel af uafhængig udtynding af $\boldsymbol{X}$.

Sætning 1.17 (ρ og $\rho^{(2)}$ under uafhængig udtynding):
Antag at $\boldsymbol{X}$ har intensitet ρ og andenordens produkttæthed $\rho^{(2)}$. Antag yderligere at $\boldsymbol{X}_{\text{thin}}$ med intensitet ρ_{thin} og andenordens produkttæthed $\rho^{(2)}_{\text{thin}}$ er en uafhængig udtynding af $\boldsymbol{X}$ med bevaringssandsynlighed $p(u)$. Så er

$$\rho_{\text{thin}}(u) = p(u)\rho(u) \qquad \text{og} \qquad \rho^{(2)}_{\text{thin}}(u,v) = p(u)p(v)\rho^{(2)}(u,v).$$

Bevis:
Ved at anvende definition 1.6 for $\boldsymbol{X}_{\text{thin}}$ og betinge på $\boldsymbol{X}$ fås for $R(u) \sim \text{Unif}(0,1)$, at

$$\begin{aligned}
\mu_{\text{thin}}(A) &= \mathbf{E}\left[\sum_{u\in\boldsymbol{X}_{\text{thin}}}\Big[\mathbf{1}[u\in A]\Big]\right] = \mathbf{E}\left[\mathbf{E}\left[\sum_{u\in\boldsymbol{X}}\Big[\mathbf{1}[u\in A, R(u)\le p(u)]\Big]\Big|\boldsymbol{X}\right]\right] \\
&= \mathbf{E}\left[\sum_{u\in\boldsymbol{X}}\Big[\mathbf{1}\,[u\in A]\Big]\mathbf{E}\Big[\mathbf{1}\,\big[R(u)\le p(u)\big]\Big|\boldsymbol{X}\Big]\right] = \mathbf{E}\left[\sum_{u\in\boldsymbol{X}}\Big[\mathbf{1}[u\in A]p(u)\Big]\right] \\
&= \int_A p(u)\rho(u)\,\mathrm{d}u.
\end{aligned}$$

Dermed er $\rho_{\text{thin}}(u) = p(u)\rho(u)$, og det følger på tilsvarende vis, at

$$\begin{aligned}
\alpha^{(2)}_{\text{thin}}(A\times B) &= \mathbf{E}\left[\sum_{u,v\in\boldsymbol{X}_{\text{thin}}}^{\neq}\Big[\mathbf{1}[(u,v)\in A\times B]\Big]\right] \\
&= \mathbf{E}\left[\sum_{u,v\in\boldsymbol{X}}^{\neq}\Big[p(u)p(v)\,\mathbf{1}[(u,v)\in A\times B]\Big]\right] \\
&= \int_A\int_B p(u)p(v)\rho^{(2)}(u,v)\,\mathrm{d}u\mathrm{d}v.
\end{aligned}$$

■

1.2.2 Parkorrelationsfunktionen g og $\mathcal{K}$-målet

Ved at normalisere andenordens produkttætheden fås den såkaldte *parkorrelationsfunktion* g, der er en af de deskriptive størrelser, der indikerer tiltrækning eller frastødning af punkterne i et punktmønster. Dette forhold præciseres nærmere i afsnit 1.3.4 i forbindelse med udregning af g for Poisson punktprocesser.

Definition 1.18 (Parkorrelationsfunktionen):
Hvis både ρ og $\rho^{(2)}$ eksisterer, så er *parkorrelationsfunktionen* givet ved

$$g(u,v) = \frac{\rho^{(2)}(u,v)}{\rho(u)\rho(v)}, \quad \text{hvor } g(u,v) := 0 \text{ for } \rho(u)\rho(v) = 0.$$

Det følger direkte fra sætning 1.17 og definition 1.18, at $g_{\text{thin}} = g$, så g er invariant under uafhængig udtynding. Bemærk at når $\boldsymbol{X}$ er stationær, så er g også invariant under parallelforskydning, $g(u,v) = g(u-v)$, og da andenordens produkttætheden er symmetrisk, så er $g(u,v) = g(v-u) = g(u-v)$ for $\boldsymbol{X}$ stationær. Denne egenskab vil blive anvendt i flere efterfølgende beviser.

Endnu et mål $\mathcal{K}(B)$ vil nu blive defineret. Det har i visse tilfælde en tæt sammenhæng med g, og det er grundlaget for flere af de andenordens deskriptive størrelser, der indføres i næste afsnit.

Definition 1.19 (Andenordens reduceret momentmål):
Antag at $\boldsymbol{X}$ har intensitet ρ, at $0 < |A| < \infty$, og at målet

$$\mathcal{K}(B) = \frac{1}{|A|}\mathbf{E}\left[\sum_{u,v\in\boldsymbol{X}}^{\neq}\left[\frac{\mathbf{1}[u\in A, v-u\in B]}{\rho(u)\rho(v)}\right]\right], \qquad A,B\subseteq\mathbb{R}^d$$

ikke afhænger af valget A, hvor led i summen sættes lig 0 for $\rho(u)\rho(v) = 0$. Så kaldes $\mathcal{K}$ for et *andenordens reduceret momentmål*, og $\boldsymbol{X}$ siges at være *andenordens intensitetsgenvægtet stationær*.

Bemærk at stationaritet af $\boldsymbol{X}$ medfører andenordens intensitetsgenvægtet stationaritet. Sammenhængen mellem g og $\mathcal{K}(B)$ fremgår af følgende resultat.

Sætning 1.20 (Sammenhængen mellem $\mathcal{K}$ og g):
Antag at g eksisterer og er invariant under parallelforskydning. Så er $\boldsymbol{X}$ andenordens intensitetsgenvægtet stationær, og

$$\mathcal{K}(B) = \int_B g(u)\,\mathrm{d}u, \qquad B\subseteq\mathbb{R}^d.$$

Bevis:
Ved anvendelse af henholdsvis definition 1.19, Campbells formler, invariansen af g under parallelforskydning samt transformationen $w = v - u$ fås, at

$$\begin{aligned}
\mathcal{K}(B) &= \frac{1}{|A|}\mathbf{E}\left[\sum_{u,v\in\boldsymbol{X}}^{\neq}\left[\frac{\mathbf{1}[u\in A, v-u\in B]}{\rho(u)\rho(v)}\right]\right] \\
&= \frac{1}{|A|}\iint \frac{\mathbf{1}[u\in A, v-u\in B]}{\rho(u)\rho(v)}\rho^{(2)}(u,v)\,\mathrm{d}u\mathrm{d}v \\
&= \frac{1}{|A|}\iint \mathbf{1}[u\in A, v-u\in B]g(v-u)\,\mathrm{d}u\mathrm{d}v \\
&= \int \mathbf{1}[w\in B]g(w)\,\mathrm{d}w \\
&= \int_B g(w)\,\mathrm{d}w.
\end{aligned}$$

■

1.2.3 Andenordens deskriptive størrelser

$\mathcal{K}$-målet anvendes oftest for en klasse af mængder såsom kugler, hvilket er grunden til følgende andenordens deskriptive størrelser.

Definition 1.21 (*K*- og *L*-funktionerne):
For andenordens intensitetsgenvægtede stationære punktprocesser er

$$K(r) = \mathcal{K}(b(0,r)) \qquad \text{og} \qquad L(r) = (K(r)/\omega_d)^{1/d},$$

hvor $r > 0$, $b(0,r)$ er en d-dimensional kugle med radius r, og ω_d er volumen af den d-dimensionale enhedskugle $b(0,1)$, dvs. $\omega_d = \pi^{d/2}/\Gamma(1+d/2)$.

For det stationære tilfælde angiver $\rho K(r)$ det forventede antal af yderligere punkter indenfor afstand r af origo givet at $\boldsymbol{X}$ har et punkt i origo. I praksis anvendes oftest L-funktionen, da den er nemmere at fortolke på et plot, se afsnit 1.3.4 for en forklaring herpå.

I sætning 1.20 blev sammenhængen mellem $\mathcal{K}$ og g fastslået, og det fremgår af det næste resultat, hvad sammenhængen er mellem K og g i de tilfælde, hvor g er isotropisk. Bemærk at i så fald er $g(u,v) = g(\|u-v\|) = g(r)$.

Sætning 1.22 (Sammenhængen mellem K og g):
Antag at g er isotropisk. Så er K-funktionen givet ved

$$K(r) = \sigma_d \int_0^r s^{d-1} g(s)\,\mathrm{d}s,$$

hvor r er radius af en d-dimensional kugle, og σ_d er overfladearealet af den d-dimensionale enhedskugle, dvs. $\sigma_d = 2\pi^{d/2}/\Gamma(d/2)$.

Inden beviset er det nyttigt at notere, at sfærisk integration af en funktion $f(r)$, som kun afhænger af radius r, er defineret ved $\sigma_d \int_0^r s^{d-1} f(s)\,\mathrm{d}s$.

Bevis:
Ud fra definition 1.21, Campbells formler, definition 1.18, transformationen $s = v - u$ og endeligt er skift til sfæriske koordinater fås, at

$$\begin{aligned}
K(r) &= \frac{1}{|A|}\mathbf{E}\left[\sum_{u,v\in\boldsymbol{X}}^{\neq}\left[\frac{\mathbf{1}[u\in A, \|v-u\|\le r]}{\rho(u)\rho(v)}\right]\right] \\
&= \frac{1}{|A|}\iint \frac{\mathbf{1}[u\in A, \|v-u\|\le r]}{\rho(u)\rho(v)}\,\rho^{(2)}(u,v)\,\mathrm{d}u\mathrm{d}v \\
&= \frac{1}{|A|}\iint \mathbf{1}[u\in A, \|v-u\|\le r]\, g(\|v-u\|)\,\mathrm{d}u\mathrm{d}v \\
&= \frac{1}{|A|}\iint \mathbf{1}[u\in A, \|s\|\le r]\, g(s)\,\mathrm{d}u\mathrm{d}s \\
&= \int \mathbf{1}[\|s\|\le r]\, g(s)\,\mathrm{d}s \\
&= \sigma_d \int_0^r s^{d-1} g(s)\,\mathrm{d}s,
\end{aligned}$$

■

1.2.4 Deskriptive størrelser baseret på afstanden mellem punkter

En anden type deskriptive størrelser er de følgende, der er baseret på afstanden mellem punkter for stationære punktprocesser.

Definition 1.23 (*F*-, *G*- og *J*-funktionerne):
Antag at $\boldsymbol{X}$ er en stationær punktproces. *Tomrumsfunktionen* F er fordelingsfunktionen for afstanden mellem $a \in \mathbb{R}^d$ og det nærmeste punkt i $\boldsymbol{X}$, dvs.

$$F(r) \;=\; P\left(\boldsymbol{X} \cap b\left(a,r\right) \neq \emptyset\right) \;=\; \mathbf{E}\left[\mathbf{1}\left[\boldsymbol{X} \cap b\left(a,r\right) \neq \emptyset\right]\right], \qquad r > 0.$$

Nærmeste nabo-funktionen G er fordelingsfunktionen for afstanden fra $u \in \boldsymbol{X}_A$ til det nærmeste nabopunkt i $\boldsymbol{X}$ for et arbitrært $A \subseteq \mathbb{R}^d$, $0 < |A| < \infty$, så

$$G(r) \;=\; P(\boldsymbol{X}\backslash\{u\} \cap b(u,r) \neq \emptyset) \;=\; \frac{1}{\rho|A|}\mathbf{E}\left[\sum_{u \in \boldsymbol{X}_A}\left[\mathbf{1}\left[(\boldsymbol{X}\backslash\{u\}) \cap b(u,r) \neq \emptyset\right]\right]\right], r > 0.$$

J-funktionen er defineret ud fra F og G til at være

$$J(r) \;=\; \frac{1-G(r)}{1-F(r)}, \qquad F(r) < 1.$$

Der kendes kun udtryk for F-, G- og J-funktionerne på lukket form for enkelte punktprocesser, herunder Poisson punktprocesser, hvilket vises i afsnit 1.3.4.

1.2.5 Ikke-parametrisk estimation af deskriptive størrelser

For alle de indførte deskriptive størrelser findes der nogle ikke-parametriske estimater, hvoraf enkelte vil blive omtalt i dette afsnit. Estimaterne er baseret på et konkret punktmønster $\boldsymbol{X}_W = \boldsymbol{x}$, som er observeret i et begrænset observationsvindue $W \subset \mathbb{R}^d$, $|W| > 0$, og det antages, at de forskellige deskriptive størrelser eksisterer i det omfang, det er nødvendigt.

Som et første estimat vises, at det oplagte bud på et estimat af ρ er centralt.

Sætning 1.24 (Centralt estimat af ρ):
I det homogene tilfælde er et centralt estimat for intensiteten ρ givet ved

$$\widehat{\rho} \;=\; \frac{n(\boldsymbol{x})}{|W|}.$$

Bevis:

$$\mathbf{E}\left[\widehat{\rho}\right] \;=\; \mathbf{E}\left[\frac{n(\boldsymbol{x})}{|W|}\right] \;=\; \frac{\mu(W)}{|W|} \;=\; \frac{1}{|W|}\int_W \rho\,\mathrm{d}u \;=\; \frac{\rho}{|W|}\int_W 1\,\mathrm{d}u \;=\; \rho.$$

■

I det inhomogene tilfælde er et ikke-parametrisk *kerneestimat* af intensiteten

$$\widehat{\rho}_b(u) \;=\; \sum_{v\in\boldsymbol{x}} \frac{k_b(u-v)}{h_{W,b}(v)}, \qquad u\in W. \tag{1.2}$$

Her er k en tæthedsfunktion, og k_b er en kerne med bredde $b>0$, således at $k_b(u)=k(u/b)/b^d$, og $h_{W,b}(v)$ er en kantkorrektionsfaktor givet ved $h_{W,b}(v)=\int_W k_b(u-v)\,\mathrm{d}u$. På baggrund heraf kan et estimat for intensitetsmålet opstilles.

Sætning 1.25 (Centralt estimat af μ):
Et centralt estimat af $\mu(W)$ er givet ved $\widehat{\mu}(W) \;=\; \displaystyle\int_W \widehat{\rho}_b(u)\,\mathrm{d}u$.

Bevis:
Fra definition 1.6, Campbells formler og (1.2) fås det, at

$$\begin{aligned}
\mathbf{E}\left[\widehat{\mu}(W)\right] \;&=\; \mathbf{E}\left[\int_W \widehat{\rho}_b(u)\,\mathrm{d}u\right] \\
&=\; \int_W \mathbf{E}\left[\widehat{\rho}_b(u)\right]\mathrm{d}u \;=\; \int_W \mathbf{E}\left[\sum_{v\in\boldsymbol{X}_W}\left[\frac{k_b(u-v)}{h_{W,b}(v)}\right]\right]\mathrm{d}u \\
&=\; \int_W\left(\int_W \frac{k_b(u-v)}{h_{W,b}(v)}\mathrm{d}u\right)\rho(v)\mathrm{d}v \;=\; \int_W \rho(v)\mathrm{d}v \;=\; \mu(W),
\end{aligned}$$

■

Sætning 1.26:
Antag at $|W\cap W_u|>0$ for alle $u\in B$, hvor $W_u=\left\{\,v+u\,|\,v\in W\,\right\}$ er parallelforskydningen af W med u, og at $\boldsymbol{X}$ er andenordens intensitetsgenvægtet stationær. Da er et centralt estimat af $\mathcal{K}$

$$\widehat{\mathcal{K}}(B) \;=\; \sum_{u,v\in\boldsymbol{x}}^{\neq}\left[\frac{\mathbf{1}[v-u\in B]}{\rho(u)\rho(v)|W\cap W_{v-u}|}\right].$$

Bevis:
Ved anvendelse af Campbells formler, definition 1.18, transformationen $w = v - u$, at $|W \cap W_w| = |W \cap W_{-w}|$ samt sætning 1.20 følger det, at

$$\begin{aligned}
\mathbf{E}\left[\widehat{\mathcal{K}}(B)\right] &= \mathbf{E}\left[\sum_{u,v\in \boldsymbol{X}_W}^{\neq}\left[\frac{\mathbf{1}[v-u\in B]}{\rho(u)\rho(v)|W\cap W_{v-u}|}\right]\right] \\
&= \iint \frac{\mathbf{1}[v-u\in B]}{|W\cap W_{v-u}|}\mathbf{1}[u\in W, v\in W]g(v-u)\,\mathrm{d}u\mathrm{d}v \\
&= \iint \frac{\mathbf{1}[u\in W, u\in W_{-w}]}{|W\cap W_w|}\mathbf{1}[w\in B]\,g(w)\,\mathrm{d}u\mathrm{d}w \\
&= \int \mathbf{1}[w\in B]\,g(w)\mathrm{d}w \\
&= \int_B g(w)\,\mathrm{d}w \;=\; \mathcal{K}(B).
\end{aligned}$$

■

Bemærk at estimatet kun er centralt, hvis $\rho(u)\rho(v)$ kendes. Hvis det er nødvendigt at anvende $\widehat{\rho}$ eller $\widehat{\rho}_b(u)$, så bliver $\widehat{\mathcal{K}}(B)$ ikke-centralt.

1.2.6 Kanteffekter og kantkorrektion

Når der fokuseres på et begrænset observationsvindue W, så kan der være en vekselvirkning mellem observerede punkter i nærheden af kanten af W og uobserverede punkter udenfor W. Dette fænomen kaldes *kanteffekter*, og der findes forskellige former for *kantkorrektioner* til at korrigere for dette problem for at modvirke bias. Ovenfor er anvendt de to kantkorrektioner $h_{W,b}(v)$ og $1/|W \cap W_{v-u}|$, der viser sig at kunne bruges til at frembringe centrale estimater.

En anden type kantkorrektion er *minus sampling*, hvor der for $r > 0$ benyttes

$$W_{\ominus r} \;=\; \left\{u \in W \mid b(u,r) \subseteq W\right\},$$

dvs. der anvendes de punkter i W, der har en afstand større end r til kanten af W. Det er bl.a. muligt at bestemme estimater for F-, G- og J-funktionerne ved brug af minus sampling. Andre kantkorrektioner og estimater af de forskellige deskriptive størrelser findes, men de vil ikke blive yderligere gennemgået her, en del vil dog blive anvendt til modelkontrol i afsnit 4.3 vha. `R`-pakken `spatstat`, der har en lang række af estimater og kantkorrektioner indbygget.

1.3 Poisson punktprocesser

En fundamental type punktproces er den såkaldte Poisson punktproces (PPP), der besidder en række interessante egenskaber, der vil blive belyst i dette afsnit. Inden den formelle indførsel af PPP'er er det dog nødvendigt først at definere den binomiale punktproces.

Definition 1.27 (Binomial punktproces):
Lad f være en tæthed på en mængde $S \subseteq \mathbb{R}^d$, og lad $n \in \mathbb{N}$. En punktproces $\boldsymbol{X}$ bestående af n i.i.d. punkter med tæthed f kaldes en *binomial punktproces* med n punkter i S med tæthed f. Dette skrives $\boldsymbol{X} \sim \text{binomial}\big(S,n,f\big)$.

Dette kaldes en binomial punktproces, da antallet af punkter $N(B)$ for $B \subseteq S$ er binomialfordelt. Sandsynligheden for, at et punkt fra en binomial punktproces er i $B \subseteq S$, er $p = \int_B f(x)\,\mathrm{d}x \left(\int_S f(x)\,\mathrm{d}x = 1\right)$, og da punkterne er i.i.d., så haves alle kravene for en binomial stokastisk variabel, dvs. $N(B) \sim \text{Bin}(n,p)$.

Definition 1.28 (Poisson punktproces):
$\boldsymbol{X}$ er en PPP med intensitet $\rho(u)$ og endeligt intensitetsmål $\mu < \infty$, hvis der for enhvert begrænset område $B \subseteq S$ med $\mu(B) > 0$ gælder følgende:

(i) $N(B) \sim \text{Poi}\big(\mu(B)\big)$, hvor $\mu(B) = 0$ betyder, at $N(B) = 0$.
(ii) Betinget på $N(B) = n$ for $n \in \mathbb{N}$, $\boldsymbol{X}_B \sim \text{binomial}\big(B,n,f\big)$ med $f \propto \rho(u)$.

Dette skrives $\boldsymbol{X} \sim \text{Poisson}(S,\rho)$.

Da middelværdien af en Poissonfordeling $\text{Poi}\big(\mu(B)\big)$ er $\mu(B)$, så er $\mathbf{E}[N(B)] = \mu(B)$, og i overensstemmelse med definition 1.7 er det derfor naturligt at benytte begreberne om intensitet for en PPP. Jf. definition 1.8 siges en PPP at være homogen eller inhomogen, men en særlig homogen PPP benyttes ofte.

Definition 1.29 (Standard Poisson punktprocessen):
$\boldsymbol{X} \sim \text{Poisson}\,(S,1)$ kaldes *standard Poisson punktprocessen.*

Når $\boldsymbol{X}$ er defineret som en homogen PPP, så er sandsynligheden $P(u \in \boldsymbol{X})$ for et arbitrært punkt $u \in S$ den samme for ethvert område $B_i \subseteq S$, når alle B_i'erne har

samme størrelse. Derfor vil en parallelforskydning eller rotation af $\boldsymbol{X}$ ikke ændre fordelingen af $\boldsymbol{X}$, så en homogen PPP på $\mathbb{R}^d$ er både stationær og isotropisk.

1.3.1 Eksistens og entydighed af Poisson punktprocesser

Der er en række trin, der skal gennemføres, for at vise eksistens og entydighed.

Trin 1 – Poisson-udvikling og eksistens af endelige PPP'er

Det første skridt er at vise den såkaldte *Poisson-udvikling*.

Sætning 1.30 (Poisson-udvikling):

(i) $\boldsymbol{X} \sim \text{Poisson}(S,\rho) \Leftrightarrow$ alle $B \subseteq S$ med $\mu(B) < \infty$ og alle $F \subseteq N_{\text{lf}}$ opfylder

$$P(\boldsymbol{X}_B \in F) = \sum_{n=0}^{\infty} \left[\frac{e^{(-\mu(B))}}{n!} \int_B \cdots \int_B \mathbf{1}\left[\{x_1,\ldots,x_n\} \in F\right] \prod_{i=1}^{n} \left[\rho(x_i)\right] \mathrm{d}x_1 \ldots \mathrm{d}x_n \right], \tag{1.3}$$

hvor integranden for $n = 0$ er $\mathbf{1}[\emptyset \in F]$.

(ii) Hvis $\boldsymbol{X} \sim \text{Poisson}\big(S,\rho\big)$, gælder for funktioner $h : N_{\text{lf}} \to \mathbb{R}$ for $B \subseteq S$ med $\mu(B) < \infty$, at

$$\mathbf{E}\left[h(\boldsymbol{X}_B)\right] = \sum_{n=0}^{\infty} \left[\frac{\exp\big(-\mu(B)\big)}{n!} \int_B \cdots \int_B h(\{x_1,\ldots,x_n\}) \prod_{i=1}^{n} \left[\rho(x_i)\right] \mathrm{d}x_1 \ldots \mathrm{d}x_n \right].$$

Bevis:

(i) Antag at $\boldsymbol{X} \sim \text{Poisson}\big(S,\rho\big)$. Ved at betinge på $N(B)$ og anvende den diskrete version af loven om total sandsynlighed fås det, at

$$P(\boldsymbol{X}_B \in F) \;=\; \sum_{n=0}^{\infty} \left[P(N(B) = n)\, P\big(\boldsymbol{X}_B \in F \,|\, N(B) = n\big) \right].$$

Da $N(B) \sim \text{Poi}\big(\mu(B)\big)$ pr. definition 1.28 (i), så kan dette skrives som

$$\sum_{n=0}^{\infty} \left[\exp\big(-\mu(B)\big) \frac{\mu(B)^n}{n!} \mathbf{E}\left[\mathbf{1}\left[\boldsymbol{X}_B \in F \,|\, N(B) = n\right]\right] \right]. \tag{1.4}$$

Middelværdien i (1.4) kan skrives som multiple integraler over B, og indikatorfunktionen forbliver, da F kan specificeres arbitrært. Da de n punkter er binomi-

alfordelt med tæthed $f \propto \rho$ pr. definition 1.28 (ii), så er normaliseringskonstanten $\int_B \rho(u)\,\mathrm{d}u = \mu(B)$. Herved kan (1.4) skrives som

$$\begin{aligned}
&\sum_{n=0}^{\infty}\left[\exp\left(-\mu(B)\right)\frac{\mu(B)^n}{n!}\int_B\cdots\int_B \mathbf{1}\left[\{x_1,\ldots,x_n\}\in F\right]\prod_{i=1}^{n}\left[\frac{\rho(x_i)}{\mu(B)}\right]\mathrm{d}x_1\ldots\mathrm{d}x_n\right]\\
=\ &\sum_{n=0}^{\infty}\left[\frac{\exp\left(-\mu(B)\right)}{n!}\int_B\cdots\int_B \mathbf{1}\left[\{x_1,\ldots,x_n\}\in F\right]\prod_{i=1}^{n}\left[\rho(x_i)\right]\mathrm{d}x_1\ldots\mathrm{d}x_n\right],
\end{aligned}$$

hvilket svarer til (1.3), hvorved *kun hvis*-delen af **(i)** er vist. For at vise *hvis*-delen antages, at (1.3) er sand. For at vise opfyldelse af den første betingelse i definition 1.28 defineres $F := \left\{\boldsymbol{x}\in N_{\mathrm{lf}} \middle| N(B)=k\right\}$, og (1.3) bruges til at beregne sandsynligheden for, at $\boldsymbol{X}_B$ indeholder præcist k punkter, hvis der er n punkter i S, dvs.

$$P(N(B)=k) = \sum_{n=0}^{\infty}\left[\frac{\exp(-\mu(S))}{n!}\int_S\cdots\int_S \mathbf{1}\left[N(B)=k\right]\prod_{i=1}^{n}\left[\rho(x_i)\right]\mathrm{d}x_1\ldots\mathrm{d}x_n\right]. \quad (1.5)$$

De første k led i summen er 0, da indikatorfunktionen ikke kan være sand, når der er færre end k punkter, hvorved indekset kan ændres til $n=k$. Ved at betragte situationen som en binomial punktproces, enten er $x_i\in B$, eller også er $x_i\in S\setminus B$, så kan (1.5) benyttes til at omskrive sandsynligheden til

$$\begin{aligned}
P(N(B)=k) &= \sum_{n=k}^{\infty}\left[\frac{\exp\left(-\mu(S)\right)}{n!}\binom{n}{k}\int_{B^k}\int_{(S\setminus B)^{n-k}}\prod_{i=1}^{n}\left[\rho(x_i)\right]\mathrm{d}x_1\ldots\mathrm{d}x_n\right]\\
&= \sum_{n=k}^{\infty}\left[\frac{\exp\left(-\mu(S)\right)}{n!}\frac{n!}{k!(n-k!)}\left(\int_B\rho(x)\,\mathrm{d}x\right)^k\left(\int_{(S\setminus B)}\rho(x)\,\mathrm{d}x\right)^{n-k}\right]\\
&= \sum_{n=k}^{\infty}\left[\frac{\exp\left(-\mu(S)\right)}{k!\,(n-k)!}\mu(B)^k\,\mu(S\setminus B)^{n-k}\right]. \quad (1.6)
\end{aligned}$$

Pr. definition er $\mu(S)=\mu(S\setminus B)+\mu(B)$ for $B\subseteq S$, så (1.6) kan omskrives til

$$\begin{aligned}
&\exp\left(-\mu(B)\right)\frac{\mu(B)^k}{k!}\sum_{n=k}^{\infty}\left[\frac{\exp\left(-\mu(S\setminus B)\right)}{(n-k)!}\mu(S\setminus B)^{n-k}\right]\\
=\ &\exp\left(-\mu(B)\right)\tfrac{\mu(B)^k}{k!}\sum_{m=0}^{\infty}\left[\tfrac{\exp(-\mu(S\setminus B))}{m!}\mu(S\setminus B)^m\right] = \exp\left(-\mu(B)\right)\tfrac{\mu(B)^k}{k!} \quad (1.7)
\end{aligned}$$

med $m := n-k$, hvorved det ses, at summen i (1.7) er lig 1, da leddene er en Poissontæthed. Resultatet er også en Poissontæthed, så $N(B)\sim \mathrm{Poi}\left(\mu(B)\right)$ som påkrævet. For at vise, at definition 1.28 (ii) også er opfyldt, vil det blive vist, at

betinget på $N(B) = k$, så er antallet at punkter $N(A) = m$ med $A \subseteq B$ og $m \leq k$ binomialfordelt. For at gøre dette er der brug for et udtryk for den simultane fordeling, og tilsvarende til ovenfor fås, at

$$P\left(N(A) = m, N(B) = k\right) = \exp\left(-\mu(A)\right) \frac{\mu(A)^m}{m!} \exp\left(-\mu(B \setminus A)\right) \frac{\mu(B \setminus A)^{k-m}}{(k-m)!}. \quad (1.8)$$

Ved brug af betinget sandsynlighed samt (1.7) og (1.8) fås nu, at

$$\begin{aligned}
P\left(N(A) = m \,|\, N(B) = k\right) &= \frac{P\left(N(A) = m, N(B) = k\right)}{P\left(N(B) = k\right)} \\
&= \frac{\exp\left(-\mu(A)\right) \frac{\mu(A)^m}{m!} \exp\left(-\mu(B \setminus A)\right) \frac{\mu(B \setminus A)^{k-m}}{(k-m)!}}{\exp\left(-\mu(B)\right) \frac{\mu(B)^k}{k!}} \\
&= \frac{k!}{m!(k-m)!} \exp\left(\mu(B) - \mu(A) - \mu(B \setminus A)\right) \frac{\mu(A)^m \mu(B \setminus A)^{k-m}}{\mu(B)^k} \\
&= \binom{k}{m} \left(\frac{\mu(A)}{\mu(B)}\right)^m \left(\frac{\mu(B \setminus A)}{\mu(B)}\right)^{k-m} \\
&= \binom{k}{m} \left(\frac{\mu(A)}{\mu(B)}\right)^m \left(\frac{\mu(B) - \mu(A)}{\mu(B)}\right)^{k-m} \\
&= \binom{k}{m} \left(\frac{\mu(A)}{\mu(B)}\right)^m \left(1 - \frac{\mu(A)}{\mu(B)}\right)^{k-m},
\end{aligned}$$

hvor $\mu(B) - \mu(A) = \mu(B \setminus A)$, da $A \subseteq B$. Det er en binomialtæthed, hvorved **(i)** er vist. Til at bevise **(ii)** anvendes standardbeviset. Lad $\boldsymbol{X} \sim \text{Poisson}(S, \rho)$, så er

$$\begin{aligned}
\mathbf{E}\left[\mathbf{1}\left[\boldsymbol{X}_B \in F\right]\right] &= P(\boldsymbol{X}_B \in F) \\
&= \sum_{n=0}^{\infty} \left[\frac{\exp(-\mu(B))}{n!} \int_B \cdots \int_B \mathbf{1}\left[\{x_1, \ldots, x_n\} \in F\right] \prod_{i=1}^{n} \left[\rho(x_i)\right] \mathrm{d}x_1 \ldots \mathrm{d}x_n \right] \\
&:= H(\mathbf{1}_F) \quad , \quad \mathbf{1}_F = \mathbf{1}\left[\boldsymbol{X}_B \in F\right].
\end{aligned}$$

Ud fra denne definition af H ses påstanden at være sand for indikatorfunktioner, og det følger af standardbeviset, at det gælder for alle h, hvilket beviser **(ii)**. ■

Eksistensen af endelige PPP'er følger af dele af beviset. Antag at S er begrænset, så $\mu(S) < \infty$, og sæt $B = S$. For en begrænset mængde $A \subset B$ fremgår det af beviset, at Poisson-udviklingen definerer en punktproces, som opfylder definition 1.28, da $N(A) \sim \text{Poi}(\mu(A))$, og $\left(N(A) \,|\, N(B) = n\right) \sim \text{Bin}(n, p)$ med $p \propto \rho$.

Trin 2 – Void-sandsynligheder

Sætning 1.31 (Void-sandsynligheder for PPP'er):
PPP'en $\boldsymbol{X} \sim \text{Poisson}(S, \rho)$ eksisterer og er entydigt bestemt ud fra sine void-sandsynligheder

$$v(B) \;=\; \exp\big(-\mu(B)\big) \quad , \quad B \subseteq S \text{ begrænset.}$$

Bevis:
Bemærk først, at for $Y \sim \text{Poi}(k)$, så er

$$P(Y=0) \;=\; \frac{k^0}{0!}\exp(-k) \;=\; \exp(-k). \tag{1.9}$$

Lad $u \in S$ være et arbitrært punkt og definer $B_i := \big\{ v \in S \mid i-1 \le \|v-u\| < i \big\}$ for $i \in \mathbb{N}$, således at B_i'erne er koncentriske ringe omkring u hver med bredde 1. Dermed er S en disjunkt forening af begrænsede B_i'er. Definer yderligere $\boldsymbol{X} := \bigcup_{i=1}^{\infty} \boldsymbol{X}_i$ med $\boldsymbol{X}_i \sim \text{Poisson}\big(B_i, \rho_i\big)$, hvor hver $\boldsymbol{X}_i$ er uafhængig og ρ_i er restriktionen af ρ til B_i. Så er void-sandsynlighederne for $\boldsymbol{X}$ for en begrænset mængde $B \subseteq S$

$$v(B) \;=\; P(\boldsymbol{X}_B = \emptyset) \;=\; \prod_{i=1}^{\infty} \Big[P(\boldsymbol{X}_i \cap B = \emptyset) \Big] \;=\; \prod_{i=1}^{\infty} \Big[P(N(B_i \cap B) = 0) \Big],$$

hvilket pr. (1.9) er lig

$$\prod_{i=1}^{\infty} \big[\exp\big(-\mu(B_i \cap B)\big) \big] \;=\; \exp\left(\sum_{i=1}^{\infty} \big[-\mu(B_i \cap B) \big] \right) \;=\; \exp\big(-\mu(B)\big).$$

Dette stemmer med en PPP med intensitetsmål μ, så entydigheden følger af sætning 1.13. ■

Bemærk at sætningens resultat kan udvides til void-sandsynlighederne for den simultane fordeling af flere uafhængige PPP'er. Givet at både $\boldsymbol{X}_1$ og $\boldsymbol{X}_2$ er PPP'er på S med intensitetsmål μ_1 og μ_2, så gælder for begrænsede $A, B \subseteq S$, at fordelingen af $(\boldsymbol{X}_1, \boldsymbol{X}_2)$ er entydigt bestemt ved void-sandsynlighederne

$$P(\boldsymbol{X}_1 \cap A = \emptyset, \boldsymbol{X}_2 \cap B = \emptyset) = P(\boldsymbol{X}_1 \cap A = \emptyset)\, P(\boldsymbol{X}_2 \cap B = \emptyset) = \exp\big(-\mu_1(A) - \mu_2(B)\big).$$

Trin 3 – Eksistens af uendelige PPP'er

Efter at have vist eksistensen og entydigheden af endelige PPP'er, så er det sidste trin at fastslå, at det også gælder for uendelige PPP'er. I beviset for sætning 1.31 er S begrænset, og B_i'erne er defineret på en bestemt måde, men en ubegrænset S, så $\mu(S) = \infty$, kunne lige så godt være anvendt ligesom en anden definition af B_i'erne, så længe det gælder, at $S = \cup_{i=1}^{\infty} B_i$ er en disjunkt forening. Ved at anvende de samme argumenter og udledninger som i beviset kan det derfor vises, at void-sandsynlighederne også vil passe for en uendelig PPP.

1.3.2 Egenskaber for Poisson punktprocesser

PPP'er har en række grundlæggende egenskaber, der er med til at fastslå vigtigheden af denne type punktproces. Det skyldes, at flere af dem er så generelle, at de kan benyttes ved konstruktionen af andre punktprocesser, eller det kan bruges til at undersøge i hvilket omfang et punktmønster ligner en PPP.

Ingen interaktion og komplet rumlig tilfældighed

Sætning 1.32 (Disjunkt opdeling af en PPP):
$\boldsymbol{X} \sim \text{Poisson}(S,\rho) \;\Rightarrow\; \boldsymbol{X}_{B_1}, \boldsymbol{X}_{B_2}, \ldots$ er uafhængige for disjunkte $B_1, B_2, \ldots \subseteq S$.

Bevis:
Det er kun nødvendigt at verificere, at $\boldsymbol{X}_{B_1}, \boldsymbol{X}_{B_2}, \ldots, \boldsymbol{X}_{B_n}$ er uafhængige for et endeligt antal disjunkte mængder $B_1, B_2, \ldots, B_n \subseteq S$. Antag at $B = B_1 \cup B_2$, hvor B_1 og B_2 er disjunkte, så følger det fra sætning 1.30, at

$$P(\boldsymbol{X}_{B_1} \in F_1, \boldsymbol{X}_{B_2} \in F_2) = \sum_{n=0}^{\infty} \left[\frac{e^{-\mu(B)}}{n!} \int_B \cdots \int_B \mathbf{1}[\boldsymbol{X}_{B_1} \in F_1, \boldsymbol{X}_{B_2} \in F_2] \prod_{i=1}^{n} \big[\rho(x_i)\big] \, \mathrm{d}x_1 \ldots \mathrm{d}x_n \right].$$

Antag at de første n_1 punkter er i $\boldsymbol{X}_{B_1}$ og resten i $\boldsymbol{X}_{B_2}$, så kan dette skrives

$$\sum_{n=0}^{\infty} \left[\frac{\exp(-\mu(B))}{n!} \sum_{n_1=0}^{n} \left[\frac{n!}{n_1!(n-n_1)!} \int_B \cdots \int_B \mathbf{1}\left[\{x_1, \ldots, x_{n_1}\} \in F_1, \{x_{n_1+1}, \ldots, x_n\} \in F_2\right] \cdot \prod_{i=1}^{n} \big[\rho(x_i)\big] \, \mathrm{d}x_1 \ldots \mathrm{d}x_n \right] \right],$$

og ved at opdele B i B_1 og B_2 fås

$$\sum_{n=0}^{\infty}\left[\sum_{n_1=0}^{n}\left[\frac{\exp(-\mu(B_1))}{n_1!}\frac{\exp(-\mu(B_2))}{(n-n_1)!}\int_B\cdots\int_B \mathbf{1}\left[\{x_1,\ldots,x_{n_1}\}\in F_1\right]\prod_{i=1}^{n_1}\left[\rho(x_i)\right]\mathrm{d}x_1\ldots\mathrm{d}x_{n_1}\cdot\right.\right.$$

$$\left.\left.\int_B\cdots\int_B \mathbf{1}\left[\{x_{n_1+1},\ldots,x_n\}\in F_2\right]\prod_{i=n_1+1}^{n}\left[\rho(x_i)\right]\mathrm{d}x_{n_1+1}\ldots\mathrm{d}x_n\right]\right].$$

Ombytning af summerne giver nu, at

$$\begin{aligned}
&\sum_{n_1=0}^{\infty}\left[\frac{\exp\left(-\mu(B_1)\right)}{n_1!}\int_B\cdots\int_B \mathbf{1}\left[\{x_1,\ldots,x_{n_1}\}\in F_1\right]\prod_{i=1}^{n_1}\left[\rho(x_i)\right]\mathrm{d}x_1\ldots\mathrm{d}x_{n_1}\right]\cdot\\
&\sum_{n=n_1}^{\infty}\left[\frac{\exp\left(-\mu(B_2)\right)}{(n-n_1)!}\int_B\cdots\int_B \mathbf{1}\left[\{x_{n_1+1},\ldots,x_n\}\in F_2\right]\prod_{i=n_1+1}^{n}\left[\rho(x_i)\right]\mathrm{d}x_{n_1+1}\ldots\mathrm{d}x_n\right]\\
&= P\left(\boldsymbol{X}_{B_1}\in F_1\right)\cdot P\left(\boldsymbol{X}_{B_2}\in F_2\right),
\end{aligned}$$

hvilket viser, at $\boldsymbol{X}_{B_1}$ og $\boldsymbol{X}_{B_2}$ er uafhængige. Pr. induktion følger det, at det også er tilfældet, når $B_1,\ldots,B_n\subseteq S$ er begrænsede og disjunkte for $n>2$. Dermed er $\boldsymbol{X}_{B_1},\boldsymbol{X}_{B_2},\ldots$ uafhængige for begrænsede og disjunkte $B_i\subseteq$, $i=1,2,\ldots$. Idet delmængder af S er tællelige foreninger af begrænsede delmængder af S, så ses påstanden at holde for alle disjunkte $B_1,\ldots,B_n\subseteq S$ og alle $n\in\mathbb{N}$. ■

Begreberne *ingen interaktion* og *komplet rumlig tilfældighed* kommer fra resultatet i sætning 1.32, fordi det medfører, at ved enhver opdeling af en PPP i andre PPP'er begrænset til disjunkte mængder, da vil de resulterende punktprocesser altid være uafhængige. Derfor er der ikke nogen interaktion imellem dem, så der er ingen tendens til hverken klyngedannelse eller regularitet, der er kun den såkaldte komplet rumlige tilfældighed.

Superpositionering og udtynding

Det viser sig, at PPP'er er lukket under både superpositionering og uafhængig udtynding, når det udføres på uafhængige PPP'er. Disse forhold vises i de følgende sætninger, hvor den første på delvist er det omvendte resultat af 1.32, fordi den viser, at det er muligt at kombinere flere uafhængige PPP'er til en ny PPP.

Sætning 1.33 (PPP'er er lukket under superpositionering):
Hvis $\boldsymbol{X}_i \sim \text{Poisson}\big(S,\rho_i\big)$ er gensidigt uafhængige for $i = 1,2,\ldots$, og $\rho = \sum_i \big[\rho_i\big]$ er lokalt integrabel, så gælder følgende for superpositionen $\boldsymbol{X} = \bigcup_{i=1}^{\infty} \boldsymbol{X}_i$:

(i) $\boldsymbol{X} \sim \text{Poisson}\big(S,\rho\big)$.

(ii) $\boldsymbol{X}$ er en disjunkt forening med sandsynlighed 1.

Bevis:
(i) Lad μ_i være intensitetsmålet for $\boldsymbol{X}_i$. For en begrænset mængde $B \subseteq S$ og $\sum_{i=1}^{\infty}\big[\mu_i(B)\big] = \mu(B)$ følger det af uafhængigheden af $\boldsymbol{X}_i$'erne, at

$$P(\boldsymbol{X}_B = \emptyset) = \prod_{i=1}^{\infty}\Big[P(\boldsymbol{X}_i \cap B = \emptyset)\Big] = \prod_{i=1}^{\infty}\Big[\exp(-\mu_i(B))\Big] = \exp\big(-\mu(B)\big), \quad (1.10)$$

og fra definition 1.7 følger, at

$$\mu(B) = \sum_{i=1}^{\infty}\big[\mu_i(B)\big] = \sum_{i=1}^{\infty}\left[\int_B \rho_i(u)\,\mathrm{d}u\right] = \int_B \sum_{i=1}^{\infty}\big[\rho_i(u)\big]\,\mathrm{d}u = \int_B \rho(u)\,\mathrm{d}u. \quad (1.11)$$

Idet ρ antages at være lokalt integrabel fås fra (1.11), at $\mu(B)$ er et intensitetsmål, og (1.10) viser, at void-sandsynlighederne for $\boldsymbol{X}$ stemmer pr. sætning 1.31.

(ii) Denne del vil blive bevist ved induktion over k. For $k = 1$ er påstanden oplagt sand, hvilket giver basistrinnet af induktionen.

Antag som induktionshypotese, at **(ii)** gælder for $k-1$. Fra dette og **(i)** er $\bigcup_{i=1}^{k-1}\boldsymbol{X}_i := \boldsymbol{X}_1' \sim \text{Poisson}(S,\rho)$ med $\rho = \sum_{i=1}^{k-1}\big[\rho_i\big]$. Det betyder, at det kun er nødvendigt at se på $\boldsymbol{X}_1'$ og en anden arbitrær PPP $\boldsymbol{X}_2'$. Lad μ_1 og μ_2 være intensitetsmålene for henholdsvis $\boldsymbol{X}_1'$ og $\boldsymbol{X}_2'$, lad ρ_1 og ρ_2 være de tilknyttede intensiteter, og lad $n(\boldsymbol{X}_1') = n_1$, $n(\boldsymbol{X}_2') = n_2$ samt $n = n_1 + n_2$. Sandsynligheden for, at superpositionen af $\boldsymbol{X}_1'$ og $\boldsymbol{X}_2'$ er disjunkt svarer til sandsynligheden for, at deres fællesmængde er tom, hvilket kan udtrykkes vha. Poisson-udviklingen som

$$\begin{aligned} P\big((\boldsymbol{X}_1' \cap \boldsymbol{X}_2') \cap B = \emptyset\big) \;=\; & \sum_{n_1=0}^{\infty}\left[\sum_{n_2=0}^{\infty}\left[\frac{\exp\big(-\mu_1(B)\big)}{n_1!}\,\frac{\exp\big(-\mu_2(B)\big)}{n_2!}\cdot\right.\right. \\ & \int_B\cdots\int_B \mathbf{1}\big[\{x_1,\ldots,x_{n_1}\}\cap\{x_{n_1+1},\ldots,x_n\} = \emptyset\big]\cdot \quad (1.12) \\ & \left.\left.\prod_{i=1}^{n_1}\big[\rho_1(x_i)\big]\prod_{i=n_1+1}^{n}\big[\rho_2(x_i)\big]\,\mathrm{d}x_1\ldots\mathrm{d}x_n\right]\right]. \end{aligned}$$

Indikatorfunktionen i (1.12) kan reelt fjernes, fordi den er 1 *næsten overalt (almost everywhere)*, og integralet i (1.12) kan skrives som

$$\begin{aligned}
&\int_B \cdots \int_B \prod_{i=1}^{n_1} \left[\rho_1(x_i)\right] \prod_{i=n_1+1}^{n} \left[\rho_2(x_i)\right] \mathrm{d}x_1 \ldots \mathrm{d}x_n \\
= &\int_B \cdots \int_B \prod_{i=1}^{n_1} \left[\rho_1(x_i)\right] \mathrm{d}x_1 \ldots \mathrm{d}x_{n_1} \cdot \int_B \cdots \int_B \prod_{i=n_1+1}^{n} \left[\rho_2(x_i)\right] \mathrm{d}x_{n_1+1} \ldots \mathrm{d}x_n \\
= &\left(\int_B \rho_1(x)\,\mathrm{d}x\right)^{n_1} \cdot \left(\int_B \rho_2(x)\,\mathrm{d}x\right)^{n_2} = \mu_1(B)^{n_1}\mu_2(B)^{n_2}. \qquad (1.13)
\end{aligned}$$

Ved at indsætte (1.13) i (1.12) ses, at sandsynligheden er

$$\begin{aligned}
P\left(\left(\boldsymbol{X}_1' \cap \boldsymbol{X}_2'\right) \cap B = \emptyset\right) &= \sum_{n_1=0}^{\infty} \left[\sum_{n_2=0}^{\infty} \left[\frac{\exp(-\mu_1(B))}{n_1!} \frac{\exp(-\mu_2(B))}{n_2!} \mu_1(B)^{n_1} \mu_2(B)^{n_2}\right]\right] \\
&= \sum_{n_1=0}^{\infty} \left[\frac{\exp(-\mu_1(B))}{n_1!} \mu_1(B)^{n_1}\right] \cdot \sum_{n_2=0}^{\infty} \left[\frac{\exp(-\mu_2(B))}{n_2!} \mu_2(B)^{n_2}\right] = 1
\end{aligned}$$

■

Sætning 1.34 (PPP'er er lukket under uafhængig udtynding):
Antag at $\boldsymbol{X} \sim \text{Poisson}\left(S, \rho\right)$ udtyndes med bevaringssandsynlighed $p(u)$ for $u \in S$. Så er

$$\rho_{\text{thin}}(u) = p(u)\,\rho(u) \quad , \quad u \in S.$$

Yderligere er både $\boldsymbol{X}_{\text{thin}}$ og $\boldsymbol{X} \setminus \boldsymbol{X}_{\text{thin}}$ uafhængige PPP'er med intensitet henholdsvis $\rho_{\text{thin}}(u)$ og $\rho(u) - \rho_{\text{thin}}(u)$.

Bevis:
Først og fremmest ses $\rho_{\text{thin}}(u) = p(u)\,\rho(u)$ at holde direkte fra sætning 1.17. Det vises nu, at $\boldsymbol{X}_{\text{thin}}$ er en PPP ved at bestemme dens void-sandsynligheder. For en begrænset mængde $C \subseteq S$ er

$$P(\boldsymbol{X}_{\text{thin}} \cap C = \emptyset) = \mathbf{E}\left[\mathbf{1}\left[\boldsymbol{X}_{\text{thin}} \cap C = \emptyset\right]\right] = \mathbf{E}\left[\mathbf{E}\left[\mathbf{1}\left[\boldsymbol{X}_{\text{thin}} \cap C = \emptyset \,|\, \boldsymbol{X}_C\right]\right]\right].$$

Ved at anvende Poisson-udviklingen er dette lig

$$\sum_{n=0}^{\infty} \left[\frac{\exp(-\mu(C))}{n!} \int_C \cdots \int_C \mathbf{E}\left[\mathbf{1}\left[\boldsymbol{X}_{\text{thin}} \cap C = \emptyset\right] \,|\, \boldsymbol{X}_C = \{x_1, \ldots, x_n\}\right] \prod_{i=1}^{n} \left[\rho(x_i)\right] \mathrm{d}x_1 \ldots \mathrm{d}x_n\right] \qquad (1.14)$$

med $h = \mathbf{E}\left[\mathbf{1}\left[\boldsymbol{X}_{\text{thin}} \cap C = \emptyset\right] \mid \boldsymbol{X}_C = \{x_1, \ldots, x_n\}\right]$. Dette kan også skrives som

$$\begin{aligned}\mathbf{E}\left[\mathbf{1}\left[\boldsymbol{X}_{\text{thin}} \cap C = \emptyset\right] \mid \boldsymbol{X}_C = \{x_1, \ldots, x_n\}\right] &= P\left(\boldsymbol{X}_{\text{thin}} \cap C = \emptyset \mid \boldsymbol{X}_C = \{x_1, \ldots, x_n\}\right) \\ &= \prod_{i=1}^{n}\left[\left(1 - p\left(x_i\right)\right)\right], \end{aligned} \tag{1.15}$$

da x_i'erne er uafhængige og indeholdt i $\boldsymbol{X}_{\text{thin}}$ med sandsynlighed $p\left(x_i\right)$. Ved indsættelse af (1.15) i (1.14) fås

$$\begin{aligned}&\sum_{n=0}^{\infty}\left[\frac{\exp\left(-\mu\left(C\right)\right)}{n!}\int_C \cdots \int_C \prod_{i=1}^{n}\left[\left(1 - p\left(x_i\right)\right)\right] \prod_{i=1}^{n}\left[\rho\left(x_i\right)\right] \mathrm{d}x_1 \ldots \mathrm{d}x_n\right] \\ = \;& \exp\left(-\mu\left(C\right)\right) \sum_{n=0}^{\infty}\left[\frac{1}{n!}\int_C \cdots \int_C \prod_{i=1}^{n}\left[\left(1 - p\left(x_i\right)\right)\rho\left(x_i\right)\right] \mathrm{d}x_1 \ldots \mathrm{d}x_n\right] \\ = \;& \exp\left(-\mu\left(C\right)\right) \sum_{n=0}^{\infty}\left[\frac{1}{n!}\left(\int_C \left(1 - p\left(x\right)\right)\rho\left(x\right)\mathrm{d}x\right)^n\right]. \end{aligned} \tag{1.16}$$

Summen i (1.16) er potensrækken for $\exp\left(\cdot\right)$, og da

$$\rho\left(u\right) - \rho_{\text{thin}}\left(u\right) = \rho\left(u\right) - p\left(u\right)\rho\left(u\right) = \left(1 - p\left(u\right)\right)\rho\left(u\right),$$

så kan definition 1.7 anvendes til at skrive summen som

$$\begin{aligned}\sum_{n=0}^{\infty}\left[\frac{1}{n!}\left(\int_C \left(\rho\left(u\right) - \rho_{\text{thin}}\left(u\right)\right)\mathrm{d}u\right)^n\right] &= \sum_{n=0}^{\infty}\left[\frac{1}{n!}\left(\mu\left(C\right) - \mu_{\text{thin}}\left(C\right)\right)^n\right] \\ &= \exp\left(\mu\left(C\right) - \mu_{\text{thin}}\left(C\right)\right). \end{aligned} \tag{1.17}$$

Ved endeligt at indsætte (1.17) i (1.16) ses void-sandsynlighederne at være

$$P\left(\boldsymbol{X}_{\text{thin}} \cap C = \emptyset\right) = \exp\left(-\mu\left(C\right) + \mu\left(C\right) - \mu_{\text{thin}}\left(C\right)\right) = \exp\left(-\mu_{\text{thin}}\left(C\right)\right). \tag{1.18}$$

Tilsvarende kan det vises for $\boldsymbol{X} \setminus \boldsymbol{X}_{\text{thin}}$ og $C \subseteq S$ begrænset, at

$$P\left(\boldsymbol{X} \setminus \boldsymbol{X}_{\text{thin}} \cap C = \emptyset\right) = \exp\left(-\left(\mu\left(C\right) - \mu_{\text{thin}}\left(C\right)\right)\right). \tag{1.19}$$

Pr. sætning 1.31 kan det konkluderes fra (1.18) og (1.19), at både $\boldsymbol{X}_{\text{thin}}$ og $\boldsymbol{X} \setminus \boldsymbol{X}_{\text{thin}}$ er PPP'er med de korrekte intensiteter som påstået i sætningen.

Til sidst skal det vises, at $\boldsymbol{X}_{\text{thin}}$ og $\boldsymbol{X} \setminus \boldsymbol{X}_{\text{thin}}$ også er uafhængige. For $A,B \subseteq S$ begrænset er

$$\begin{aligned}&P\left(\boldsymbol{X}_{\text{thin}} \cap A = \emptyset\ ,\ \boldsymbol{X} \setminus \boldsymbol{X}_{\text{thin}} \cap B = \emptyset\right) \\ = \;& P\left(\boldsymbol{X} \cap \left(A \cap B\right) = \emptyset\ ,\ \boldsymbol{X}_{\text{thin}} \cap \left(A \setminus B\right) = \emptyset\ ,\ \boldsymbol{X} \setminus \boldsymbol{X}_{\text{thin}} \cap \left(B \setminus A\right) = \emptyset\right). \end{aligned}$$

Det vides, at $\boldsymbol{X}$, $\boldsymbol{X}_{\text{thin}}$ og $\boldsymbol{X} \setminus \boldsymbol{X}_{\text{thin}}$ er PPP'er, og idet de nu er begrænset til disjunkte mængder $A \cap B$, $A \setminus B$ og $B \setminus A$, så er de uafhængige pr. sætning 1.32, og derfor er deres simultane tæthed produktet

$$\begin{aligned}
& P(\boldsymbol{X} \cap (A \cap B) = \emptyset)\ P(\boldsymbol{X}_{\text{thin}} \cap (A \setminus B) = \emptyset)\ P(\boldsymbol{X} \setminus \boldsymbol{X}_{\text{thin}} \cap (B \setminus A) = \emptyset) \\
&= \exp\big(-\mu(A \cap B)\big)\ \exp\big(-\mu_{\text{thin}}(A \setminus B)\big)\ \exp\big(-\big(\mu(B \setminus A) - \mu_{\text{thin}}(B \setminus A)\big)\big) \\
&= \exp\big(-\mu(A \cap B) - \mu_{\text{thin}}(A \setminus B) - \big(\mu(B \setminus A) - \mu_{\text{thin}}(B \setminus A)\big)\big) \\
&= \exp\big(-\mu_{\text{thin}}(A) - \big(\mu(B) - \mu_{\text{thin}}(B)\big)\big) \\
&= P(\boldsymbol{X}_{\text{thin}} \cap A = \emptyset)\ P(\boldsymbol{X} \setminus \boldsymbol{X}_{\text{thin}} \cap B = \emptyset). \qquad (1.20)
\end{aligned}$$

Da den simultane fordeling er lig produktet af de marginale fordelinger pr. (1.20), så følger uafhængigheden af $\boldsymbol{X}_{\text{thin}}$ og $\boldsymbol{X} \setminus \boldsymbol{X}_{\text{thin}}$. ■

En praktisk anvendelse af resultatet er, at det kan benyttes til at konstruere en inhomogen PPP fra en homogen i visse tilfælde. Dette fremgår at følgende korollar, som følger direkte af sætningen.

Korollar 1.35 (Inhomogen PPP fra homogen PPP):
Antag at $\boldsymbol{X} \sim \text{Poisson}\big(S, \rho(u)\big)$, $S \subseteq \mathbb{R}^d$, med intensitet ρ, der er begrænset af en konstant $c < \infty$. Så er $\boldsymbol{X}$ fordelt som en uafhængig udtynding af $\text{Poisson}(S, c)$ med bevaringssandsynlighed $p(u) = \frac{\rho(u)}{c}$.

Slivnyak-Meckes sætninger

Poisson-udviklingen i sætning 1.30 er en brugbar karakterisering af Poisson punktprocesser, og ligeså er sætning 1.31. En yderligere praktisk karakteristik gives i de følgende Slivnyak-Mecke sætninger.

Sætning 1.36 (Slivnyak-Mecke):
Lad $\boldsymbol{X} \sim \text{Poisson}(S, \rho)$ og $h : S \times N_{\text{lf}} \to [0, \infty)$. Så er

$$\mathbf{E}\left[\sum_{u \in \boldsymbol{X}} \big[\, h(u, \boldsymbol{X} \setminus \{u\}) \,\big]\right] = \int_S \mathbf{E}\big[\, h(u, \boldsymbol{X}) \,\big] \rho(u)\, \mathrm{d}u.$$

Bevis:
Der vises tilfældet, hvor $\boldsymbol{X}$ er indeholdt i en begrænset mængde $B \subseteq S$, da det generelle tilfælde i sætningen kræver målteoretiske detaljer, som ligger udenfor dette speciales fokus. Det vises således, at

$$\mathbf{E}\left[\sum_{u\in\boldsymbol{X}_B}\left[h(u,\boldsymbol{X}_B\setminus\{u\})\right]\right] = \int_B \mathbf{E}\left[h(u,\boldsymbol{X}_B)\right]\rho(u)\,\mathrm{d}u. \tag{1.21}$$

Venstresiden af (1.21) kan pr. sætning 1.30 (ii) skrives som

$$\sum_{n=1}^{\infty}\left[\frac{e^{-\mu(B)}}{n!}\int_B\cdots\int_B\sum_{x_i\in\boldsymbol{X}_B}\left[h(x_i,\{x_1,\ldots,x_n\}\setminus\{x_i\})\right]\prod_{i=1}^{n}\left[\rho(x_i)\right]\mathrm{d}x_1\ldots\mathrm{d}x_n\right], \tag{1.22}$$

hvor summen nu starter fra $n=1$, da tilfældet $n=0$ ikke giver mening. Summen i integralet i (1.22) kan fjernes ved at sætte det første argument i h til altid at være x_n og gange med n pga. de n forskellige permutationer. Derved bliver (1.22) til

$$\sum_{n=1}^{\infty}\left[\frac{\exp\left(-\mu(B)\right)}{(n-1)!}\int_B\cdots\int_B h(x_n,\{x_1,\ldots,x_{n-1}\})\prod_{i=1}^{n}\left[\rho(x_i)\right]\mathrm{d}x_1\ldots\mathrm{d}x_n\right].$$

Ved at sætte $n=k+1$, $x_n=u$ samt gange med $\frac{\mu(B)^k}{\mu(B)^k}$ bliver dette til

$$\int_B\sum_{k=0}^{\infty}\left[\tfrac{e^{-\mu(B)}}{k!}\mu(B)^k\int_B\cdots\int_B h(u,\{x_1,\ldots,x_k\})\prod_{i=1}^{k}\left[\tfrac{\rho(x_i)}{\mu(B)}\right]\mathrm{d}x_1\ldots\mathrm{d}x_k\right]\rho(u)\,\mathrm{d}u. \tag{1.23}$$

Punkterne i en PPP er uafhængige, og betinget på antallet er de binomialfordelt, så tætheden for $\{x_1,\ldots,x_k\}$ er $\prod_{i=1}^{k}\left[\frac{\rho(x_i)}{\mu(B)}\right]$. Derved kan integralerne i (1.23) skrives som en middelværdi, og det samme kan summen, da $\frac{\exp(-\mu(B))}{k!}\mu(B)^k$ er en Poissontæthed. Derved kan (1.23) omskrives til

$$\begin{aligned}
&\int_B\sum_{k=0}^{\infty}\left[\frac{\exp\left(-\mu(B)\right)}{k!}\mu(B)^k\mathbf{E}\left[h(u,\{x_1,\ldots,x_k\})\,|\,N(B)=k\right]\right]\rho(u)\,\mathrm{d}u\\
=\;&\int_B\mathbf{E}\Big[\mathbf{E}\big[h(u,\{x_1,\ldots,x_k\})\,|\,N(B)=k\big]\Big]\rho(u)\,\mathrm{d}u\\
=\;&\int_B\mathbf{E}\left[h(u,\boldsymbol{X}_B)\right]\rho(u)\,\mathrm{d}u.
\end{aligned}$$

■

En udvidelse af sætningen fås på følgende vis.

Sætning 1.37 (Udvidet Slivnyak-Mecke):
Lad $\boldsymbol{X} \sim \text{Poisson}(S,\rho)$ og $h : S^n \times N_{\text{lf}} \to [0,\infty)$. Så gælder for alle $n \in \mathbb{N}$, at

$$\begin{aligned} &\mathbf{E}\left[\sum_{u_1,\ldots,u_n \in \boldsymbol{X}}^{\neq} \big[h(u_1,\ldots,u_n, \boldsymbol{X} \setminus \{u_1,\ldots,u_n\}) \big] \right] \\ = \quad &\int_S \cdots \int_S \mathbf{E}\big[h(u_1,\ldots,u_n,\boldsymbol{X}) \big] \prod_{i=1}^{n} \big[\rho(u_i)\big] \, \mathrm{d}u_1 \ldots \mathrm{d}u_n, \end{aligned}$$

Bevis:
Som før vises tilfældet, hvor $\boldsymbol{X}$ er indeholdt i en begrænset mængde $B \subseteq S$, dvs. at

$$\begin{aligned} &\mathbf{E}\left[\sum_{u_1,\ldots,u_n \in \boldsymbol{X}_B}^{\neq} \big[h(u_1,\ldots,u_n, \boldsymbol{X}_B \setminus \{u_1,\ldots,u_n\}) \big] \right] \\ = \quad &\int_B \cdots \int_B \mathbf{E}\big[h(u_1,\ldots,u_n,\boldsymbol{X}_B) \big] \prod_{i=1}^{n} \big[\rho(u_i)\big] \, \mathrm{d}u_1 \ldots \mathrm{d}u_n. \end{aligned} \tag{1.24}$$

Beviset udføres ved induktion over n, og basistrinnet $n = 1$ følger trivialt af sætning 1.36. Antag som induktionshypotese, at (1.24) er sand for $n-1$, og definer

$$\widetilde{h}(u,\mathbf{x}) \;:=\; \sum_{u_2,\ldots,u_n \in \boldsymbol{x}}^{\neq} \Big[h(u,u_2,\ldots,u_n,\boldsymbol{x} \setminus \{u_2,\ldots,u_n\}) \Big]$$

Bemærk at pr. sætning 1.36, så er

$$\begin{aligned} &\mathbf{E}\left[\sum_{u \in \boldsymbol{X}_B} \big[\widetilde{h}(u, \boldsymbol{X}_B \setminus \{u\}) \big] \right] \;=\; \int_B \mathbf{E}\Big[\widetilde{h}(u,\boldsymbol{X}_B) \Big] \rho(u) \, \mathrm{d}u \\ = \quad &\int_B \mathbf{E}\left[\sum_{u_2,\ldots,u_n \in \boldsymbol{X}}^{\neq} h(u,u_2,\ldots,u_n,\boldsymbol{X}_B \setminus \{u_2,\ldots,u_n\}) \right] \rho(u) \, \mathrm{d}u. \end{aligned} \tag{1.25}$$

Middelværdien i (1.25) indeholder kun de $n-1$ ukendte $u_2,\ldots,u_n$, så pr. induktionshypotesen er det lig

$$\int_B \cdots \int_B \mathbf{E}\,[h(u,u_2,\ldots,u_n,\boldsymbol{X}_B)] \prod_{i=2}^{n} \big[\rho(u_i)\big] \, \mathrm{d}u_2 \ldots \mathrm{d}u_n. \tag{1.26}$$

Ved at kombinere (1.25) med resultatet i (1.26) og sætte $u = u_1$ fås, at

$$\begin{aligned} &\int_B \cdots \int_B \mathbf{E}\,[h(u,u_2,\ldots,u_n,\boldsymbol{X}_B)] \, \rho(u_2) \cdots \rho(u_n) \, \mathrm{d}u_2 \ldots \mathrm{d}u_n \, \rho(u) \, \mathrm{d}u \\ = \quad &\int_B \cdots \int_B \mathbf{E}\,[h(u_1,u_2,\ldots,u_n,\boldsymbol{X}_B)] \prod_{i=1}^{n} \big[\rho(u_i)\big] \, \mathrm{d}u_1 \ldots \mathrm{d}u_n \end{aligned}$$

som påkrævet for at fuldføre induktionen. ■

Absolut kontinuitet og tætheden for en Poisson punktproces

Pga. den matematiske kompleksitet af punktprocesser, så er det i almindelighed ikke muligt at bestemme en tæthed for dem mht. det sædvanlige Lebesguemål eller tællemål. I visse tilfælde er det dog muligt at udtrykke tætheden for en punktproces mht. en anden punktproces, hvilket kræver det følgende begreb.

Definition 1.38 (Absolut kontinuitet af punktprocesser):
For to punktprocesser $\boldsymbol{X}_1$ og $\boldsymbol{X}_2$ på det samme rum S, da siges fordelingen af $\boldsymbol{X}_1$ at være *absolut kontinuert* mht. fordelingen af $\boldsymbol{X}_2$, hvis og kun hvis

(i) $P(\boldsymbol{X}_2 \in F) = 0 \;\Rightarrow\; P(\boldsymbol{X}_1 \in F) = 0$ for alle $F \subseteq N_{\text{lf}}$, eller ækvivalent

(ii) $f : N_{\text{lf}} \to [0,\infty]$ findes, så $P(\boldsymbol{X}_1 \in F) = \mathbf{E}\left[\mathbf{1}[\boldsymbol{X}_2 \in F]\, f(\boldsymbol{X}_2)\right]$ for $F \subseteq N_{\text{lf}}$

holder. Funktionen f kaldes tætheden for $\boldsymbol{X}_1$ med hensyn til $\boldsymbol{X}_2$.

Punkt **(i)** i definitionen er den traditionelle version, punkt **(ii)** skyldes *Radon-Nikodyms sætning*, [Rudin, 1987, sæt. 6.10], og den version viser sig nyttig, når det nu skal opstilles tætheder for PPP'er.

Sætning 1.39 (Absolut kontinuitet og tæthed for PPP'er):

(i) For alle $\rho_1, \rho_2 \in \mathbb{R}_+$, da er $\boldsymbol{X}_1 \sim \text{Poisson}(\mathbb{R}^d, \rho_1)$ absolut kontinuert med hensyn til $\boldsymbol{X}_2 \sim \text{Poisson}(\mathbb{R}^d, \rho_2)$ hvis og kun hvis $\rho_1 = \rho_2$.

(ii) Lad $\rho_1, \rho_2 : S \to [0,\infty)$ være intensiteten for henholdsvis $\boldsymbol{X}_1$ og $\boldsymbol{X}_2$, og lad $\mu_1, \mu_2 < \infty$. Hvis $\rho_1(u) > 0 \Rightarrow \rho_2(u) > 0$, så er $\boldsymbol{X}_1 \sim \text{Poisson}(S, \rho_1)$ absolut kontinuert med hensyn til $\boldsymbol{X}_2 \sim \text{Poisson}(S, \rho_2)$ med tæthed

$$f(\boldsymbol{x}) \;=\; \exp\big(\mu_2(S) - \mu_1(S)\big) \prod_{u \in \boldsymbol{x}} \left[\frac{\rho_1(u)}{\rho_2(u)}\right]$$

for lokalt endelige $\boldsymbol{x} \subset S$. ($0/0 := 0$ i denne sammenhæng)

Bevis:
(i) Hvis $\rho_1 = \rho_2$, følger resultatet direkte fra definition 1.38 (i). Antag omvendt, at $\rho_1 \neq \rho_2$, og tag disjunkte delmængder $A_i \subseteq \mathbb{R}^d$ med $|A_i| = 1$ for $i \in \mathbb{N}$, og lad

$$F \;=\; \left\{ \boldsymbol{x} \in N_{\text{lf}} \;\middle|\; \lim_{m \to \infty} \left(\sum_{i=1}^{m} \left[\frac{n\left(\boldsymbol{x}_{A_i}\right)}{m} \right] \right) = \rho_1 \right\}, \tag{1.27}$$

dvs. F er mængden af punktmønstre, hvor hvert område med enhedsstørrelse i gennemsnit har ρ_1 punkter. Bemærk at $\mathbf{E}[n(\boldsymbol{X}_1 \cap A_i)] = \mu_1(A_i) = \int_{A_i} \rho_1 \, \mathrm{d}u = \rho_1$, og at $\mathbf{E}[n(\boldsymbol{X}_2 \cap A_i)] = \rho_2$. Pga. af F i (1.27) og de store tals stærke lov, så er $P(\boldsymbol{X}_1 \in F) = 1$, men $P(\boldsymbol{X}_2 \in F) = 0$, hvilket er en modstrid, så $\rho_1 = \rho_2$.

(ii) Ved at anvende begge punkter i sætning 1.30 vises det ved følgende udregninger, at med den angivne tæthed $f(\boldsymbol{x})$, så er definition 1.38 (ii) opfyldt.

$$
\begin{aligned}
P(\boldsymbol{X}_1 \in F) &= \sum_{n=0}^{\infty} \left[\frac{\exp\left(-\mu_1(S)\right)}{n!} \int_S \cdots \int_S \mathbf{1}[\boldsymbol{X}_1 \in F] \prod_{i=1}^{n} \left[\rho_1(x_i)\right] \mathrm{d}x_1 \ldots \mathrm{d}x_n \right] \\
&= \sum_{n=0}^{\infty} \left[\frac{\exp(-\mu_1(S))}{n!} \int_S \cdots \int_S \mathbf{1}[\boldsymbol{X}_1 \in F] \frac{\exp(-\mu_2(S)) \prod_{i=1}^{n} [\rho_2(x_i)]}{\exp(-\mu_2(S)) \prod_{i=1}^{n} [\rho_2(x_i)]} \prod_{i=1}^{n} \left[\rho_1(x_i)\right] \mathrm{d}x_1 \ldots \mathrm{d}x_n \right] \\
&= \sum_{n=0}^{\infty} \left[\frac{\exp(-\mu_2(S))}{n!} \int_S \cdots \int_S \mathbf{1}[\boldsymbol{X}_1 \in F] \, e^{\mu_2(S) - \mu_1(S)} \prod_{i=1}^{n} \left[\frac{\rho_1(x_i)}{\rho_2(x_i)} \right] \prod_{i=1}^{n} \left[\rho_2(x_i)\right] \mathrm{d}x_1 \ldots \mathrm{d}x_n \right] \\
&= \sum_{n=0}^{\infty} \left[\frac{\exp\left(-\mu_2(S)\right)}{n!} \int_S \cdots \int_S \mathbf{1}[\boldsymbol{X}_1 \in F] \, f(\boldsymbol{X}_2) \prod_{i=1}^{n} \left[\rho_2(x_i)\right] \mathrm{d}x_1 \ldots \mathrm{d}x_n \right] \\
&= \mathbf{E}\left[\mathbf{1}[\boldsymbol{X}_2 \in F] \, f(\boldsymbol{X}_2)\right]
\end{aligned}
$$

■

Direkte fra resultatet fås følgende vigtige korollar.

Korollar 1.40 (Tæthed for en PPP mht. standard PPP'en):
Lad $\boldsymbol{X} \sim \text{Poisson}(S, \rho)$ med S begrænset. Så er $\boldsymbol{X}$ absolut kontinuert mht. standard Poisson punktprocessen Poisson$(S,1)$, og tætheden for $\boldsymbol{X}$ er

$$
f(\boldsymbol{x}) = \exp\left(|S| - \mu(S)\right) \prod_{u \in \boldsymbol{x}} \left[\rho(u)\right].
$$

Bemærk at korollaret angiver, at *enhver* PPP er absolut kontinuert mht. Poisson$(S,1)$ for begrænset S, mens det ikke i almindelighed gælder, at to PPP'er er absolut kontinuert mht. hinanden.

Resultatet i korollaret er vigtigt, fordi det angiver en konkret tæthed, hvilket gør det muligt at opstille mere komplicerede punktprocesser, hvor der ikke nødvendigvis kan findes en eksplicit tæthed, og det gør det bl.a. muligt at bestemme en tæthed for den model, der anvendes i de efterfølgende kapitler.

1.3.3 Mærkede Poisson punktprocesser

Bemærk at der i dette afsnit om mærkede PPP'er anvendes en let ændret notation for at understrege, at de er forskellige fra ordinære PPP'er.

Definition 1.41 (Mærket Poisson punktproces):
Lad $\boldsymbol{Y} \sim \text{Poisson}(T,\varphi)$ med lokalt integrabel intensitet φ. Hvis mærkerne $\{m_u \,|\, u \in \boldsymbol{Y}\}$ betinget på $\boldsymbol{Y}$ er gensidigt uafhængige, så siges $\boldsymbol{X} = \{(u,m_u) \,|\, u \in \boldsymbol{Y}\}$ at være en *mærket PPP* med punkter i T og mærker i M.

Det viser sig, at nogle mærkede PPP'er kan kombineres med mærkerne til at lave en ny PPP, og at selve mærkerne i visse tilfælde er en PPP.

Sætning 1.42:
Lad $\boldsymbol{X}$ være en mærket PPP med punkter i T og mærker i $M \subseteq \mathbb{R}^p$, $p \geq 1$. Hvis hvert mærke m_u betinget på $\boldsymbol{Y} \sim \text{Poisson}(T,\varphi)$ har en tæthed p_u, som ikke afhænger af $\boldsymbol{Y} \setminus \{u\}$, så gælder følgende:

(i) $\boldsymbol{X} \sim \text{Poisson}\big(T \times M, \rho\big)$, hvor $\rho\,(u,m_u) = \varphi\,(u)\,p_u\,(m_u)$.

(ii) Hvis intensiteten på M givet ved $\kappa\,(m_u) := \int_T \rho\,(u,m_u)\,\mathrm{d}u$ er lokalt integrabel, så er mærkerne $\{m_u \,|\, u \in \boldsymbol{Y}\} \sim \text{Poisson}\,(M,\kappa)$.

Bevis:
(i) Bemærk først, at sandsynligheden for $\boldsymbol{X} \in F$ for $F \subseteq N_{\text{lf}}$ kan skrives som

$$P\,(\boldsymbol{X} \in F) \;=\; \mathbf{E}\,[\mathbf{1}\,[\boldsymbol{X} \in F]] \;=\; \mathbf{E}\big[\mathbf{E}\,[\mathbf{1}\,[\boldsymbol{X} \in F]] \,|\, \boldsymbol{Y}\big] \;=\; \mathbf{E}\big[P\big(\boldsymbol{X} \in F \,|\, \boldsymbol{Y}\big)\big]. \tag{1.28}$$

Ved at betragte den betingede sandsynlighed som en funktion af $\boldsymbol{Y} \cap B_T$, $B_T \subseteq T$, så kan sætning 1.30 (ii) anvendes til at skrive (1.28) som

$$\begin{aligned}
&\sum_{n=0}^{\infty}\left[\frac{e^{-\mu(B_T)}}{n!}\int_{B_T}\cdots\int_{B_T} P\left(\{(u_1,m_{u_1}),\ldots,(u_n,m_{u_n})\} \in F\right)\prod_{i=1}^{n}\left[\varphi\,(u_i)\right]\mathrm{d}u_1\ldots\mathrm{d}u_n\right] \\
&= \sum_{n=0}^{\infty}\left[\frac{e^{-\mu(B_T)}}{n!}\int_{B_T}\cdots\int_{B_T} \mathbf{E}\left[\mathbf{1}\left[\left(\{(u_1,m_{u_1}),\ldots,(u_n,m_{u_n})\} \in F\right)\right]\right]\prod_{i=1}^{n}\left[\varphi\,(u_i)\right]\mathrm{d}u_1\ldots\mathrm{d}u_n\right].
\end{aligned} \tag{1.29}$$

Middelværdien tages mht. mærkerne, og $B_M \subseteq M$ indføres til at skrive (1.29) som

$$\sum_{n=0}^{\infty}\left[\frac{\exp\left(-\mu\left(B_T\right)\right)}{n!}\int_{B_T}\cdots\int_{B_T}\int_{B_M}\cdots\int_{B_M}\mathbf{1}\left[\left\{\left(u_1,m_{u_1}\right),\ldots,\left(u_n,m_{u_n}\right)\right\}\in F\right]\cdot\right.$$
$$\left.\prod_{i=1}^{n}\left[p_u\left(m_{u_i}\right)\right]\mathrm{d}m_{u_1}\ldots\mathrm{d}m_{u_n}\prod_{i=1}^{n}\left[\varphi\left(u_i\right)\right]\mathrm{d}u_1\ldots\mathrm{d}u_n\right]. \tag{1.30}$$

Bemærk at hvis $B := B_T \times B_M$, så kan *Fubinis sætning*, [Rudin, 1987, sæt. 8.8], anvendes til at skrive intensitetsmålet af B_T som

$$\begin{aligned}\mu\left(B_T\right) &= \int_{B_T}\varphi\left(u\right)\mathrm{d}u = \int_{B_M}p_u\left(m_u\right)\mathrm{d}m_u\int_{B_T}\varphi\left(u\right)\mathrm{d}u\\ &= \int_{B_T}\int_{B_M}p_u\left(m_u\right)\varphi\left(u\right)\mathrm{d}m_u\mathrm{d}u = \int_B\rho\left(u,m_u\right)\mathrm{d}\left(u,m_u\right) = \mu\left(B\right).\end{aligned}$$

Ved brug af Fubinis sætning og kombinering af produkterne kan (1.30) skrives

$$\sum_{n=0}^{\infty}\left[\frac{\exp\left(-\mu\left(B\right)\right)}{n!}\int_{B}\cdots\int_{B}\mathbf{1}\left[\left\{\left(u_1,m_{u_1}\right),\ldots,\left(u_n,m_{u_n}\right)\right\}\in F\right]\cdot\right.$$
$$\left.\prod_{i=1}^{n}\left[\varphi\left(u_i\right)p_u\left(m_{u_i}\right)\right]\mathrm{d}\left(u_1,m_{u_1}\right)\ldots\mathrm{d}\left(u_n,m_{u_n}\right)\right],$$

så $\boldsymbol{X}\sim\text{Poisson}\left(T\times M,\varphi\left(u\right)p_u\left(m_u\right)\right)$ pr. sætning 1.30 (i).

(ii) For $B\subseteq M$ er void-sandsynlighederne for mærkerne

$$\begin{aligned}P\left(\left\{m_u\,|\,u\in\boldsymbol{Y}\right\}\cap B=\emptyset\right) &= P\left(N\left(\boldsymbol{X}\cap\left(T\times B\right)\right)=0\right) = \exp\left(-\mu\left(\boldsymbol{X}\cap\left(T\times B\right)\right)\right)\\ &= \exp\left(-\int_B\int_T\rho\left(u,m_u\right)\mathrm{d}u\mathrm{d}m_u\right) = \exp\left(-\int_B\kappa\left(m_u\right)\mathrm{d}m_u\right),\end{aligned}$$

hvilket viser, at mærkerne er en PPP på M med intensitet κ pr. sætning 1.31. ■

Tilfældig uafhængig flytning af punkterne i en PPP

Antag at $\boldsymbol{Y}'=\left\{u+m_u\,|\,u\in\boldsymbol{Y}\right\}$ fås ved en *tilfældig uafhængig flytning* af punkterne i $\boldsymbol{Y}\sim\text{Poisson}(\mathbb{R}^d,\rho)$, hvor m_u'erne betinget på $\boldsymbol{Y}$ er uafhængige og følger en tæthed p_u på $\mathbb{R}^d$, som ikke afhænger af $\boldsymbol{Y}\setminus\{u\}$. Ved at lade $\boldsymbol{X}'$ være en mærket PPP med punkter u og mærker $u+m_u$, så følger det af sætning 1.42 (ii), at $\boldsymbol{Y}'$ er en PPP med intensitet

$$\rho'(v) = \int\rho(u)p_u(v-u)\,\mathrm{d}u,$$

givet at ρ' er lokalt integrabel. Hvis ρ er konstant, og p_u ikke afhænger af u, så er $\boldsymbol{Y}'$ en stationær PPP med intensitet $\rho'=\rho$.

1.3.4 Deskriptive størrelser for Poisson punktprocesser

Sætning 1.43 (Parkorrelationsfunktionen for PPP'er):
Lad $\boldsymbol{X} \sim \text{Poisson}(S,\rho)$. Så er parkorrelationsfunktionen givet ved $g(u,v) = 1$.

Bevis:
Det vises først, at $\rho^{(2)}(u,v) = \rho(u)\rho(v)$ for en PPP. Antag at $\boldsymbol{X} \sim \text{Poisson}(S,\rho)$ med intensitet ρ, og at $A,B \in \mathbb{R}^d$. Så fås fra definition 1.15 og sætning 1.37, at

$$\begin{aligned}
\alpha^{(2)}(A \times B) &= \mathbf{E}\left[\sum_{u,v \in \boldsymbol{X}}^{\neq} \left[\mathbf{1}[u \in A, v \in B]\right]\right] \\
&= \int_S \int_S \mathbf{E}\left[\mathbf{1}[u \in A, v \in B]\right] \rho(u)\rho(v)\,\mathrm{d}u\mathrm{d}v \\
&= \int_B \int_A \rho(u)\rho(v)\,\mathrm{d}u\mathrm{d}v \;=\; \int_B \int_A \rho^{(2)}(u,v)\,\mathrm{d}u\mathrm{d}v,
\end{aligned}$$

og det følger af definition 1.18, at $g = 1$. ■

Idet $g = 1$ for en PPP, så indikerer det komplet rumlig tilfældighed, mens $g(u,v) > 1$ indikerer, at par af punkter er mere tilbøjelige til at forekomme simultant, dvs. der er tendens til klyngedannelse, og omvendt indikerer $g(u,v) < 1$ regularitet.

Sætning 1.44 (L-funktionen for PPP'er):
Lad $\boldsymbol{X}$ være en PPP med K-funktion $K(r)$. Så er L-funktionen $L(r) = r$.

Bevis:
Fra definition 1.21, Campbells formler, sætning 1.43, transformationen $s = v - u$ samt et skift til sfæriske koordinater fås det, at

$$\begin{aligned}
L(r) &= \left(\frac{1}{\omega_d |A|}\mathbf{E}\left[\sum_{u,v \in \boldsymbol{X}}^{\neq}\left[\frac{\mathbf{1}[u \in A, \|v-u\| \le r]}{\rho(u)\rho(v)}\right]\right]\right)^{(1/d)} \\
&= \left(\frac{1}{\omega_d |A|}\iint \mathbf{1}[u \in A, \|v-u\| \le r]\,\mathrm{d}u\mathrm{d}v\right)^{(1/d)} \\
&= \left(\frac{1}{\omega_d}\int \mathbf{1}\left[\|s\| \le r\right]\mathrm{d}s\right)^{(1/d)} \\
&= \left(\frac{\sigma_d}{\omega_d}\int_0^r s^{d-1}\,\mathrm{d}s\right)^{(1/d)} \;=\; \left(d\frac{r^d}{d}\right)^{(1/d)} \;=\; r.
\end{aligned}$$

■

I praksis anvendes dette resultat oftest til at plotte r i forhold til $L(r)-r$, fordi det for en PPP vil være 0-funktionen, der er nem at sammenligne med visuelt. På tilsvarende vis som for g, så gælder for tilpas små r, at $L(r)-r>0$ indikerer klyngedannelse, og $L(r)-r>0$ indikerer regularitet.

Sætning 1.45 (F-, G- og J-funktionerne for PPP'er):
For en stationær PPP på $\mathbb{R}^d$ med intensitet $\rho<\infty$, da er

$$F(r) = G(r) = 1-\exp(\rho\omega_d r^d) \quad \text{og} \quad J(r) = 1.$$

Bevis:
At $F(r)=G(r)$ følger ved at anvende definition 1.23, sætning 1.36, og at det for en stationær PPP er muligt at udskifte $b(u,r)$ med $b(v,r)$. Så er nemlig

$$\begin{aligned} G(r) &= \frac{1}{\rho|A|}\mathbf{E}\left[\sum_{u\in\boldsymbol{X}_A}\left[\mathbf{1}[(\boldsymbol{X}\backslash\{u\})\cap b(u,r)\neq\emptyset]\right]\right] \\ &= \frac{1}{\rho|A|}\int_A \mathbf{E}\left[\mathbf{1}[\boldsymbol{X}\cap b(v,r)\neq\emptyset]\right]\rho\,\mathrm{d}u \\ &= \mathbf{E}\left[\mathbf{1}[\boldsymbol{X}\cap b(v,r)\neq\emptyset]\right]\frac{1}{\rho|A|}\int_A \rho\,\mathrm{d}u \\ &= \mathbf{E}\left[\mathbf{1}[\boldsymbol{X}\cap b(v,r)\neq\emptyset]\right] = F(r). \end{aligned}$$

Det følger heraf direkte fra definition 1.23, at $J(r)=1$. For at bestemme et udtryk for $F(r)$ og $G(r)$ benyttes sætning 1.31, stationariten af $\boldsymbol{X}$ og et skift til sfæriske koordinater, hvorved det fås, at

$$\begin{aligned} F(r) &= \mathbf{E}\left[\mathbf{1}[\boldsymbol{X}\cap b(v,r)\neq\emptyset\right] = 1-P[\boldsymbol{X}\cap b(v,r) = \emptyset] \\ &= 1-\exp\left(-\mu\left(b(0,r)\right)\right) = 1-\exp\left(-\sigma_d\int_0^r s^{d-1}\rho\,\mathrm{d}s\right) \\ &= 1-\exp(-\rho\omega_d r^d). \end{aligned}$$

■

Også dette resultat er nyttigt til at vurdere en punktproces eller et punktmønster ud fra, da det som nævnt ovenfor er simpelt at sammenligne med en konstant funktion. Værdien af J-funktionen indikerer også henholdsvis klyngetendenser og regularitet, men i modsætning til g og L, så er det $J(r)<1$, der indikerer klyngedannelse, mens $J(r)>1$ indikerer regularitet.

1.4 Cox punktprocesser

En PPP er i praksis ofte ikke velegnet til at modellere virkelige datasæt, hvorfor der er udviklet mere fleksible typer af punktprocesser, der dog i flere tilfælde bygger på egenskaber ved PPP'er. En af svaghederne ved PPP'er er den totale rumlige tilfældighed, hvor punkterne ikke har nogen form for indbyrdes indflydelse på hinanden, hvilket kun sjældent er tilfældet i konkrete datasæt. En type af interaktion mellem punkter er klyngedannelse, og dette fænomen kan bl.a. repræsenteres ved brug af Cox punktprocesser, der er emnet i dette afsnit

1.4.1 Definition og grundlæggende egenskaber

Definition 1.46 (Cox punktproces):
Antag at $Z = \{ Z(u) \,|\, u \in S \}$ med $S \subseteq \mathbb{R}^d$ er en ikke-negativ stokastisk funktion, så $u \to Z(u)$ er lokalt integrabel med sandsynlighed 1. Hvis den betingede fordeling af $\boldsymbol{X} \,|\, Z$ er en PPP på S med intensitet Z, så siges $\boldsymbol{X}$ at være en *Cox punktproces drevet af* Z.

For at udlede egenskaber ved Cox punktprocesser kan det benyttes, at $\boldsymbol{X} \,|\, Z$ er en PPP. Det gør det bl.a. muligt at bestemme udtryk for ρ, $\rho^{(2)}$ og g.

Sætning 1.47 (ρ og $\rho^{(2)}$ for Cox punktprocesser):
Antag at $\boldsymbol{X} \subseteq S$ og $u, v \in S$. Hvis $\boldsymbol{X}$ er en Cox punktproces drevet af Z, så er

$$\rho(u) = \mathbf{E}[Z(u)] \quad \text{og} \quad \rho^{(2)}(u,v) = \mathbf{E}[Z(u)Z(v)].$$

Bevis:
Fra definition 1.7 og 1.46 samt ved at betinge på Z fås, at

$$\begin{aligned}
\mu(B) &= \mathbf{E}\left[\sum_{u \in \boldsymbol{X}} \Big[\mathbf{1}[u \in B] \Big] \right] \\
&= \mathbf{E}\left[\mathbf{E}\left[\sum_{u \in \boldsymbol{X}} \Big[\mathbf{1}[u \in B] \Big] \Big| Z \right] \right] \\
&= \mathbf{E}\left[\int_B Z(u) \, \mathrm{d}u \right] = \int_B \mathbf{E}[Z(u)] \, \mathrm{d}u.
\end{aligned}$$

For at vise den anden lighed benyttes definition 1.15 og 1.46, Campbells formler samt at $\rho^{(2)}_{\boldsymbol{X}|Z}(u,v) = Z(u)Z(v)$ jf. beviset for sætning 1.43. Så er

$$\begin{aligned}
\alpha^{(2)}(A\times B) &= \mathbf{E}\left[\mathbf{E}\left[\sum_{u,v\in\boldsymbol{X}}^{\neq}\left[\mathbf{1}[(u,v)\in A\times B]\right]\middle| Z\right]\right] \\
&= \mathbf{E}\left[\iint \mathbf{1}[(u,v)\in A\times B]Z(u)Z(v)\,\mathrm{d}u\mathrm{d}v\right] \\
&= \int_B\int_A \mathbf{E}[Z(u)Z(v)]\,\mathrm{d}u\mathrm{d}v.
\end{aligned}$$

■

Følgende korollar om parkorrelationsfunktionen følger direkte fra resultatet.

Korollar 1.48 (Parkorrelationsfunktionen for Cox punktprocesser):
Parkorrelationsfunktionen for en Cox punktproces drevet af Z er givet ved

$$g(u,v) = \frac{\mathbf{E}[Z(u)Z(v)]}{\mathbf{E}[Z(u)]\mathbf{E}[Z(v)]},$$

såfremt $Z(u)$ har endelig varians for alle $u\in S$.

For de fleste konkrete Cox punktprocesser er $g > 1$, der jf. afsnit 1.3.4 indikerer klyngedannelse. En anden indikation herpå fås af følgende sætning.

Sætning 1.49 (Overspredning af Cox punktprocesser):
For $B\subseteq S$, da gælder for en Cox punktproces, at $\mathbf{Var}[N(B)] \geq \mathbf{E}[N(B)]$.

Bevis:
Lad $\boldsymbol{X}$ være en Cox punktproces på S, $B\subseteq S$, og benyt, at $\mathbf{Var}[A] = \mathbf{E}[\mathbf{Var}[A|Y] + \mathbf{Var}[\mathbf{E}[A|Y]]$ samt at $\mathbf{Var}[\mathrm{Poi}(\lambda)] = \mathbf{E}[\mathrm{Poi}(\lambda)]$. Så er

$$\begin{aligned}
\mathbf{Var}[N(B)] &= \mathbf{E}\left[\mathbf{Var}[N(B)|Z]\right] + \mathbf{Var}\left[\mathbf{E}[N(B)|Z]\right] \\
&\geq \mathbf{E}\left[\mathbf{Var}[N(B)|Z]\right] \\
&= \mathbf{E}[N(B)].
\end{aligned}$$

■

Fænomenet i sætningen kaldes overspredning, fordi variansen af antallet af punkter i et område B kan være større end middelværdien. Det betyder, at der er sandsynlighed for forskelligt antal punkter i områder, der er lige store, og det indikerer derfor, at der kan opstå klyngetendenser. Tætheden for en Cox punktproces fremgår af det næste resultat, som følger af korollar 1.40.

Sætning 1.50 (Tæthed for en Cox punktproces):
Antag at $\boldsymbol{X}$ er en Cox punktproces begrænset til $B \subseteq S$ med $|B| < \infty$. Så er tætheden af $\boldsymbol{X}_B$ mht. Poisson$(S,1)$ for endelige punktmønstre $\boldsymbol{x} \subset B$ givet ved

$$f(\boldsymbol{x}) \;=\; \mathbf{E}\left[\exp\left(|B| - \int_B Z(u)\,\mathrm{d}u\right) \prod_{u \in \boldsymbol{X}_B} \Big[Z(u)\Big]\right].$$

1.4.2 Neyman-Scott punktprocesser som Cox punktprocesser

Som en konkret type Cox punktprocesser behandles her typen Neyman-Scott.

Definition 1.51 (Neyman-Scott punktproces):
Lad $\boldsymbol{C}$ være en stationær PPP på $\mathbb{R}^d$ med intensitet $\kappa > 0$. Betinget på $\boldsymbol{C}$, lad da $\boldsymbol{X}_c$ med $c \in \boldsymbol{C}$ være uafhængige PPP'er på $\mathbb{R}^d$, hvor $\boldsymbol{X}_c$ har intensitet

$$\rho_c(u) \;=\; \alpha k(u-c),$$

hvor $\alpha > 0$ er en parameter, og k er en kerne, dvs. for alle $c \in \mathbb{R}^d$, da er $u \to k(u-c)$ en tæthed. Så er $\boldsymbol{X} = \bigcup_{c \in \boldsymbol{C}} \boldsymbol{X}_c$ et specialtilfælde af en *Neyman-Scott punktproces* med klyngecentre $\boldsymbol{C}$ og klynger $\boldsymbol{X}_c$, $c \in \boldsymbol{C}$.

I den generelle definition på en Neyman-Scott punktproces antages klyngeprocessen at være en stationær PPP som i definitionen, men betinget på en given klynge, så antages punkterne i klyngen at være i.i.d. omkring klyngecenteret, men de behøver ikke nødvendigvist at være resultatet af en PPP. Bemærk i øvrigt, at klyngecentrene $c \in \boldsymbol{C}$ *ikke* er en del af det genererede punktmønster.

Frem for at betragte $\boldsymbol{X}$ som foreningen af klyngepunktprocesserne, $\boldsymbol{X} = \bigcup_{c \in \boldsymbol{C}} \boldsymbol{X}_c$, så kan $\boldsymbol{X}$ også defineres som en Cox punktproces som vist i følgende resultat.

Sætning 1.52 (Neyman-Scott punktprocesser som Cox punktprocesser):
Pr. sætning 1.33 er $\boldsymbol{X}$ som i definition 1.51 en Cox punktproces drevet af

$$Z(u) = \sum_{c \in \boldsymbol{C}} \alpha k(u-c),$$

hvor $\alpha > 0$ er en parameter og k en kerne. Z er stationær, og $\rho = \alpha\kappa$ er lokalt integrabel. Yderligere er Z isotropisk, når $k(u) = k(\|u\|)$ er isotropisk.

Bevis:
Stationariteten af Z vises ved brug af Campbells formler, og at $\rho(c) = \kappa$. Så er

$$\mathbf{E}[Z(u)] \;=\; \mathbf{E}\left[\sum_{c \in \boldsymbol{C}} \Big[\alpha k(u-c)\Big]\right] \;=\; \alpha \int k(u-c)\kappa \,\mathrm{d}c \;=\; \alpha\kappa \int k(u-c)\,\mathrm{d}c \;=\; \alpha\kappa.$$

■

Det er i almindelighed svært at bestemme deskriptive størrelser for Cox punktprocesser, men i tilfældet Neyman-Scott er det bl.a. muligt at bestemme g.

Sætning 1.53 (Parkorrelationsfunktion for Neyman-Scott punktproces):
Parkorrelationsfunktionen for en Cox punktproces $\boldsymbol{X}$ som i sætning 1.52 er

$$g(u) = 1 + \frac{1}{\kappa} \int k(v)k(u+v)\,\mathrm{d}v.$$

Bevis:
Først udledes et udtryk for $\rho^{(2)}(u,v) = \mathbf{E}[Z(u)Z(v)]$ vha. Campbells formler:

$$\begin{aligned}
\mathbf{E}[Z(u)Z(v)] \;&=\; \mathbf{E}\left[\sum_{c \in C} \Big[\alpha k(u-c)\Big] \sum_{c \in C} \Big[\alpha k(v-c)\Big]\right] \\
&=\; \alpha^2 \mathbf{E}\left[\sum_{c,\widetilde{c} \in C}^{\neq} \Big[k(u-c)k(v-\widetilde{c})\Big]\right] + \alpha^2 \mathbf{E}\left[\sum_{c \in C} \Big[k(u-c)k(v-c)\Big]\right] \\
&=\; \alpha^2 \iint k(u-c)k(v-\widetilde{c})\kappa^2 \,\mathrm{d}c\mathrm{d}\widetilde{c} + \alpha^2 \int k(u-c)k(v-c)\kappa \,\mathrm{d}c \\
&=\; \alpha^2\kappa^2 + \alpha^2\kappa \int k(u-c)k(v-c)\,\mathrm{d}c
\end{aligned}$$

Fra korollar 1.48 og sætning 1.52 fås, at $g(u,v) = 1 + \kappa^{-1} \int k(u-c)k(v-c)\,\mathrm{d}c$. Da k er stationær, så er $g(u,v) = g(u-v) = g(s) = 1 + \kappa^{-1} \int k(s+v-c)k(v-c)\,\mathrm{d}c$, og ved at sætte $v - c = t$ fås det, at $g(u,v) = g(s) = 1 + \kappa^{-1} \int k(s+t)k(t)\,\mathrm{d}t$. ■

Bemærk at det fremgår af resultatet, at $g > 1$, da k og κ begge er positive, hvilket bekræfter, at denne type model har klyngetendenser. Valget af kernen k adskiller forskellige typer af Neyman-Scott punktprocesser, og ved valget af en normalfordeling fås den såkaldte Thomas punktproces, der skal vise sig at være en væsentlig inspiration for den model, som opstilles i kapitel 2.

Thomas punktprocesser

En Thomas punktproces fås ved at vælge kernen

$$k(u) = \frac{1}{(2\pi\omega^2)^{d/2}} \exp\left(\frac{-\|u\|^2}{2\omega^2}\right),$$

hvilket ses at være tætheden for $N_d(0, \omega^2 I_d)$, hvor der haves d i.i.d. variable hver med middelværdi 0 og varians ω^2. På baggrund af sætning 1.53 kan det vises, at

$$g(u) = 1 + \frac{1}{\kappa(4\pi\omega^2)^{d/2}} \exp\left(-\frac{\|u\|^2}{4\omega^2}\right), \tag{1.31}$$

og da kernen er isotropisk, så er g det også. Det er dermed muligt analytisk at bestemme et udtryk for K-funktionen, hvilket nu gøres for $d = 2$.

Sætning 1.54 (K-funktionen for Thomas punktprocesser):
For en Thomas punktproces i 2 dimensioner er K-funktionen givet ved

$$K(r) = \pi r^2 + \frac{1}{\kappa}\left(1 - \exp\left(-\frac{r^2}{4\omega^2}\right)\right).$$

Bevis:
Idet g er isotropisk, så følger det af sætning 1.22 og (1.31), at for $d = 2$, da er

$$\begin{aligned} K(r) &= 2\pi \int_0^r s g(s)\,\mathrm{d}s = 2\pi \int_0^r \left(1 + \frac{1}{\kappa(4\pi\omega^2)} \exp\left(-\frac{\|s\|^2}{4\omega^2}\right)\right) s\,\mathrm{d}s \\ &= 2\pi \left(\int_0^r s\,\mathrm{d}s + \frac{1}{\kappa(4\pi\omega^2)} \int_0^r \exp\left(-\frac{\|s\|^2}{4\omega^2}\right) s\,\mathrm{d}s\right) \\ &= 2\pi \left[\frac{1}{2}s^2 - \frac{1}{2\pi\kappa} \exp\left(-\frac{\|s\|^2}{4\omega^2}\right)\right]_0^r = \pi r^2 + \frac{1}{\kappa}\left(1 - \exp\left(-\frac{r^2}{4\omega^2}\right)\right). \end{aligned}$$

■

1.5 Markovkæde Monte Carlo og Gibbs sampling

Det er ofte ikke muligt analytisk at bestemme de marginale fordelinger for parametrene i komplekse og flerdimensionale statistiske modeller, da de involverede tætheder kan være for komplicerede. Eksempelvis kan modeller indeholdende tætheden for en inhomogen PPP eller en Cox punktproces være for komplekse til, at det umiddelbart er muligt at udføre inferens for de indgående parametre. For at muliggøre udførsel af eksempelvis inferens i sådanne situationer er der udviklet approksimative metoder, som bl.a. er baseret på sandsynlighedsteoretiske forhold. En væsentlig type af metoder er af typen Markovkæde Monte Carlo (MCMC), hvor der dannes en Markovkæde bestående af udtræk fra en bestemt fordeling.

I dette afsnit præsenteres en MCMC-metode med navnet Metropolis-Hastings samt et specialtilfælde i form af Gibbs sampling. Yderligere gennemgås, hvorledes Metropolis-Hastings kan anvendes til at simulere punktprocesser med et henholdsvis varierende og fast antal punkter. Afsnittet er primært baseret på [Walsh, 2004], [Casella & George, 1992], [Berthelsen & Møller, 2004] samt [Møller & Waagepetersen, 2004, kap. 7], der alle fire indeholder mere uddybende baggrundsteori vedr. Markovkæder og vurdering af fejlstørrelsen, hvilket ikke vil blive præsenteret her.

1.5.1 Monte Carlo integration

Monte Carlo-delen i MCMC-metoder kommer fra en særlig tilgang til at approksimere vanskelige integraler. Hvis det for et integrale $\int_a^b g(x)\,\mathrm{d}x$ er muligt at opskrive integranden $g(x)$ som et produkt af en funktion $f(x)$ og en tæthed $p(x)$, der er defineret og positiv på $(a;b)$, så er

$$\int_a^b g(x)\,\mathrm{d}x \;=\; \int_a^b f(x)p(x)\,\mathrm{d}x \;=\; \mathbf{E}\big[f(x)\big].$$

Hvis X er en stokastisk variabel, som følger tætheden $p(x)$, så kan der tilfældigt udvælges realisationer $x_1,\ldots,x_n$, og en approksimation vil være

$$\int_a^b g(x)\,\mathrm{d}x \;=\; \mathbf{E}\big[f(x)\big] \;\approx\; \frac{1}{n}\sum_{i=1}^{n}\Big[f\left(x_i\right)\Big].$$

Det er denne metode, der kendes som *Monte Carlo integration*, og den kan også benyttes til at approksimere integraler med betingede udtryk, eksempelvis kan

$$I(y) \;=\; \int f\left(y|x\right)p(x)\,\mathrm{d}x \quad \text{approksimeres ved} \quad \widehat{I}(y) \;=\; \frac{1}{n}\sum_{i=1}^{n}\Big[f\left(y|x_i\right)\Big].$$

1.5.2 Metropolis-Hastings

For at kunne udtrække værdier fra en kompliceret fordeling, hvor normaliseringskonstanten ikke kendes, kan der anvendes en særlig type Markovkæde. Hvis en Markovkæde opfylder at være *aperiodisk*, *irreducibel* og *reversibel*, så vil den konvergere mod en entydig stationær fordeling π, og formålet med at anvende Metropolis-Hastings er, at det genererer en sådan kæde, der vil konvergere mod en valgfri fordeling π. Den generaliserede metode hertil er introduceret i [Hastings, 1970], der er en videreudvikling af metoden i [Metropolis et al., 1953].

Givet at kæden er i en tilstand X_i, $i \in \mathbb{N}_0$, lad da $\pi = h(x)/c$, hvor c er en normaliseringskonstant, være den tæthed, der ønskes at konvergere mod, og lad $h(x_0) > 0$. Lad yderligere $q(\cdot|\cdot)$ være en *forslagsfordeling*, som anvendes til at udtrække et forslag til den næste tilstand X_{i+1}. Til at afgøre, hvad den næste tilstand skal være, benyttes

$$\textit{Hastingsforholdet } H\left(x,x'\right) \;=\; \frac{\pi\left(x'\right) q\left(x\,|\,x'\right)}{\pi\left(x\right) q\left(x'\,|\,x\right)} \;=\; \frac{h\left(x'\right) q\left(x\,|\,x'\right)}{h\left(x\right) q\left(x'\,|\,x\right)}, \qquad (1.32)$$

hvor x er den aktuelle værdi i tilstand i, og x' er forslaget til værdien i tilstand $i+1$. Hvis $h\left(x\right) q\left(x'\,|\,x\right) = 0$, så sættes $H\left(x,x'\right) = \infty$. Bemærk at c forkortes ud i Hastingsforholdet, hvorfor det kun er nødvendigt at kende π op til proportionalitet. Forslaget x' accepteres med

$$\textit{acceptsandsynligheden } a\left(x,x'\right) \;=\; \min\left\{1, H\left(x,x'\right)\right\},$$

hvilket i praksis gøres ved at lave en evaluering af $r < H\left(x,x'\right)$, hvor $r \sim \text{Unif}(0,1)$. Hvis x er endimensional, og der anvendes eksempelvis en normalfordeling $\text{N}\left(x',\sigma\right)$ som forslagstætheden $q\left(x\,|\,x'\right)$, så er

$$q\left(x\,|\,x'\right) \;=\; \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{\left(x-x'\right)^2}{2\sigma^2}\right) \;=\; q\left(x'\,|\,x\right),$$

dvs. $q(\cdot|\cdot)$ er symmetrisk i x og x', således at den også kan forkortes ud i (1.32). For symmetriske forslagsfordelinger bliver Hastingsforholdet derfor reduceret til

$$\textit{Metropolisforholdet } H\left(x,x'\right) \;=\; \frac{h\left(x'\right)}{h\left(x\right)},$$

som blev anvendt i den oprindelige Metropolis-algoritme. Bemærk at i dette tilfælde vil forslaget altid blive accepteret, hvis $h(x')/h(x) \geq 1$, dvs. hvis x' er mere sandsynlig end x. Algoritmisk kan forløbet ved anvendelse af Metropolis-Hastings beskrives som følger.

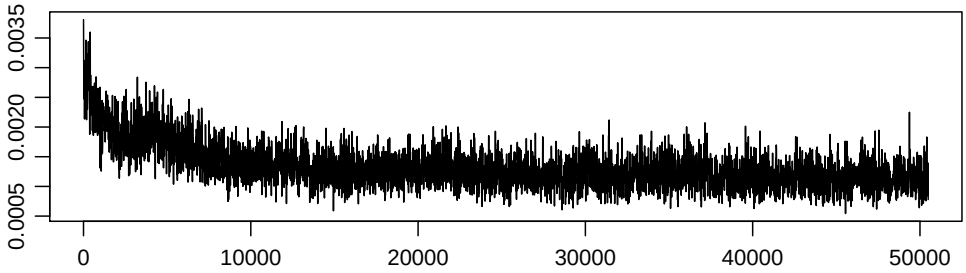
Algoritme 1.55 (Metropolis-Hastings):

1) Vælg en startværdi $X_0 = x_0$, hvor $h(x_0) > 0$, og gør for alle $i = 0, \ldots, N, N \in \mathbb{N}$:
2) Udtræk et forslag x' fra $q(x' \mid x_i)$
3) Beregn $H(x_i, x')$
4) Hvis $r < H(x_i, x')$ sæt $X_{i+1} = x'$, ellers sæt $X_{i+1} = x_i$

□

Ved konkret anvendelse af algoritme 1.55 vil det tage et antal iterationer, før kæden kan forventes i tilfredsstillende grad at ligne sin stationære fordeling π. Derfor bør værdierne i kæden først anvendes efter et såkaldt *burn-in* $k < N, k \in \mathbb{N}$, som er det antal led i starten af kæden, der sorteres fra, dvs. der benyttes kun led $X_j, j = k+1, k+2, \ldots, N$.

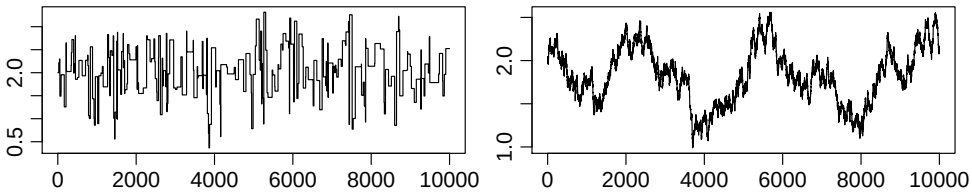
Der kan være store forskelle på hvor mange iterationer, det tager for en kæde at nærme sig sin stationære fordeling, og dermed hvilket k, der skal vælges. Én måde at vurdere det på, hvis x er endimensional, er at betragte et *traceplot*, som viser værdierne i kæden i forhold til iterationsnummeret. På figur 1.1 ses et eksempel på et traceplot, hvor det fremgår, at det tager ca. 25.000 iterationer før kæden bliver stabiliseret, så i dette tilfælde vil $k = 25.000$ være et passende burn-in. Er x multidimensional, så kan der laves et traceplot for hver komponent, og så må det vurderes, hvornår der er konvergens for alle komponenter. Der findes mere formelle og decideret kvantitative tilgange til at bestemme et velegnet burn-in, men disse vil ikke blive anvendt i dette speciale.



***Figur 1.1** – Eksempel på traceplot.*

Forskellen på værdien imellem tilstanden X_i og forslaget til X_{i+1} afgøres af spredningen af $q(\cdot \mid \cdot)$. Det er derfor vigtigt, at den har en passende værdi, således at de såkaldte spring er af en tilpas størrelse, da der ellers vil opstå en høj autokorrelation imellem tilstandene, og mange af værdierne vil være fra et lille område af π frem for hele dens domæne. Dette omtales som kædens *mixing*, hvor god mixing er når størstedelen af domænet for π indgår i kæden. Til at opnå en hensigtsmæssig forslagsspredning anvendes bl.a. *acceptraten*, dvs. antallet af accepterede forslag i forhold til antallet af iterationer.

En generel retningslinje er, at acceptraten bør ligge i intervallet [0,2 ; 0,4], og det er vist i [Roberts et al., 1997], at den teoretisk optimale acceptrate er 0,234. Acceptraten for eksemplet på figur 1.1 er 0,29, hvilket medvirker til at give en god mixing, som er kendetegnet ved, at traceplottet ligner hvid støj. Hvis acceptraten er for lav, er spredningen af $q(\cdot|\cdot)$ for høj, dvs. for få forslag bliver accepteret, og det fremstår på et traceplot som mange vandrette segmenter. Omvendt betyder en for høj acceptrate, at spredningen er for lille, og det tilfælde får et traceplot til at ligne en random walk. Disse to forhold kan ses illustreret på figur 1.2.



Figur 1.2 *– Traceplot med en acceptrate på henholdsvis 0,02 (venstre) og 0,98 (højre).*

1.5.3 Gibbs sampling og Metropolis-i-Gibbs

Ved brug af Metropolis-Hastings kan der godt anvendes en multidimensional x, men det kan i nogle situationer være besværligt at konstruere en Markovkæde med god mixing ved direkte anvendelse af den simultane tæthed. Det kan eksempelvis være tilfældet, hvis parametrene i tætheden er indbyrdes afhængige, eller det kan vise sig at være problematisk at opnå en fornuftig acceptrate, når flere parametre skal opdateres simultant. Derfor er der i litteraturen fremkommet forskellige bud på løsning af denne problemstilling, og i [Geman & Geman, 1984] introduceres konceptet *Gibbs sampling* som et specialtilfælde af Metropolis-Hastings. Ideen heri er, at der frem for at bruge simultane tætheder kun anvendes betingede tætheder af én parameter, således at der betinges på alle øvrige parametre, da det forsimpler de nødvendige beregninger.

Lad $x = (x_1, \ldots, x_n)$ være en multidimensional parameter, der anvendes i en tæthed, og lad $x^i = (x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_n)$ være x uden komponenten x_i. I Gibbs sampling foretages en opdatering for hver komponent i x ud fra den betingede tæthed $p\left(x_i \,|\, x^i\right)$, og i en klassisk Gibbs sampler vil acceptsandsynligheden være 1. Det skyldes, at brugen af de betingede tætheder ofte vil resultere i velkendte fordelinger, hvor der findes metoder til at udtrække i.i.d. værdier fra. Hvis det ikke er tilfældet, så kan opdateringer foretages på samme vis som i Metropolis-Hastings ud fra Hastingsforholdet, der tales i så fald om *Metropolis-i-Gibbs*, og det er netop denne type opdateringer, som vil blive anvendt i kapitel 3.

Da der i Gibbs sampling skal opdateres flere forskellige parametre, så er der forskellige rækkefølger at gøre det i. Én måde er et *cyklisk opdateringsskema*, hvor parametrene opdateres fortløbende fra 1 til n, og derefter startes forfra. I så fald vil forløbet af Metropolis-i-Gibbs med parametre $x = (x_1, x_2, x_3)$ se ud som følger.

Algoritme 1.56 (Gibbs sampling med cyklisk opdateringsskema):

1) Vælg en startværdi $X_0 = x_0$, hvor $h(x_0) > 0$, og gør for alle $j = 0, \ldots, N, N \in \mathbb{N}$:
2) Udtræk et forslag x_1' fra $q\left(x_1' \mid x_{1\,j}\right)$ og udregn $H\left(x_1, x_1'\right) = \dfrac{h\left(x_1' \mid x_{2\,j}, x_{3\,j}\right)}{h\left(x_1 \mid x_{2\,j}, x_{3\,j}\right)}$
3) Hvis $r < H\left(x_1, x_1'\right)$ sæt $x_{1\,j+1} = x_1'$, ellers sæt $x_{1\,j+1} = x_{1\,j}$
4) Udtræk et forslag x_2' fra $q\left(x_2' \mid x_{2\,j}\right)$ og udregn $H\left(x_2, x_2'\right) = \dfrac{h\left(x_2' \mid x_{1\,j+1}, x_{3\,j}\right)}{h\left(x_2 \mid x_{1\,j+1}, x_{3\,j}\right)}$
5) Hvis $r < H\left(x_2, x_2'\right)$ sæt $x_{2\,j+1} = x_2'$, ellers sæt $x_{2\,j+1} = x_{2\,j}$
6) Udtræk et forslag x_3' fra $q\left(x_3' \mid x_{3\,j}\right)$ og udregn $H\left(x_3, x_3'\right) = \dfrac{h\left(x_3' \mid x_{1\,j+1}, x_{2\,j+1}\right)}{h\left(x_3 \mid x_{1\,j+1}, x_{2\,j+1}\right)}$
7) Hvis $r < H\left(x_3, x_3'\right)$ sæt $x_{3\,j+1} = x_3'$, ellers sæt $x_{3\,j+1} = x_{3\,j}$

□

I visse situationer vil denne type opdateringsskema ikke give en reversibel Markovkæde. Er der brug for det, så kan der anvendes et andet type opdateringsskema kaldet *forwards-backwards*. Her opdateres der i rækkefølgen $x_1, \ldots x_{n-1}, x_n, x_{n-1}, \ldots, x_1$, dvs. at i forhold til algoritme 1.56 vil trin 7) blive efterfulgt af trin 4) og 5) (hvor den netop opdaterede værdi af x_3 benyttes i beregningen af $H(x_2, x_2')$), og derefter følger trin 2) og 3) inden iterationen afsluttes. En anden mulighed for at lave reversibel Gibbs sampling er at opdatere parametrene i tilfældig rækkefølge i hver iteration.

1.5.4 Simulering af punktprocesser vha. Metropolis-Hastings

Da punktprocesser ofte har komplicerede tætheder, hvor normaliseringskonstanten ikke nødvendigvis er kendt, så besværliggøres simulation af dem af samme årsager som i ovenstående tilfælde. Det er derfor nødvendigt at anvende en metode til at generere konkrete punktmønstre for en given punktproces, og her kan Metropolis-Hastings med visse modifikationer anvendes. Når der arbejdes med punktprocesser frem for tætheder i almindelighed, kan der anvendes en algoritme af typen *Birth-Death-Move* (BDM), hvor Birth-Death betyder, at der kan tilføjes og fjernes punkter i et punktmønster, og Move betyder, at de enkelte punkter kan flyttes.

Lad $\boldsymbol{Y}$ være en punktproces med tæthed $f(\boldsymbol{y}) = \pi(\boldsymbol{y})$ mht. Poisson$(B,1)$, $B \subset \mathbb{R}^d$, $|B|$ endelig, $\boldsymbol{y} \in N_{\text{lf}}$, og lad $\pi(\boldsymbol{y}) = h(\boldsymbol{y})/c$, hvor c er en normaliseringskonstant. I en BDM-algoritme benyttes henholdsvis $q(\boldsymbol{y}) \in [0\,;1]$ og $p(\boldsymbol{y}) \in [0\,;1]$ som sandsynligheder for, om der i den enkelte iteration skal laves en opdatering, hvor der forsøges henholdsvis at tilføje, fjerne eller slette et punkt. Bemærk at $q(\boldsymbol{y})$ ikke skal forveksles med den tidligere nævnte forslagsfordeling $q(\cdot|\cdot)$. $q(\boldsymbol{y})$ afgør, om der forsøges tilføjelse/sletning eller flytning, og i tilfælde af førstnævnte, så afgør $p(\boldsymbol{y})$, om der forsøges tilføjelse eller sletning, dvs. at hvis der vælges konstanter $q = 1/3$ og $p = 1/2$, så er alle tre typer opdateringer lige sandsynlige. Hvad de konkrete valg bør være afhænger af den enkelte situation, i [Geyer & Møller, 1994] omtales de nævnte værdier som passende for molekylære systemer, men i artiklens eget eksempel viste andre valg sig at være mere hensigtsmæssige.

Hver type af opdatering har en tilknyttet forslagsfordeling og Hastingsforhold, der adskiller sig fra de tidligere anvendte. For en move-opdatering udvælges et punkt $y_i, i \sim \text{Unif}(\{1,\ldots,n(\boldsymbol{y})\})$ fra det aktuelle $\boldsymbol{y} = (y_1,\ldots,y_{n(\boldsymbol{y})})$, og der genereres et nyt punkt y' vha. forslagsfordelingen $q_m(y'|\boldsymbol{y})$. q_m er en tæthedsfunktion på B, eksempelvis en uniform tæthed på hele B eller i et rektangel omkring y_i. Det nye punkt y' indsættes i forslaget $\boldsymbol{y}' = (y_1,\ldots y_{i-1}, y', y_{i+1},\ldots,y_{n(\boldsymbol{y})})$, og

$$\text{Hastingsforholdet er givet ved } H_m(\boldsymbol{y},\boldsymbol{y}') \;=\; \frac{h(\boldsymbol{y}')\,q_m(y_i|\boldsymbol{y}')}{h(\boldsymbol{y})\,q_m(y'|\boldsymbol{y})}.$$

Bemærk at hvis q_m er en uniform tæthed på hele B, så er $q_m = \frac{1}{|B|}$, hvorved den kan forkortes ud i Hastingsforholdet, som derved bliver et Metropolisforhold. Ved en birth-opdatering tilføjes et nyt punkt y' ud fra $q_b(y'|\boldsymbol{y})$, og for en death-opdatering udvælges et punkt y' til sletning vha. den diskrete forslagsfordeling $q_d(y'|\boldsymbol{y})$. Hastingsforholdene for de to typer opdateringer er da givet ved

$$H_b(\boldsymbol{y},y') \;=\; \frac{h(\boldsymbol{y}\cup y')\,(1-p(\boldsymbol{y}))\,q_d(y'|\boldsymbol{y}\cup y')}{h(\boldsymbol{y})\,p(\boldsymbol{y})\,q_b(y'|\boldsymbol{y})} \quad \text{samt}$$

$$H_d(\boldsymbol{y},y') \;=\; \frac{h(\boldsymbol{y}\setminus y')\,p(\boldsymbol{y})\,q_b(y'|\boldsymbol{y}\setminus y')}{h(\boldsymbol{y})\,(1-p(\boldsymbol{y}))\,q_d(y'|\boldsymbol{y})}.$$

Acceptsandsynlighederne for en BDM-algoritme er som tidligere, dvs. $a_m(\boldsymbol{y},\boldsymbol{y}') = \min\{1, H_m(\boldsymbol{y},\boldsymbol{y}')\}$, $a_b(\boldsymbol{y},y') = \min\{1, H_b(\boldsymbol{y},\boldsymbol{y})\}$ og $a_d(\boldsymbol{y},y') = \min\{1, H_d(\boldsymbol{y},\boldsymbol{y})\}$. Algoritme 1.57 beskriver det samlede forløb af algoritmen.

Algoritme 1.57 (Metropolis-Hastings for punktprocesser (BDM)):
Lad $Y_i = \left(y_1, \ldots, y_{n(\boldsymbol{y})}\right)$ være den aktuelle tilstand i skridt i, hvor Y_0 er en valgt starttilstand, som afhænger af den enkelte situation. Lad yderligere

$$r_{bdm}, r_m, r_{bd}, r_b, r_d, \sim \text{Unif}(0,1) \text{ og } I \sim \text{Unif}\left(\{1, \ldots, n\left(\boldsymbol{y}\right)\}\right).$$

I hver iteration $i = 1, \ldots, N, N \in \mathbb{N}$, udføres følgende:

- Hvis $r_{bdm} < q\left(\boldsymbol{y}\right)$ gå i move, ellers gå i birth-death

  a) Move:
  - Fjern linjestykke y_I og tilføj nyt ud fra $q_m(\cdot|\cdot)$ til forslag $\boldsymbol{y}'$
  - Hvis $r_m < (H_m(\boldsymbol{y}, \boldsymbol{y}')$ sæt $Y_i = \boldsymbol{y}'$, ellers sæt $Y_i = Y_{i-1}$

  b) Birth-death:
  - Hvis $r_{bd} < p\left(\boldsymbol{y}\right)$ gå i birth, ellers gå i death

    b.1) Birth:
    - Generer linjestykke y' ud fra $q_b(\cdot|\cdot)$ som nyt forslag
    - Hvis $r_b < (H_b(\boldsymbol{y}, y')$ sæt $Y_i = \boldsymbol{y} \cup y'$, ellers sæt $Y_i = Y_{i-1}$

    b.2) Death:
    - Udtræk linjestykke y' ud fra $q_d(\cdot|\cdot)$ som forslag til sletning
    - Hvis $r_d < H_d(\boldsymbol{y}, y')$ sæt $Y_i = \boldsymbol{y} \setminus y'$, ellers sæt $Y_i = Y_{i-1}$

□

Som starttilstanden Y_0 kan eksempelvis anvendes en homogen PPP, da en sådan er simpel at simulere. To specialtilfælde af algoritme 1.57 fås ved valg af henholdsvis $q = 0$ og $q = 1$. Hvis $q = 0$ haves tilfældet, hvor der udelukkende laves birth- og death-opdateringer, mens der for $q = 1$ fås situationen, hvor der udelukkende foretages move-opdateringer. $q = 1$ svarer dermed til at simulere $f\left(\boldsymbol{y} \,|\, n\left(\boldsymbol{y}\right) = n\right), n \in \mathbb{N}$, dvs. der betinges på et bestemt antal punkter for de simulerede punktmønstre. Bemærk at tilstandsrummet reduceres ved valget af $q = 1$, fordi det ikke længere er muligt at ændre på antallet af punkter, og Markovkæden er dermed ikke længere irreducibel i forhold til hele tilstandsrummet.

De beskrevne algoritmer af typerne Metropolis-Hastings, Gibbs sampling og BDM er meget fleksible algoritmer, og det er muligt at sammensætte dem på mange måder. Således vil der i kapitel 3 blive gjort brug af en kombination af dem, da det viser sig nødvendigt at simulere både en punktproces og en simultan tæthed på samme tid.

Opstilling og implementation af model 2

I dette kapitel specificeres det, hvordan specialets model specifikt opstilles. Der vil først blive redegjort for, hvordan den er opbygget ud fra punktprocesser og forskellige fordelinger, og der vil herefter blive givet en mere detaljeret beskrivelse af, hvordan den teknisk kan implementeres. Den overordnede kode til at generere realisationer af modellen vil blive præsenteret, og der vil blive givet eksempler på punktmønstre, der er genereret af modellen. Endeligt vil der blive opstillet tætheder for modellen til senere brug til inferens for modellens parametre.

2.1 Model-specifikation

De grundlæggende punktprocesser, der blev præsenteret i kapitel 1, er i sig selv ikke velegnede til at repræsentere datasættet om gravhøje. Der er tydelige klyngetendenser i datasættet, og da en PPP har komplet rumlig tilfældighed, så er det ikke en anvendelig type punktproces. Cox punktprocesser kan repræsentere forskellige typer af klynger, men de omtalte Neyman-Scott punktprocesser kan ikke umiddelbart repræsentere de tydelige lineære strukturer, som datasættet indeholder. Derfor er der brug for en mere kompleks punktproces, og den foreslåede model er inspireret af Thomas punktprocessens brug af en normalfordeling til at forskyde punkter i forhold til hvert klyngecentrum, og den består af en kombination af PPP'er, der samlet giver en Cox punktproces.

Modellen består overordnet beskrevet af to punktprocesser $\boldsymbol{X}$ og $\boldsymbol{Y}$, hvor $\boldsymbol{Y}$ er en underliggende proces af linjestykker, som repræsenterer de antagede veje i datasættet, og en realisation af $\boldsymbol{Y}$ bruges som input til $\boldsymbol{X}$, der kommer til at indeholde de egentlige punkter. Det opnås ved at generere et antal punkter på hvert af linjestykkerne i realisationen af $\boldsymbol{Y}$, hvorefter punkterne forskydes i forhold til det linjestykke, punktet er blevet genereret på. Yderligere anvendes en punktpro-

ces til at indføre såkaldte baggrundspunkter, som ikke har nogen tilknytning til linjestykkerne, og endeligt foretages en superposition af alle de forskudte punkter samt baggrundspunkterne til den samlede realisation af $\boldsymbol{X}$.

$\boldsymbol{Y}$ er en mærket homogen PPP på et observationsvindue B med intensitet λ, hvor et punkt angiver midtpunktet m_i for linjestykket y_i, og mærkerne består af henholdsvis en vinkel v_i i forhold til vandret og en længde l_i af hvert linjestykke. Begge typer mærker er i.i.d., vinklerne er uniformt fordelte i intervallet $(0,\pi)$, og længderne er eksponentialfordelte med parameter β, dvs. $\frac{1}{\beta}$ er middelværdien af linjestykkernes længder. Samlet består realisationer af $\boldsymbol{Y}$ dermed af et antal linjestykker med kendt placering, orientering og længde.

På hvert linjestykke y_i i en realisation af $\boldsymbol{Y}$ anvendes uafhængige homogene PPP'er $\boldsymbol{X}_i$ med intensitet γ, hvilket resulterer i, at der genereres et antal punkter beliggende på hvert linjestykke. For hvert punkt x_j udtrækkes i.i.d. værdier d_j fra en normalfordeling med middelværdi 0 og spredning σ, og hvert x_j forskydes afstanden d_j vinkelret i forhold til linjestykket y_i, hvilket giver de endelige punkter til realisationen af $\boldsymbol{X}_i$. Forskydningen er en tilfældig uafhængig flytning, så $\boldsymbol{X}_i$ er jf. afsnit 1.3.3 stadigvæk en PPP. Ud over forskydelsen af punkterne, så anvendes en homogen PPP $\boldsymbol{X}_0$ med intensitet δ til at repræsentere baggrundspunkterne, og til sidst foretages en superposition, således at $\boldsymbol{X} = \bigcup_{i=0}^{n(\boldsymbol{y})} \boldsymbol{X}_i$, og det er jf. sætning 1.33 en PPP, fordi $\boldsymbol{X}_i$'erne er uafhængige. Da $\boldsymbol{X} \mid \boldsymbol{Y}$ dermed er en PPP, så bliver $\boldsymbol{X}$ en Cox punktproces drevet af $\boldsymbol{Y}$ jf. definition 1.46. I tabel 2.1 ses en sammenfattende oversigt over, hvordan modellen er opbygget.

Parametre: $\theta = (\lambda, \beta, \gamma, \delta, \sigma)$	
$\boldsymbol{Y}$	Punkter $m_i \sim \text{Poisson}(B,\lambda)$
	Vinkler $v_i \sim \text{Unif}(0,\pi)$
	Længder $l_i \sim \text{Exp}(\beta)$
$\boldsymbol{X}_0$	Baggrundspunkter $\boldsymbol{X}_0 \sim \text{Poisson}(B,\delta)$
$\boldsymbol{X}_i$	Punkter før forskydning: $x_j \sim \text{Poisson}(y_i,\gamma)$
	x_j forskydes vinkelret med $d_j \sim \text{N}(0,\sigma)$
Superposition: $\boldsymbol{X} = \bigcup_{i=0}^{n(\boldsymbol{y})} \boldsymbol{X}_i$	

Tabel 2.1 *– Oversigt over modellens opbygning.*

Bemærk at pga. anvendelsen af homogene PPP'er, ligefordelte vinkler og en normalfordeling, der er symmetrisk, så kan modellen simuleres på hele $\mathbb{R}^2$. Yderligere vil realisationer udvise den samme opførsel overalt på $\mathbb{R}^2$ for et givent sæt af parametre, så modellen er både stationær og isotropisk. Modellen har i alt fem forskellige parametre, der under ét vil blive betegnet med $\theta = (\lambda,\beta,\gamma,\delta,\sigma)$, og parameterrummet er $\mathbb{R}^5_+$. De to første parametre er knyttet til $\boldsymbol{Y}$, og de tre sidste parametre er knyttet til $\boldsymbol{X}$, så det kan undertiden være hensigtsmæssigt at anvende henholdsvis $\theta_{\boldsymbol{Y}} = (\lambda,\beta)$ og $\theta_{\boldsymbol{X}} = (\gamma,\delta,\sigma)$ frem for den samlede parameter θ.

2.2 Implementation af modellen

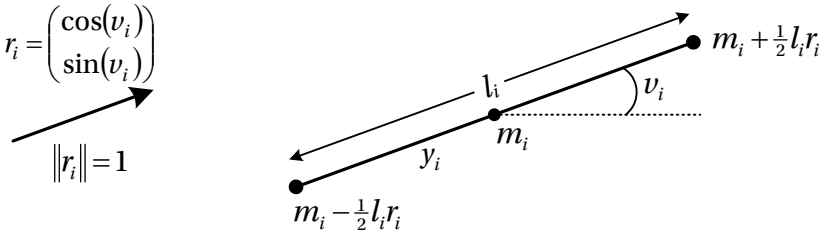
Der er en række forskellige tekniske aspekter, der må afklares, for at kunne lave en implementation af modellen i `R` ud fra ovenstående specifikation. Eksempelvis skal det afklares, hvordan forskydningen af punkter i forhold til linjestykkerne præcist skal fortolkes, samt hvordan mulige kanteffekter skal håndteres, og i dette afsnit gennemgås nogle af disse praktiske udfordringer.

2.2.1 Implementation af $\boldsymbol{Y}$

For at kunne generere $\boldsymbol{Y}$ skal der bruges $\theta_{\boldsymbol{Y}}$ samt et rektangulært observationsvindue $B \subseteq \mathbb{R}^2$, der angives vha. to vektorer, som indeholder henholdsvis minimum og maksimum for henholdsvis længden x og bredden y. Det vides fra afsnit 1.1.2, at intensiteten λ for en homogen punktproces svarer til det forventede antal punkter pr. enhedsareal, og pr. definition 1.28 fås det dermed, at antallet af punkter $n \sim \text{Poi}(\lambda|B|)$. Punkterne i B er uniformt fordelte givet antallet, og derved kan selve punkterne $m_i, i = 1,\ldots,n$, i $\boldsymbol{Y}$ genereres ved at udtrække en uniformt fordelt x- og y-koordinat som midtpunkt for hvert linjestykke. Til hvert m_i tilknyttes der efterfølgende henholdsvis vinkel v_i og længde l_i som mærker, og det hele gemmes i en matrix, hvor hver række indeholder x- og y-koordinater for punktet samt det tilknyttede mærke.

2.2.2 Implementation af $\boldsymbol{X}$

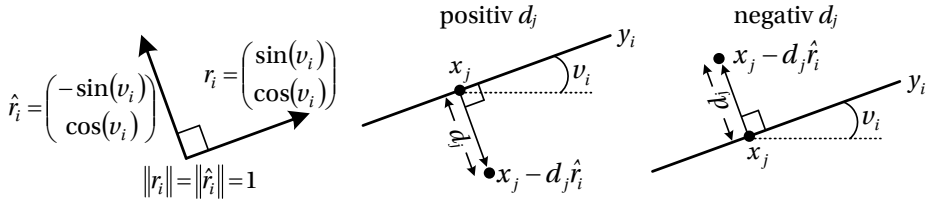
Til at generere en realisation af $\boldsymbol{X}$ benyttes ud over $\theta_{\boldsymbol{X}}$ også koordinater samt mærker i en realisation af $\boldsymbol{Y}$ til i første omgang at beregne koordinaterne til start- og slut-punktet af hvert linjestykke. Dette gøres ved at benytte, at $r_i = \begin{pmatrix}\cos(v_i)\\ \sin(v_i)\end{pmatrix}$ vil være en retningsvektor for linjestykket y_i med vinkel v_i og have enhedslængde. Ved at anvende stedvektoren for midtpunktet af linjestykket $m_i \pm \frac{1}{2} l_i r_i$, så vil henholdsvis slut- og start-koordinaterne kunne beregnes, se figur 2.2.



Figur 2.2 *– Illustration af beregningen af linjestykket y_i's start- og slut-koordinater ved brug af midtpunkt m_i, vinkel v_i, længde l_i samt retningsvektor r_i.*

På hvert linjestykke y_i skal der simuleres en homogen PPP $\boldsymbol{X}_i$ med intensitet γ, og som ved genereringen af $\boldsymbol{Y}$ vides, at antallet af punkter for det enkelte linjestykke $m \sim \text{Poi}(\gamma l_i)$. For hvert y_i skal de m genererede punkter placeres et sted uniformt på linjestykket, hvilket kan gøres ud fra de beregnede start- og slut-koordinater.

Herefter skal de genererede punkter $x_j, j = 1,\ldots,m$, forskydes i forhold til y_i. I specifikationen står, at dette gøres vinkelret i forhold til y_i, og det skal fortolkes på den måde, at hvis den genererede værdi fra $d_j \sim \text{N}(0,\sigma)$ er positiv, så skal punktet forskydes d_j langs retningsvektoren for y_i roteret 90° i negativ omløbsretning. Dvs. at for eksempelvis et lodret linjestykke y_i med $v_i = \frac{\pi}{2}$, da skal et punkt x_j forskydes d_j vandret til højre for $d_j > 0$ og vandret til venstre for $d_j < 0$. Da der arbejdes i $\mathbb{R}^2$, så kan tværvektoren $\widehat{r_i} = \begin{pmatrix} -\sin(v_i) \\ \cos(v_i) \end{pmatrix}$ benyttes til at foretage forskydningen. En tværvektor $\widehat{r_i}$ er roteret 90° i positiv omløbsretning i forhold til r_i, så den endelige placering af et punkt kan beregnes ved at gange størrelsen af forskydningen d_j med $-\widehat{r_i}$, se figur 2.3.



Figur 2.3 *– Illustration af, hvordan et punkt x_j på linjestykket y_i forskydes afstanden d_j vinkelret i forhold til y_i ved brug af tværvektoren $\widehat{r_i}$.*

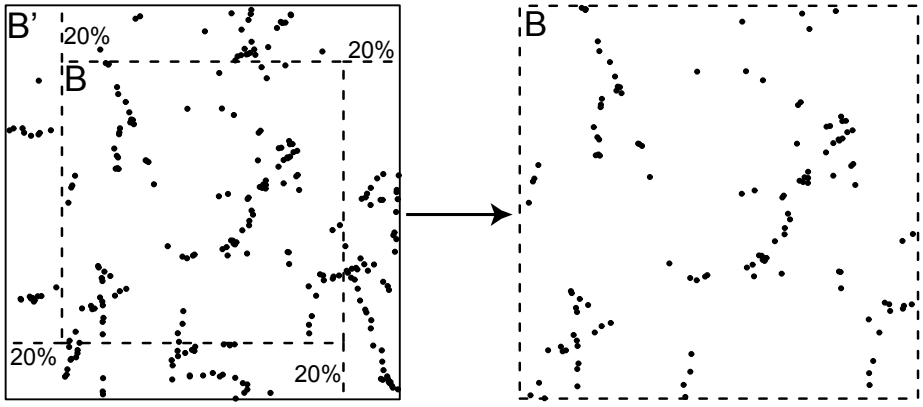
Endeligt tilføjes baggrundspunkterne til sidst ved at simulere $o \sim \text{Poi}(\delta|B|)$ punkter uniformt i B. De resulterende punkter x_j's x- og y-koordinater efter forskydningen samt baggrundspunkterne gemmes i en matrix med 2 søjler, hvor hver række angiver koordinaterne til ét punkt.

2.2.3 Håndtering af kanteffekter

Midtpunkterne m_i for linjestykkerne vil ligge indenfor observationsvinduet B, men på grund af linjestykkernes tilfældige længder og vinkler, så kan dele af det enkelte y_i komme til at ligge udenfor B. Ved genereringen af hver $\boldsymbol{X}_i$ kan der derfor opstå punkter, som befinder sig udenfor B, enten fordi punktet x_j er genereret på en del af y_i udenfor B, eller fordi det efter den afsluttende forskydning er blevet placeret udenfor B. Det er dermed nødvendigt at vælge en fremgangsmåde for, hvordan der skal tages hensyn til disse kanteffekter.

Det er valgt at håndtere kanteffekterne ved at udvide observationsvinduet, når $\boldsymbol{Y}$ genereres, hvilket så også medfører, at $\boldsymbol{X}$ genereres i et udvidet observationsvindue $B' \supset B$. Efterfølgende fjernes de punkter, som ligger i $B' \setminus B$, således at $\boldsymbol{X}$ kun indeholder punkter i B. Et argument for at anvende denne tilgang er, at modellen jf. afsnit 2.1 er stationær og isotropisk, og der vil dermed ikke være forskel på, om der simuleres på B eller B', men det forventes at reducere indflydelsen fra kanteffekter at benytte et forstørret observationsvindue under simulationen.

Konkret udvides B med 20% i hver side, således at B' har 40% større længde og bredde og 96% større areal end B. Efter genereringen af $\boldsymbol{X}$ på B' beskæres $\boldsymbol{X}$ ved at sammenligne med grænserne for B. $\boldsymbol{Y}$ vil fortsat indeholde alle data fra genereringen på B', men ved plot af linjestykkerne i $\boldsymbol{Y}$ fjernes de dele, som ligger udenfor B. På figur 2.4 ses en illustration af fremgangsmåden.



Figur 2.4 *– Illustration af håndteringen af kanteffekter, hvor der simuleres på et udvidet observationsvindue B', som efterfølgende beskæres til B.*

2.2.4 Overordnet kode

I R-kode 2.5 ses den overordnede kode, som benyttes til at lave en realisation af modellen. Den indeholder kald til de underliggende funktioner, der tilsammen laver $\boldsymbol{Y}$ og $\boldsymbol{X}$ som beskrevet ovenfor. Det kan bemærkes, at der også indgår en matrix Z, som indeholder informationer om linjestykkernes koordinater og antallet af punkter for hvert linjestykke, men den gemmes ikke, da det kun er midlertidige informationer, der er brug for under genereringen, og ikke en del af $\boldsymbol{Y}$ eller $\boldsymbol{X}$. Koden til den resterende implementation kan findes i bilag A.1, som indeholder koden til de funktioner, der kaldes fra henholdsvis `generer_Y` og `generer_X`.

```
1  generer_model <- function(lambda, beta, gamma, delta, sigma, xrange,
       yrange) { # master-funktion, der genererer en komplet realisation af modellen
2    Y <<- generer_Y(lambda, beta, xrange, yrange)
3    X <<- generer_X(Y, gamma, delta, sigma, xrange, yrange)
4  } # end function
5
6  generer_Y <- function(lambda, beta, xrange, yrange) { # generering af linjestykker
7    antal_veje <- generer_antal_veje(lambda, xrange, yrange)
8    Y <- generer_veje(antal_veje, xrange, yrange)
9    mærker <- generer_mærker(antal_veje, beta)
10   Y <- cbind(Y,mærker)
11 } # end function
12
13 generer_X <- function(Y, gamma, delta, sigma, xrange, yrange) { # punkterne
14   Z <- beregn_vej_start_slut(Y)
15   Z <- generer_punkter_pr_vej(Y, Z, gamma)
16   X <- generer_punkter(Y, Z)
17   X <- generer_forskydninger(X, Y, Z, sigma)
18   X <- generer_baggrundspunkter(X, delta, xrange, yrange)
19   X <- beskæring(X, xrange, yrange)
20 } # end function
```

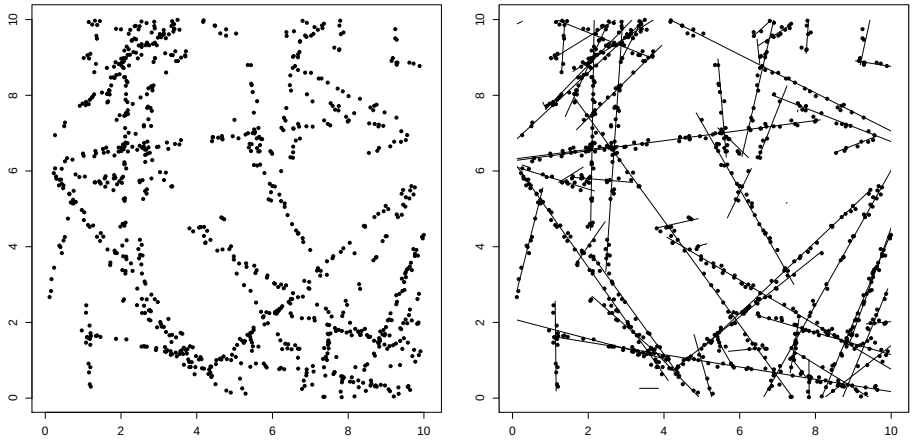
R-kode 2.5 *– `model-master.R`, overordnet kode til realisering af modellen.*

Der er anvendt en funktions-orienteret tilgang til implementationen, hvor hver fase i genereringen af en realisation af modellen er lavet i sin egen funktion, hvilket bidrager til at give et større overblik over koden. En anden fordel er, at det også giver en forholdsvis stor modularitet, hvor der nemt kan udskiftes dele af modellen, hvis der eksempelvis ønskes en anden type fordeling for længden af linjestykkerne eller for forskydningen af punkterne til sidst. Således er den valgte implementationsform med til at vise, at modellen indeholder en hvis fleksibilitet, fordi den giver mulighed for anvendelse af andre typer fordelinger end de her anvendte eksponentialfordeling og normalfordeling, uden at der ændres fundamentalt på modellens opbygning.

2.3 Eksempler på genererede punktmønstre

I dette afsnit gives der eksempler på forskellige punktmønstre, der er blevet genereret af modellen for at konstatere, om der som ønsket opnås punktmønstre med lineære strukturer. Faktisk er der allerede ovenfor i figur 2.4 anvendt konkret output fra modellen, men her vil der blive set mere detaljeret på to væsentligt forskellige punktmønstre for bl.a. at undersøge samspillet mellem parametrene nærmere.

Det første punktmønster ses på figur 2.6, hvor parametrene er valgt til at være $\theta = (\lambda,\beta,\gamma,\delta,\sigma) = (0{,}5\ ;\ 1/5\ ;\ 5\ ;\ 0\ ;\ 0{,}05)$ og $B = [0\ ;\ 10] \times [0\ ;\ 10]$. Bemærk at δ er valgt til at være 0 her, således at der ingen baggrundspunkter er, for at holde fokus på de lineære strukturer. På plottet til venstre ses der tydelige tendenser til, at punkterne ligger langs rette linjestykker, hvilket også er forventeligt pga. kombinationen af det lave σ og høje γ, som giver mange punkter, der ligger forholdsvis tæt på det underliggende linjestykke. Det synes også evident, at linjestykkernes orientering er vidt forskellig pga. den uniforme fordeling af deres vinkler, og at der er en forholdsvis stor variation i deres længder pga. eksponentialfordelingens varians.

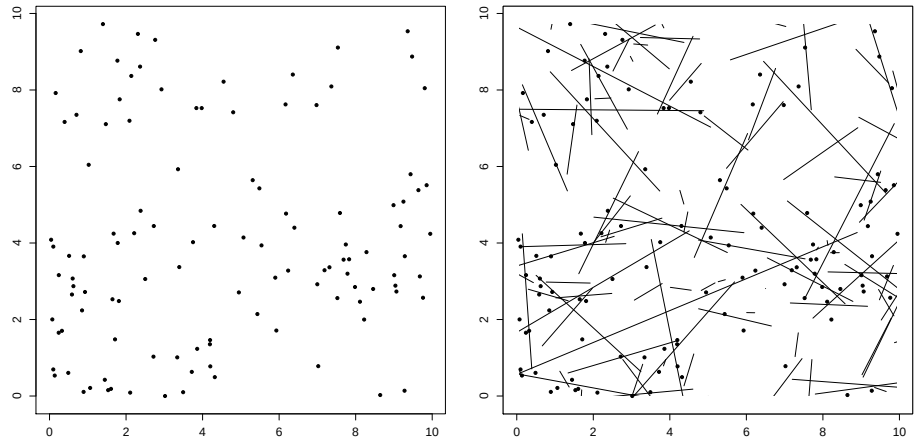


Figur 2.6 *– Til venstre et punktmønster* $\boldsymbol{x}$ *genereret ved kørsel af* `generer_model(0.5, 1/5, 5, 0, 0.05, c(0,10), c(0,10))`. *Til højre* $\boldsymbol{x}$ *sammen med det tilhørende* $\boldsymbol{y}$.

Disse forhold bekræftes ved at betragte plottet til højre, der både indeholder punktmønsteret og de underliggende linjestykker. Det antydes også, at håndteringen af kanteffekter spiller en aktiv rolle, da en del af linjestykkerne lader til at

slutte udenfor observationsvinduet, og der er et forholdsvist lille antal, som er komplet indeholdt i observationsvinduet B.

På figur 2.7 ses et andet punktmønster, hvor parametrene nu er $\theta = (\lambda,\beta,\gamma,\delta,\sigma) = (0{,}8\,;\,1/3\,;\,0{,}5\,;\,0\,;\,0{,}15)$ og $B = [0\,;\,10] \times [0\,;\,10]$. Det ses at være væsentligt forskelligt fra det tidligere og dermed medvirke til at understrege modellens fleksibilitet til at kunne frembringe forskellige typer af punktmønstre. Der er i dette eksempel langt færre punkter, og de lineære strukturer fornemmes kun svagt. Dette skyldes bl.a., at der er et lavere γ end før, så der er kun 0,5 punkter pr. enhedslængde. Når der yderligere er gennemsnitligt kortere længder af linjestykkerne pga. et større β, og punkterne forskydes længere væk pga. højere σ, så er det ikke overraskende, at punktmønsteret i dette tilfælde er mindre kompakt og ikke udviser så visuelt tydelige lineære tendenser.



Figur 2.7 *– Til venstre et punktmønster* $\boldsymbol{x}$ *genereret ved kørsel af* `generer_model(0.8, 1/3, 0.5, 0, 0.15, c(0,10), c(0,10))`. *Til højre* $\boldsymbol{x}$ *sammen med det tilhørende* $\boldsymbol{y}$.

Til højre, hvor linjestykkerne også er sat på plottet, ses det, at der faktisk er genereret en del flere linjestykker end før, hvilket skyldes, at λ er forøget. Alligevel er der dog forholdsvis få punkter pga. det lave γ, så resultatet er, at der slet ikke er genereret nogle punkter på en del af linjestykkerne. En interessant detalje er, at det kunne se ud til, at flere af de lineære strukturer, som synes at fremgå af plottet til venstre, i virkeligheden er opstået tilfældigt, idet punkterne lader til at være genereret på flere forskellige linjestykker.

2.4 Tætheder for modellen

For at kunne lave inferens for θ, så er det nødvendigt at bestemme udtryk for modellens tætheder for henholdsvis $\boldsymbol{Y}|\theta_{\boldsymbol{Y}}$ og $\boldsymbol{X}|\boldsymbol{Y},\theta_{\boldsymbol{X}}$. De kan benyttes til at bestemme den simultane tæthed for $\boldsymbol{X},\boldsymbol{Y}|\theta$, som vil indeholde alle modellens parametre og derfor kunne anvendes til inferens i næste kapitel. Til at gøre dette kan resultaterne fra korollar 1.40 og sætning 1.42 benyttes til at angive tæthederne med hensyn til Poisson(S,1).

$\boldsymbol{Y}$ er en mærket PPP med konstant intensitet λ på et observationsvindue B og mærker $l_i \sim \text{Exp}(\beta)$ og $v_i \sim \text{Unif}(0,\pi)$. Dermed bliver tætheden for $\boldsymbol{Y}$

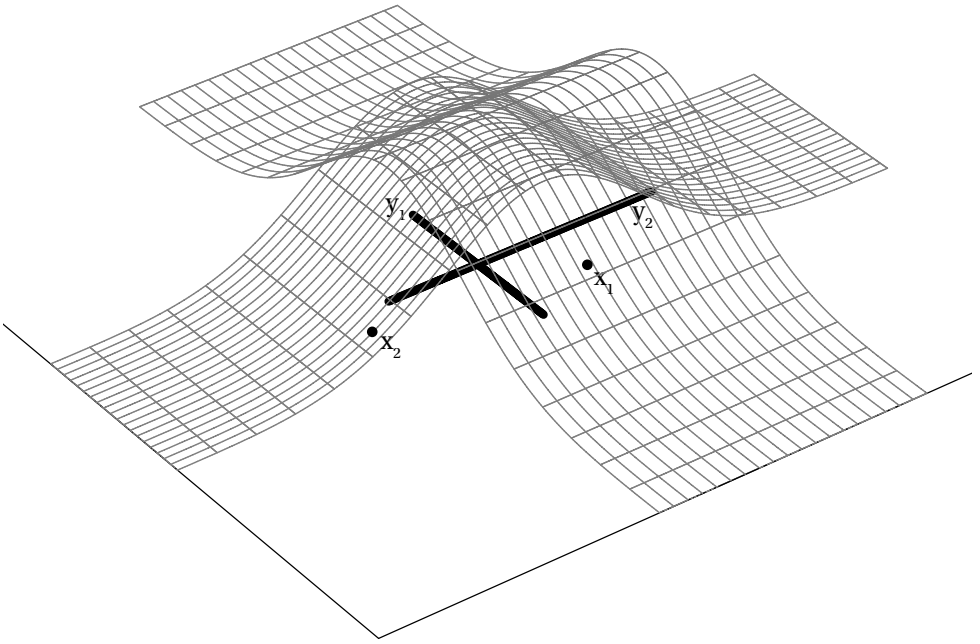
$$\begin{aligned} p\left(\boldsymbol{Y}|\lambda,\beta\right) &= \exp\left(|B|-\mu(B)\right)\prod_{u\in\boldsymbol{y}}\left[\rho(u)\right] \\ &= \exp\left(|B|-\lambda|B|\right)\prod_{i=1}^{n(\boldsymbol{y})}\left[\lambda\cdot\beta\exp\left(-\beta l_i\right)\cdot\tfrac{1}{\pi}\right] \\ &= \exp\left(|B|-\lambda|B|\right)\left(\frac{\lambda\beta}{\pi}\right)^{n(\boldsymbol{y})}\exp\left(-\beta\sum_{i=1}^{n(\boldsymbol{y})}\left[l_i\right]\right) \end{aligned} \tag{2.1}$$

mht. Poisson(B,1). $\boldsymbol{X}|\boldsymbol{Y}$ er en PPP, der pga. af forskydningen af punkter væk fra linjestykkerne bliver inhomogen, og dens intensitet $\rho(\boldsymbol{x})$ afhænger ud over af γ og δ også af de normalfordelinger, der er placeret langs hvert linjestykke y_j. Hver af disse vil bidrage til intensiteten afhængigt af, om punktet x_i ligger placeret, så der eksisterer en vinkelret afstand til hver y_j. Dette er illustreret på figur 2.8, hvor punktet x_1 bidrager til intensiteten i forhold til normalfordelingen hørende til henholdsvis y_1 og y_2, mens x_2 kun bidrager til intensiteten i forhold til normalfordelingen hørende til y_1. Bemærk at normalfordelingen hørende til y_2 har større volumen end normalfordelingen hørende til y_1, da y_2 er længere end y_1 og dermed bør have flere punkter tilknyttet end y_1.

Den betingede tæthed for $\boldsymbol{X}|\boldsymbol{Y}$ mht. Poisson(S,1) bliver således

$$\begin{aligned} p(\boldsymbol{X}|\boldsymbol{Y},\gamma,\delta,\sigma) &= \exp\left(|B|-\mu(B)\right)\prod_{i=1}^{n(\boldsymbol{x})}\left[\rho(x_i)\right], \text{ hvor} \\ \rho\left(x_i\right) &= \gamma\sum_{j=1}^{n(\boldsymbol{y})}\left[\rho_{y_j}\left(x_i\right)\right]+\delta \quad \text{og} \quad \rho_{y_j}\left(x_i\right) = \frac{1}{\sqrt{2\pi}\sigma}\exp\left(-\frac{d_{i,j}^2}{2\sigma^2}\right), \end{aligned} \tag{2.2}$$

og hvor $d_{i,j}$ er afstanden fra punktet x_i til linjestykket y_j. Hvis en vinkelret afstand ikke eksisterer, så sættes $\rho_{y_j}(x_i)=0$, fordi der ikke kommer noget bidrag



Figur 2.8 *– Illustration af to linjestykker y_1 og y_2 med tilhørende normalfordelinger samt to punkter x_1 og x_2.*

fra pågældende linjestykkes normalfordeling. Regneteknisk kan dette gøres ved at sætte $d_{i,j} = \infty$, da ρ_{y_j} så bliver 0. Pr. definition af betinget sandsynlighed er $p(\boldsymbol{X},\boldsymbol{Y}\,|\,\theta) = p(\boldsymbol{Y}\,|\,\lambda,\beta)\cdot p(\boldsymbol{X}\,|\,\boldsymbol{Y},\gamma,\sigma)$, så det fås fra (2.2) og (2.1), at

$$\begin{aligned} p(\boldsymbol{X},\boldsymbol{Y}\,|\,\theta) \;=\; & \exp\left(|B|-\lambda|B|\right)\left(\frac{\lambda\beta}{\pi}\right)^{n(\boldsymbol{y})}\exp\left(-\beta\sum_{i=1}^{n(\boldsymbol{y})}\left[l_i\right]\right)\cdot \\ & \exp\left(|B|-\mu(B)\right)\prod_{i=1}^{n(\boldsymbol{x})}\left[\gamma\sum_{j=1}^{n(\boldsymbol{y})}\left[\frac{1}{\sqrt{2\pi}\sigma}\exp\left(-\frac{d_{i,j}^2}{2\sigma^2}\right)\right]+\delta\right], \end{aligned}$$

der kan omskrives til

$$\exp\left((2-\lambda)|B|-\beta\sum_{i=1}^{n(\boldsymbol{y})}\left[l_i\right]-\mu(B)\right)\left(\frac{\lambda\beta}{\pi}\right)^{n(\boldsymbol{y})}\prod_{i=1}^{n(\boldsymbol{x})}\left[\gamma\sum_{j=1}^{n(\boldsymbol{y})}\left[\frac{1}{\sqrt{2\pi}\sigma}\exp\left(-\frac{d_{i,j}^2}{2\sigma^2}\right)\right]+\delta\right]. \tag{2.3}$$

Inferens for modellens parametre 3

I dette kapitel er formålet at udføre inferens for at bestemme værdier af modellens parametre, der kan frembringe et punktmønster med samme egenskaber som datasættets. Til det formål konstrueres en algoritme, som overordnet består af to dele i form af henholdsvis simulering af linjestykker og estimation af parametre. Datasættet indeholder kun gravhøjenes placering, der er ingen information om de forhistoriske veje, som de menes at have ligget langs, dvs. i denne sammenhæng er $\boldsymbol{X}$ kendt, men indholdet af $\boldsymbol{Y}$ er ukendt.

For at gøre det muligt at estimere parametrene i den simultane model $\boldsymbol{X},\boldsymbol{Y}$ er det derfor nødvendigt at simulere linjestykkerne i $\boldsymbol{Y}$, og det gøres ved brug af en BDM-algoritme som beskrevet i afsnit 1.5.4. Ud fra simulationen af $\boldsymbol{Y}$ kan der derefter laves estimation af parametrene i den simultane model ved at anvende almindelig Metropolis-Hastings. Dette gøres konkret ved i hver iteration af algoritmen først at opdatere indholdet i $\boldsymbol{Y}$ og derefter opdatere parametrene på baggrund af den aktuelle tilstand af $\boldsymbol{Y}$.

Da der er valgt at anvende en bayesiansk tilgang, så skal der til konstruktion af algoritmen ud over likelihoods også opstilles priors og posteriors. Dette foregår i det efterfølgende afsnit, og resultaterne derfra vil blive anvendt til at bestemme de Hastingsforhold, som anvendes til opdatering af parametre og linjestykker, således at en komplet algoritme kan specificeres. Dernæst vil der blive gennemgået en række tekniske aspekter vedr. implementationen af algoritmen, inden kapitlet afsluttes med en præsentation af resultater af anvendelse af algoritmen på henholdsvis simulerede data og på datasættet om gravhøje.

3.1 Likelihood og posterior

For at estimere θ skal tætheden i (2.3) betragtes som en likelihood $L(\theta \mid \boldsymbol{X},\boldsymbol{Y})$, men for at kunne anvende den til konkrete beregninger, så skal der vælges en måde at approksimere $\mu(B)$ på.

3.1.1 Approksimation af $\mu(B)$

Idet $\boldsymbol{X}\,|\,\boldsymbol{Y}$ er en PPP, så er $\mu(B) = \int_B \rho(z)\,\mathrm{d}z$, men dette integrale kan ikke umiddelbart udregnes pga. den komplicerede ρ-funktion, så der må anvendes en approksimation.

$\int_B \rho(z)\,\mathrm{d}z$ er den samlede volumen over B af de normalfordelinger, der er tilknyttet linjestykkerne y_i, samt baggrundsintensiteten δ. Da normalfordelingerne er tætheder, så integrerer deres areal til $1\cdot\gamma$, da de er skalerede med γ, og deres volumen over $\mathbb{R}^2$ må derfor være $\gamma\cdot l_i$, hvilket kan bruges til en approksimation. Denne vil være større end det egentlige volumen over B, men størstedelen af volumen vil være koncentreret tæt på linjestykkerne i B, så det vurderes som værende en anvendelig approksimation. Ud over dette, så bidrager δ med $\delta|B|$ til volumen, og der fås samlet approksimationen

$$\mu(B) \;=\; \int_B \rho(z)\,\mathrm{d}z \;\approx\; \sum_{i=1}^{n(\boldsymbol{y})} \Big[\gamma l_i\Big] + \delta|B| \;=\; \gamma \sum_{i=1}^{n(\boldsymbol{y})} \Big[l_i\Big] + \delta|B|. \tag{3.1}$$

3.1.2 Simultan likelihood og posterior

På baggrund af (3.1) samt tæthederne fra afsnit 2.4 fås det, at likelihoodfunktionen for parametrene bliver

$$\begin{aligned} L(\theta\,|\,\boldsymbol{X},\boldsymbol{Y}) \;=\; & \exp\left((2-\lambda)|B| - \beta \sum_{i=1}^{n(\boldsymbol{y})} \Big[l_i\Big] - \gamma \sum_{i=1}^{n(\boldsymbol{y})} \Big[l_i\Big] - \delta|B|\right)\left(\frac{\lambda\beta}{\pi}\right)^{n(\boldsymbol{y})} \cdot \\ & \prod_{i=1}^{n(\boldsymbol{x})} \left[\gamma \sum_{j=1}^{n(\boldsymbol{y})} \left[\frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{d_{i,j}^2}{2\sigma^2}\right)\right] + \delta\right]. \end{aligned}$$

Som priors anvendes for hver parameter en gammafordeling $\Gamma(\alpha,\kappa)$, hvor $\alpha > 0$ er en formparameter og $\kappa > 0$ en skalaparameter. Anvendelsen af gammafordelte frem for normalfordelte priors har bl.a. den fordel, at de kun er defineret for positive tal på samme vis som parametrene, og der er også mulighed for at tilpasse spredningen efter behov som ved en normalfordeling. Det præciseres nærmere i afsnit 3.3.1, hvordan disse priors anvendes, og hvordan de i de efterfølgende kørsler af algoritmen vælges for det enkelte tilfælde. Således indføres det, at

$$\begin{gathered} \lambda \sim \Gamma(\lambda_\alpha,\lambda_\kappa) \quad , \quad \beta \sim \Gamma\left(\beta_\alpha\,\beta_\kappa\right) \quad , \quad \gamma \sim \Gamma\left(\gamma_\alpha,\gamma_\kappa\right), \\ \delta \sim \Gamma(\delta_\alpha,\delta_\kappa) \quad , \quad \sigma \sim \Gamma(\sigma_\alpha,\sigma_\kappa), \end{gathered}$$

og da der er uafhængighed mellem parametrene, så er

$$p(\theta) \;=\; p(\lambda)\,p\left(\beta\right)\,p\left(\gamma\right)\,p(\delta)\,p(\sigma).$$

Ved at sætte $l(\boldsymbol{y}) = \sum_{i=1}^{n(\boldsymbol{y})} \left[l_i \right]$ fås derved en simultan posterior

$$p(\theta \mid \boldsymbol{X}, \boldsymbol{Y}) \propto p(\theta) \cdot L(\theta \mid \boldsymbol{X}, \boldsymbol{Y}) \propto$$
$$p(\theta) \exp\left(-(\lambda+\delta)|B| - (\beta+\gamma) l(\boldsymbol{y})\right) (\lambda\beta)^{n(\boldsymbol{y})} \prod_{i=1}^{n(\boldsymbol{x})} \left[\gamma \sum_{j=1}^{n(\boldsymbol{y})} \left[\frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{d_{i,j}^2}{2\sigma^2} \right) \right] + \delta \right], \tag{3.2}$$

hvor konstanter nu er reduceret væk pga. den bayesianske tilgang.

3.1.3 Betingede posteriors

Til brug for estimering af parametrene i θ skal de betingede posteriors bestemmes, hvor der betinges på de øvrige parametre samt data. Det gøres på baggrund af den simultane posterior i (3.2), og det fås, at

$$p(\lambda \mid \boldsymbol{X}, \boldsymbol{Y}, \beta, \gamma, \delta, \sigma) \propto p(\lambda) \exp(-\lambda|B|) \lambda^{n(\boldsymbol{y})},$$

$$p(\beta \mid \boldsymbol{X}, \boldsymbol{Y}, \lambda, \gamma, \delta, \sigma) \propto p(\beta) \exp\left(-\beta l(\boldsymbol{y})\right) \beta^{n(\boldsymbol{y})},$$

$$p(\gamma \mid \boldsymbol{X}, \boldsymbol{Y}, \lambda, \beta, \delta, \sigma) \propto p(\gamma) \exp\left(-\gamma l(\boldsymbol{y})\right) \prod_{i=1}^{n(\boldsymbol{x})} \left[\gamma \sum_{j=1}^{n(\boldsymbol{y})} \left[\frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{d_{i,j}^2}{2\sigma^2} \right) \right] + \delta \right],$$

$$p(\delta \mid \boldsymbol{X}, \boldsymbol{Y}, \lambda, \beta, \gamma, \sigma) \propto p(\delta) \exp(-\delta|B|) \prod_{i=1}^{n(\boldsymbol{x})} \left[\gamma \sum_{j=1}^{n(\boldsymbol{y})} \left[\frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{d_{i,j}^2}{2\sigma^2} \right) \right] + \delta \right],$$

$$p(\sigma \mid \boldsymbol{X}, \boldsymbol{Y}, \lambda, \beta, \delta, \gamma) \propto p(\sigma) \prod_{i=1}^{n(\boldsymbol{x})} \left[\gamma \sum_{j=1}^{n(\boldsymbol{y})} \left[\frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{d_{i,j}^2}{2\sigma^2} \right) \right] + \delta \right],$$

idet udtryk, der kun afhænger af de givne parametre, kan reduceres væk på grund af proportionaliteten.

3.2 Opbygning af algoritmen

Algoritmen opbygges som en Gibbs sampler, således at hver parameter opdateres uafhængigt af de øvrige parametre ud fra de betingede posteriors, og selve

opdateringen foretages vha. Metropolis-Hastings, dvs. der er tale om Metropolis-i-Gibbs. Til opdatering af hver af parametrene skal der anvendes en (evt. unormaliseret) tæthed $h(\cdot)$ jf. afsnit 1.5.2, og hertil kan de betingede posteriors i ovenstående afsnit anvendes, så derved bliver

$$h_\lambda(\lambda) = p(\lambda|\cdot) \quad , \quad h_\beta(\beta) = p(\beta|\cdot) \quad , \quad h_\gamma(\gamma) = p(\gamma|\cdot),$$
$$h_\delta(\delta) = p(\delta|\cdot) \quad , \quad h_\sigma(\sigma) = p(\sigma|\cdot).$$

3.2.1 Hastingsforhold for parametrene

Til estimation af parametrene vælges en normalfordeling som forslagstætheden $q(\cdot|\cdot)$, således at forslaget φ' udtrækkes fra en normalfordeling med den aktuelle værdi af φ som middelværdi og en spredning ψ. Dette valg af forslagstæthed er symmetrisk i φ og φ', og dermed reduceres Hastingsforholdet

$$H(\varphi,\varphi') = \frac{h(\varphi')\,q(\varphi|\varphi')}{h(\varphi)\,q(\varphi'|\varphi)} \quad \text{til Metropolisforholdet} \quad H(\varphi,\varphi') = \frac{h(\varphi')}{h(\varphi)}.$$

Derved fås for de fem parametre, at

$$H(\lambda,\lambda') = \frac{h_\lambda(\lambda')}{h_\lambda(\lambda)} \quad , \quad H(\beta,\beta') = \frac{h_\beta(\beta')}{h_\beta(\beta)} \quad , \quad H(\gamma,\gamma') = \frac{h_\gamma(\gamma')}{h_\gamma(\gamma)},$$
$$H(\delta,\delta') = \frac{h_\delta(\delta')}{h_\delta(\delta)} \quad \text{samt at} \quad H(\sigma,\sigma') = \frac{h_\sigma(\sigma')}{h_\sigma(\sigma)}.$$

Det er nødvendigt at anvende logaritmen af disse for at få konkret anvendelige udtryk til brug i `R`, da de ovenstående udtryk giver numeriske problemer. Det skyldes især produktet i de betingede posteriors for γ, δ og σ, idet det vil blive uhåndterligt stort allerede ved forholdsvis små datasæt, hvis der ikke tages logaritmen til udtrykkene. De Hastingsforhold, som anvendes i praksis, bliver derfor

$$\begin{aligned}
\log\big(H(\lambda,\lambda')\big) &= \log\left(\frac{h_\lambda(\lambda')}{h_\lambda(\lambda)}\right) = \log\left(\frac{p(\lambda')\exp\big(-\lambda'|B|\big)\,(\lambda')^{n(\boldsymbol{y})}}{p(\lambda)\exp\big(-\lambda|B|\big)\,\lambda^{n(\boldsymbol{y})}}\right) \\
&= \log\left(\frac{p(\lambda')}{p(\lambda)}\right) + (\lambda-\lambda')\,|B| + n(\boldsymbol{y})\log\left(\frac{\lambda'}{\lambda}\right), \\
\log\big(H(\beta,\beta')\big) &= \log\left(\frac{h_\beta(\beta')}{h_\beta(\beta)}\right) = \log\left(\frac{p(\beta')\exp\big(-\beta' l(\boldsymbol{y})\big)\,(\beta')^{n(\boldsymbol{y})}}{p(\beta)\exp\big(-\beta l(\boldsymbol{y})\big)\,\beta^{n(\boldsymbol{y})}}\right) \\
&= \log\left(\frac{p(\beta')}{p(\beta)}\right) + (\beta-\beta')\,l(\boldsymbol{y}) + n(\boldsymbol{y})\log\left(\frac{\beta'}{\beta}\right), \qquad (3.3)
\end{aligned}$$

$$\log\left(H\left(\gamma,\gamma'\right)\right) = \log\left(\frac{h_\gamma(\gamma')}{h_\gamma(\gamma)}\right)$$

$$= \log\left(\frac{p\left(\gamma'\right)\exp\left(-\gamma' l\left(\boldsymbol{y}\right)\right)\prod_{i=1}^{n(\boldsymbol{x})}\left[\gamma'\sum_{j=1}^{n(\boldsymbol{y})}\left[\frac{1}{\sqrt{2\pi}\sigma}\exp\left(-\frac{d_{i,j}^2}{2\sigma^2}\right)\right]+\delta\right]}{p\left(\gamma\right)\exp\left(-\gamma l\left(\boldsymbol{y}\right)\right)\prod_{i=1}^{n(\boldsymbol{x})}\left[\gamma\sum_{j=1}^{n(\boldsymbol{y})}\left[\frac{1}{\sqrt{2\pi}\sigma}\exp\left(-\frac{d_{i,j}^2}{2\sigma^2}\right)\right]+\delta\right]}\right)$$

$$= \log\left(\frac{p\left(\gamma'\right)}{p\left(\gamma\right)}\right)+\left(\gamma-\gamma'\right)l\left(\boldsymbol{y}\right)+\rho\left(\boldsymbol{y},\gamma',\delta,\sigma\right)-\rho\left(\boldsymbol{y},\gamma,\delta,\sigma\right), \tag{3.4}$$

$$\log\left(H\left(\delta,\delta'\right)\right) = \log\left(\frac{h_\delta(\delta')}{h_\delta(\delta)}\right)$$

$$= \log\left(\frac{p\left(\delta'\right)\exp\left(-\delta'|B|\right)\prod_{i=1}^{n(\boldsymbol{x})}\left[\gamma'\sum_{j=1}^{n(\boldsymbol{y})}\left[\frac{1}{\sqrt{2\pi}\sigma}\exp\left(-\frac{d_{i,j}^2}{2\sigma^2}\right)\right]+\delta'\right]}{p\left(\delta\right)\exp\left(-\delta|B|\right)\prod_{i=1}^{n(\boldsymbol{x})}\left[\gamma\sum_{j=1}^{n(\boldsymbol{y})}\left[\frac{1}{\sqrt{2\pi}\sigma}\exp\left(-\frac{d_{i,j}^2}{2\sigma^2}\right)\right]+\delta\right]}\right)$$

$$= \log\left(\frac{p\left(\delta'\right)}{p\left(\delta\right)}\right)+\left(\delta-\delta'\right)|B|+\rho\left(\boldsymbol{y},\gamma,\delta',\sigma\right)-\rho\left(\boldsymbol{y},\gamma,\delta,\sigma\right) \text{ samt}$$

$$\log\left(H\left(\sigma,\sigma'\right)\right) = \log\left(\frac{h_\sigma(\sigma')}{h_\sigma(\sigma)}\right)$$

$$= \log\left(\frac{p\left(\sigma'\right)\prod_{i=1}^{n(\boldsymbol{x})}\left[\gamma\sum_{j=1}^{n(\boldsymbol{y})}\left[\frac{1}{\sqrt{2\pi}\sigma'}\exp\left(-\frac{d_{i,j}^2}{2(\sigma')^2}\right)\right]+\delta\right]}{p\left(\sigma\right)\prod_{i=1}^{n(\boldsymbol{x})}\left[\gamma\sum_{j=1}^{n(\boldsymbol{y})}\left[\frac{1}{\sqrt{2\pi}\sigma}\exp\left(-\frac{d_{i,j}^2}{2\sigma^2}\right)\right]+\delta\right]}\right)$$

$$= \log\left(\frac{p\left(\sigma'\right)}{p\left(\sigma\right)}\right)+\rho\left(\boldsymbol{y},\gamma,\delta,\sigma'\right)-\rho\left(\boldsymbol{y},\gamma,\delta,\sigma\right), \text{ hvor}$$

$$\rho(\boldsymbol{y},\gamma,\delta,\sigma) = \sum_{i=1}^{n(\boldsymbol{x})}\left[\log\left(\gamma\sum_{j=1}^{n(\boldsymbol{y})}\left[\frac{1}{\sqrt{2\pi}\sigma}\exp\left(-\frac{d_{i,j}^2}{2\sigma^2}\right)\right]+\delta\right)\right]. \tag{3.5}$$

3.2.2 Hastingsforhold for BDM-delen

Til BDM-delen af algoritmen kan (3.2) igen benyttes som den unormaliserede $h(\cdot)$. Denne gang kan faktorer, der er uafhængige af linjestykkerne, forkortes ud, og dermed er

$$h_l(\boldsymbol{y}) \;=\; \exp\left(-\left(\beta+\gamma\right)l\left(\boldsymbol{y}\right)\right)\left(\lambda\beta\right)^{n(\boldsymbol{y})}\prod_{i=1}^{n(\boldsymbol{x})}\left[\gamma\sum_{j=1}^{n(\boldsymbol{y})}\left[\frac{1}{\sqrt{2\pi}\sigma}\exp\left(-\frac{d_{i,j}^2}{2\sigma^2}\right)\right]+\delta\right].$$

Som forslagstæthed q_m for flytning af et linjestykke vælges en ligefordeling på hele observationsvinduet B, dvs. $q_m = \frac{1}{|B|}$. Ud over at selve midtpunktet for linjestykket flyttes, så genereres også nye mærker i form af længde og vinkel som specificeret i modellen i forrige kapitel, og Hastingsforholdet bliver også her reduceret til et Metropolisforhold pga. den valgte forslagstæthed, idet

$$H_m(\boldsymbol{y},\boldsymbol{y}') \;=\; \frac{h_l(\boldsymbol{y}')q_m(\cdot\,|\,\cdot)}{h_l(\boldsymbol{y})q_m(\cdot\,|\,\cdot)} \;=\; \frac{h_l(\boldsymbol{y}')\frac{1}{|B|}}{h_l(\boldsymbol{y})\frac{1}{|B|}} \;=\; \frac{h_l(\boldsymbol{y}')}{h_l(\boldsymbol{y})}.$$

Som før skal logaritmen til forholdet bestemmes, og

$$\log\left(H_m(\boldsymbol{y},\boldsymbol{y}')\right) \;=\; \log\left(\frac{h_l(\boldsymbol{y}')}{h_l(\boldsymbol{y})}\right)$$

$$= \log\left(\frac{\exp\left(-\left(\beta+\gamma\right)l\left(\boldsymbol{y}'\right)\right)\left(\lambda\beta\right)^{n(\boldsymbol{y}')}\prod_{i=1}^{n(\boldsymbol{x})}\left[\gamma\sum_{j=1}^{n(\boldsymbol{y}')}\left[\frac{1}{\sqrt{2\pi}\sigma}\exp\left(-\frac{d_{i,j}^2}{2\sigma^2}\right)\right]+\delta\right]}{\exp\left(-\left(\beta+\gamma\right)l\left(\boldsymbol{y}\right)\right)\left(\lambda\beta\right)^{n(\boldsymbol{y})}\prod_{i=1}^{n(\boldsymbol{x})}\left[\gamma\sum_{j=1}^{n(\boldsymbol{y})}\left[\frac{1}{\sqrt{2\pi}\sigma}\exp\left(-\frac{d_{i,j}^2}{2\sigma^2}\right)\right]+\delta\right]}\right)$$

$$= \left(\beta+\gamma\right)\left(l(\boldsymbol{y})-l(\boldsymbol{y}')\right)+\rho\left(\boldsymbol{y}',\gamma,\delta,\sigma\right)-\rho\left(\boldsymbol{y},\gamma,\delta,\sigma\right),$$

idet $\log\left((\lambda\beta)^{n(\boldsymbol{y}')}/(\lambda\beta)^{n(\boldsymbol{y})}\right)=0$, da $n\left(\boldsymbol{y}'\right)=n\left(\boldsymbol{y}\right)$ for move-opdateringer.

Til at afgøre, om algoritmen skal forsøge at flytte et linjestykke med en move-opdatering, eller om der skal påbegyndes birth-death, bruges en konstant sandsynlighed q, og hvis der påbegyndes birth-death, så anvendes en konstant sandsynlighed p til at afgøre, om der skal forsøges birth eller death. For tilføjelse af et nyt linjestykke y' vælges midtpunktet som ved flytning uniformt i $|B|$, så $q_b = \frac{1}{|B|}$, mens der ved fjernelse af et linjestykke y' vælges uniformt mellem de eksisterende, dvs. $q_d(\boldsymbol{y}) = \frac{1}{n(\boldsymbol{y})}$. Det fås så jf. afsnit 1.5.4, at

$$H_b(\boldsymbol{y},y') = \frac{h_l(\boldsymbol{y}\cup y')(1-p)\frac{1}{n(\boldsymbol{y})+1}}{h_l(\boldsymbol{y})p\frac{1}{|B|}} = \frac{h_l(\boldsymbol{y}\cup y')(1-p)|B|}{h_l(\boldsymbol{y})p(n(\boldsymbol{y})+1)}, \text{ og at}$$

$$H_d(\boldsymbol{y},y') = \frac{h_l(\boldsymbol{y}\setminus y')p\frac{1}{|B|}}{h_l(\boldsymbol{y})(1-p)\frac{1}{n(\boldsymbol{y})}} = \frac{h_l(\boldsymbol{y}\setminus y')pn(\boldsymbol{y})}{h_l(\boldsymbol{y})(1-p)|B|}.$$

Logaritmen af disse to Hastingsforhold bliver henholdsvis

$$\log\left(H_b\left(\boldsymbol{y},y'\right)\right) = \log\left(\frac{h_l(\boldsymbol{y}\cup y')(1-p)|B|}{h_l(\boldsymbol{y})p(n(\boldsymbol{y})+1)}\right)$$

$$= \log\left(\frac{e^{-(\beta+\gamma)l(\boldsymbol{y}\cup y')}(\lambda\beta)^{n(\boldsymbol{y})+1}\prod_{i=1}^{n(\boldsymbol{x})}\left[\gamma\sum_{j=1}^{n(\boldsymbol{y})+1}\left[\frac{1}{\sqrt{2\pi}\sigma}\exp\left(-\frac{d_{i,j}^2}{2\sigma^2}\right)\right]+\delta\right](1-p)|B|}{e^{-(\beta+\gamma)l(\boldsymbol{y})}(\lambda\beta)^{n(\boldsymbol{y})}\prod_{i=1}^{n(\boldsymbol{x})}\left[\gamma\sum_{j=1}^{n(\boldsymbol{y})}\left[\frac{1}{\sqrt{2\pi}\sigma}\exp\left(-\frac{d_{i,j}^2}{2\sigma^2}\right)\right]+\delta\right]p(n(\boldsymbol{y})+1)}\right)$$

$$= \left(\beta+\gamma\right)\left(l\left(\boldsymbol{y}\right)-l\left(\boldsymbol{y}\cup y'\right)\right)+\log\left(\frac{\lambda\beta(1-p)|B|}{p(n(\boldsymbol{y})+1)}\right)+\rho(\boldsymbol{y}\cup y',\gamma,\delta,\sigma)-\rho(\boldsymbol{y},\gamma,\delta,\sigma) \text{ og}$$

$$\log\left(H_d\left(\boldsymbol{y},y'\right)\right) = \log\left(\frac{h_l(\boldsymbol{y}\setminus y')pn(\boldsymbol{y})}{h_l(\boldsymbol{y})(1-p)|B|}\right)$$

$$= \log\left(\frac{e^{-(\beta+\gamma)l(\boldsymbol{y}\setminus y')}\left(\lambda\beta\right)^{n(\boldsymbol{y})-1}\prod_{i=1}^{n(\boldsymbol{x})}\left[\gamma\sum_{j=1}^{n(\boldsymbol{y})-1}\left[\frac{1}{\sqrt{2\pi}\sigma}\exp\left(-\frac{d_{i,j}^2}{2\sigma^2}\right)\right]+\delta\right]pn(\boldsymbol{y})}{e^{-(\beta+\gamma)l(\boldsymbol{y})}(\lambda\beta)^{n(\boldsymbol{y})}\prod_{i=1}^{n(\boldsymbol{x})}\left[\gamma\sum_{j=1}^{n(\boldsymbol{y})}\left[\frac{1}{\sqrt{2\pi}\sigma}\exp\left(-\frac{d_{i,j}^2}{2\sigma^2}\right)\right]+\delta\right](1-p)|B|}\right)$$

$$= \left(\beta+\gamma\right)\left(l\left(\boldsymbol{y}\right)-l\left(\boldsymbol{y}\setminus y'\right)\right)+\log\left(\frac{pn\left(\boldsymbol{y}\right)}{\lambda\beta(1-p)|B|}\right)+\rho(\boldsymbol{y}\setminus y',\gamma,\delta,\sigma)-\rho(\boldsymbol{y},\gamma,\delta,\sigma).$$

3.2.3 Den samlede algoritme

På baggrund af de ovenstående Hastingsforhold er det nu muligt at specificere en samlet oversigt over forløbet af estimationen, hvilket ses i algoritme 3.1.

Algoritme 3.1 (Estimationsalgoritme):
Lad $Y_i = \left(y_1, \ldots, y_{n(\boldsymbol{y})}\right)$ og $X_i = \left(\lambda_i, \beta_i, \gamma_i, \delta_i, \sigma_i\right)$ være den aktuelle tilstand i skridt i for henholdsvis linjestykker og parametre, hvor Y_0 og X_0 er valgte starttilstande, som afhænger af den enkelte situation. Lad yderligere

$$r_{bdm}, r_m, r_{bd}, r_b, r_d, r_\lambda, r_\beta, r_\gamma, r_\delta, r_\sigma \sim \text{Unif}(0,1) \text{ og } I_m, I_d \sim \text{Unif}\left(\{1, \ldots, n\left(\boldsymbol{y}\right)\}\right).$$

I hver iteration $i = 1, \ldots, N, N \in \mathbb{N}$, udføres følgende:

1) Birth-Death-Move:
   - Hvis $r_{bdm} < q$ gå i move, ellers gå i birth-death

   1.1) Move:
   - Fjern linjestykke y_{I_m} og tilføj nyt til forslag $\boldsymbol{y}'$
   - Hvis $\log(r_m) < \log\left(H_m(\boldsymbol{y}, \boldsymbol{y}')\right)$ sæt $Y_i = \boldsymbol{y}'$, ellers sæt $Y_i = Y_{i-1}$

   1.2) Birth-death:
   - Hvis $r_{bd} < p$ gå i birth, ellers gå i death

   1.2.1) Birth:
   - Generer linjestykke y' som nyt forslag
   - Hvis $\log(r_b) < \log\left(H_b(\boldsymbol{y}, y')\right)$ sæt $Y_i = \boldsymbol{y} \cup y'$, ellers sæt $Y_i = Y_{i-1}$

   1.2.2) Death:
   - Udtræk linjestykke $y' = y_{I_d}$ som forslag til sletning
   - Hvis $\log(r_d) < \log\left(H_d(\boldsymbol{y}, y')\right)$ sæt $Y_i = \boldsymbol{y} \setminus y'$, ellers sæt $Y_i = Y_{i-1}$

2) Opdatering af λ:
   - Generer nyt forslag $\lambda' \sim \text{N}\left(\lambda_{i-1}, \lambda_\psi\right)$
   - Hvis $\log(r_\lambda) < \log\left(H_\lambda\left(\lambda_{i-1}, \lambda'\right)\right)$ sæt $\lambda_i = \lambda'$, ellers sæt $\lambda_i = \lambda_{i-1}$

3) Opdatering af β:
   - Generer nyt forslag $\beta' \sim \text{N}\left(\beta_{i-1}, \beta_\psi\right)$
   - Hvis $\log\left(r_\beta\right) < \log\left(H_\beta\left(\beta_{i-1}, \beta'\right)\right)$ sæt $\beta_i = \beta'$, ellers sæt $\beta_i = \beta_{i-1}$

4) Opdatering af γ:
   - Generer nyt forslag $\gamma' \sim \text{N}\left(\gamma_{i-1}, \gamma_\psi\right)$
   - Hvis $\log\left(r_\gamma\right) < \log\left(H_\gamma\left(\gamma_{i-1}, \gamma'\right)\right)$ sæt $\gamma_i = \gamma'$, ellers sæt $\gamma_i = \gamma_{i-1}$

5) Opdatering af δ:
   - Generer nyt forslag $\delta' \sim \text{N}\left(\delta_{i-1}, \delta_\psi\right)$
   - Hvis $\log(r_\delta) < \log\left(H_\delta\left(\delta_{i-1}, \delta'\right)\right)$ sæt $\delta_i = \delta'$, ellers sæt $\delta_i = \delta_{i-1}$

6) Opdatering af σ:
   - Generer nyt forslag $\sigma' \sim \mathrm{N}(\sigma_{i-1}, \sigma_\psi)$
   - Hvis $\log(r_\sigma) < \log\left(H_\sigma(\sigma_{i-1}, \sigma')\right)$ sæt $\sigma_i = \sigma'$, ellers sæt $\sigma_i = \sigma_{i-1}$

□

Efter de N iterationer udgør X en Markovkæde, der efter et passende burn-in k kan forventes at være tilpas tilnærmet sin stationære fordeling. Da alle opdateringer foregår vha. Metropolis-Hastings, så er kæden aperiodisk, og der er ingen restriktioner, som forhindrer hverken parameter-estimationen eller BDM-delen i at nå ethvert sted i tilstandsrummet, så kæden vil også være irreducibel.

3.3 Implementation af algoritmen

Der er lavet en implementation i `R` af algoritmen på baggrund af resultaterne i forrige afsnit, og dette afsnit har til formål at præsentere løsningerne på en del af de praktiske problemstillinger, der er forbundet hermed.

3.3.1 Startværdier

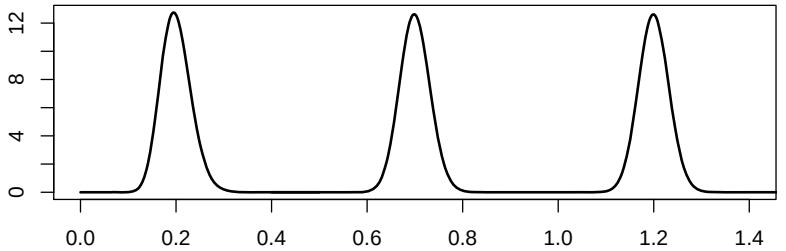
Ved starten af algoritmen er der brug for en række af startværdier, som bl.a. skal indeholde valg af antal iterationer, sandsynligheder for at gå i henholdsvis move, birth og death samt indhold til Y_0 og X_0 – tabel 3.1 indeholder en samlet oversigt.

Sandsynligheder for BDM	$q \in [0\,;1]$ – sandsynlighed for move
	$p \in [0\,;1]$ – sandsynlighed for birth
Iterationer	$N \in \mathbb{N}$ – antal iterationer
Y_0 – startlinjestykker	$y_1, \ldots, y_n, n \geq 2$
X_0 – startværdier for parametre	$\lambda_0, \beta_0, \gamma_0, \delta_0, \sigma_0 \in \mathbb{R}_+$
Forslagsspredninger	$\lambda_\psi, \beta_\psi, \gamma_\psi, \delta_\psi, \sigma_\psi \in \mathbb{R}_+$
Formparametre for priors	$\lambda_\alpha, \beta_\alpha, \gamma_\alpha, \delta_\alpha, \sigma_\alpha \in \mathbb{R}_+$

Tabel 3.1 *– Oversigt over startværdier til estimations-algoritmen.*

Middelværdien af den gammafordelte prior $\Gamma(\alpha,\kappa)$ er $\alpha\kappa$, og ud fra det enkelte tilfælde vælges et α, og κ sættes til at være eksempelvis $\lambda_\kappa = \lambda_0/\lambda_\alpha$, således at middelværdien er λ_0. Ved at forhøje værdien af α koncentreres en større del af sandsynlighedsmassen omkring middelværdien, og det kan dermed bruges til at

til at justere priorens betydning for estimationen. Det skal dog bemærkes, at ens formparametre ikke giver samme form af tætheden, idet variansen er $\alpha\kappa^2$, og derfor vil en højere middelværdi kræve et højere α for at have samme varians. Dette forhold er illustreret på figur 3.2, hvor der ses tre eksempler på, hvor stor α skal være for at opnå ens varians ved tre forskellige middelværdier.



Figur 3.2 – *Illustration af tætheden for* $\Gamma(40;0{,}2/40)$ *(venstre),* $\Gamma(490;0{,}7/490)$ *(midt) samt* $\Gamma(1440;1{,}2/1440)$ *(højre).*

Bemærk også vedr. indholdet af Y_0, at der kræves $n \geq 2$ pga. tekniske årsager i `R`, da datatypen matrix bliver konverteret til en vektor, hvis der kun er 1 række. Denne begrænsning har ikke nogen praktisk betydning, idet det for datasættet om gravhøje ikke er realistisk kun at have brug for ét linjestykke. Der kan enten benyttes n tilfældigt genererede linjestykker, eller der kan indlæses en given mængde af linjestykker inden estimationen startes. I afsnit 3.4 belyses fordele ved den tilgang frem for at starte med tilfældigt genererede linjestykker.

3.3.2 Optimeringer

Estimationen er meget beregningstung pga. særligt $\rho\left(\boldsymbol{y},\gamma,\delta,\sigma\right)$-funktionen i (3.5), som optræder i størstedelen af Hastingsforholdene. Den indeholder $n(\boldsymbol{x}) \times n\left(\boldsymbol{y}\right)$ led, hvor der i hvert led skal bestemmes afstanden mellem punkt og linjestykke, og der skal beregnes den tilsvarende tæthed i en normalfordeling for pågældende afstand. Dette skal gøres 8 gange pr. iteration, så det kan for selv forholdsvis små datasæt blive til over en million beregninger for hver iteration, og profilering af en kørsel viser også, at over 90% af kørselstiden bruges på beregning af $\rho(\cdot)$.

For at gøre algoritmen praktisk anvendelig har det derfor været nødvendigt at lave en række af optimeringer frem for blot at implementere algoritmen direkte ud fra specifikationen i afsnit 3.2.3. Først og fremmest har de fleste datastrukturer globalt scope, så der ikke anvendes unødig tid på sende data rundt mellem funktionskald, men der arbejdes direkte på dataene af de fleste funktioner. Da det interessante er indholdet af Markovkæden X efter en kørsel, så er det ikke nødvendigt at gemme information om tidligere tilstande af Y, det er kun nødvendigt at

kende den aktuelle tilstand, idet Y også er en Markovkæde. Derfor gemmes kun den aktuelle tilstand af Y, og den ændres kun, hvis der accepteres en opdatering af Y i BDM-delen af algoritmen. Ved ændring af Y opdateres også afstandene $d_{i,j}$ mellem alle punkter x_j og det nye linjestykke y_i baseret på [Bourke, 1988], således at afstandene kun skal beregnes én gang for hvert linjestykke.

En anden væsentlig optimering er at genbruge $\rho(\cdot)$ i de tilfælde, hvor det er muligt. I hver iteration optræder den 8 gange, men de 4 af gangene er på samme form $\rho\left(\boldsymbol{y},\gamma,\delta,\sigma\right)$, der kan genbruges fra tidligere beregning af Hastingsforhold afhængigt af, om forslaget er blevet accepteret eller ej. Inden påbegyndelse af estimationen beregnes den første $\rho(\cdot)$ ud fra start-værdier og -linjestykker, og herefter vil den ved hver accepteret opdatering, hvor den indgår i Hastingsforholdet, blive sat til forslagets $\rho(\cdot)$, hvilket effektivt halverer antallet af nødvendige beregninger.

I enkelte tilfælde vil yderligere optimering være mulig, da der ved eksempelvis beregning af $\rho(\cdot)$ til forslag for opdatering af γ og δ ikke vil være ændret på den indre sum i forhold til forrige beregning, men denne type optimering er dog ikke blevet implementeret.

3.3.3 Overordnet kode

I `R`-kode 3.3 ses den overordnede kode for estimations-algoritmen, og den ses at afspejle det øverste niveau af pseudo-koden i afsnit 3.2.3 med enkelte tilføjelser af teknisk karakter. Bemærk at ingen af funktionskaldene tager nogle parametre, da der som omtalt ovenfor primært arbejdes på globale datastrukturer, og de nødvendige startværdier fra tabel 3.1 indlæses også globalt, så estimationen påbegyndes ved blot at køre `estimation()`.

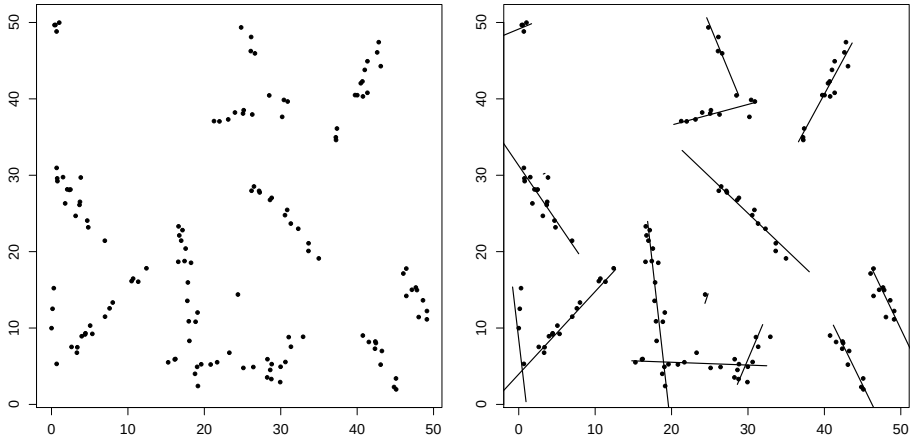
```
1  estimation <- function() { # master–funktion til estimation af parametre
2    initialiser() # opret datastrukturer og opret/indlæs start–veje inden selve algoritmen startes
3    for (i in 1:N) { # heri kører hver iteration af algoritmen
4      i <<- i # optimering for at have globalt tilgængeligt i
5      if (runif(1) < q) { move() } else { birth_death() } # kør M eller BD
6      opdater_lambda() # opdater parametrene
7      opdater_beta()
8      opdater_gamma()
9      opdater_delta()
10     opdater_sigma()
11     status() # lav statusopdatering hver n skridt
12   } # end for
13   afslutning() # udskriv resultat af kørslen til sidst
14 } # end function
```

R-kode 3.3 *– `estimation-master.R`, overordnet kode til estimation af parametre.*

Bilag A.2 indeholder koden for de funktioner, som kaldes i `estimation()` samt tilhørende hjælpefunktioner. Heri kan det yderligere bl.a. ses, hvorledes de nævnte optimeringer er lavet i praksis, og hvordan der beregnes løbende acceptrater for opdateringen af hver parameter, således at det er muligt at justere på spredningen ψ af forslagsfordelingen for at opnå en acceptrate i intervallet $[0{,}2\,;\,0{,}4]$, hvilket er med til at give en bedre mixing af Markovkæden jf. afsnit 1.5.2.

3.4 Inferens for simulerede data

For at undersøge algoritmens effektivitet vil den i dette afsnit blive anvendt til at estimere parametrene for et punktmønster, der er genereret af modellen. Figur 3.4 viser punktmønsteret, og det vil igennem fire typer kørsler blive undersøgt i hvilken grad, algoritmen er i stand til at estimere de parametre, der har frembragt punkterne.



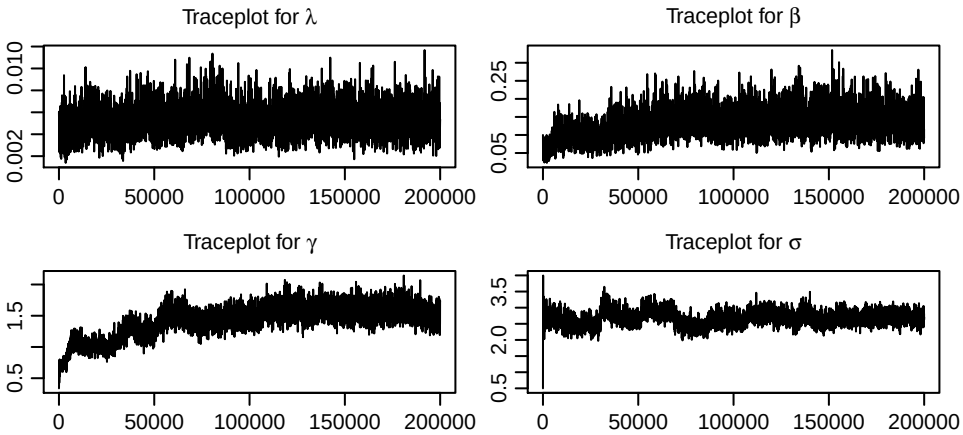
Figur 3.4 *– Til venstre et punktmønster* $\boldsymbol{x}$ *genereret ved kørsel af* `generer_model(0.006, 0.1, 0.8, 0, 0.5,c(0,50),c(0,50))`. *Til højre* $\boldsymbol{x}$ *sammen med det tilhørende* $\boldsymbol{y}$.

3.4.1 Kørsel 1 og 2 – betydningen af priors

Den første kørsel har til formål at undersøge udfaldet af en kørsel, hvor der anvendes priors med en formparameter på 2, dvs. forholdsvis flade priors, som ikke forventes at påvirke resultatet nævneværdigt. Som startværdier for parametrene er valgt de samme som ved genereringen af punktmønsteret for λ, β og σ, mens δ har fået en startværdi på 0,001, da den af regnetekniske årsager skal være positiv. q og p er valgt til henholdsvis 0,8 og 0,5, dvs. der i BDM-delen af estimationen vil være en fordeling på 80% move, 10% birth og 10% death. Disse størrelser er valgt

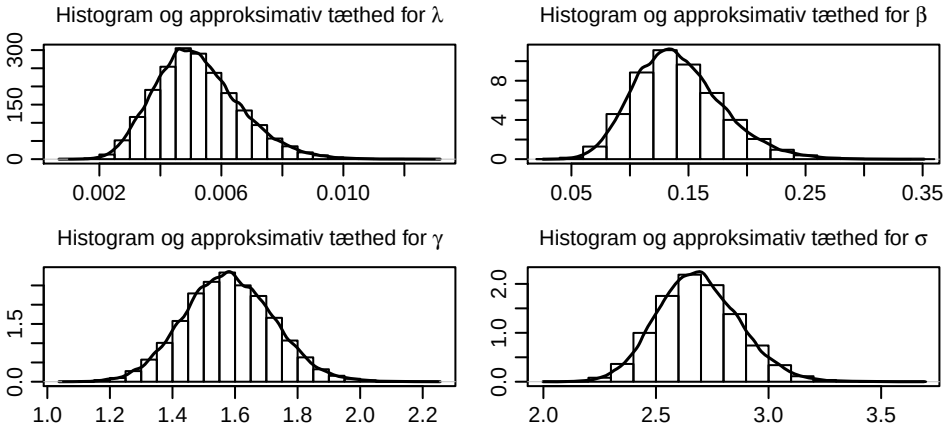
ud fra en forventning om, at antallet af linjestykker vil blive stabiliseret i løbet af afviklingen af algoritmen, og derfor vil det være hensigtsmæssigt at have en stor andel af move-opdateringer frem for at forsøge at ændre på antallet af linjestykker hele tiden. Der er forsøgt andre kombinationer af q og p for at undersøge, om det leder til hurtigere konvergens, men det synes ikke at være tilfældet, tværtimod forøger væsentligt lavere værdier af q problemet omtalt i afsnit 3.4.3.

Antal iterationer er valgt til $N = 200.000$, da eksperimenter med algoritmen har vist, at det kan tage op mod 100.000 iterationer før kæden virker til at være tilfredsstillende stabiliseret. Endeligt er der blevet justeret på spredningen af forslagsfordelingerne, indtil der er opnået acceptrater i intervallet $[0{,}2\,;\,0{,}4]$. På figur 3.5 ses traceplots af kørslen for de fire parametre λ,β,γ og σ, mens δ er udeladt, da den er af mindre betydning her. Det ses, at der kræves mange iterationer, før særligt γ og σ bliver stabiliserede, men der synes at være tilstrækkelig stabilitet efter 100.000 iterationer, så der vælges et burn-in $k = 100.000$.



Figur 3.5 *– Traceplots for kørsel 1.*

Efter burn-in fås histogrammerne på figur 3.6, hvor der også er indlagt en approksimeret tæthed for den enkelte parameter. For alle fire parametre fås en næsten symmetrisk og unimodal tæthed, hvilket er med til at bekræfte forventningen om, at kæden har nået sin stationære fordeling. Traceplots og histogrammer for de efterfølgende kørsler ligner i høj grad disse eksempler, så denne type illustrationer bringes kun for denne første kørsel.



Figur 3.6 *– Histogrammer og approksimerede tætheder for kørsel 1.*

En oversigt over valgte startværdier og resultatet kan ses i tabel 3.7. Heraf fremgår, at CPI'erne for λ og β indeholder den sande værdi, mens det ikke er tilfældet for γ og i endnu mindre grad for σ. En årsag hertil kan være, at de indsatte linjestykker fra BDM-delen af algoritmen ikke passer så godt som de sande linjestykker fra genereringen af punktmønsteret.

Kørsel 1 – Startværdier

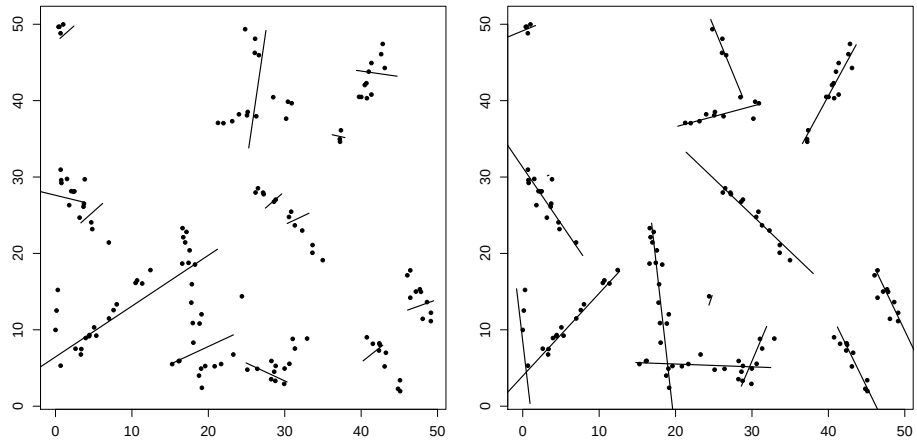
$q = 0{,}8$	$p = 0{,}5$	$N = 200000$	$n = 15$	$k = 100000$
$\lambda_0 = 0{,}006$	$\beta_0 = 0{,}1$	$\gamma_0 = 0{,}8$	$\delta_0 = 0{,}001$	$\sigma_0 = 0{,}5$
$\lambda_\psi = 0{,}01$	$\beta_\psi = 0{,}18$	$\gamma_\psi = 0{,}48$	$\delta_\psi = 0{,}004$	$\sigma_\psi = 0{,}65$
$\lambda_\alpha = 2$	$\beta_\alpha = 2$	$\gamma_\alpha = 2$	$\delta_\alpha = 2$	$\sigma_\alpha = 2$

Kørsel 1 – Resultater

Parameter	Gennemsnit	95% CPI	Acceptrate
λ	0,00524	[0,00289 ; 0,00830]	0,235
β	0,148	[0,0785 ; 0,224]	0,289
γ	1,58	[1,31 ; 1,86]	0,299
δ	0,000860	[0,000117 ; 0,00223]	0,240
σ	2,69	[2,36 ; 3,06]	0,301

Tabel 3.7 *– Oversigt over valgte startværdier og resultatet af kørsel 1.*

Sammenlignes de to plots i figur 3.8, så ses linjestykkerne i estimationen i flere tilfælde at gå på tværs af de oprindelige linjestykker. De er også generelt kortere, og det kan forklare det forhøjede gennemsnit af β og derigennem, at estimatet af γ ligger højere end den sande værdi. Det væsentligt forhøjede estimat af σ kan



Figur 3.8 *– Linjestykker ved afslutning af kørsel 1 (venstre) og linjestykker fra generering af punktmønsteret (højre – samme plot som det højre i figur 3.4).*

forklares af samme årsag, da det tydeligt ses på plottet til venstre, at punkterne ligger betragteligt længere væk fra linjestykkerne end i plottet til højre.

For at forsøge og opnå et bedre resultat, er det undersøgt, i hvilken grad det har effekt at ændre på formparameteren for de fem priors. Det har vist sig, at særligt en smal prior for σ kan føre til et bedre resultat af estimationen, idet det så at sige tvinger σ til at ligge indenfor et lille interval, og det gør, at BDM-delen ikke er så tilbøjelig til at placere linjestykker på samme måde, som det blev gjort ved kørsel 1. Således er kørsel 2 foretaget på samme måde som kørsel 1 men med ændrede formparametre samt forslagsspredninger, og en oversigt kan ses i tabel 3.9.

Kørsel 2 – Startværdier

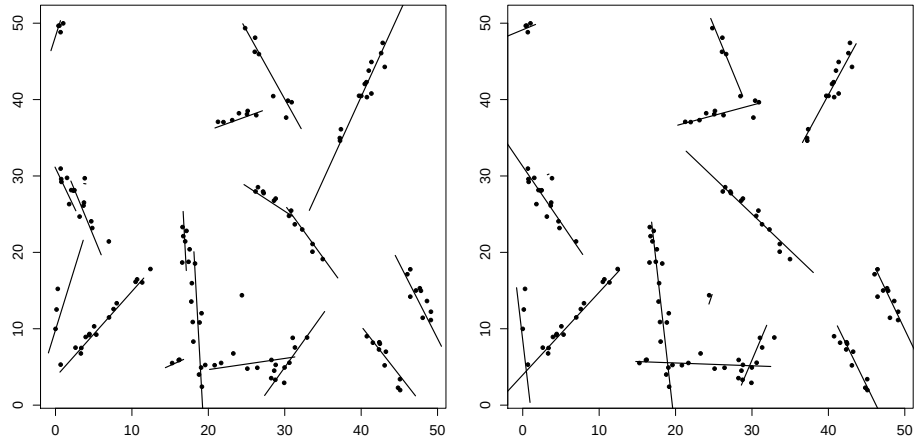
$q = 0{,}8$	$p = 0{,}5$	$N = 200000$	$n = 15$	$k = 100000$
$\lambda_0 = 0{,}006$	$\beta_0 = 0{,}1$	$\gamma_0 = 0{,}8$	$\delta_0 = 0{,}001$	$\sigma_0 = 0{,}5$
$\lambda_\psi = 0{,}006$	$\beta_\psi = 0{,}07$	$\gamma_\psi = 0{,}23$	$\delta_\psi = 0{,}0012$	$\sigma_\psi = 0{,}1$
$\lambda_\alpha = 20$	$\beta_\alpha = 20$	$\gamma_\alpha = 20$	$\delta_\alpha = 20$	$\sigma_\alpha = 500$

Kørsel 2 – Resultater

Parameter	Gennemsnit	95% CPI	Acceptrate
λ	0,00664	[0,00466 ; 0,00894]	0,251
β	0,0963	[0,0679 ; 0,130]	0,289
γ	0,674	[0,568 ; 0,789]	0,279
δ	0,00112	[0,000117 ; 0,00162]	0,280
σ	0,523	[0,485 ; 0,562]	0,232

Tabel 3.9 *– Oversigt over valgte startværdier og resultatet af kørsel 2.*

Denne gang er det kun γ hvis CPI ikke indeholder den sande værdi, men det er dog ikke langt fra, og forklaringen kan være, at gennemsnittet for β nu er mindre end 0,1. Det bør også bemærkes, at CPI'erne i kørsel 2 er kortere end i kørsel 1, hvilket er forventeligt pga. de forhøjede formparametre. Hvis der som før foretages en sammenligning af linjestykkerne, som ses på figur 3.10, så fremgår det, at deres placering denne gang i høj grad ligner den oprindelige placering. Disse velegnede placeringer lader til primært at kunne tilskrives valget af $\sigma_\alpha = 500$, og det kan også ses under selve kørslen, at stort set samtlige nyindsatte linjestykker er i stil med dem, som blev brugt under genereringen af punktmønsteret.



Figur 3.10 *– Linjestykker ved afslutning af kørsel 2 (venstre) og linjestykker fra genererering af punktmønsteret (højre – samme plot som det højre i figur 3.4).*

På baggrund af de to kørsler må det konstateres, at der ikke nødvendigvis opnås korrekte estimater, men det er muligt at opnå et mere præcist resultat ved justering af priors. Den tilgang er ikke velegnet til alle situationer, idet der ikke altid kan forventes at være nok a priori viden til at kunne forsvare brugen af så smal en prior, som det viste sig nødvendigt her. Derfor er der forsøgt et alternativ, hvor der, i stedet for at generere tilfældige linjestykker ved starten af algoritmen, benyttes linjestykkerne fra genereringen af punktmønsteret.

3.4.2 Kørsel 3 og 4 – anvendelse af linjestykker fra genereringen

I kørsel 3 er der ved starten af kørslen samme situation som på figur 3.4, idet der anvendes de samme linjestykker. Denne tilgang forsøges ud fra forventningen om, at der er en forholdsvis lille sandsynlighed for, at disse linjestykker vil blive flyttet eller fjernet af BDM-delen, og at der samtidigt opnås en lille sandsynlighed for indsættelse af linjestykker, der som i kørsel 1 ligger på tværs af mere optimale

linjestykker. Af disse årsager er det også i første omgang forsøgt at anvende priors med en formparameter på 2, det har dog vist sig at være nødvendigt at benytte en væsentligt højere β_α og γ_α for at få estimatet af disse to parametre til at passe. En mulig forklaring på, at det er nødvendigt, gives senere i forbindelse med kørsel 4.

Tabel 3.11 indeholder startværdier og resultatet af kørslen, og det ses i rimelig grad at stemme overens med resultatet fra kørsel 2, idet gennemsnittet ligger tæt på den sande værdi, der også er klart indeholdt i CPI'erne. Der er yderligere fo-

Kørsel 3 – Startværdier

$q = 0{,}8$	$p = 0{,}5$	$N = 200000$	$n = —$	$k = 100000$
$\lambda_0 = 0{,}006$	$\beta_0 = 0{,}1$	$\gamma_0 = 0{,}8$	$\delta_0 = 0{,}001$	$\sigma_0 = 0{,}5$
$\lambda_\psi = 0{,}007$	$\beta_\psi = 0{,}05$	$\gamma_\psi = 0{,}14$	$\delta_\psi = 0{,}005$	$\sigma_\psi = 0{,}15$
$\lambda_\alpha = 2$	$\beta_\alpha = 100$	$\gamma_\alpha = 500$	$\delta_\alpha = 2$	$\sigma_\alpha = 2$

Kørsel 3 – Resultater

Parameter	Gennemsnit	95% CPI	Acceptrate
λ	0,00585	[0,00333 ; 0,00905]	0,303
β	0,0952	[0,0786 ; 0,114]	0,222
γ	0,766	[0,705 ; 0,828]	0,260
δ	0,00151	[0,000353 ; 0,00321]	0,236
σ	0,555	[0,484 ; 0,634]	0,295

***Tabel 3.11** – Oversigt over valgte startværdier og resultatet af kørsel 3.*

retaget to lignende kørsler kaldet kørsel 3' og kørsel 3". For de to kørsler er der anvendt de samme startværdier som for kørsel 3, men der er inden starten af algoritmen kun indlæst henholdsvis halvdelen og en tredjedel af linjestykkerne fra genereringen af punktmønsteret. Det har til formål af afprøve den situation, hvor der haves viden om en del af linjestykkerne, hvilket kan være relevant for datasættet om gravhøje, da nogle forhistoriske veje er kendte. Disse to kørsler giver lignende resultater som kørsel 3 for λ, β og γ, men for σ indeholder CPI'et ikke den sande værdi, og det menes at kunne tilskrives de samme årsager som beskrevet vedr. kørsel 1, dvs. placering af linjestykker, som passer dårligt til de korrekte linjer, af BDM-delen.

Efter at have erfaret de forskellige problemstillinger ved de tre forskellige typer kørsler er der endeligt forsøgt med en kørsel 4, hvor der fra starten anvendes de samme linjestykker som i kørsel 3, men denne gang fastholdes de, dvs. BDM-delen er frakoblet i kørsel 4. Resultatet kan ses i tabel 3.12. Bemærk at formparameteren er sat til 2 for alle priors, da der ikke bør være brug for en smal prior i denne situation, når linjemønsteret fastholdes. Resultatet viser, at λ og σ ligger pænt centralt i CPI'et, men β og γ er igen problematiske.

Kørsel 4 – Startværdier

$q = —$	$p = —$	$N = 200000$	$n = —$	$k = 100000$
$\lambda_0 = 0{,}006$	$\beta_0 = 0{,}1$	$\gamma_0 = 0{,}8$	$\delta_0 = 0{,}001$	$\sigma_0 = 0{,}5$
$\lambda_\psi = 0{,}007$	$\beta_\psi = 0{,}05$	$\gamma_\psi = 0{,}14$	$\delta_\psi = 0{,}005$	$\sigma_\psi = 0{,}15$
$\lambda_\alpha = 2$	$\beta_\alpha = 2$	$\gamma_\alpha = 2$	$\delta_\alpha = 2$	$\sigma_\alpha = 2$

Kørsel 4 – Resultater

Parameter	Gennemsnit	95% CPI	Acceptrate
λ	0,00612	[0,00359 ; 0,00929]	0,309
β	0,0652	[0,0379 ; 0,0995]	0,392
γ	0,560	[0,470 ; 0,657]	0,381
δ	0,000877	[0,000158 ; 0,00215]	0,193
σ	0,5444	[0,477 ; 0,622]	0,285

Tabel 3.12 *– Oversigt over valgte startværdier og resultatet af kørsel 4.*

Ved at se nærmere på de anvendte linjestykker, så viser det sig, at en del af dem går langt udenfor observationsvinduet, og det er deres samlede længde $l(\boldsymbol{y})$, der anvendes til beregninger. Da Hastingsforholdene for netop β og γ afhænger af $l(\boldsymbol{y})$ jf. (3.3) og (3.4), så må det være her årsagen til de dårlige estimater skal findes, dvs. det skyldes kanteffekter. De pågældende linjestykker har stor tilbøjelighed til at blive liggende under hele kørsel 3 og ligger fast under kørsel 4, og den forøgede samlede længde af linjestykkerne stemmer overens med det for lave estimat af både β og γ.

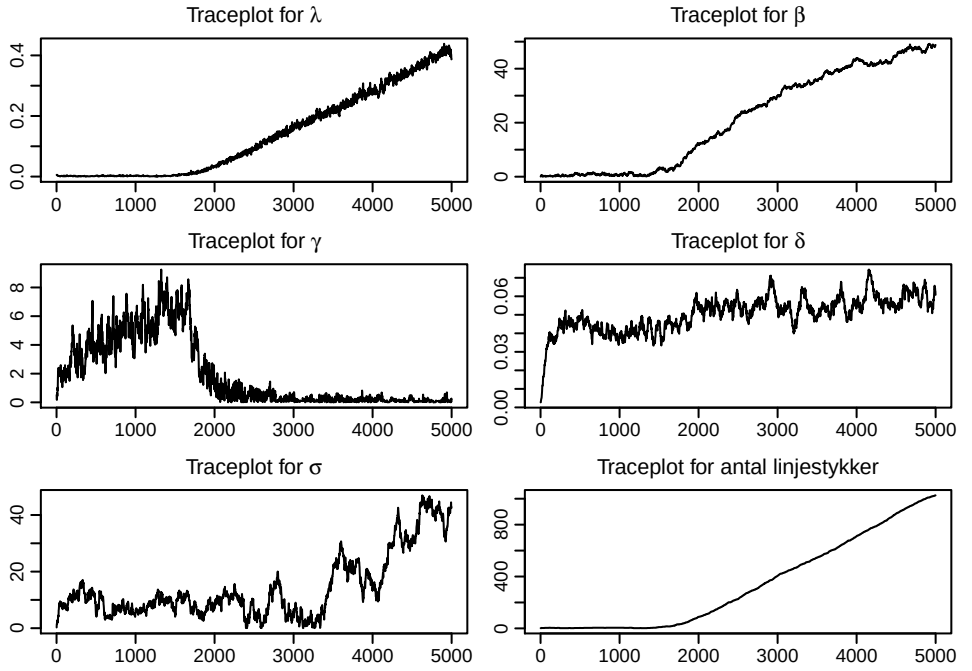
Det kan på baggrund af dette fund synes nødvendigt at modificere algoritmen til at se bort fra den del af linjestykkerne, der måtte være udenfor observationsvinduet, men det er dog valgt ikke at foretage en sådan ændring. Det skyldes, at problemstillingen har væsentlig mindre betydning, når der startes med tilfældige linjestykker som ved kørsel 1 og 2, idet de kun i sparsom grad vil have andele udenfor observationsvinduet, og der er lille sandsynlighed for accept af nye linjestykker, som i nævneværdig grad ligger udenfor observationsvinduet. Anvendes der bestemte linjestykker ved algoritmens start på samme vis som i kørsel 3 og 4, så kan de blot tilpasses til at være indeholdt i observationsvinduet for at forebygge problemet.

Konklusionen på disse eksperimenter er, at det kræver varsomhed at benytte algoritmen til estimation, da der er flere faldgruber, som kan føre til uhensigtsmæssige resultater. Ved estimation af parametrene for datasættet om gravhøje vil der derfor blive forsøgt at finde en velegnet blanding af henholdsvis priors og linjestykker ved algoritmens start.

3.4.3 Vigtigheden af egentlig og tilpas smal prior

Undervejs i implementationen af algoritmen og udarbejdelsen af de forskellige kørsler har der ved flere lejligheder vist sig et påfaldende fænomen, hvor estimationen pludselig begynder at få en kraftig ændring i både antal linjestykker og parameterestimater. Inden der blev implementeret brugen af de gammafordelte priors, dvs. ved brug af en uegentlig prior $p(\theta) \propto 1$, opstod der flere gange en situation som illustreret af traceplots på figur 3.13. Her har algoritmen kørt i ca. 1500 iterationer, hvorefter λ, β, γ og antallet af linjestykker begynder at ændre sig voldsomt og antage stærkt usandsynlige værdier i forhold til både dataene, der er de samme som i kørsel 1–4, og den intuitive opfattelse af, hvordan algoritmen bør udvikle sig.

Desværre har det ikke været muligt at observere, om den kraftige ændring stabiliseres på et tidspunkt, da algoritmen praktisk talt går i stå pga. det store antal linjestykker, der mangedobler antallet af beregninger for at bestemme $\rho(\cdot)$. Det har taget over 5 timers kørselstid blot at nå op på de 5000 iterationer, til sammenligning er kørselstiden typisk 2–3 timer for de 200.000 iterationer i kørsel 1–4.



Figur 3.13 *– Traceplots for testkørsel med kraftige ændringer efter ca. 1500 iterationer.*

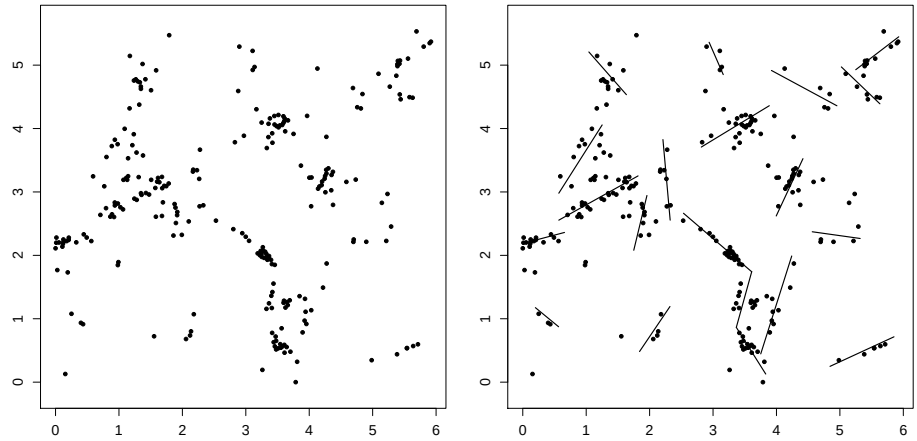
Umiddelbart giver dette mistanke om fejl i implementationen eller i de opstillede Hastingsforhold, men der er ikke blevet fundet nogle fejl herved. Ved nærmere undersøgelse af fænomenet er det observeret, at det ser ud til at opstå, hvis værdien af β bliver tilpas stor. Det betyder, at de nye forslag til linjestykker i BDM-delen bliver mindre og mindre, og så kan tingene begynde at løbe løbsk. Der indsættes mange meget korte linjestykker, og acceptraterne på alting begynder også at stige, fordi alle forslag er stort set lige sandsynlige eller rettere sagt usandsynlige, hvis først denne ustabile tilstand er nået.

At en sådan situation kan opstå må skyldes BDM-delen af algoritmen, hvor de varierende linjestykker forårsager, at estimaterne særligt i begyndelsen af en kørsel fluktuerer. Problemet synes som nævnt især at være knyttet til værdien af β, og problemet har ikke vist sig under arbejdet med kørsel 1–4 siden de egentlige priors blev indført og implementeret. Under de indledende forsøg med at anvende algoritmen på datasættet om gravhøje viste problemet sig dog igen på samme vis, når $\beta_\alpha = 2$, så det kan konstateres, at det ikke alene er vigtigt at anvende egentlige priors, det er også vigtigt, at de er tilpas smalle, da estimaterne ellers risikerer at spiralere ud af kontrol. Når situation også kan opstå på trods af brugen af en egentlig prior, så vurderes det, at det skyldes datasættet ikke i lige så høj grad som de simulerede data passer til modellen.

3.5 Inferens for datasættet om gravhøje

På trods af de optimeringer, som er omtalt i afsnit 3.3.2, så har det vist sig, at kørselstiden for at lave inferens for hele datasættet er for stor til, at det har været praktisk muligt at gennemføre. Derfor er der udvalgt en mindre del af datasættet, som den egentlige inferens udføres for. Figur 3.14 viser det reducerede datasæt, der består af i alt 250 punkter ud af de oprindelige 1147 punkter, og det dækker over et kvadratisk område på $6{,}35 \times 6{,}35 = 40{,}3225$ km^2. Ud over, at det er udvalgt ud fra målet om et håndterligt antal punkter, så vurderes det også i passende grad at repræsentere de samme karakteristika som hele datasættet i form af tydelige lineære strukturer af varierende kompakthed samt et antal isolerede punkter.

På baggrund af erfaringerne fra afsnit 3.4 er der indlagt i alt 20 linjestykker frem for at starte estimationen med tilfældigt genererede linjestykker. Disse linjestykker ses sammen med punktmønsteret på det højre plot i figur 3.14. Ideelt set skulle dette gøres ved at anvende kendte forhistoriske veje, men da sådanne oplysninger ikke haves, så er linjestykkerne indlagt manuelt ud fra en visuel betragtning af, hvad der vil passe hensigtsmæssigt i forhold til punktmønsteret.

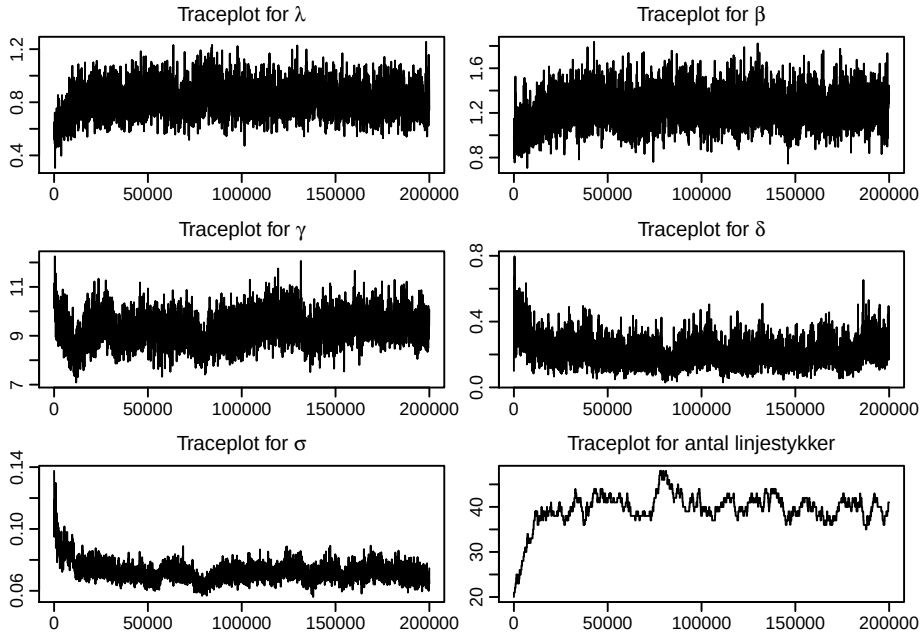


Figur 3.14 *– Til venstre det reducerede datasæt, til højre det reducerede datasæt med indlagte linjestykker, som anvendes ved starten af estimationen.*

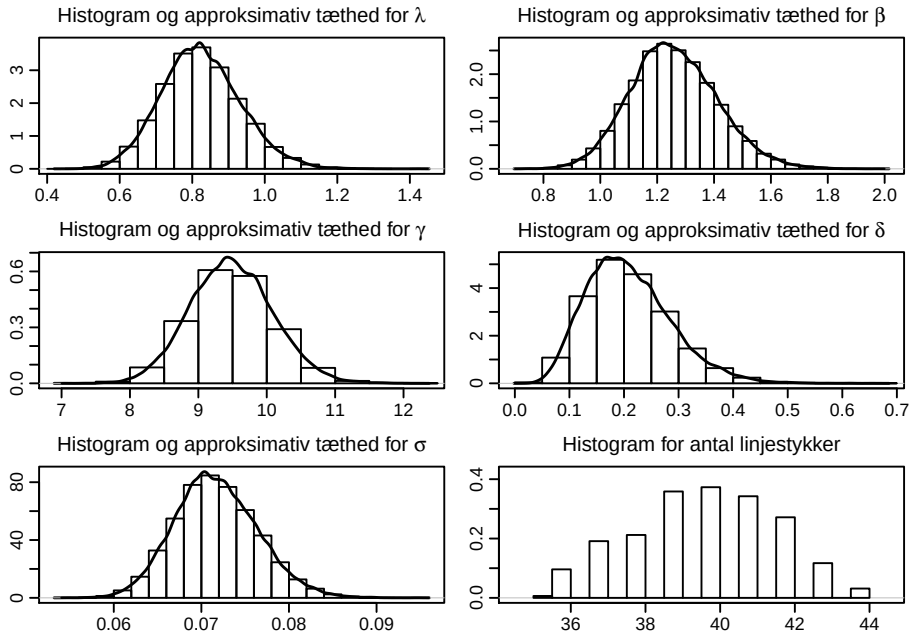
Det vurderes, at der vil være brug for flere veje for at beskrive punktmønsteret optimalt, i alt 25 veje, og derfor vælges λ_0 som 25/40,3225 = 0,62, og der forventes en middelvejlængde på ca. 1 km, så β_0 sættes til 1. Der er 250 punkter i datasættet, og valgene af β og λ giver i alt en forventning på 25 km vej, så γ_0 vælges til at være 10. δ_0 sættes til 0,1, dvs. det skal forklare ca. 4 isolerede punkter, og endeligt sættes $\sigma_0 = 0{,}1$, da størstedelen af punkterne forventes at være placeret indenfor 200 m af en vej. α og ψ er valgt på baggrund af en række testkørsler og er sat til værdier, der giver en passende konvergens og acceptrate.

Figur 3.15 viser traceplots for kørslen for alle parametre samt antallet af linjestykker. Det ses bl.a., at der ikke er så tydelig konvergens som ved kørsel 1–4, men det vurderes alligevel, at den er tilpas efter 100.000 iterationer, så der vælges som tidligere et burn-in $k = 100.000$. Det ses også, at antallet af linjestykker er knapt det dobbelte af det forventede antal, og det skyldes primært det samme fænomen som omtalt ved kørsel 1, nemlig at en del linjestykker placeres på tværs af, hvad der visuelt ser ud til at være den optimale placering, og dermed påvirkes især λ og β.

Efter burn-in kan resultatet vurderes ud fra histogrammerne og de approksimerede tætheder på figur 3.16. Det ses, at kæden må være tilpas konvergeret efter de 100.000 iterationer, da der i alle tilfælde haves forholdsvis symmetriske og unimodale resultater. For visse af parametrene ses en lille tendens til, at den approksimerede tæthed har nogle udsving i nærheden af maksimum, men det synes ikke at være af større betydning.



Figur 3.15 *– Traceplots for estimation af parametrene for (reduceret version af) datasættet om gravhøje.*



Figur 3.16 *– Histogrammer og approksimerede tætheder for estimation af parametrene for (reduceret version af) datasættet om gravhøje.*

I tabel 3.17 ses et overblik over startværdier og resultatet af estimationen, og det fremgår som allerede omtalt, at flere af parametrene ligger et stykke fra det forventede, der blev brugt som startværdi. For λ og σ indeholder CPI'et ikke startværdien, hvor det for λ kan tilskrives den allerede omtalte placering af linjestykker af BDM-delen, mens det for σ må konstateres, at punkterne ligger tættere på linjestykkerne end forventet. Yderligere ses det ud fra CPI for δ, at den skal forklare mellem ca. 3–15 isolerede punkter, hvilket synes at stemme overens med punktmønsteret. Samlet set virker resultatet derfor ikke overraskende i forhold til det forventede.

Estimation for reduceret datasæt – Startværdier

$q = 0{,}8$	$p = 0{,}5$	$N = 200000$	$n = —$	$k = 100000$
$\lambda_0 = 0{,}62$	$\beta_0 = 1$	$\gamma_0 = 10$	$\delta_0 = 0{,}1$	$\sigma_0 = 0{,}1$
$\lambda_\psi = 0{,}4$	$\beta_\psi = 0{,}7$	$\gamma_\psi = 2$	$\delta_\psi = 0{,}4$	$\sigma_\psi = 0{,}015$
$\lambda_\alpha = 20$	$\beta_\alpha = 30$	$\gamma_\alpha = 50$	$\delta_\alpha = 2$	$\sigma_\alpha = 8$

Estimation for reduceret datasæt – Resultater

Parameter	Gennemsnit	95% CPI	Acceptrate
λ	0,825	[0,622 ; 1,05]	0,316
β	1,26	[0,981 ; 1,58]	0,267
γ	9,49	[8,34 ; 10,7]	0,322
δ	0,209	[0,0837 ; 0,382]	0,278
σ	0,0717	[0,0630 ; 0,0815]	0,308

***Tabel 3.17** – Oversigt over valgte startværdier og resultatet af estimation for (reduceret version af) datasættet om gravhøje.*

Modelkontrol 4

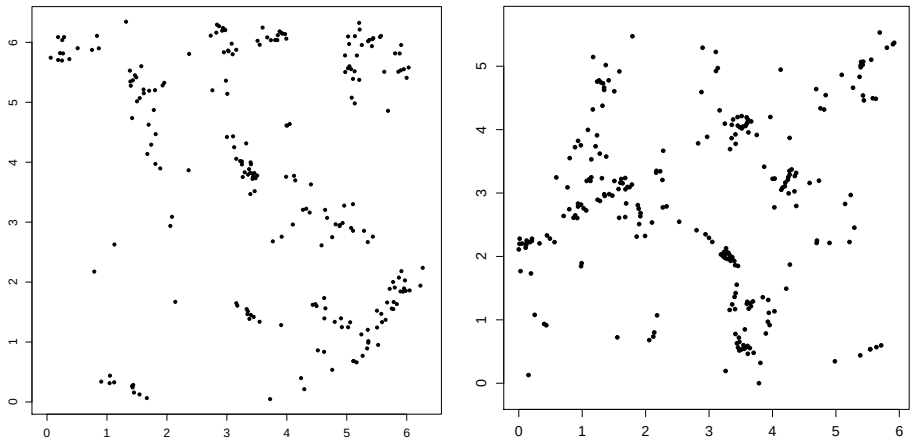
I dette kapitel vil der blive anvendt en række fremgangsmåder til at undersøge i hvilken udstrækning punktmønstre genereret af modellen er i overensstemmelse med datasættet om gravhøje. På baggrund af resultaterne fra afsnit 3.5 foretages først forskellige visuelle sammenligninger for at få et indledende indtryk af ligheder og forskelle. Derefter vil det blive undersøgt, om fordelingen af vinkler mellem et punkt og dets to nærmeste naboer for henholdsvis model og datasæt har sammenfald, da de bør ligne hinanden pga. de lineære strukturer. Til sidst anvendes de deskriptive størrelser fra afsnit 1.2 til sammenligning. Idet der er udført inferens for et reduceret datasæt, så vil de forskellige sammenligninger blive foretaget både for det reducerede og det fulde datasæt med henblik på at belyse, om resultaterne kan anvendes i begge tilfælde.

4.1 Visuelle sammenligninger

Et første indtryk af modellens egnethed fås ved at sammenligne punktmønstre genereret af modellen og datasættet. På figur 4.1 ses et punktmønster $\boldsymbol{x}$ genereret ved brug af de estimerede gennemsnit fra tabel 3.17 sammen med det reducerede datasæt. Ved en visuel sammenligning synes der ikke at være større uoverensstemmelse mellem strukturen af de to plot. Det simulerede punktmønster indeholder tydelige lineære strukturer af varierende længde og intensitet af punkter, hvilket er tilsvarende det reducerede datasæt. Det kan også ses, hvordan der på begge plot er omtrent lige mange isolerede punkter, som må formodes ikke at være tilknyttet nogen vej.

På disse væsentlige faktorer er der således sammenfald mellem de to punktmønstre, men der synes dog at optræde en hvis forskel på de to plot i form af de synlige klyngetendenser. Der optræder enkelte synlige klynger på det simulerede punktmønster, men der er flere klynger i det reducerede datasæt, og intensiteten af dem synes at være højere. Det er ikke overraskende, at modellen ikke umiddelbart er i stand til at repræsentere denne type klynger, når både linjestykker og punkter (før forskydning) genereres vha. en PPP og derfor er uniformt fordelte. Det viser sig da

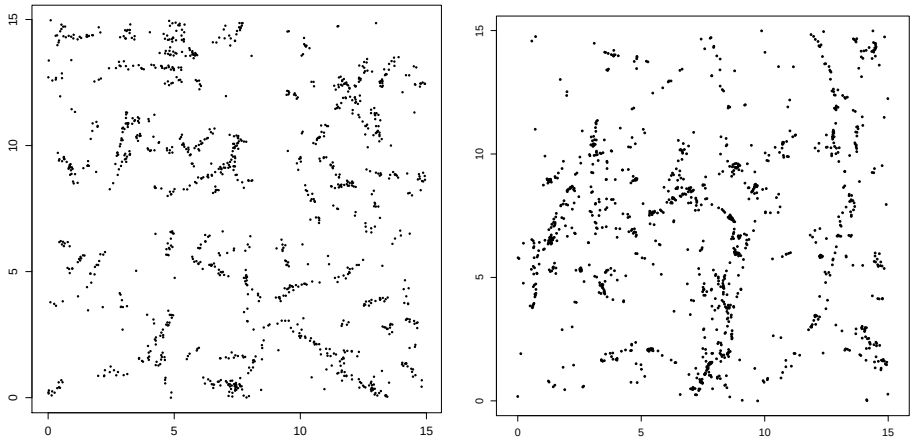
også ved at betragte mønsteret af linjestykker $\boldsymbol{y}$, at klyngerne for $\boldsymbol{x}$ hovedsageligt ligger ved skæringer mellem linjestykkerne.



Figur 4.1 *– Til venstre et punktmønster $\boldsymbol{x}$ på $[0\,;6.35]\times[0\,;6.35]$ genereret af modellen ved brug af de estimerede gennemsnit, til højre det reducerede datasæt.*

Mht. antal punkter, så indeholder $\boldsymbol{x}$ 229 punkter i forhold til de 250 for det reducerede datasæt. For at kunne vurdere variationen i antallet af punkter, så kan modellen simuleres et antal gange for at bestemme intervaller herfor. Baseret på 5000 simulationer fås et 95%-interval for det reducerede datasæt til $[155\,;388]$, og medianen er 256, så også her er der god overensstemmelse mellem simulationer og datasæt.

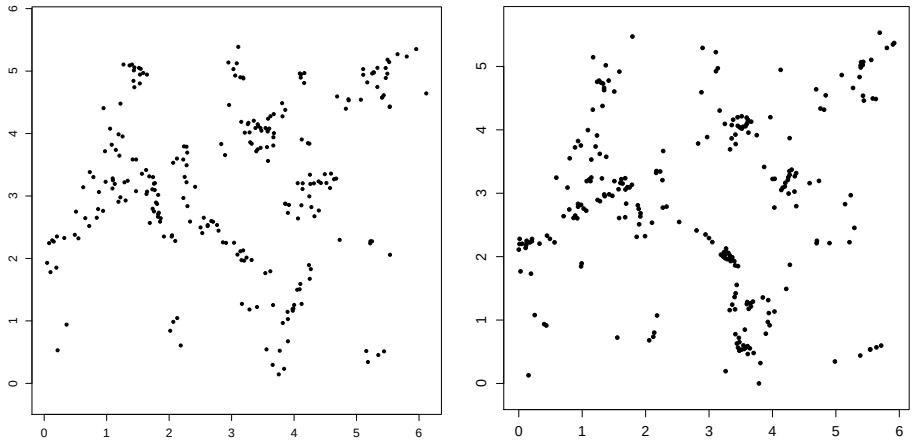
På figur 4.2 ses en simulation af et punktmønster $\boldsymbol{x}$ på hele området, og det sammenlignes nu med hele datasættet. Umiddelbart kan der observeres de samme



Figur 4.2 *– Til venstre et punktmønster $\boldsymbol{x}$ på $[0\,;15]\times[0\,;15]$ genereret af modellen ved brug af de estimerede gennemsnit, til højre det komplette datasæt.*

fænomener som for punktmønstrene i figur 4.1, men for særligt antallet af punkter viser der sig at være dårligere korrespondens mellem model og datasæt. Det viste punktmønster $\boldsymbol{x}$ har 1184 punkter i forhold til datasættets 1147, hvilket ikke synes at være nogen nævneværdig stor forskel, men det er et fåtal af simulationerne, der har så få punkter. For 5000 simulationer fås et 95%-interval $[1170\,;1740]$ og median 1440, så de estimerede parametre for det reducerede datasæt viser sig på dette område mindre velegnede til simulation på hele området.

I de to ovenstående sammenligninger er der anvendt den simultane model $\boldsymbol{X},\boldsymbol{Y}$, men det kan også være interessant at simulere den betingede model $\boldsymbol{X}|\boldsymbol{Y}$, hvor $\boldsymbol{x}$ så er en realisation af $\boldsymbol{X}$ givet et mønster af linjestykker $\boldsymbol{y}$, som vælges til at være det $\boldsymbol{y}$, der haves ved afslutningen af estimationen. Et resultat af at gøre dette ses på figur 4.3, og det skal bemærkes, at der ved denne tilgang ikke blot bør være den samme type tendenser på begge punktmønstre. De to bør være næsten visuelt identiske, hvis den betingede models punktmønstre skal siges at være i overensstemmelse med data.



Figur 4.3 *– Til venstre et punktmønster $\boldsymbol{x}$ på $[0\,;6.35]\times[0\,;6.35]$ genereret af modellen ved brug af de estimerede gennemsnit for γ,δ,σ samt $\boldsymbol{y}$ ved afslutningen af estimationen, til højre det reducerede datasæt.*

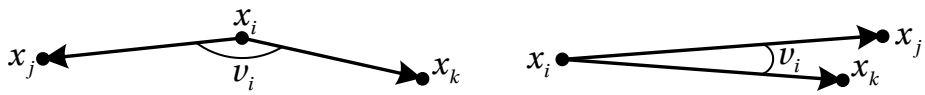
Ved sammenligning af de to punktmønstre ses det, at omend der er visse tydelige forskelle, så er der en høj grad af sammenhæng mht. placeringen af de områder, hvor punkterne ligger. Dette kan tilskrives kombinationen af de manuelt indlagte linjestykker ved begyndelsen af estimationen samt BDM-delen af algoritmen. Den har fjernet knap halvdelen af de indlagte linjestykker, men de indsatte linjestykker har i tilpas grad en placering, der giver sammenfald mellem simulerede punktmønstre og data. For antallet af punkter er der også i høj grad kor-

respondens, idet $\boldsymbol{x}$ indeholder 254 punkter, og et 95%-interval baseret på 5000 simulationer er $[220\,;281]$ med en median på 250, hvilket netop er antallet af punkter for det reducerede datasæt. Den eneste tydelige forskel er som tidligere klyngetendenserne, hvor modellen også i denne situation ikke i særlig stor grad er i stand til at producere samme type klynger som i datasættet.

4.2 Fordeling af vinkler

De forskellige deskriptive størrelser i afsnit 1.2 vedrører ikke direkte lineære strukturer, men da det er en væsentlig egenskab for modellen, så vil det være gavnligt at have en metode til at vurdere disse. En fremgangsmåde til dette angives i [Møller & Rasmussen, 2010], hvor der undersøges størrelsen af vinklerne mellem hvert punkt og de to nærmeste nabopunkter.

Lad x_i, $i = 1,\ldots,n(\boldsymbol{x})$ være et punkt i et punktmønster $\boldsymbol{x}$, og lad henholdsvis x_j og x_k være de to nærmeste nabopunkter til x_i i $\boldsymbol{x}$. Lad yderligere vinklen $v_i \in [0\,;\pi]$ være vinklen mellem vektorerne $x_j - x_i$ og $x_k - x_i$. Hvis de tre punkter ligger præcist på en ret linje, så vil v_i være 0 eller π afhængig af den indbyrdes placering af punkterne. Dermed må der forventes at være et flertal af vinkler i nærheden af 0 og π, hvis punkterne ligger omtrent langs en ret linje, som de menes at gøre i datasættet og ved simulationer af modellen – figur 4.4 illustrerer idéen i metoden.

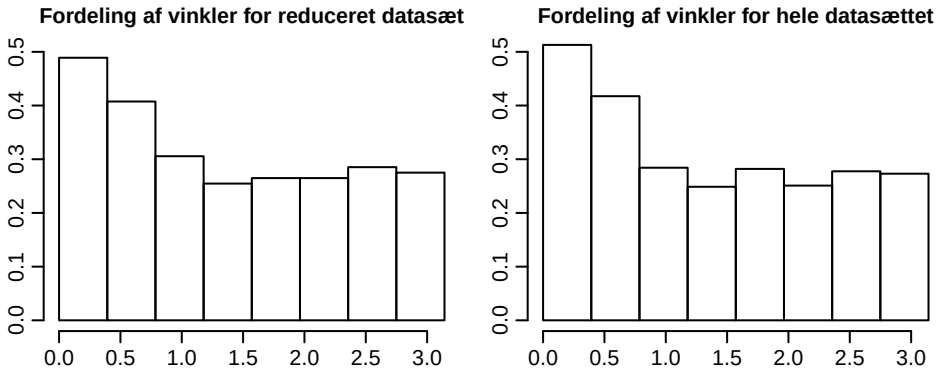


Figur 4.4 *– Illustration af, at vinklerne mellem tre punkter, der omtrent følger en ret linje, vil være forholdsvis tæt på 0 eller π.*

Metoden anvendes først til at vurdere, om datasættet vitterligt indeholder de lineære strukturer, der tydeligt synes at fremgå ved en visuel betragtning. Figur 4.5 viser et histogram over vinkelfordelingen for henholdes det reducerede og det fulde datasæt, hvor hver søjle er $\pi/8$ bred. Der anvendes tæthed på y-aksen, således at de to histogrammer er direkte sammenlignelige på trods af forskellig hyppighed af vinkler i hvert interval.

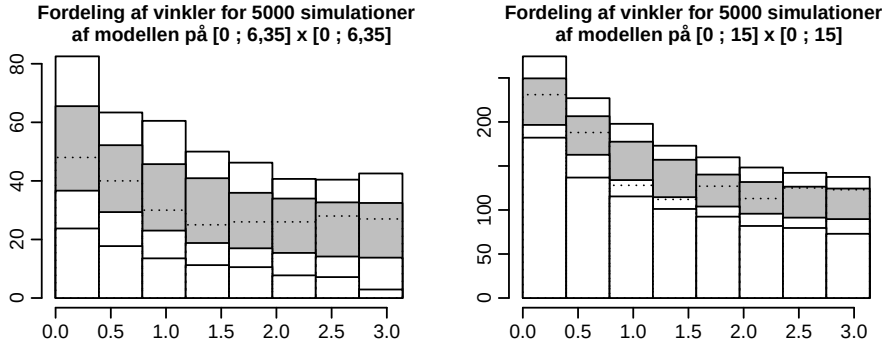
Først og fremmest bemærkes det, at de to histogrammer er praktisk talt ens, dvs. der i begge tilfælde udvises den samme grad af lineære strukturer, og det er med til at bekræfte den tidligere påstand fra afsnit 3.5 om, at det reducerede datasæt udviser samme tendenser som hele datasættet. Det ses også, at der en smule

overraskende kun er en ret begrænset overvægt af vinkler i de første to søjler nær 0, og der er ingen overvægt af vinkler nær π. En forklaring herpå er, at det sammen med en hvis afstand mellem punkter og linjestykker skyldes de isolerede punkter og klyngetendenserne, som kan påvirke fordelingen af vinkler væsentligt.



Figur 4.5 *– Fordelingen af vinkler mellem hvert punkt og dets to nærmeste naboer for henholdsvis det reducerede datasæt (venstre) og hele datasættet (højre).*

For at sammenholde vinkelfordelingen for datasættet med genererede punktmønstre er der kørt et antal simulationer, og for hver af disse er fordelingen af vinkler blevet beregnet med henblik på at opstille et 95%-interval. Pga. det varierende antal punkter i hver simulation udregnes det i %, hvor mange vinkler, der er i hvert interval, hvorved der fås et normaliseret resultat i forhold til datasættet.



Figur 4.6 *– Fordelingen af vinkler for henholdsvis det reducerede datasæt (venstre) og hele datasættet (højre). Toppen af hver søjle viser maksimum, og den nederste streg viser minimum for de simulerede punktmønstre. Det grå område angiver et 95%-interval, og den prikkede streg angiver værdien for pågældende interval for datasættet.*

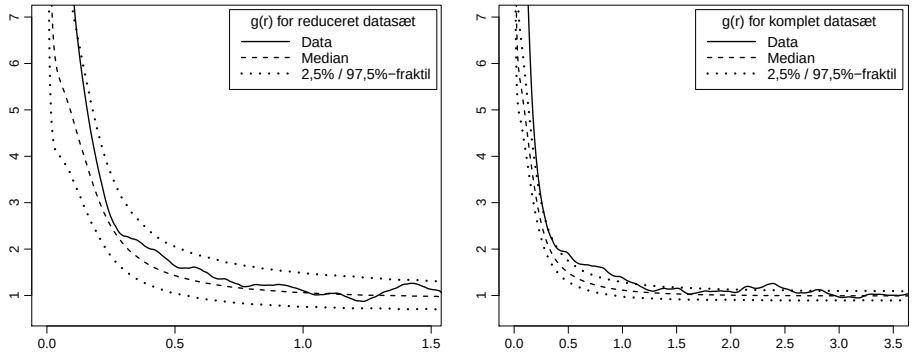
Resultatet ses på figur 4.6, hvor der til simulation af modellen igen er anvendt de estimerede gennemsnit fra tabel 3.17. Denne type modelkontrol er blevet implementeret i `R`, og bilag A.3.1 indeholder koden, som har genereret det venstre plot. For det reducerede datasæt ses der at være god sammenhæng mellem model og data, da værdien for datasættet i alle tilfælde ligger klart indenfor 95%-intervallet. For hele datasættet er sammenhængen dog ikke helt så god, idet datasættets vinkelfordeling for interval 3 og 4 ligger udenfor 95%-intervallet, og for de sidste to intervaller ligger det kun lige netop indenfor 95%-intervallet og kan være decideret svært at se på figuren. Samlet fås det dermed, at denne type modelkontrol giver et tilsvarende resultat som i forrige afsnit i form af, at der er god korrespondens for det reducerede datasæt, men den er mindre god for det komplette datasæt.

4.3 Deskriptive størrelser

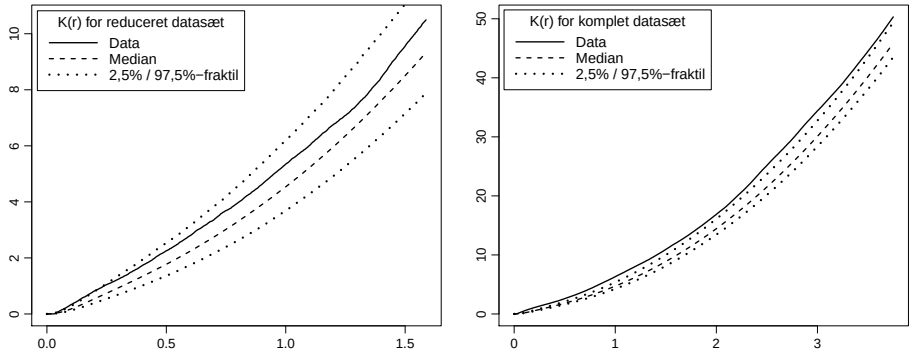
I dette afsnit vil de deskriptive størrelser fra afsnit 1.2 blive anvendt til at undersøge sammenhængen mellem modellen og datasættet. Pga. den komplicerede tæthed for modellen (se (2.3)), så vurderes det ikke, at det umiddelbart er muligt analytisk at bestemme udtryk for de forskellige deskriptive størrelser, og der anvendes derfor kun estimater. Da modellen er stationær, så er det muligt at estimere alle de 6 deskriptive størrelser $g(r), K(r), L(r), F(r), G(r)$ samt $J(r)$, og det gøres i praksis ved brug af `spatstat`-pakkens kommandoer `pcf.ppp()`, `Kest()`, `Lest()`, `Fest()`, `Gest()` og `Jest()`. I bilag A.3.2 ses `R`-koden til bestemmelse af parkorrelationsfunktionen $g(r)$ samt fraktiler for det reducerede datasæt, de øvrige estimater er lavet på tilsvarende vis.

På de følgende to sider ses på figur 4.7–4.12 plot af de deskriptive størrelser for henholdsvis det reducerede og det komplette datasæt. Der er ud over den estimerede funktion for data også angivet et 95%-interval samt median baseret på 5000 simulationer af modellen, hvor der som tidligere er anvendt de estimerede parametre fra tabel 3.17. De forskellige intervaller af r er i hvert tilfælde sat til det interval, som anbefales af den enkelte estimationsfunktion i `spatstat`. Som interval for funktionsværdien er i hvert tilfælde valgt en passende værdi, så mest mulig information ses på plottet. Mht. kantkorrektion er der for g, K og L anvendt *Ripley's isotropic*, og for F, G og J benyttes *Kaplan-Meier*, da de er mest velegnede til rektangulære observationsvinduer ifølge dokumentationen til `spatstat`.

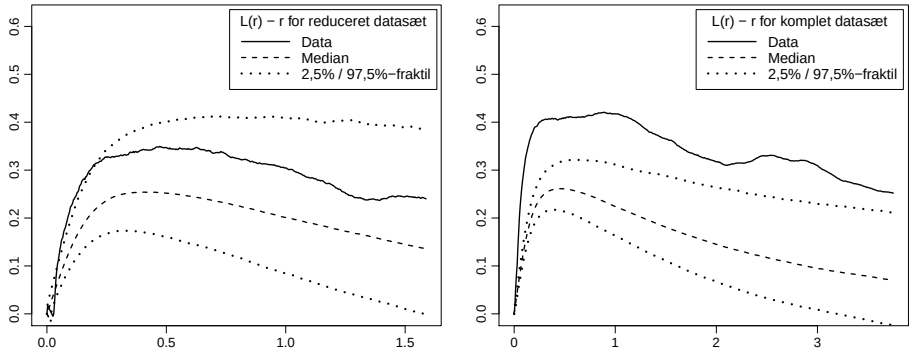
Parkorrelationen $g(r)$ ses for det reducerede datasæt at være i god overensstemmelse med modellen for $r > 0{,}2$, hvor den ligger indenfor 95%-intervallet. For



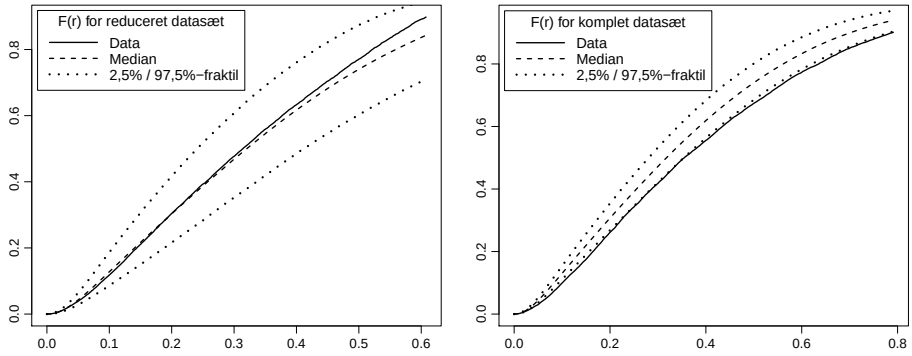
Figur 4.7 *– Parkorrelationsfunktionen $g(r)$ for det reducerede (venstre) og det komplette datasæt (højre). Fraktilerne er baseret på 5000 simulationer.*



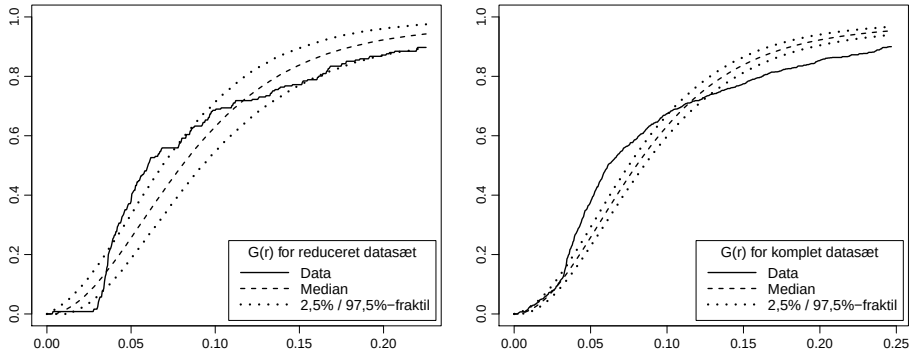
Figur 4.8 *– $K(r)$ for det reducerede (venstre) og det komplette datasæt (højre). Fraktilerne er baseret på 5000 simulationer.*



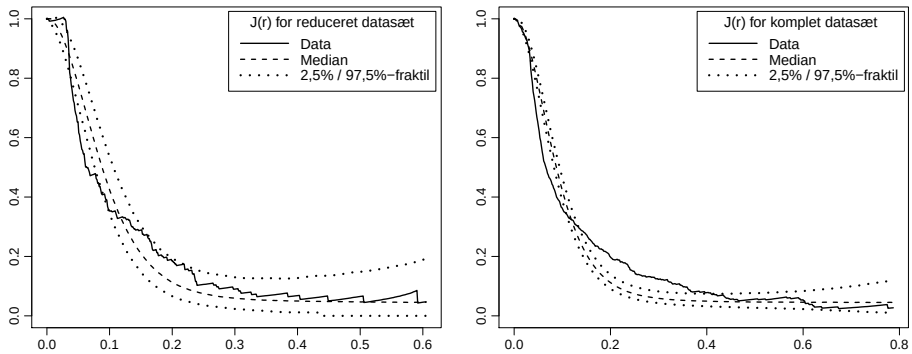
Figur 4.9 *– $L(r)-r$ for det reducerede (venstre) og det komplette datasæt (højre). Fraktilerne er baseret på 5000 simulationer.*



Figur 4.10 *– Tomrumsfunktionen $F(r)$ for det reducerede (venstre) og det komplette datasæt (højre). Fraktilerne er baseret på 5000 simulationer.*



Figur 4.11 *– Nærmeste nabo-funktionen $G(r)$ for det reducerede (venstre) og det komplette datasæt (højre). Fraktilerne er baseret på 5000 simulationer.*



Figur 4.12 *– $J(r)$ for det reducerede (venstre) og det komplette datasæt (højre). Fraktilerne er baseret på 5000 simulationer.*

$r < 0{,}2$ er $g(r)$ for data større, og det kan skyldes de tidligere omtalte klyngetendenser, der netop optræder for små r flere steder, og som modellen er mindre velegnet til at repræsentere. For hele datasættet er der en mindre god sammenhæng, idet $g(r)$ ligger udenfor eller tæt på kanten af 95%-intervallet for de fleste værdier af r. Stort set tilsvarende forhold ses for K-funktionen og for $L(r)-r$, hvilket ikke er overraskende, da der jf. sætning 1.20 og 1.22 er sammenhæng mellem $\mathcal{K}$, K og g, når g er invariant under parallelforskydning/isotropisk, og L er defineret ud fra K. Ses der på selve funktionsværdierne for data, så fremgår det som forventet, at der er klyngetendenser for tilpas små r, hvor henholdsvis $g(r) > 1$ og $L(r) - r > 0$.

De afstandsbaserede deskriptive størrelser F, G og J udviser også en blandet grad af korrespondens mellem data og model. For det reducerede datasæt ses F for det reducerede datasæt i høj grad at være sammenfaldende med modellen, idet den ligger tæt på medianen for alle r, men for det komplette datasæt ligger den lige udenfor 95%-intervallet. G er derimod for både det reducerede og komplette datasæt kun i god overensstemmelse med modellen for enkelte værdier af r, og det samme er også altovervejende tilfældet for J, der dog for tilpas store r ligger indenfor 95%-intervallet. Mht. funktionsværdierne for J, så ses igen indikation af klyngetendenser, idet $J(r) < 1$ for stort set alle r, dog ses der for det reducerede datasæt et lille interval tæt på $r = 0$, hvor $J(r)$ lige netop er større end 1.

Samlet set giver de deskriptive størrelser et tilsvarende billede som i de to forrige afsnit, nemlig at der for de fleste tilfælde er en god sammenhæng mellem data og model for det reducerede datasæt, mens den er mindre god for det fulde datasæt. Det må på baggrund af disse forhold derfor konstateres, at modellen i rimelig udtrækning er i stand til at producere punktmønstre af samme type som det reducerede datasæt, men de estimerede parameterværdier er ikke velegnede til at blive overført til det fulde datasæt.

Konklusion og diskussion

Der er i dette speciale forsøgt en ny tilgang til at opstille en parametrisk model for rumlige punktmønstre med lineære strukturer med udgangspunkt i et datasæt om gravhøjes placering i Vestjylland. Det har igennem arbejdet med at udføre inferens for modellen i kapitel 3 vist sig, at der er flere typer af udfordringer, der skal tackles, for at få et brugbart resultat. Derfor har det været nødvendigt at fokusere på en mindre del af datasættet, men den udførte inferens herpå har gennem modelkontrollen i kapitel 4 dog også i flere henseender vist sig at lede til god overensstemmelse mellem model og data. Ved forsøg på at anvende de estimerede parametre på det fulde datasæt er der konstateret en mindre god sammenhæng, men på baggrund af de fornuftige resultater for det reducerede datasæt må det forventes, at der vil være en bedre sammenhæng mellem model og data, hvis der blev gennemført estimation for hele datasættet.

For at komme nærmere en muliggørelse af dette, så er der bl.a. en række implementations-tekniske forhold, der kan vise sig fordelagtige at undersøge nærmere. I løbet at speciale-perioden er `R` 2.14 udkommet, hvor en af de væsentlige nyheder er, at alle indbygge funktioner og de fleste pakker nu er eller kan kompileres til såkaldt bytekode, der afvikles hurtigere end `R`-kode fortolket fra kommandolinjen. Eksperimentering med estimations-algoritmen har dog kun givet små hastighedsforbedringer på få procent. En anden potentiel brugbar udvikling indenfor `R` er den stigende fokus på udnyttelse af flere kerner, hvor der er flere bud på udnyttelse af, at mange computere i dag har flere kerner.

En af dem er pakken `multicore`, [Urbanek, 2011], som giver en multikerneversion af `lapply()`, der netop er en af de centrale og mest tidsforbrugene funktioner i estimations-algoritmen. Brugere af pakken rapporterer om stort set lineær skalering i forhold til antallet af kerner, men en test på estimationsalgoritmen viser på trods af forøget cpu-forbrug tværtimod en forringelse af hastigheden på ca. 50%, så det må konstateres, at denne pakke i sin nuværende form ikke er anvendelig i denne situation. Den indbyrdes afhængighed mellem trinene i Metropolis-Hastings og Gibbs sampling gør det svært at parallelisere algoritmen, og der er derfor lille forventning til, at sådanne generelle pakker vil være i

stand til at give afgørende hastighedsforbedringer. Da der dermed ikke har været nogen succes med disse nye teknologiske muligheder, så tyder det på, at det her og nu vil kræve andre tilgange at udføre inferens for hele datasættet. Eksempelvis er der stadig mulighed for yderligere optimering af beregningen af $\rho(\cdot)$, og det forventes også at kunne hjælpe, hvis der manuelt indlægges linjestykker for hele datasættet til brug ved starten af algoritmen. Det kan også forsøges at indhente information om kendte forhistoriske veje, som kan indlægges og evt. fastholdes, således at der fås en hybrid BDM-algoritme, hvor der både er faste og varierende linjestykker.

I stedet for kun at fokusere på ren forøgelse af hastigheden ad teknologisk vej, så vil der også være muligheder i at undersøge effekten af mere fundamentale ændringer af algoritmens virkemåde. En af udfordringerne er, at der indsættes langt flere linjestykker, end hvad der intuitivt forekommer realistisk, og der er flere bud på, hvordan det kan modvirkes. En simpel mulighed er at lade $q(\cdot)$ være en stigende funktion af antallet af iterationer, således at der i starten af algoritmen er mange birth-death-opdateringer, og der gradvist fås en større andel af move-opdateringer. På denne vis bør antallet af linjestykker hurtigere blive stabiliseret, og ved løbende at nedsætte sandsynligheden for birth-death, vil antallet på sigt variere minimalt.

Måden move-opdateringer laves på er også en kandidat til eksperimentation, og i [Allen & Tildesley, 1989] anbefales det at foretage en flytning indenfor et rektangel omkring det udvalgte y_i. En sådan tilgang vil måske kunne give flere mindre justeringer af linjestykkernes placering, hvor de lokalt bliver flyttet til mere optimale steder frem for, af der ved en birth-death-opdatering oprettes nye linjestykker i nærheden. Denne teknik vil yderligere kunne udbygges til at lave opdateringen i tre trin, hvor linjestykket først parallelforskydes, og derefter kan der evt. ændres på vinkel og længde, hvor hver type ændring foretages vha. Metropolis-Hastings. Som supplement eller alternativ kan *multiple-try metropolis*, [Liu et al., 2000], forsøges, hvor der ved hver opdatering udtrækkes flere forslag. Eksempelvis kan der udtrækkes et antal vinkler og vejlængder, hvor så det mest sandsynlige par bruges til at afgøre, om opdateringen accepteres. Dette forventes at kunne medvirke til at undgå, at nye linjestykker placeres på tværs af et intuitiv bedre linjestykke. Til at vurdere effekten af disse forslag til ændringer, kan der ud over betragtning af traceplots også benyttes andre metoder. Eksempelvis er der specificeret en test i [Geweke, 1991], hvor der anvendes en del af hver ende af Markovkæden efter burn-in. Hvis der er tilpas stor lighed imellem de to, så må kæden være konvergeret, og således kan testen benyttes til at vurdere et valgt burn-in.

Den største udfordring for at udføre inferens har vist sig at være BDM-delen af algoritmen, hvor β har tendens til nogle gange at stige i en grad, så der oprettes mange meget korte linjestykker. Dette er i sig selv ikke et argument for at ændre på modellen, men modelkontrollen for det reducerede datasæt viste både forekomst af god og mindre god korrespondens mellem model og data, så der er grund til også at forsøge og justere på modellen. Det forekommer som en oplagt mulighed at undersøge effekten af at udskifte ekponentialfordelingen af vejlængder med noget andet. Eksempelvis vil brugen af en normal- eller ligefordeling give mulighed for at undgå problemet med de mange korte linjestykker, da deres middelværdi kan fastholdes, og spredningen kan justeres efter behov.

En ændring af denne type vil ikke påvirke modellens virkemåde i særlig grad, og det vil være relativt nemt at forsøge. Til gengæld er der to andre forhold, som vil kræve væsentligt mere fundamentale ændringer at få indeholdt i modellen. Det drejer sig bl.a. om selve placeringen af linjestykkerne, hvor der anvendes en mærket homogen PPP til at placere dem i modellen, mens det i datasættet fremstår tydeligt, at der er tale om en inhomogen placering. Derfor vil det være interessant, om der kunne vælges en mere velegnet metode til at placere linjestykkerne, men det kan risikere at komplicere modellen i en grad, så udførsel af inferens bliver for vanskelig. Yderligere kan modellen blive for specifikt rettet mod det konkrete datasæt og miste for meget generalitet til stadig at være interessant i andre sammenhænge. Tilsvarende forhold gør sig gældende omkring klyngetendenserne, hvor en klyngeproces godt vil kunne indbygges i modellen til at få mere kompakte klynger, men det giver samme risici mht. kompleksitet og generalitet.

Når der i denne diskussion har været så stor fokus på at belyse mulige forbedringer til estimation og model, så skyldes det anvendelsesperspektiverne for modellen. De ovennævnte betragtninger virker måske umiddelbart mest interessante i en ren matematisk-statistisk sammenhæng, men de bunder også i høj grad i ønsket om at kunne producere resultater med relation til virkeligheden. Flere gange gennem specialet er det nævnt, at der er en problematik omkring den manglende realisme omkring placeringen af linjestykker, og det hæmmer anvendeligheden af modellen til et virkeligt formål. En anvendelse inspireret af datasættet kan være i arkæologisk sammenhæng, hvor estimaterne for parametrene kan bruges til at anslå forhold om veje eller gravhøje i andre områder. Det er kun muligt at gøre på forsvarlig vis, hvis de enkelte parametre bliver estimeret til en værdi, der har en reel sammenhæng med virkeligheden. Derfor er det vigtigt ikke kun at betragte modellen ud fra et teoretisk synspunkt, da statistik nu engang i overvejende grad er en anvendt disciplin.

Referencer

[Allen & Tildesley, 1989] M. P. Allen og D. J. Tildesley. *Computer Simulation of Liquids*. Oxford University Press, 1. udgave, 1989. ISBN: 0-19-855645-4.

[Baddeley & Turner, 2005] Adrian Baddeley og Rolf Turner. *Spatstat: an R package for analyzing spatial point patterns*. Journal of Statistical Software, 12, nr. 6, s. 1–42, 2005. URL: http://www.jstatsoft.org/v12/i06. ISSN 1548-7660.

[Berthelsen & Møller, 2004] Kasper K. Berthelsen og Jesper Møller. *A short diversion into the theory of Markov chains, with a view to Markov chain Monte Carlo methods*, June 2004. URL: http://people.math.aau.dk/~kkb/Gulrapport.pdf.

[Bourke, 1988] Paul Bourke. *Minimum Distance between a Point and a Line*, October 1988. URL: http://paulbourke.net/geometry/pointline/.

[Casella & George, 1992] Geoge Casella og Edward I. George. *Explaining the Gibbs Sampler*. The American Statistician, 46, nr. 3, s. 167–174, 1992. URL: http://www.jstor.org/pss/2685208.

[Geman & Geman, 1984] Stuart Geman og Donald Geman. *Stochastic Relaxation, Gibbs Distributions, and the Bayesian Restoration of Images*. IEEE Transactions on Pattern Analysis and Machine Intelligence, 6, nr. 6, s. 721–741, 1984. URL: http://dx.doi.org/10.1109/TPAMI.1984.4767596.

[Geweke, 1991] John Geweke. *evaluating the accuracy of sampling-based approaches to the calculation of posterior moments*. Staff Report 148, Federal Reserve Bank of Minneapolis, 1991. URL: http://ideas.repec.org/p/fip/fedmsr/148.html.

[Geyer & Møller, 1994] Charles H. Geyer og Jesper Møller. *Simulation Procedures and Likelihood Inference for Spatial Point Processes*. Scandinavian Journal of Statistics, 21, nr. 4, s. 359–373, 1994. URL: http://www.jstor.org/stable/4616323.

[Haals & Jensen, 2009] Niels Peter Haals og Per Jensen. *Rumlige punkt- og linjeprocesser – En introduktion med henblik på beskrivelse af punktmønstre med linjetendenser og en modelopstilling for datasæt af gravhøje*. Speciale, Aalborg Universitet, 2009. URL: http://projekter.aau.dk/projekter/files/17652662/master.pdf.

[Hastings, 1970] W. K. Hastings. *Monte Carlo Sampling Methods Using Markov Chains and Their Applications*. Biometrika, 57, nr. 1, s. 97–109, 1970. URL: http://www.jstor.org/stable/2334940.

[Lee, 2004] Peter M. Lee. *Bayesian Statistics – an introduction*. John Wiley & Sons, Ltd., 3. udgave, 2004. ISBN: 978-0-470-68920-2. URL: http://www-users.york.ac.uk/~pml1/bayes/book.htm.

[Liu et al., 2000] Jun S. Liu, Faming Liang og Wing Hung Wong. *The Multiple-Try Method and Local Optimization in Metropolis Sampling*. Journal of the American Statistical Association, 95, nr. 449, s. 121–134, 2000. URL: http://www.jstor.org/stable/2669532.

[Metropolis et al., 1953] Nicholas Metropolis, Arianna W. Rosenbluth, Marshall N. Rosenbluth, Augusta H. Teller og Edward Teller. *Equation of State Calculations by Fast Computing Machines*. Journal of chemical physics, 21, nr. 6, s. 1087–1092, 1953. URL: http://link.aip.org/link/doi/10.1063/1.1699114.

[Møller & Rasmussen, 2010] Jesper Møller og Jakob Gulddahl Rasmussen. *A Sequential Point Process Model and Bayesian Inference for Spatial Point Patterns with Linear Structures*. Centre for Stochastic Geometry and Advanced Bioimaging, CSGB Research Reports no. 06, Juli 2010. URL: http://data.imf.au.dk/publications/csgb/2010/imf-csgb-2010-06.pdf.

[Møller & Waagepetersen, 2004] Jesper Møller og Rasmus Plenge Waagepetersen. *Statistical Inference and Simulation for Spatial Point Processes*. Chapman & Hall/CRC, 1. udgave, 2004. ISBN: 1-58488-265-4. URL: http://www.crcnetbase.com/isbn/9780203496930.

[Olofsson, 2005] Peter Olofsson. *Probability, Statistics, and Stochastic Processes*. John Wiley & Sons, Inc., 1. udgave, 2005. ISBN: 978-0-471-67969-1.

[Plummer et al., 2006] Martyn Plummer, Nicky Best, Kate Cowles og Karen Vines. *CODA: Convergence Diagnosis and Output Analysis for MCMC*. R News, 6, nr. 1, s. 7–11, 2006. URL: http://CRAN.R-project.org/doc/Rnews/Rnews_2006-1.pdf.

[R Development Core Team, 2011] R Development Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2011. URL: http://www.R-project.org. ISBN 3-900051-07-0.

[Roberts et al., 1997] G. O. Roberts, A. Gelman og W. R. Giles. *Weak convergence and optimal scaling of random walk Metropolis algorithms*. The Annals of Applied Probability, 7, nr. 1, s. 110–120, 1997. URL: http://www.jstor.org/stable/2245134.

[Rudin, 1987] Walter Rudin. *Real and Complex Analysis*. WCB/McGraw-Hill, 3. udgave, 1987. ISBN: 978-0-07-054234-1.

[Urbanek, 2011] Simon Urbanek. *multicore: Parallel processing of R code on machines with multiple cores or CPUs*, 2011. URL: http://www.rforge.net/multicore/.

[Walsh, 2004] Bruce Walsh. *Markov Chain Monte Carlo and Gibbs Sampling*. Lecture notes, april 2004. URL: http://nitro.biosci.arizona.edu/courses/EEB581-2006/handouts/Gibbs.pdf.

R-kode A

A.1 R-kode til simulation af modellen

```
1  generer_antal_veje <- function(lambda, xrange, yrange) {
2    areal <- (xrange[2] - xrange[1]) * (yrange[2] - yrange[1]) * 1.96 # arealet at
         B' (arealet vokser med faktor 1+4n+4n^2, når der tillægges n% i hver "ende")
3    lambda <- lambda * areal # lav nyt lambda = forventet antal veje for hele arealet
4    antal_veje <- 0 # initialiser, så while-løkke kan bruges
5    while (antal_veje < 1) {
6      antal_veje <- rpois(n=1,lambda=lambda) } # generer antallet af veje
7      return(antal_veje) } # end function
8
9  generer_veje <- function(antal_veje, xrange, yrange) { # generering af midtpunkter
10   Y <- matrix(nrow=antal_veje, ncol=2,
         dimnames=list(c(),c("midt_x","midt_y"))) # opret matrix til koordinater
11   xlængde <- xrange[2]-xrange[1] ; ylængde <- yrange[2]-yrange[1] # udvid B til B'
12   xrange_ny <- c(xrange[1] - xlængde*0.2 , xrange[2] + xlængde*0.2)
13   yrange_ny <- c(yrange[1] - ylængde*0.2 , yrange[2] + ylængde*0.2)
14   for(i in 1:antal_veje) { # for hver vej
15     temp_x <- runif(1,min=xrange_ny[1],max=xrange_ny[2]) # generer uniform x-værdi
16     temp_y <- runif(1,min=yrange_ny[1],max=yrange_ny[2]) # generer uniform y-værdi
17     Y[i,] <- c(temp_x,temp_y) # gem i Y
18   } # end for
19   return(Y) } # end function
20
21 generer_mærker <- function(antal_veje, beta) { # generering af vinkler og længder
22   vinkler <- runif(antal_veje, min=0, max=pi) # generer vinkler
23   længder <- rexp(antal_veje, rate=beta) # generer længder
24   mærker <- cbind(vinkler,længder) # sæt dem sammen
25 } # end function
26
27 beregn_vej_start_slut <- function(Y) { # beregning af vejenes start- og slut-koordinater
28   Z <- matrix(nrow=length(Y[,1]), ncol=4, dimnames=list(c(),
         c("start_x","start_y","slut_x","slut_y"))) # opret hjælpematrix
29   for(i in 1:length(Y[,1])) { # beregn for hver vej
30     Z[i,1:2] <- c(Y[i,1] - 1/2*Y[i,4]*cos(Y[i,3]) , Y[i,2] -
           1/2*Y[i,4]*sin(Y[i,3])) # startpunktet er midtpunktet minus den halve længde gange
           en enhedsretningsvektor
31     Z[i,3:4] <- c(Y[i,1] + 1/2*Y[i,4]*cos(Y[i,3]) , Y[i,2] +
           1/2*Y[i,4]*sin(Y[i,3])) # slutpunktet er midtpunktet plus den halve længde gange en
           enhedsretningsvektor
32   } # end for
33   return(Z) } # end function
34
```

```
35 generer_punkter_pr_vej <- function(Y, Z, gamma) { # generer antal punkter for hver vej
36   vejpunkter <- matrix(nrow=length(Y[,1]), ncol=1,
         dimnames=list(c(),c("vejpunkter"))) # opret som matrix til senere brug
37   for(i in 1:length(Y[,1])) { # for hver vej
38     gamma_ny <- gamma * Y[i,4] # lav nyt gamma = middelværdi for antal punkter for hele vejen
39     vejpunkter[i] <- rpois(1,gamma_ny) } # end for
40   Z <- cbind(Z,vejpunkter) } # end function
41
42 generer_punkter <- function(Y, Z) { # generering af punkter på vejene
43   X <- matrix(ncol=2,nrow=0,dimnames=list(c(),c("x", "y"))) # til koordinaterne
44   for(i in 1:length(Y[,1])) { # for hver vej
45     if(Z[i,5] > 0) { # hvis der skal genereres noget
46       for(j in 1:Z[i,5]) { # generer så mange punkter
47         temp <- runif(1,min=0,max=Y[i,4]) # generer uniformt sted på linjen
48         X <- rbind(X,c(Z[i,1] + temp*cos(Y[i,3]) ,Z[i,2] +
               temp*sin(Y[i,3]))) # beregn koordinaten til dette sted fra linjens begyndelse
49       } #end for
50     } # end if
51   } #end for
52   return(X) } # end function
53
54 generer_forskydninger <- function(X, Y, Z, sigma) { # forskydning af punkterne
55   k <- 1 # hjælpetæller
56   for(i in 1:length(Y[,1])) { # for hver vej
57     if(Z[i,5] > 0) { # hvis den har nogle punkter
58       tværvektor <- c(-sin(Y[i,3]) , cos(Y[i,3])) # beregn tværvektoren
59       for(j in k:(k+Z[i,5])-1) { # minus 1, da der ellers vil være overlap
60         temp <- rnorm(1,sd=sigma) # generer normalfordelt forskydning
61         X[j,] <- temp*(-tværvektor) + c(X[j,1],X[j,2]) # gem de forskudte koordinater
62       } # end for
63       k <- k + Z[i,5] } # end if (ryk frem til starten af næste vejs punkter)
64   } # end for
65   return(X) } # end function
66
67 generer_baggrundspunkter <- function(X, delta, xrange, yrange) { # homogen
       PPP med intensitet delta til baggrundspunkterne (behøver ikke at blive simuleret på B')
68   areal <- (xrange[2] - xrange[1]) * (yrange[2] - yrange[1]) # areal af B
69   delta <- delta * areal # det forventede antal punkter for hele arealet
70   antal_punkter <- rpois(n=1,lambda=delta) # generer antallet af baggrundspunkter
71   if ( antal_punkter > 0) { # if da lykken ellers kører 2 gange
72     for(i in 1:antal_punkter) { # simuler punkterne
73       temp_x <- runif(1,min=xrange[1],max=xrange[2]) # generer uniform x-værdi
74       temp_y <- runif(1,min=yrange[1],max=yrange[2]) # generer uniform y-værdi
75       X <- rbind(X,c(temp_x,temp_y)) # tilføj til X
76     } # end for
77   } # end if
78   return(X) } # end function
79
80 beskæring <- function(X, xrange, yrange) { # Beskær punkter udenfor observationsvinduet
81   X <- X[(X[,1] >= xrange[1] & X[,1] <= xrange[2] & X[,2] >= yrange[1] &
       X[,2] <= yrange[2]),] } # end function
```

R-kode A.1 *– model-funktioner.R, kode til de funktioner, som genererer simulationer af modellen.*

A.2 R-kode til estimation af parametre

```
initialiser <- function() { # oprettelse af datastrukturer mv.
  Y <<- matrix(,,8, dimnames=list(c(), c("midt_x", "midt_y", "vinkel",
       "længde", "start_x", "start_y", "slut_x", "slut_y")))
  D2 <<- matrix(,1,length(X[,1])) ; D2 <<- D2[-1,] # opret D2 til at indeholde d_[i,j]
  resultat <<- matrix(, N, 6, dimnames=list(c(), c("lambda", "beta",
       "gamma", "delta", "sigma", "linjer"))) # matrix til resultatet
  acceptrater <<- matrix(, N, 5, dimnames=list(c(), c("lambda", "beta",
       "gamma", "delta", "sigma"))) # matrix til løbende acceptrater
  accept.lambda <<- accept.beta <<- accept.gamma <<- accept.delta <<-
       accept.sigma <<- 0 # tællere til antal accepterede opdateringer af parametrene
  lambda.aktuel <<- lambda.start ; beta.aktuel <<- beta.start ;
       gamma.aktuel <<- gamma.start # indlæs startværdier
  delta.aktuel <<- delta.start ;  sigma.aktuel <<- sigma.start
  Y[1,] <<- generer_linjestykke() # lav n linjestykker til at starte med – lav første linjestykke
  D2_beregning() # udregn d_{i,j} for første linjestykke
  for (i in 1:(n-1)) { # lav de resterende linjestykker
    Y <<- rbind(Y,generer_linjestykke()) ; D2_beregning() } # end for
  rho.aktuel<<-rho(Y,D2,gamma.aktuel,sigma.aktuel,delta.aktuel) # optimering
} #end function

d <- function(px, py, x1, y1, x2, y2) { # beregning af afstand mellem punkt og linjestykke
  u <- (((px - x1) * (x2 - x1)) + ((py - y1) * (y2 - y1))) /
       ((x2-x1)^2+(y2-y1)^2) # beregning af parameteren for skæringspunktet
  if((u < 0) || (u > 1)) { # punktet har ikke nogen afstand til linjestykket
    return(Inf) # Inf er passende, da dnorm(Inf,...) = 0 (hardcoded i dnorm.c)
  } # end if
  else { # punktet har en afstand til linjestykket
    ix <- x1 + u * (x2 - x1) ; iy <- y1 + u * (y2 - y1) # normalvektors koordinater
    afstand <- sqrt((ix-px)^2+(iy-py)^2) # længden af normalvektor fra linjestykke til punkt
  } # end else
} # end function

generer_linjestykke <- function(){
  midt_x <- runif(1,min=xrange[1],max=xrange[2]) # generer uniform x–værdi
  midt_y <- runif(1,min=yrange[1],max=yrange[2]) # generer uniform y–værdi
  vinkel <- runif(1, min=0, max=pi) # generer vinkel
  længde <- rexp(1, rate=beta.aktuel) # generer længde
  start_x <- midt_x - 1/2*længde*cos(vinkel) # beregning af start– og slut–koordinater
  start_y <- midt_y - 1/2*længde*sin(vinkel)
  slut_x <- midt_x + 1/2*længde*cos(vinkel)
  slut_y <- midt_y + 1/2*længde*sin(vinkel)
  return (c(midt_x,midt_y,vinkel,længde,start_x,start_y,slut_x,slut_y))
} # end function

D2_beregning <- function() { # Tag den sidste række i Y og tilføj en ny række i D2
  lin <- length(Y[,1]) # det index, der skal arbejdes på
  temp <- numeric(length(X[,1])) # vektor til længderne
  for (i in 1:length(X[,1])) { # beregn længden til linjestykket for hvert punkt
    temp[i] <- d(X[i,1], X[i,2], Y[lin,5], Y[lin,6], Y[lin,7], Y[lin,8])
  } # end for
  D2 <<- rbind(D2,temp) # opdater D2 med resultatet
} # end function
```

```
47
48 D1_beregning <- function(Y1,D1) { # Tag den sidste række i Y1 og lav en ny række i D1
49   lin <- length(Y1[,1]) # D1_beregning() er nødvendig pga. optimering med globale datastrukturer
50   temp <- numeric(length(X[,1]))
51   for (i in 1:length(X[,1])) {
52     temp[i] <- d(X[i,1], X[i,2], Y1[lin,5], Y1[lin,6], Y1[lin,7], Y1[lin,8])
53   } # end for
54   return(rbind(D1,temp))
55 } # end function
56
57 l <- function(Y) { sum(Y[,4]) } # beregn den samlede længde af linjestykkerne i Y
58
59 rho <- function(Y, D2, gamma, sigma, delta) { # rho() i hidtil hurtigste udgave
60   rho <- unlist(lapply(D2, dnorm, sd=sigma)) # laver én-dimensional vektor
61   dim(rho) <- c(length(Y[,1]),length(X[,1])) # konverter den til matrix
62   indre.sum <- colSums(rho) # summer søjlerne for at få den indre sum
63   sum(log(gamma*indre.sum+delta)) # beregn den ydre sum
64 } # end function
65
66 move <- function() { # move-opdatering
67   Im <- sample(1:length(Y[,1]),1) # udtræk linjestykke til flytning
68   Y1 <- Y[-Im,] ; Y1 <- rbind(Y1,generer_linjestykke()) # fjern det og lav et nyt
69   D1 <- D2[-Im,] ; D1 <- D1_beregning(Y1,D1) # fjern afstande og udregn nye til forslaget
70   rho.forslag <<- rho(Y1,D1,gamma.aktuel,sigma.aktuel,delta.aktuel)
71   log.hf <- (beta.aktuel+gamma.aktuel)*(l(Y) - l(Y1)) + rho.forslag -
         rho.aktuel # beregn Hastingsforholdet
72
73   if(log(runif(1)) <= log.hf) { # hvis Hastingsforholdet er stort nok
74     Y <<- Y1 ; D2 <<- D1 ; rho.aktuel <<- rho.forslag # accepter forslaget og gem
75   } # end if
76   resultat[i,6] <<- length(Y[,1]) # opdater antal linjer
77 } # end function
78
79 birth_death <- function() { # birth-death-opdatering
80   if(runif(1) <= p){ # kør birth-opdatering
81     Y1 <- rbind(Y,generer_linjestykke()) # tilføj det nye forslag
82     D1 <- D2 ; D1 <- D1_beregning(Y1,D1) # kopier D2 og udregn nye afstande
83     rho.forslag <<- rho(Y1,D1,gamma.aktuel,sigma.aktuel,delta.aktuel)
84     log.hf <- (beta.aktuel+gamma.aktuel)*(l(Y) - l(Y1)) + log((1-p)*areal)
           - log(p*(length(Y[,1])+1)) + rho.forslag - rho.aktuel +
           log(lambda.aktuel*beta.aktuel) # beregn Hastingsforholdet
85
86     if(runif(1) <= log.hf) { # hvis Hastingsforholdet er stort nok
87       Y <<- Y1 ; D2 <<- D1 ; rho.aktuel <<- rho.forslag # accepter forslaget og gem
88     } # end if
89   } # end if
90
91   else { # kør death-opdatering
92     if(length(Y[,1]) == 2){ } # skip hvis kun 2 linjestykker (R ændrer Y til vektor ved kun 1 række)
93     else {
94       Im <- sample(1:length(Y[,1]),1) # # udtræk linjestykke til død
95       Y1 <- Y[-Im,] ; D1 <- D2[-Im,] # fjern det samt de tilhørende afstande
96       rho.forslag <<- rho(Y1,D1,gamma.aktuel,sigma.aktuel,delta.aktuel)
```

```
97      log.hf <- (beta.aktuel+gamma.aktuel)*(l(Y) - l(Y1)) +
            log(p*length(Y[,1])) - log((1-p)*areal) + rho.forslag - rho.aktuel
            - log(lambda.aktuel*beta.aktuel) # beregn Hastingsforholdet
98
99      if(runif(1) <= log.hf) { # hvis Hastingsforholdet er stort nok
100       Y <<- Y[-Im,] ; D2 <<- D2[-Im,] ; rho.aktuel <<- rho.forslag # accepter
101     } #  end if
102   } # end else
103   } # end else
104   resultat[i,6] <<- length(Y[,1]) # opdater antal linjer
105 } # end function
106
107 opdater_lambda <- function() { # opdatering af lambda
108   lambda.forslag <<- -1 # initialiser, så while-løkke kan bruges til at sikre positivt forslag
109   while (lambda.forslag < 0) { lambda.forslag <<-
          rnorm(1,lambda.aktuel,lambda.psi) } # udtræk forslag
110   prior <- dgamma(lambda.forslag,lambda.form,,lambda.start/lambda.form,T) -
          dgamma(lambda.aktuel,lambda.form,,lambda.start/lambda.form,T)
111   log.hf <- (lambda.aktuel - lambda.forslag)*areal +
          length(Y[,1])*log(lambda.forslag/lambda.aktuel) + prior # Hastingsforholdet
112   if(log(runif(1)) <= log.hf) { # opdater hvis Hastingsforholdet er stort nok
113     lambda.aktuel <<- lambda.forslag ; accept.lambda <<- accept.lambda + 1
114   } # end if
115   acceptrater[i,1] <<- accept.lambda / i ; resultat[i,1] <<- lambda.aktuel
116 } # end function
117
118 opdater_beta <- function() { # opdatering af beta
119   beta.forslag <<- -1 # initialiser, så while-løkke kan bruges til at sikre positivt forslag
120   while (beta.forslag < 0) { beta.forslag <<- rnorm(1,beta.aktuel,beta.psi)
          } # udtræk forslag
121   prior <- dgamma(beta.forslag,beta.form,,beta.start/beta.form,T) -
          dgamma(beta.aktuel,beta.form,,beta.start/beta.form,T)
122   log.hf <- (beta.aktuel - beta.forslag)*l(Y) +
          length(Y[,1])*log(beta.forslag/beta.aktuel) + prior # Hastingsforholdet
123   if(log(runif(1)) <= log.hf) { # opdater hvis Hastingsforholdet er stort nok
124     beta.aktuel <<- beta.forslag ; accept.beta <<- accept.beta + 1
125   } # end if
126   acceptrater[i,2] <<- accept.beta / i ; resultat[i,2] <<- beta.aktuel
127 } # end function
128
129 opdater_gamma <- function() { # opdatering af gamma
130   gamma.forslag <<- -1 # initialiser, så while-løkke kan bruges til at sikre positivt forslag
131   while (gamma.forslag < 0) { gamma.forslag <<-
          rnorm(1,gamma.aktuel,gamma.psi) } # udtræk forslag
132   rho.forslag <<- rho(Y,D2,gamma.forslag,sigma.aktuel,delta.aktuel)
133   prior <- dgamma(gamma.forslag,gamma.form,,gamma.start/gamma.form,T) -
          dgamma(gamma.aktuel,gamma.form,,gamma.start/gamma.form,T)
134   log.hf <- (gamma.aktuel - gamma.forslag)*l(Y) + rho.forslag - rho.aktuel
          + prior # Hastingsforholdet
135   if(log(runif(1)) <= log.hf){ # opdater hvis Hastingsforholdet er stort nok
136     gamma.aktuel <<- gamma.forslag ; accept.gamma <<- accept.gamma + 1 ;
            rho.aktuel <<- rho.forslag
137   } # end if
138   acceptrater[i,3] <<- accept.gamma / i ; resultat[i,3] <<- gamma.aktuel
139 } # end function
```

```
140
141 opdater_delta <- function() { # opdatering af delta
142   delta.forslag <<- -1 # initialiser, så while-løkke kan bruges til at sikre positivt forslag
143   while (delta.forslag < 0) { delta.forslag <<-
          rnorm(1,delta.aktuel,delta.psi) } # udtræk forslag
144   rho.forslag <<- rho(Y,D2,gamma.aktuel,sigma.aktuel,delta.forslag)
145   prior <- dgamma(delta.forslag,delta.form,,delta.start/delta.form,T) -
          dgamma(delta.aktuel,delta.form,,delta.start/delta.form,T)
146   log.hf <- (delta.aktuel - delta.forslag)*areal + rho.forslag - rho.aktuel
          + prior # Hastingsforholdet
147   if(log(runif(1)) <= log.hf){ # opdater hvis Hastingsforholdet er stort nok
148     delta.aktuel <<- delta.forslag ; accept.delta <<- accept.delta + 1 ;
            rho.aktuel <<- rho.forslag
149   } # end if
150   acceptrater[i,4] <<- accept.delta / i ; resultat[i,4] <<- delta.aktuel
151 } # end function
152
153 opdater_sigma <- function() { # opdatering af sigma
154   sigma.forslag <<- -1 # initialiser, så while-løkke kan bruges til at sikre positivt forslag
155   while (sigma.forslag < 0) { sigma.forslag <<-
          rnorm(1,sigma.aktuel,sigma.psi) } # udtræk forslag
156   rho.forslag <<- rho(Y,D2,gamma.aktuel,sigma.forslag,delta.aktuel)
157   prior <- dgamma(sigma.forslag,sigma.form,,sigma.start/sigma.form,T) -
          dgamma(sigma.aktuel,sigma.form,,sigma.start/sigma.form,T)
158   log.hf <- rho.forslag - rho.aktuel + prior # Hastingsforholdet
159   if(log(runif(1)) <= log.hf){ # opdater hvis Hastingsforholdet er stort nok
160     sigma.aktuel <<- sigma.forslag ; accept.sigma <<- accept.sigma + 1 ;
            rho.aktuel <<- rho.forslag
161   } # end if
162   acceptrater[i,5] <<- accept.sigma / i ; resultat[i,5] <<- sigma.aktuel
163 } # end function
164
165 burnin <- function(n) { # skær alt væk mindre end burn-in
166   resultat <<- resultat[-(1:n),] ; resultat <<- mcmc(resultat)
167   acceptrater <<- acceptrater[-(1:n),] ; acceptrater <<- mcmc(acceptrater)
168 } # end function
169
170 status <- function(n) { # print en status-opdatering
171   if (i %% s == 0) {
172     print(paste(i, lambda.aktuel, beta.aktuel, gamma.aktuel, delta.aktuel,
            sigma.aktuel), quote=F) # udskriv aktuel værdi af parametrene + aktuelt linjemønster
173     plot(X,pch=20,cex=0.75) ; segments(Y[,5],Y[,6],Y[,7],Y[,8], lwd=1.5)
174     flush.console() # sikrer at konsol-output kan ses i GUI-mode
175   } # end if
176 } # end function
177
178 afslutning <- function() { # lav til coda-mcmc-objeckter for nem håndtering af plots mv.
179   library(coda) ; resultat <<- mcmc(resultat)
180   acceptrater <<- mcmc(acceptrater) ; print(summary(resultat))
181 } # end function
```

R-kode A.2 – *estimation-funktioner.R, kode til de funktioner, der anvendes til estimation af parametrene.*

A.3 R-kode til modelkontrol

A.3.1 Fordeling af vinkler

```
library(spatstat) # spatstat skal bruges pga. nnwhich()
source("main-model.R") # indlæs koden til at simulere modellen

beregn_naboer <- function(X) { # lav matrix med hvert punkts 2 nærmeste naboer
  naboer <- matrix(,length(X[,1]),2,dimnames=list(c(),c("nabo_1","nabo_2")))
  naboer[,1] <- nnwhich(X,k=1) ; naboer[,2] <- nnwhich(X,k=2)
  return(naboer)
} # end function

beregn_vinkler <- function(X, naboer) { # beregn vinkel mellem x_i og 2 nærmeste naboer
  vinkler <- numeric(length(X[,1])) # vektor til resultatet
  for (i in 1:length(X[,1])) { # opstil vektorerne mellem punkterne
    a <- c(X[naboer[i,1],1] - X[i,1] , X[naboer[i,1],2] - X[i,2])
    b <- c(X[naboer[i,2],1] - X[i,1] , X[naboer[i,2],2] - X[i,2])
    vinkler[i] <- acos((a%*%b) / (sqrt(a%*%a) * sqrt(b%*%b))) # beregn vinklen
  } # end for
  return(vinkler)
} # end function

simuler_vinkler_reduceret <- function(n) { # lav n simuleringer og beregn fraktiler
  vinkelfordeling <- matrix(,n,8) # matrix til antal vinkler i hvert interval
  for ( i in 1:n) { # lav n simuleringer og beregn hvor mange vinkler, der er i hvert interval
    generer_model(0.825,1.26,9.49,0.209,0.0717,c(0,6.35),c(0,6.35))
    naboer <- beregn_naboer(X) ; vinkler <- beregn_vinkler(X, naboer)
    x <- hist(vinkler,breaks=seq(0,pi,pi/8),plot=F) # udnyt hist() til resultatet
    vinkelfordeling[i,] <- x$counts/sum(x$counts) # % pga. varierende antal punkter
  } # end for

  fraktiler <- matrix(,8,4) # matrix til resultatet
  for (i in 1:8) { # beregn min/0.025/0.975/max for hvert interval
    fraktiler[i,] <- quantile(vinkelfordeling[,i],c(0,0.025,0.975,1))
  } # end for
  return(fraktiler*250) # omregn tilbage til hyppighed
} # end function

plot_vinkler_reduceret <- function(n) { # plot simuleret resultat vs. datasæt
  fraktiler <- simuler_vinkler_reduceret(n) # lav først n simulationer
  y <- hist(1,breaks=seq(0,pi,pi/8),plot=F) ; y$counts <- fraktiler[,4]
  plot(y,ylim=c(0,max(fraktiler)), main=paste("Fordeling af vinkler
      for",n,"simulationer \n af modellen på [0 ; 6,35] x [0 ; 6,35]"))
  y$counts <- fraktiler[,3] ; plot(y,add=T,col="grey75")
  y$counts <- fraktiler[,2] ; plot(y,add=T,col="white")
  y$counts <- fraktiler[,1] ; plot(y,add=T,col="white")

  source("estimation-data-datasaet.R") # indlæs det reducerede datasæt
  naboer <- beregn_naboer(X) ; vinkler <- beregn_vinkler(X, naboer)
  hist(vinkler,breaks=seq(0,pi,pi/8),add=T,col=NULL,lty=3) } # end function
```

R-kode A.3 *– modelkontrol-vinkelfordeling.R, kode til bestemmelse af vinkelfordelingen for henholdsvis datasæt og simulerede punktmønstre.*

A.3.2 Deskriptive størrelser

```
1  library(spatstat) # spatstats estimerings-kommandoer skal bruges
2  source("main-model.R") # indlæs koden til at simulere modellen
3
4  source("estimation-data-datasaet.R") # indlæs det reducerede datasæt
5  X <- as.ppp(X, W=c(0,6.35,0,6.35)) # konverter til spatstat-punktmønster
6  x <- pcf.ppp(X,correction="Ripley", r=seq(0,1.5875,1.5875/500)) # estimer g(r)
7
8  plot(x$iso ~ x$r , type="l" , xlim=c(0,1.48), ylim=c(0.6,7)) # plot estimatet i
       passende størrelse vindue
9  legend(x = 0.713, y = 7.16, legend = c("Data","Median","2,5% /
       97,5%-fraktil"), lty = c("solid","dashed","dotted"), title="g(r) for
       reduceret datasæt") # tilføj signaturforklaring
10
11 n <- 5000 # antal simulationer
12 pcf <- matrix(,n,501) # matrix til resultatet
13
14 for ( i in 1:n ) { # estimer g(r) for n simulationer af modellen
15   generer_model(0.825,1.26,9.49,0.209,0.0717,c(0,6.35),c(0,6.35))
16   X <- as.ppp(X, W=c(0,6.35,0,6.35)) # konverter til spatstat-punktmønster
17   x <- pcf.ppp(X,correction="Ripley", r=seq(0,1.5875,1.5875/500)) # estimer g(r)
18   pcf[i,] <- x$iso # gem resultatet
19 } # end for
20
21 fraktiler <- matrix(,501,3) # matrix til resultatet
22
23 for (i in 1:501) { # beregn 0.025/0.5/0.975-fraktiler for hver r-værdi
24   fraktiler[i,] <- quantile(pcf[,i],c(0.025,0.5,0.975))
25 } # end for
26
27 lines(fraktiler[,1]~x$r,lty=3,lwd=2.5) # tilføj 95%-interval og median til plottet
28 lines(fraktiler[,2]~x$r,lty=2,lwd=1.5)
29 lines(fraktiler[,3]~x$r,lty=3,lwd=2.5)
```

R-kode A.4 *– modelkontrol-pcf.R, kode til bestemmelse af $g(r)$ for henholdsvis datasæt og simulerede punktmønstre.*