UNIVERSITY OF CALIFORNIA

Los Angeles

Applications and Properties of Point Processes

A dissertation submitted in partial satisfaction
of the requirements for the degree
Doctor of Philosophy in Statistics

by

Conor Joseph Kresin

2023

ABSTRACT OF THE DISSERTATION

Applications and Properties
of Point Processes

by

Conor Joseph Kresin
Doctor of Philosophy in Statistics
University of California, Los Angeles, 2023
Professor Frederic R. Paik Schoenberg, Chair

This dissertation discusses the properties of point process models for epidemic diseases and other clustered phenomena. We present (1) a novel computationally efficient estimator for the parameters of conditional intensity functions used to model point process data, (2) a comparison of compartmental models and Hawkes-type models for predicting the spread of COVID-19, (3) a potential outcomes framework for point process data, and (4) a novel methodology for bounding the complexity of sparse Boolean-valued tensors represented as point processes, discussed here in the context of tomographic images of fractured silicon materials.

The dissertation of Conor Joseph Kresin is approved.

Mark Stephen Handcock

Yingnian Wu

Andrea Bertozzi

Frederic R. Paik Schoenberg, Committee Chair

University of California, Los Angeles

2023

To Sanaz, the love of my life

&

Max, may I make you proud.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGMENTS

Rick Schoenberg - you are my academic role model, and a role model for me in many other ways. Thank you for opening doors for me, and thank you for waiting so patiently for me to cross through. I hope that I can continue to learn from you for the rest of my life. **Nicolas Christou** - without your belief in me, I would have never even dreamt of getting my doctorate. Thank you for sparking the courage in me to try.

My gracious collaborators are acknowledged in Sections 2.9, 3.7, and 5.4.

VITA

2011–2016 BS Applied Mathematics, BA Piano Performance, University of California, Los Angeles.

2016–2018 KPMG – Associate Statistical Consultant in Economic and Valuation Services, Los Angeles.

2018–2020 RSM – Senior Statistician, Los Angeles.

2018– Ph.D. Student, Statistics, University of California, Los Angeles. Advisor: Frederic Schoenberg.

2018–2023 Teaching Assistant, Lecturer, University of California, Los Angeles.

2019–2023 Graduate Student Researcher, in collaboration with the Army Research Office. University of California, Los Angeles.

PUBLICATIONS

Parametric estimation of spatial- temporal point processes using the Stoyan-Grabarnik Statistic. Conor Kresin and Frederic Schoenberg. Annals of the Institute of Statistical Mathematics, 2023.

Nonparametric estimation of recursive point processes with application to Mumps in Pennsylvania. Andrew Kaplan, Junhyung Park, Conor Kresin, and Frederic Schoenberg. Biometrical Journal, 2021.

Comparison of Hawkes and SEIR models for the spread of COVID-19. Conor Kresin, Frederic Schoenberg, and George Mohler. Advances and Applications in Statistics, 2020.

CHAPTER 1

Introduction

1.1 Overview

This dissertation discusses the properties of point process models for epidemic diseases, crimes, and other clustered phenomena. Chapter 1 presents a novel computationally efficient estimator for the parameters of conditional intensity functions used to model point process data. We call this estimator the Stoyan Grabarnik estimator, and prove its consistency under quite general conditions. Maximum-likelihood estimation for intensity parameters is too computationally costly for many data sets of interest. The Stoyan Grabarnik estimator offers a simple and computationally tractable alternative to likelihood-based approaches, and in particular facilitates parametric point process modeling for "big data."

Chapter 2 begins with an introduction to Hawkes and Hawkes-type models. A comparison of SEIR compartmental models and Hawkes-type models is then presented in the context of predicting the spread of COVID-19 and other infectious diseases. The chapter weighs the physical plausibility of the SEIR model against the parsimony and flexibility of the Hawkes model. The chapter concludes with detailing the mathematical connection between HawkesN and SEIR models.

Chapter 3 presents a potential outcomes framework for point process data which does not require discretization or restrictive modelling assumptions. This chapter is primarily definitional, and contains a small simulation study. Finally, Chapter 4 describes a novel methodology for bounding the complexity of point process data. To do this, we first represent

a sparse Boolean-valued tensor as a realization of an unparameterized point process, and then apply iterative superpositions of a correctly scaled homogenous Poisson process. The work within this chapter relies on leveraging superposition limit theorems for Poisson processes. The chapter concludes with applying the novel methodology to a data set of tomographic images capturing bullet-struck silicon materials.

Generally, this dissertation argues that there are important connections between point process theory and information theory and causality that are underdeveloped in the current literature. It is our hope that computationally efficient estimators like the Stoyan Grabarnik estimator allow us to model large data and complex probabilistic structures while leveraging the rich theoretical framework of point processes. In the following chapters, we discuss the mathematical properties of point processes and methodologies for fitting a model given a realization of a point process. Point process notation is quite varied across the literature, and therefore we begin with notational definitions and basic characterizations of the properties of point processes.

1.2 Preliminaries

Point processes have been characterized in various ways in the foundational point process literature. For univariate point processes, it is tempting to characterize a point process as a non-decreasing integer-valued stochastic processes, but such a characterization is not sufficient for point processes on $\mathbb{R}^d$ where $d > 1$. Point processes have been alternately characterized as a finite collection of points, but such a characterization suffices only for finite point processes (Moller and Waagepetersen, 2003).

Most present point process literature characterizes a point process as a $\mathbb{Z}^+$ valued random measure (counting measure), which extends naturally to high-dimensional and infinite point processes (Daley and Jones, 2003), and a realization of a point process as a collection of points. Such measures are assumed to be boundedly finite, *i.e.* a finite number of points

fall inside any bounded set. Formally, we notate a point process N characterized by random measure $\xi(\cdot)$ on state space $\mathcal{X}$ where $\mathcal{X}$ is a complete separable metric space. Often, we let $\mathcal{X} = \mathbb{R}^d$, or in the context of spatial temporal point processes $\mathcal{X} = \mathbb{R}^+ \times \mathbb{R}^d$. In the context of Janossy densities, we denote $\mathcal{X}^{(n)}$ as the n-fold (Cartesian) product space of $\mathcal{X} \times \ldots \times \mathcal{X}$ which is useful to describe the joint distribution of the points of a realization of N, given that there are n points. The canonical point process state space (accommodating infinite point processes) is notated $\mathcal{N}_{\mathcal{X}}^{\#}$ and represents the space of all bounded finite counting measures on $\mathcal{X}$. We can then define N as the random counting measure mapping from probability space (Ω, ξ, N) into $(\mathcal{N}_{\mathcal{X}}^{\#}, \mathcal{B}(\mathcal{N}_{\mathcal{X}}^{\#}))$ where $\mathcal{B}(\mathcal{N}_{\mathcal{X}}^{\#})$ represents the family of Borel sets that can be used to define measures on $\mathcal{N}_{\mathcal{X}}^{\#}$. In short, every distinct probability measure on $(\mathcal{N}_{\mathcal{X}}^{\#}, \mathcal{B}(\mathcal{N}_{\mathcal{X}}^{\#}))$ defines a point process (Daley and Vere-Jones, 2007, 2008). In general, throughout this dissertation, we strive to maintain the notation of Daley and Vere-Jones (2008).

When modelling a point process, we often use the *conditional intensity* characterization which, if it exists, is equal to

$$\lambda = \lim_{h \downarrow 0} \frac{\mathbb{E}[N(t+h) - N(t) | \mathcal{H}_{t-}]}{h}. \tag{1.1}$$

where $\mathcal{H}_{t-}$ represents the history of counting process N up to time t. More formally, $\mathcal{H}_{t-}(\cdot)$ is a filtration, *i.e.* an increasing sequence of σ-algebras. Therefore $\mathcal{H}_{t-}$ is the σ-algebra of events occuring at times up to but not including t. Such a characterization of intensity can be easily extended to spatio-temporal point processes, and point processes with dimension $d > 3$. When the intensity is finite, we note that $\mathbb{P}(N(x, x+h] > 0) = \lambda h + o(h)$ (Daley and Jones, 2003).

We denote $\Lambda(\cdot)$ the intensity measure characterizing point process N, and note that $\Lambda(B) = \mathbb{E}[N(B)]$ for Borel set B, which is to say that $\Lambda(B)$ is the mean number of points in B. Given that the underlying state space $\mathcal{X}$ is a completely seperable metric space (*i.e.* the point cannot be on a non-continuous lattice), $\Lambda(\cdot)$ has density $\lambda(\cdot)$ which in the point process literature is referred to as the intensity function of N. It follows that $\Lambda(B) = \int_B \lambda(x) dx$.

We can intuit the intensity function $\lambda(x)dx$ of N as the probability that a point is in an infinitesimal n-ball with volume dx centered on x (Chiu et al., 2013).

The Campbell Theorem, an analogue of Fubini's theorem, states that for any non negative and measureable function $f : \mathbb{R}^d \mapsto \mathbb{R}^+$

$$\mathbb{E}\left[\sum_x f(x)\right] = \int_{\mathbb{R}^d} f(x)\Lambda(dx) = \int_{\mathbb{R}^d} f(x)\lambda(x)dx \tag{1.2}$$

given that $\lambda(x)$ exists (Cronie and Van Lieshout, 2018; Chiu et al., 2013). For a proof of this theorem, see (Błaszczyszyn, 2017, Section 7.1.3).

A point process is *simple* when $\mathbb{P}(N(\{t\}) = \{0 \vee 1\} \forall t) = 1$. Crucially, Proposition 7.2.IV of (Daley and Jones, 2003) states that the conditional intensity determines the probability structure (finite-dimensional distribution) of any simple point process uniquely. This is true because the conditional intensity determines the family of conditional hazard functions, and these in turn determine the Janossy densities (discussed in detail in Section 5.1). Therefore, in modeling point processes, the process is typically assumed simple, and a model for λ is satisfactory.

Additionally, many results in the below work assume that a given point process is *stationary*. Intuitively, a point process is stationary if the parameters and structure governing the process do not vary over space, *i.e.* if they are translation invariant. If the process is on the real line, we can think of this as temporal invariance. Formally, a point process is considered stationary when for all bounded Borel subsets $A_1, \ldots, A_r$ of the real line, the joint distribution of $\{N(A_1 + t), \ldots, N(A_r + t)\}$ is independent of $t \in \mathbb{R}$ (Daley and Jones, 2003). This definition generalizes to finite dimensional point processes, *i.e.* a point process in $\mathbb{R}^d$ is stationary if its finite dimensional[1] distributions are invariant under simultaneous shifts (translations): a point process N has the same characteristics on A as bounded subsets of

[1]Finite dimensional distributions, often referred to as fidi distributions in the point process literature are probabilities of the form $\mathbb{P}(N(A_1) = n_1, \ldots, N(A_k) = n_k)$ for positive integers $n_1, \ldots, n_k$ and bounded Borel subets $A_1, \ldots, A_k$. The distribution of N is uniquely determined by the system of all these values for $k = 1, 2, \ldots$ (Chiu et al., 2013).

$\mathbb{R}^d$ translated, $A + u = \{x + u : x \in A, u \in \mathbb{R}^d\}$ (Chiu et al., 2013).

We note that stationarity is often confused with *homogeniety*, even in point process literature (for instance, (Chiu et al., 2013) states that the two are synonymous). A point process is homogenous if the intensity λ is constant. For instance, if N is a homogenous Poisson point process, $\mathbb{E}[N(B)] = \Lambda(B) = \lambda\mu(B)$ where B is a Borel set and $\mu(\cdot)$ is the Lebesgue measure.

CHAPTER 2

The Stoyan Grabarnik Intensity Estimator

2.1 Background

In this chapter, we propose a novel estimator for the parameters governing spatial-temporal point processes. Unlike the maximum likelihood estimator, the proposed estimator is fast and easy to compute, and does not require the computation or approximation of a computationally expensive integral. This parametric estimator is based on the Stoyan-Grabarnik (sum of inverse intensity) statistic, and is shown to be consistent, under quite general conditions. Simulations are presented demonstrating the performance of the estimator.

This chapter is structured as follows: We begin with notational definitions necessary for this chapter in Section 2.3. Section 2.4 formally introduces the Stoyan-Grabarnik statistic and estimator, and in Section 2.5, we prove the consistency of two Stoyan-Grabarnik-type estimators. Section 2.6 provides some discussion and examples of the analytical properties and extensions of the estimator, and Section 2.7 contains a brief simulation study.

2.2 Maximum Likelihood Estimation

A realization of a spatial-temporal point process is often characterized via its conditional intensity λ, the parameters of which are typically fit via maximum likelihood estimation (MLE) or Markov chain Monte Carlo (MCMC) methods. Specifically, for a realization $\{(t_i, x_i, y_i)\}_{i=1}^n = \{\tau_i\}_{i=1}^n$ of the point process N, one typically estimates the parameter

vector θ by computing

$$\hat{\theta}_{\text{MLE}} = \arg\max_{\theta \in \Theta} \left(\sum_i \log \lambda(\tau_i; \theta) - \int_0^T \int \int \lambda(\tau; \theta) dt dx dy \right). \tag{2.1}$$

Such estimates are, under quite general conditions, consistent, asymptotically normal, asymptotically unbiased, and efficient, with standard errors readily constructed using the diagonal elements of the inverse of the Hessian (Krickeberg, 1982; Ogata, 1978). Unfortunately, for many point processes, the integral term on the right in Equation (2.1) is often extremely difficult to compute (Harte, 2010; Ogata, 1998) especially when the conditional intensity λ is highly volatile, as in this situation the user must approximate the integral of a highly variable and often high-dimensional stochastic process, which is not at all easy to do.

Approximation methods proposed for certain processes such as Hawkes processes suggest a computationally intensive numerical integration method (Ogata and Katsura, 1988; Schoenberg, 2013), but in general the problem of computation or estimation of the integral term in the log-likelihood can be burdensome (Harte, 2010; Reinhart, 2018). Despite computational limitations, maximum likelihood remains the most common method for estimating the parameters of point process intensities (Reinhart, 2018).

We propose an alternative class of estimators based on the Stoyan-Grabarnik summed inverse intensity statistic introduced in Stoyan and Grabarnik (1991). The Stoyan-Grabarnik ("SG") statistic

$$\bar{m} = \frac{1}{\lambda} \tag{2.2}$$

was introduced as the exponential "mean mark" in the context of the Palm distribution of marked Gibbs processes (Stoyan and Grabarnik, 1991). As a primary property of Equation (2.2), it is noted in Stoyan and Grabarnik (1991) that the expectation of the sum of the exponential marks corresponding to the points observed in some region is equal to the Lebesgue measure $\mu(\cdot)$ of that region. For the purposes of this paper, we define the SG statistic corresponding to a parameter vector θ and a realization $\{\tau_i\}_{i=1}^n$ of the point process

N on spatial-temporal region $\mathcal{I}$ as

$$S_{\mathcal{I}}(\theta) = \sum_{i:\tau_i \in \mathcal{I}} \frac{1}{\lambda(\tau_i;\theta)}.$$

The SG statistic has been suggested as a goodness-of-fit model diagnostic for point processes (Baddeley et al., 2005), and more recently has been proposed for finding the optimum bandwidth for kernel smoothing to estimate the intensity of a spatial Poisson process (Cronie and Van Lieshout, 2018). Here, we consider a general spatial-temporal point process and suggest dividing the observation region into cells and estimating the parameters of the process by minimizing the sum of squared differences between the Stoyan-Grabarnik statistic and its expected value. We show that the resulting estimator is generally consistent and far easier to compute than the MLE.

2.3 Formal Setting

For the purposes of the remainder of this chapter, a point process is a measurable mapping from a filtered probability space $(\Omega, \mathcal{F}, \mathcal{P})$ onto $\mathcal{N}$, the set of $\mathbb{Z}^+$-valued random measures (counting measures) on a complete separable metric space (CSMS) $\mathcal{X}$ (Daley and Jones, 2003), where $\mathbb{Z}^+$ denotes the set of positive integers. Following convention (e.g. Daley and Jones (2003)), we will restrict our attention to point processes that are boundedly finite, i.e. processes having only a finite number of points inside any bounded set. For a spatial-temporal point process, $\mathcal{X}$ is a portion of $\mathbb{R}^+ \times \mathbb{R}^2$ or $\mathbb{R}^+ \times \mathbb{R}^3$ where $\mathbb{R}^+$ and $\mathbb{R}^d$ represent the set of positive real numbers and d-dimensional Euclidean space, respectively. The point process is assumed to be adapted to the filtration $\{\mathcal{F}_t\}_{t\geq 0}$ containing all information on the process N at all locations and all times up to and including time t. In what follows we will assume the spatial domain of the point process $\mathcal{S}$ is a finite and bounded portion of the plane $\mathbb{R}^2$ and denote point i of the process as $\tau_i = (t_i, x_i, y_i)$, though the results here extend in obvious ways to the case where the spatial domain is a portion of $\mathbb{R}^3$.

A process is $\mathcal{F}$-predictable if it is adapted to the filtration generated by the left continuous

processes $\mathcal{F}_{(-)}$. Intuitively, $\mathcal{F}_{(-)}$ represents the history of a process up to, but not including time t. A rigorous definition of $\mathcal{F}_{(-)}$ can be found in Daley and Vere-Jones (2007). Assuming it exists, the $\mathcal{F}$-conditional intensity λ of N is an integrable, non-negative, $\mathcal{F}$-predictable process, such that

$$\lambda(\tau) = \lim_{h,\delta \downarrow 0} \frac{\mathbb{E}[N\left([t, t+h) \times \mathbb{B}_{(x,y),\delta}\right) | \mathcal{F}_{t-}]}{h\pi\delta^2}.$$

where $\mathbb{B}_{(x,y),\delta}$ is a ball centered at location (x, y) with radius δ, and $\mathcal{F}_{t-}$ represents the history of the process N up to but not including time t.

A point process is simple if with probability one, all the points are distinct. Since the conditional intensity λ uniquely determines the finite-dimensional distributions of any simple point process (Proposition 7.2.IV of Daley and Jones (2003)), one typically models a simple spatial-temporal point process by specifying a model for λ. A point process is stationary if the specified model has a structure which is invariant over shifts in space or time.

An important spatial-temporal point process result sometimes called the martingale formula states that, for any non-negative predictable process f,

$$\mathbb{E}\left[\sum_i f(\tau_i)\right] = \mathbb{E}\left[\int_{\mathbb{R}^d} f(\tau)\lambda(\tau)d\mu\right];$$

where the expectation is with respect to $\mathcal{P}$. For a rigorous derivation of the martingale formula using Campbell measures, see Proposition 14.2.1 of Daley and Vere-Jones (2007). This result is the motivating impetus for exploring the Stoyan-Grabarnik estimator below. The martingale formula is a generalization of the Campbell formula which accommodates a non-negative deterministic function f (Cronie and Van Lieshout, 2018) and the Georgii-Nyugen-Zessin formula which accommodates an analogous equality using Papangelou intensities in a purely spatial context (Baddeley et al., 2005).

2.4 The Stoyan-Grabarnik Estimator

Suppose the spatial-temporal domain $\mathcal{X}$ is partitioned into p cells $\{\mathcal{I}_j\}_{j=1}^p$. Define the estimator

$$\begin{aligned}\hat{\theta} &= \arg\min_{\theta\in\Theta} \sum_{j=1}^{p} \left(\sum_{i:(\tau_i)\in\mathcal{I}_j} \frac{1}{\lambda(\tau_i;\theta)} - \mathbb{E}\left[\sum_{i:(\tau_i)\in\mathcal{I}_j} \frac{1}{\lambda(\tau_i;\theta)} \right] \right)^2 \\ &= \arg\min_{\theta\in\Theta} \sum_{j=1}^{p} \left(S_{\mathcal{I}_j}(\theta) - \mathbb{E}\left[S_{\mathcal{I}_j}(\theta) \right] \right)^2 . \end{aligned} \tag{2.3}$$

Because λ is non-negative and predictable, so is $1/\lambda$, and therefore, by the martingale formula, at the true value of the parameter vector θ^*,

$$\mathbb{E}\left[\sum_{i:(\tau_i)\in\mathcal{I}_j} \frac{1}{\lambda(\tau_i;\theta^*)} \right] = \mathbb{E}\left[\int_{\mathcal{I}_j} \frac{\lambda(\tau;\theta^*)}{\lambda(\tau;\theta^*)} d\mu \right] = \mu(\mathcal{I}_j)$$

where the expectation is with respect to $\mathcal{P}$. Thus the computationally intensive integral term necessary to find the MLE is replaced with a term which is computationally trivial to compute, namely the volume of the cell $\mathcal{I}_j$. Therefore, in practice it is convenient to plug in the volume of $\mathcal{I}_j$ for $\mathbb{E}\left[S_{\mathcal{I}_j}(\theta)\right]$, and thus define the SG estimator as

$$\begin{aligned}\tilde{\theta} &= \arg\min_{\theta\in\Theta} \sum_{j=1}^{p} \left(\sum_{i:\tau_i\in\mathcal{I}_j} \frac{1}{\lambda(\tau_i;\theta)} - \mathbb{E}\left[\sum_{i:\tau_i\in\mathcal{I}_j} \frac{1}{\lambda(\tau_i;\theta^*)} \right] \right)^2 \\ &= \arg\min_{\theta\in\Theta} \sum_{j=1}^{p} \left(S_{\mathcal{I}_j}(\theta) - \mathbb{E}[S_{\mathcal{I}_j}(\theta^*)] \right)^2 \\ &= \arg\min_{\theta\in\Theta} \sum_{j=1}^{p} \left(S_{\mathcal{I}_j}(\theta) - |\mathcal{I}_j| \right)^2 . \end{aligned} \tag{2.4}$$

The SG estimator is closely related to the scaled residual random field described in Baddeley et al. (2005). Specifically, for a fixed spatial-temporal kernel density $\mathcal{K}(\cdot)$ with fixed bandwidth b, let

$$Q(s) = \sum_{i=1}^{n} \frac{\mathcal{K}(s-\tau_i)}{\lambda(\tau_i;\theta)} - 1,$$

for s any location in space-time. Then if $\mathcal{X}$ is the observation window,

$$\begin{aligned}
\mathbb{E}\left[\int_{\mathcal{X}} Q(s)d\mu\right] =& \mathbb{E}\left[\int_{\mathcal{X}} \sum_{i=1}^{n} \frac{\mathcal{K}(s-\tau_i)}{\lambda(\tau_i;\theta)} d\mu(s)\right] - |\mathcal{X}| \qquad (2.5)\\
=& \mathbb{E}\left[\sum_{i=1}^{n} \frac{1}{\lambda(\tau_i;\theta)} \int_{\mathcal{X}} \mathcal{K}(s-\tau_i)d\mu(s)\right] - |\mathcal{X}| \\
\approx& \mathbb{E}\left[\sum_{i=1}^{n} \frac{1}{\lambda(\tau_i;\theta)}\right] - |\mathcal{X}| \qquad (2.6)\\
=& |\mathcal{X}| - |\mathcal{X}| = 0,
\end{aligned}$$

where the approximation in (2.6) stems from the fact that the integral over $\mathcal{X}$ of the kernel density will be close to unity provided the bandwidth is sufficiently small in relation to the size of the observation window $\mathcal{X}$. Ignoring such edge effects, the SG estimator minimizes the sum of squares of the integral of this residual field over cells in the partition, but one may alternatively find parameters θ minimizing some other criterion, such as for example the integral of $Q^2(s)$ over $\mathcal{X}$, or over cells of the partition. Given unbiased edge correction, (2.5) is exactly equal to zero.

2.5 Consistency

This section establishes the consistency of $\hat{\theta}$ and $\tilde{\theta}$, for a simple and stationary spatial-temporal point process N with conditional intensity $\lambda(\tau;\theta)$, where $\tau = \{t, x, y\}$ is a location in space-time, and λ depends on the parameter vector θ which is an element of some parameter space Θ. Let θ^* denote the true parameter vector, and suppose N is observed on the spatial-temporal domain $\mathcal{X} = [0, T) \times \mathcal{S}$, where $\mathcal{S}$ represents the spatial domain equipped with Borel measure μ, and $\mathcal{X}$ is some CSMS. The following assumptions regarding N, Θ and $\mathcal{S}$ are useful in establishing consistency of the estimators.

2.5.1 Assumptions

Assumption A1: The spatial observation region $\mathcal{S}$ allows a partitioning scheme

$$\mathcal{S} = \bigcup_{j=1}^{p} \mathcal{S}_j$$

such that $\mu(\mathcal{S}_j) > 0 \; \forall j \in \{1, \ldots, p\}$, for some fixed finite number p. We further assume that p is large enough that for any θ_1 and θ_2, if $\theta_1 \neq \theta_2$, then

$$\mathbb{E}[S_{\mathcal{I}_j}(\theta_1)] \neq \mathbb{E}[S_{\mathcal{I}_j}(\theta_2)] \tag{2.7}$$

or equivalently

$$\mathbb{E}\left[\int_{\mathcal{I}_j} \frac{\lambda(\theta^*)}{\lambda(\theta_1)} d\mu\right] \neq \mathbb{E}\left[\int_{\mathcal{I}_j} \frac{\lambda(\theta^*)}{\lambda(\theta_2)} d\mu\right] \tag{2.8}$$

$\forall j \in \{1, \ldots, p\}$, where $\mathcal{I}_j = \mathcal{S}_j \times [0, T)$.

Note on Assumption A1: The assumption that p is sufficiently large that condition (2.7) or equivalently (2.8) holds is needed for the identifiability of $\hat{\theta}$ and $\tilde{\theta}$. The minimal value of p to satisfy this condition appears to depend on the underlying structure of the conditional intensity λ. In practice, a large value of p can be selected to ensure that condition (2.7) is met, although the computational expense of the estimator increases as p increases, and more importantly, the efficiency of the estimator appears to decrease as p grows (see Figure 2.5). For finite datasets p must not be chosen to be too small so as to ensure that $N(\mathcal{I}_j) > 0 \; \forall j$. Note also that the cells $\mathcal{S}_j$ need not necessarily be connected, closed, or otherwise regular.

Assumption A2: Θ is a complete separable metric space and $\theta^* \subset \Theta$. Further, Θ admits a finite partition of compact subsets $\{\Theta_T^1, \ldots, \Theta_T^q\}$ such that $\lambda(\tau; \theta)$ is a continuous function of θ within $\Theta_T^j \; \forall j \in \{1, \ldots, q\}$.

Note on Assumption A2: A2 ensures that $\tilde{\theta}, \hat{\theta} \in \Theta$, *i.e.* that our estimator for θ^* exists within the parameter space.

Assumption A3: Given an open neighborhood $\mathcal{U}(\theta^*)$ around θ^*, $\lambda(\tau; \theta^*) - \lambda(\tau; \theta)$ is uniformly bounded away from zero for $\theta \notin \mathcal{U}(\theta^*)$.

Note on Assumption A3: A3 ensures that θ^* is identifiable. In particular, this assumption excludes the case where λ does not depend on θ.

Assumption A4: λ is finite and bounded away from zero across all cells $\mathcal{I}_j$, *i.e.* $\exists \zeta > 0$ such that

$$\zeta < \int_{\mathcal{I}_j} \lambda(\theta) d\mu < \infty$$

for j in $1, 2, ..., p$.

Note on Assumption A4: This assumption is needed for uniform integrability, and precludes cases such as $\lambda(\tau; \alpha) = \exp(-\alpha t)$ where only finitely many points occur as $T \to \infty$, and therefore α is not consistently estimable via the SG estimator (or via MLE, for that matter). Similarly, because we restrict to stationary point processes, we similarly ensure that there are never finitely many points that occur as $T \to \infty$ which a parameter to be estimated is dependent on.

2.5.2 Results

Under Assumptions A1-A4, the estimate $\hat{\theta}$ defined in (2.3) is a consistent estimator of θ^*.

Proof. For any $\epsilon > 0$ and any neighborhood $\mathcal{U}(\theta^*)$ around θ^*, for all sufficiently large T,

$$\mathbb{P}(\hat{\theta}_T \notin \mathcal{U}(\theta^*)) < \epsilon.$$

We begin with demonstrating that

$$M(\theta, T) = \sum_{j=1}^{p} \left(\sum_{i:\tau_i \in \mathcal{I}_j} \frac{1}{\lambda(\tau_i; \theta)} - \mathbb{E}\left[\sum_{i:\tau_i \in \mathcal{I}_j} \frac{1}{\lambda(\tau_i; \theta)} \right] \right)^2 \overset{\text{a.s.}}{\to} E[M(\theta, T)]$$

for $\theta \in \Theta$ as $T \to \infty$. For a partition of $\mathcal{X}$ with index j, let

$$C_j(\theta, T) = \sum_{i:\tau_i \in \mathcal{I}_j} \frac{1}{\lambda(\tau_i; \theta)} - \mathbb{E}\left[\sum_{i:\tau_i \in \mathcal{I}_j} \frac{1}{\lambda(\tau_i; \theta)} \right].$$

$C_j(\theta, T)$ is a $\mathcal{F}$-martingale since $1/\lambda$ is $\mathcal{F}-$predictable. By Jensen's inequality, $C_j(\theta, T)^2$ is a $\mathcal{F}-$sub-martingale as $g(x) = x^2$ is a convex function. Letting

$$M(\theta, T) = \sum_{j=1}^{p} C_j(\theta, T)^2,$$

M is a $\mathcal{F}-$sub-martingale. It follows from martingale convergence, and the fact that λ is absolutely continuous as a function of θ from Assumptions A2 and A4, that $M(\theta, T) \to \mathbb{E}[M(\theta, T)]$ uniformly.

We next demonstrate that

$$\theta^* = \arg\min_{\theta \in \Theta} \mathbb{E}[M(\theta, T)], \tag{2.9}$$

concluding this result in lines (2.18) and (2.19). Note that for a given cell j in the partition,

$$\mathbb{E}[C_j(\theta, T)] = \mathbb{E}\left[\sum_{i:\tau_i \in \mathcal{I}_j} \frac{1}{\lambda(\tau_i; \theta)} - \mathbb{E}\left[\sum_{i:\tau_i \in \mathcal{I}_j} \frac{1}{\lambda(\tau_i; \theta)}\right]\right] = 0$$

for all $\theta \in \Theta$. One can find the second moment, as follows:

$$\mathbb{E}[C_j(\theta, T)^2] = \text{var}(C_j(\theta, T)) + \mathbb{E}[C_j(\theta, T)]^2 = \text{var}(C_j(\theta, T)).$$

If $\theta = \theta^*$, then

$$\begin{aligned}
\text{var}(C_j(\theta, T)) =& \text{var}\left(\sum_{i:\tau_i \in \mathcal{I}_j} \frac{1}{\lambda(\tau_i; \theta^*)} - \mathbb{E}\left[\sum_{i:\tau_i \in \mathcal{I}_j} \frac{1}{\lambda(\tau_i; \theta^*)}\right]\right) \\
=& \text{var}\left(\sum_{i:\tau_i \in \mathcal{I}_j} \frac{1}{\lambda(\tau_i; \theta^*)} - |\mathcal{I}_j|\right) \\
=& \text{var}\left(\sum_{i:\tau_i \in \mathcal{I}_j} \frac{1}{\lambda(\tau_i; \theta^*)}\right) \\
=& \mathbb{E}\left[\left(\sum_{i:\tau_i \in \mathcal{I}_j} \frac{1}{\lambda(\tau_i; \theta^*)}\right)^2\right] - \left[\mathbb{E}\sum_{i:\tau_i \in \mathcal{I}_j} \frac{1}{\lambda(\tau_i; \theta^*)}\right]^2 \\
=& \mathbb{E}\left[\sum_{i:\tau_i \in \mathcal{I}_j} \left(\frac{1}{\lambda(\tau_i; \theta^*)}\right)^2 + \sum_{i:\tau_i \in \mathcal{I}_j} \sum_{k:\tau_k \in \mathcal{I}_j, k \neq i} \frac{1}{\lambda(\tau_i; \theta^*)\lambda(\tau_k; \theta^*)}\right]
\end{aligned}$$

$$- \left[\mathbb{E} \sum_{i:\tau_i \in \mathcal{I}_j} \frac{1}{\lambda(\tau; \theta^*)} \right]^2 \tag{2.10}$$

$$= \mathbb{E} \left[\int_{\mathcal{I}_j} \frac{1}{\lambda(\theta^*)} d\mu \right] + \mathbb{E} \left[\sum_{i:\tau_i \in \mathcal{I}_j} \sum_{k:\tau_k \in \mathcal{I}_j, k \neq i} \frac{1}{\lambda(\tau_i; \theta^*)\lambda(\tau_k; \theta^*)} \right] - \left[\mathbb{E} \int_{\mathcal{I}_j} d\mu \right]^2, \tag{2.11}$$

by applying the Martingale formula to both the first and last terms in (2.10). The middle cross-term can be evaluated as follows:

$$\mathbb{E} \left[\sum_{i:\tau_i \in \mathcal{I}_j} \sum_{k:\tau_k \in \mathcal{I}_j, k \neq i} \frac{1}{\lambda(\tau_i; \theta^*)\lambda(\tau_k; \theta^*)} \right]$$

$$= \mathbb{E} \left[\int_{\mathcal{I}_j} \underbrace{\int_{\mathcal{I}_j : t<u} \frac{1}{\lambda(\theta^*, t)\lambda(\theta^*, u)} dN(t)}_{\text{Predictable w.r.t. filtration } \mathcal{F}_{t<u}} dN(u) \right] \tag{2.12}$$

$$= \mathbb{E} \left[\int_{\mathcal{I}_j} \int_{\mathcal{I}_j : t<u} \frac{\lambda(\theta^*, u)}{\lambda(\theta^*, t)\lambda(\theta^*, u)} dN(t) d\mu(u) \right]$$

$$= \int_{\mathcal{I}_j} \mathbb{E} \left[\int_{\mathcal{I}_j : t<u} \frac{1}{\lambda(\theta^*, t)} dN(t) \right] d\mu(u)$$

$$= \int_{\mathcal{I}_j} \mathbb{E} \left[\mu(\mathcal{S}_j) \cdot u \right] d\mu(u) \tag{2.13}$$

$$= \mu(\mathcal{S}_j) \mathbb{E} \left[\int_{\mathcal{I}_j} u d\mu(u) \right]$$

$$= \mu(\mathcal{S}_j)^2 \frac{T^2}{2}$$

$$= \frac{|\mathcal{I}_j|^2}{2}. \tag{2.14}$$

Therefore, combining (2.11) and (2.14),

$$\mathbb{E}[C_j^2(\theta, T) | \theta = \theta^*] = \mathbb{E} \left[\int_{\mathcal{I}_j} \frac{1}{\lambda(\theta^*)} d\mu \right] - \frac{|\mathcal{I}_j|^2}{2}$$

Solving for the second moment of $C_j(\theta, T)$ when $\theta \neq \theta^*$, one similarly obtains

$$\begin{aligned}
\mathbb{E}[C_j(\theta,T)^2|\theta\neq\theta^*] =& \text{var}\left(\sum_{i:\tau_i\in\mathcal{I}_j}\frac{1}{\lambda(\tau_i;\theta)} - \mathbb{E}\left[\sum_{i:\tau_i\in\mathcal{I}_j}\frac{1}{\lambda(\tau_i;\theta)}\right]\right)\\
=& \text{var}\left(\sum_{i:\tau_i\in\mathcal{I}_j}\frac{1}{\lambda(\tau_i;\theta)}\right)\\
=& \mathbb{E}\left[\left(\sum_{i:\tau_i\in\mathcal{I}_j}\frac{1}{\lambda(\tau_i;\theta)}\right)^2\right] - \left[\mathbb{E}\sum_{i:\tau_i\in\mathcal{I}_j}\frac{1}{\lambda(\tau_i;\theta)}\right]^2\\
=& \mathbb{E}\left[\sum_{i:\tau_i\in\mathcal{I}_j}\left(\frac{1}{\lambda(\tau_i;\theta)}\right)^2\right]\\
&+ \mathbb{E}\left[\sum_{i:\tau_i\in\mathcal{I}_j}\sum_{k:\tau_k\in\mathcal{I}_j,k\neq i}\frac{1}{\lambda(\tau_i;\theta)\lambda(\tau_k;\theta)}\right]\\
&- \mathbb{E}\left[\sum_{i:\tau_i\in\mathcal{I}_j}\frac{1}{\lambda(\tau;\theta)}\right]^2 \qquad (2.15)\\
=& \mathbb{E}\left[\int_{\mathcal{I}_j}\frac{\lambda(\theta^*)}{\lambda(\theta)^2}d\mu\right] + \mathbb{E}\left[\sum_{i:\tau_i\in\mathcal{I}_j}\sum_{k:\tau_k\in\mathcal{I}_j,k\neq i}\frac{1}{\lambda(\tau_i;\theta)\lambda(\tau_k;\theta)}\right]\\
&- \mathbb{E}\left[\int_{\mathcal{I}_j}\frac{\lambda(\theta^*)}{\lambda(\theta)}d\mu\right]^2 \qquad (2.16)\\
=& \mathbb{E}\left[\int_{\mathcal{I}_j}\frac{\lambda(\theta^*)}{\lambda(\theta)^2}d\mu\right]\\
&+ \mathbb{E}\left[\int_{\mathcal{I}_j}\frac{\lambda(\theta^*,u)}{\lambda(\theta,u)}\int_{\mathcal{I}_j:t<u}\frac{\lambda(\theta^*,t)}{\lambda(\theta,t)}d\mu(t)d\mu(u)\right]\\
&- \mathbb{E}\left[\int_{\mathcal{I}_j}\frac{\lambda(\theta^*)}{\lambda(\theta)}d\mu\right]^2 \equiv g(\theta,\theta^*,\mathcal{I}_j), \qquad (2.17)
\end{aligned}$$

again applying the Martingale formula to the first and third terms in (2.15). Equation (2.17) is obtained from (2.16) using the same logic as in lines 2.12-2.13.

Consider the division of $\mathcal{X}$ into two regions: the spatial-temporal locations where

$$1 < \frac{\lambda(\theta^*, \tau)}{\lambda(\theta, \tau)} \qquad \text{Case C1}$$

and

$$0 < \delta_1 < \frac{\lambda(\theta^*, \tau)}{\lambda(\theta, \tau)} \leq 1 - \delta_2 \qquad \text{Case C2}$$

for $\delta_1 + \delta_2 < 1$. That is, we can express $g(\theta, \theta^*, \mathcal{I}_j)$ as the sum of three integrals:

$$\begin{aligned} g(\theta, \theta^*, \mathcal{I}_j) &= \sum_{h=1}^{3} g(\theta, \theta^*, \mathcal{I}_j \cap \mathcal{A}_h) \\ &= \sum_{h=1}^{2} g(\theta, \theta^*, \mathcal{I}_j \cap \mathcal{A}_h) \end{aligned}$$

where

$$\begin{aligned} \mathcal{A}_1 &= \{\mathcal{X} \cap \{\lambda(\theta, \tau) < \lambda(\theta^*, \tau)\}\} \\ \mathcal{A}_2 &= \{\mathcal{X} \cap \{\lambda(\theta, \tau) > \lambda(\theta^*, \tau)\}\} \\ \mathcal{A}_3 &= \{\mathcal{X} \cap \{\lambda(\theta, \tau) = \lambda(\theta^*, \tau)\}\} = \emptyset. \end{aligned}$$

We proceed by evaluating cases C1 and C2 separately for notational simplicity. In Case C1, we show that $\mathbb{E}[C_j(\theta, T)^2 | \theta = \theta^*] < \mathbb{E}[C_j(\theta, T)^2 | \theta \neq \theta^*]$ as follows:

$$\begin{aligned} \mathbb{E}[C_j(\theta, T)^2 | \theta \neq \theta^*] =& \mathbb{E}\left[\int_{\mathcal{I}_j} \frac{\lambda(\theta^*)}{\lambda(\theta)} \cdot \frac{1}{\lambda(\theta)} d\mu\right] \\ &+ \mathbb{E}\left[\int_{\mathcal{I}_j} \frac{\lambda(\theta^*, u)}{\lambda(\theta, u)} \int_{\mathcal{I}_j : t<u} \frac{\lambda(\theta^*, t)}{\lambda(\theta, t)} d\mu(t) d\mu(u)\right] \\ &- \mathbb{E}\left[\int_{\mathcal{I}_j} \frac{\lambda(\theta^*)}{\lambda(\theta)} d\mu\right]^2 \\ >& \mathbb{E}\left[\int_{\mathcal{I}_j} \frac{1}{\lambda(\theta)} d\mu\right] + \mathbb{E}\left[\int_{\mathcal{I}_j} 1 \cdot \int_{\mathcal{I}_j : t<u} 1 \cdot d\mu(t) d\mu(u)\right] \\ &- \mathbb{E}\left[\int_{\mathcal{I}_j} \frac{\lambda(\theta^*)}{\lambda(\theta)} d\mu\right]^2 \end{aligned}$$

$$= \mathbb{E}\left[\int_{\mathcal{I}_j} \frac{1}{\lambda(\theta)} d\mu\right] + \frac{|\mathcal{I}_j|^2}{2} - \mathbb{E}\left[\int_{\mathcal{I}_j} \frac{\lambda(\theta^*)}{\lambda(\theta)} d\mu\right]^2. \tag{2.18}$$

Therefore, $\mathbb{E}[C_j(\theta,T)^2|\theta=\theta^*] < \mathbb{E}[C_j(\theta,T)^2|\theta\neq\theta^*]$, since given the assumptions of Case C1,

$$\mathbb{E}\left[\int_{\mathcal{I}_j} \frac{1}{\lambda(\theta)} d\mu\right] + \frac{|\mathcal{I}_j|^2}{2} - \mathbb{E}\left[\int_{\mathcal{I}_j} \frac{\lambda(\theta^*)}{\lambda(\theta)} d\mu\right]^2$$
$$> \mathbb{E}\left[\int_{\mathcal{I}_j} \frac{1}{\lambda(\theta^*)} d\mu\right] + \frac{|\mathcal{I}_j|^2}{2} - \left(\int_{\mathcal{I}_j} d\mu\right)^2.$$

Equivalently,

$$\mathbb{E}\left[\int_{\mathcal{I}_j} \frac{\lambda(\theta^*)-\lambda(\theta)}{\lambda(\theta^*)\lambda(\theta)} d\mu\right] > \mathbb{E}\left[\int_{\mathcal{I}_j} \frac{\lambda(\theta^*)}{\lambda(\theta)} d\mu\right]^2 - \mathbb{E}\left[\int_{\mathcal{I}_j} \frac{\lambda(\theta^*)}{\lambda(\theta^*)} d\mu\right]^2,$$

and by the assumption of Case C1,

$$\mathbb{E}\left[\int_{\mathcal{I}_j} \frac{\lambda(\theta^*)}{\lambda(\theta)} d\mu\right]^2 - \mathbb{E}\left[\int_{\mathcal{I}_j} \frac{\lambda(\theta^*)}{\lambda(\theta^*)} d\mu\right]^2 > 0.$$

Assumption A3 guarantees that $\exists\delta_0 > 0$ such that $\lambda(\theta^*) - \lambda(\theta) > \delta_0$ and therefore this condition is satisfied given Assumption A4.

In Case C2, as $T \to \infty$

$$\mathbb{E}[C_j(\theta,T)^2|\theta\neq\theta^*] > \mathbb{E}\left[\int_{\mathcal{I}_j} \frac{\delta_1}{\lambda(\theta)} d\mu\right] + \frac{(\delta_1\cdot|\mathcal{I}_j|)^2}{2} - \mathbb{E}\left[\int_{\mathcal{I}_j} (1-\delta_2) d\mu\right]^2 \tag{2.19}$$

and therefore $\mathbb{E}[C_j(\theta,T)^2|\theta=\theta^*] < \mathbb{E}[C_j(\theta,T)^2|\theta\neq\theta^*]$, since

$$\mathbb{E}\left[\int_{\mathcal{I}_j} \frac{\delta_1}{\lambda(\theta)} d\mu\right] + \frac{(\delta_1\cdot|\mathcal{I}_j|)^2}{2} - 2(1-\delta_2)^2\frac{|\mathcal{I}_j|^2}{2} > \mathbb{E}\left[\int_{\mathcal{I}_j} \frac{1}{\lambda(\theta^*)} d\mu\right] - \frac{|\mathcal{I}_j|^2}{2}$$
$$|\mathcal{I}_j|^2\left(\frac{\delta_1^2+2\delta_2-1}{2}\right) > \mathbb{E}\left[\int_{\mathcal{I}_j} \frac{\lambda(\theta)-\delta_1\cdot\lambda(\theta^*)}{\lambda(\theta^*)\lambda(\theta)} d\mu\right]. \tag{2.20}$$

Note that $\forall\delta_1 \in (0,1)$, $\exists\delta_2 \in \left(2^{-1}\left(1-\sqrt{2}\sqrt{\delta_1^2+1}\right), 1\right)$, so the LHS of relation (2.20) is positive. The RHS is non zero by the assumption of Case C2 and the fact that $\int\lambda(\theta)d\mu$ is

non-zero as given by Assumption A4. As $M(\theta, T)$ is the sum of $C_j(\theta, T)^2$ for each partition $j \in \{1, \ldots, p\}$, we can therefore conclude that for any $\check{\theta} \notin \mathcal{U}(\theta^*)$, $\exists \delta > 0$ such that

$$\inf_{\theta \in \Theta} \left\{ \mathbb{E}[M(\check{\theta}, T) - M(\theta^*, T)] \right\} > \delta.$$

Finally, by Assumption A2, and given that $M(\hat{\theta}, T) \to \mathbb{E}[M(\theta^*, T)]$ uniformly, and $\inf_{\theta \in \Theta} \left\{ \mathbb{E}[M(\check{\theta}, T) - M(\theta^*, T)] \right\} > \delta$ as proven above, we conclude that for sufficiently large T (or equivalently, sufficiently large space-time volume $|\mathcal{X}|$) and $\forall \alpha, \epsilon > 0$,

$$\begin{aligned}
\mathbb{P}(\hat{\theta} \notin \mathcal{U}(\theta^*)) =& \mathbb{P}\left(M(\hat{\theta}, T) \leq \inf_{\theta \in \mathcal{U}(\theta^*)} \{M(\theta^*, T)\} \right) \\
<& \mathbb{P}\left(M(\hat{\theta}, T) \leq M(\theta^*, T) - \alpha \right) \\
=& \mathbb{P}\left(M(\theta^*, T) - M(\hat{\theta}, T) \geq \alpha \right) \\
\leq& \mathbb{P}\left(M(\theta^*, T) - \mathbb{E}[M(\theta^*, T)] \geq \frac{\alpha}{3} \right) \\
& + \mathbb{P}\left(M(\hat{\theta}, T) - \mathbb{E}[M(\hat{\theta}, T)] \geq \frac{\alpha}{3} \right) \\
& + \mathbb{P}\left(\mathbb{E}[M(\theta^*, T) - M(\hat{\theta}, T)] \geq \frac{\alpha}{3} \right) \\
=& \frac{\epsilon}{2} + \frac{\epsilon}{2} + 0.
\end{aligned}$$

□ □

The estimator

$$\tilde{\theta} = \arg\min_{\theta \in \Theta} \sum_{j=1}^{p} \left(\sum_{i:\tau_i \in \mathcal{I}_j} \frac{1}{\lambda(\tau_i; \theta)} - |\mathcal{I}_j| \right)^2$$

is a consistent estimator for θ^*. This estimator will be henceforth referred to as the SG estimator.

Proof. This results can be proven using the same method as in the proof of Theorem 1. A brief sketch of the proof is given below. When $\theta = \theta^*$,

$$\mathbb{E}\left[\sum_{i:\tau_i \in \mathcal{I}_j} \frac{1}{\lambda(\tau_i; \theta)} \right] = |\mathcal{I}_j|.$$

Define

$$\tilde{M}(\theta, T) = \sum_{j=1}^{p} \left(\sum_{i:\tau_i \in \mathcal{I}_j} \frac{1}{\lambda(\tau_i; \theta)} - |\mathcal{I}_j| \right)^2,$$

and note that although $\tilde{M}(\theta, T)$ is not generally a sub-martingale, $\tilde{M}(\theta^*, T)$ is. It follows as in the proof of Theorem 1 that $\tilde{M}(\theta^*, T) \xrightarrow{\text{a.s.}} \mathbb{E}[\tilde{M}(\theta^*, T)]$, and by absolute continuity of λ with respect to θ, this convergence is uniform. Similarly,

$$\arg\min_{\theta \in \Theta} \mathbb{E}[M(\theta, T)] = \arg\min_{\theta \in \Theta} \mathbb{E}[\tilde{M}(\theta, T)] = \theta^*$$

because

$$\mathbb{E}[\tilde{C}_j(\theta, T)^2 | \theta = \theta^*] = \text{var}(\tilde{C}_j(\theta, T)^2 | \theta = \theta^*)$$

where $\tilde{C}_j$ is defined analogously to C_j in Theorem 1, and

$$\begin{aligned}
\mathbb{E}[\tilde{C}_j(\theta, T)^2 | \theta \neq \theta^*] =& \mathbb{E}\left[\int_{\mathcal{I}_j} \frac{\lambda(\theta^*)}{\lambda(\theta)^2} d\mu \right] \\
&+ \mathbb{E}\left[\int_{\mathcal{I}_j} \frac{\lambda(\theta^*, u)}{\lambda(\theta, u)} \int_{\mathcal{I}_j : t<u} \frac{\lambda(\theta^*, t)}{\lambda(\theta, t)} d\mu(t) d\mu(u) \right] \\
&- 2|\mathcal{I}_j| \mathbb{E}\left[\int_{\mathcal{I}_j} \frac{\lambda(\theta^*)}{\lambda(\theta)} d\mu \right] + |\mathcal{I}_j|^2 \\
&\geq \mathbb{E}[C_j(\theta, T)^2 | \theta \neq \theta^*]. \qquad (2.21)
\end{aligned}$$

Relation (2.21) follows directly from the fact that

$$|\mathcal{I}_j|^2 \geq 2|\mathcal{I}_j| \mathbb{E}\left[\int_{\mathcal{I}_j} \frac{\lambda(\theta^*)}{\lambda(\theta)} d\mu \right] - \mathbb{E}\left[\int_{\mathcal{I}_j} \frac{\lambda(\theta^*)}{\lambda(\theta)} d\mu \right]^2.$$

From this one concludes exactly as in Theorem 1 that for any $\epsilon > 0$, for sufficiently large T, $\mathbb{P}(\tilde{\theta} \notin \mathcal{U}(\theta^*)) < \epsilon$. □ □

2.5.3 Discussion

In practice, a partitioning scheme and a set value of p must be decided upon before computing $\tilde{\theta}$ for realization N given a specified model λ. Analogous partitioning problems in

the context of quadrature schemes needed for numerical approximation of likelihoods have been discussed, see Berman and Turner (1992); Baddeley et al. (2004). A general solution or methodology for constructing a partitioning scheme which yields maximally accurate SG estimates is a difficult problem and future work.

Asymptotically, a very general class of partitioning schemes is sufficient to produce consistent SG-type estimates of the parameters of conditional intensity functions. As previously noted, cells are not assumed to be connected, closed, regular, or disjoint. The primary consideration for choosing a partitioning scheme in an asymptotic context is finding p large enough such that Assumption A1 is met and identifiability is ensured.

We therefore suggest that practitioners choose a simple partitioning scheme (*e.g.* a grid or Voronoï tessellation based on some subset of points in N) and some $p > 2c$ where c is the cardinality of θ. For relatively larger realizations of a process, $p > c^2$ may be an appropriate choice. This suggestion is only informed by trial and error via simulation of Hawkes, Cox and Poisson processes across various p for a given partitioning scheme. In the case of Poisson processes, it appears that for a Poisson intensity expressed as a polynomial, $p = c + 1$ and any grid partitioning scheme is sufficient to produce consistent SG estimates, where c is the number of polynomial coefficients to be estimated. We note that in general, computational expense increases as p increases. Further there appears to be a bias-variance trade off wherein larger p results in less bias but more variance, see Figure 2.5. Resultant bias and variance as a function of the number of parameters estimated, number of points realized, and selected p is the subject of future work.

2.6 Examples: Estimation of Poisson Processes

2.6.1 Homogeneous Poisson Process

Suppose N is a homogeneous Poisson process, *i.e.* $\lambda = \theta$ for some $\theta \in \mathbb{R}^+$. In this simple case an analytical solution for the SG estimator θ can be derived.

$$\begin{aligned}\tilde{\theta} &= \arg\min_{\theta\in\mathbb{R}^+} \sum_{j=1}^{p} \left(\sum_{i:\tau_i\in\mathcal{I}_j} \frac{1}{\theta} - |\mathcal{I}_j| \right)^2 \\ &= \arg\min_{\theta\in\mathbb{R}^\infty} \sum_{j=1}^{p} \left(\frac{N(\mathcal{I}_j)}{\theta} - |\mathcal{I}_j| \right)^2\end{aligned}$$

and setting the derivative to zero:

$$\begin{aligned}0 \stackrel{!}{=} &\frac{\partial}{\partial\theta} \left(\sum_{j=1}^{p} \left(\frac{N(\mathcal{I}_j)}{\theta} - |\mathcal{I}_j| \right)^2 \right) = -2 \sum_{j=1}^{p} \left(\frac{N(\mathcal{I}_j)}{\theta} - |\mathcal{I}_j| \right) \left(\frac{N(\mathcal{I}_j)}{\theta^2} \right) \\ = &\sum_{j=1}^{p} \left(\frac{N(\mathcal{I}_j)^2}{\theta^3} - \frac{N(\mathcal{I}_j)\cdot|\mathcal{I}_j|}{\theta^2} \right).\end{aligned}$$

Thus $\tilde{\theta}$ satisfies

$$\begin{aligned}\frac{\sum_{j=1}^{p} N(\mathcal{I}_j)^2}{\tilde{\theta}^3} &= \frac{\sum_{j=1}^{p} N(\mathcal{I}_j)\cdot|\mathcal{I}_j|}{\tilde{\theta}^2}. \\ \frac{1}{\tilde{\theta}} \sum_{j=1}^{p} N(\mathcal{I}_j)^2 &= \sum_{j=1}^{p} N(\mathcal{I}_j)\cdot|\mathcal{I}_j|. \\ \tilde{\theta} &= \frac{\sum_{j=1}^{p} N(\mathcal{I}_j)^2}{\sum_{j=1}^{p} N(\mathcal{I}_j)\cdot|\mathcal{I}_j|}\end{aligned} \tag{2.22}$$

Equation (2.22) has an interesting geometric interpretation. For the positive integer vector $N = N(\mathcal{I}_1), \ldots, N(\mathcal{I}_p)$ and the positive real vector $I = |\mathcal{I}_1|, \ldots, |\mathcal{I}_p|$ we can express $\lambda(\tilde{\theta})$ as

$$\begin{aligned}\frac{\sum_{j=1}^{p} N(\mathcal{I}_j)^2}{\sum_{j=1}^{p} N(\mathcal{I}_j)\cdot|\mathcal{I}_j|} &= \frac{||N||_2^2}{N\cdot I} \\ &= \frac{||N||_2^2}{||N||_2||I||_2\cos(\alpha)}\end{aligned}$$

$$= \frac{||N||_2}{||I||_2 \cos(\alpha)}. \tag{2.23}$$

Note that $\cos(\alpha)$, the angle between N and I, is constrained to $0 \leq \cos(\alpha) \leq 1$ due to the signs of N and I.

Equation (2.23) provides insight into the nature of the partitioning scheme chosen. As N and I become closer to orthogonal, $\cos(\alpha)$ approaches 0, forcing $\lambda(\tilde{\theta})$ to become arbitrarily large. Alternatively, if N and I are parallel, $\cos(\alpha) = 1$ and in this case

$$\lambda(\tilde{\theta}) = \frac{||N||_2}{||I||_2} = \sqrt{\frac{\sum_{j=1}^{p} N(\mathcal{I}_j)^2}{\sum_{j=1}^{p} |\mathcal{I}_j|^2}}. \tag{2.24}$$

Equation (2.24) achieves the minimum value that $\lambda(\tilde{\theta})$ can attain over $\alpha \in [0, 1]$, and is possible if there exists $\beta \in \mathbb{R}$ such that $N(\mathcal{I}_j) = \beta \cdot |\mathcal{I}_j|$ for all $j \in \{1, \ldots, p\}$. It immediately follows that a partitioning scheme P minimizes Equation (2.24) if it is chosen such that $N(\mathcal{I}_j) \propto |\mathcal{I}_j|$ for all j. This suggests that in the homogeneous Poisson case, ideally the partition will have roughly equal numbers of points per unit area in each cell.

Note a special case of Equation (2.24). If $p = 1$, then

$$\frac{\sum_{j=1}^{p} N(\mathcal{I}_j)^2}{\sum_{j=1}^{p} N(\mathcal{I}_j) \cdot |\mathcal{I}_j|} = \frac{N(\mathcal{X})}{|\mathcal{X}|} = \hat{\theta}_{\text{MLE}}.$$

In this special case, the SG estimator is equivalent to the MLE and therefore inherits the desirable properties of the MLE such as consistency, asymptotic normality, asymptotic unbiasedness and efficiency (Ogata, 1978). For instance, if N has 100 points in an observed spatial-temporal region $\mathcal{X}$ such that $\mu(\mathcal{X}) = 20$, then $\hat{\theta} = 100/20 = 5$, as expected.

2.6.1.1 Inhomogeneous Poisson with Step Function Intensity

We now assume that N has conditional intensity

$$\lambda(\tau; \theta) = \sum_{j=1}^{p} \gamma_j \mathbb{1}\{\tau \in \mathcal{I}_j\}$$

for $\gamma_j \in \mathbb{R}^+$ and $\theta = \{\gamma_1, ..., \gamma_p\}$. Thus N is homogeneous Poisson within each cell, but with an intensity varying from cell to cell.

The properties of similar processes have been discussed in the context of Poisson Voronoi Tessellations (PVTs) (Błaszczyszyn and Schott, 2003, 2005). Total variation error bounds for approximation of an inhomogeneous Poisson process via a mixture of locally homogeneous Poisson processes are provided in Błaszczyszyn and Schott (2003), where the error is due to the "spill-over" or overlap of optimal cell partitioning. Further, the existence of an approximation for such a decomposition is described using a modulated PVT (Błaszczyszyn and Schott, 2003, Proposition 4.1).

In this case, the SG estimator must satisfy

$$\tilde{\gamma} = \arg\min_{\gamma \in \mathbb{R}_+^p} \sum_{j=1}^{p} \left(\sum_{i:\tau_i \in \mathcal{I}_j} \left(\sum_{j=1}^{p} \gamma_j \mathbb{1}\{\tau \in \mathcal{I}_p\} \right)^{-1} - |\mathcal{I}_j| \right)^2 .$$

$\tilde{\gamma}$ in this case is a vector of the p estimates $\tilde{\gamma_j}$. Each $\tilde{\gamma_j}$ is itself a SG estimator corresponding to a disjoint homogeneous Poisson process on the observation region $\mathcal{I}_j$. Following the same reasoning as in the homogeneous Poisson case, the resulting estimator reduces to when the partitioning scheme is such that $\mathcal{I}_j$ is the only cell, *i.e.* the observation region is equal to a single cell and $p = 1$. We can therefore express the solution for the estimated coefficient within a single cell as

$$\tilde{\gamma_j} = \frac{N(\mathcal{I}_j)}{|\mathcal{I}_j|}$$

and again is equivalent to the MLE and therefore in this case the SG estimator, like the MLE, is consistent, asymptotically normal, asymptotic unbiased and efficient (Ogata, 1978). As each estimator $\tilde{\gamma_j}$ is consistent, we can conclude that the sum $\tilde{\gamma}$ is also consistent by Slutsky's Theorem.

2.6.2 Inhomogeneous Poisson with Polynomial Intensity

Suppose now that N is a Poisson process with polynomial intensity

$$\lambda(\theta) = \sum_{k=0}^{\nu} \theta_k \tau^k$$

with $\theta_k \in \mathbb{R}\ \forall k$. In practice, we can assume that λ is a polynomial of degree ν. Intensities of this class, *i.e.* intensities represented as an infinite polynomial, represent all possible inhomogeneous Poisson intensities, as intensities are by definition continuous functions. Should we start with some function λ which we wish to approximate as polynomial ρ, we could construct such a function via Bernstein polynomials.

A direct proof demonstrating that $\lambda(\theta)$ can be consistently estimated using the SG estimator is not obvious. This is because the first (below underbracketed) term of Equation 2.25 cannot be simplified nicely.

$$\{\tilde{\theta}_0, \tilde{\theta}_1, \ldots\} = \arg\min_{\theta \in \mathbb{R}^\infty} \sum_{j=1}^{p} \left(\underbrace{\sum_{i:\tau_i \in \mathcal{I}_j} \frac{1}{\sum_{k=0}^{\infty} \theta_k \tau_i^k}}_{} - |\mathcal{I}_j| \right)^2 \tag{2.25}$$

It would be desirable to achieve a simplification of the following nature:

$$= \arg\min_{\theta \in \mathbb{R}^\infty} \sum_{j=1}^{p} \left(\frac{N(\mathcal{I}_j)}{\sum_{k=0}^{\infty} \theta_k \tau_j^k} - |\mathcal{I}_j| \right)^2 \tag{2.26}$$

which occurs if all points are located within the same place within each partition. If we can assume the points are located in the same place in a partition (or that some point within is sufficiently "characteristic" of the location of all points in a partition), we can really simplify Equation 2.25 further to

$$= \arg\min_{\gamma \in \mathbb{R}_+^p} \sum_{j=1}^{p} \left(\frac{N(\mathcal{I}_j)}{\gamma_j} - |\mathcal{I}_j| \right)^2$$

i.e. the homogeneous within a partition approach. This is because $\sum_{k=0}^{\infty} \theta_k \tau_j^k$ has some constant value within a partition given this assumption.

In order to achieve a simplification like Equation 2.26, all points within a given partition need to be located at a single coordinate. Such a process would likely not be of interest to us in practice, and would also violate simplicity assumptions. We therefore consider this case as likely implemented as an approximation. Given the assumption that it is reasonable to assume all points within a given partition can be aggregated meaningfully, a single coordinate within such a partition must be chosen to be representative. Options for aggregate locations within cell $\mathcal{I}_j$ include $\bar{\tau}_j$ i.e. the mean of points in partition j, median(τ_j), or τ_j°, the coordinates of the centroid of the partition. Section 2.7 provides results which demonstrate $\bar{\tau}_j$ and τ_j° produce consistent estimates, but for notational simplicity, we choose τ_j°, *i.e.*

$$\tau_1^\circ = G(\mathcal{I}_1), \tau_2^\circ = G(\mathcal{I}_2), \ldots, \tau_p^\circ = G(\mathcal{I}_p)$$

where $G(A)$ represents the coordinates of the centroid of (convex) A. Note this convexity assumption is a restriction upon the more general partitioning scheme described in Assumption A1 above. Such a restriction is not necessary if we choose the mean or median aggregating statistic. Despite this, choosing centroid coordinates as representative of a partition is intuitively appealing given we are effectively assuming that the points within that partition are homogeneous.

We can then express Equation 2.26 as follows, assuming some polynomial intensity function of degree $\nu < \infty$:

$$\{\tilde{\theta}_0, \tilde{\theta}_1, \ldots, \tilde{\theta}_\nu\} \approx \arg\min_{\theta \in \mathbb{R}^\nu} \sum_{j=1}^{p} \left(\frac{N(\mathcal{I}_j)}{\sum_{k=0}^{\nu} \theta_k \left(\tau_j^\circ\right)^k} - |\mathcal{I}_j| \right)^2$$

and note that within any given partition j, $\sum_{k=0}^{\nu} \theta_k \left(\tau_j^\circ\right)^k = \gamma_j$ for some constant $\gamma_j \in \mathbb{R}$.

$$\{\tilde{\gamma}_0, \tilde{\gamma}_1, \ldots, \tilde{\gamma}_p\} \approx \arg\min_{\gamma \in \mathbb{R}^\nu} \sum_{j=1}^{p} \left(\frac{N(\mathcal{I}_j)}{\gamma_j} - |\mathcal{I}_j| \right)^2$$

and as seen in Section 2.6.1.1, we can solve directly to find

$$\tilde{\gamma}_j = \frac{N(\mathcal{I}_j)}{|\mathcal{I}_j|}$$

for all $j \in \{1, \ldots, p\}$ indexed partitions. Now, we denote $\gamma^\top = \begin{bmatrix} \tilde{\gamma}_1 & \ldots & \tilde{\gamma}_p \end{bmatrix}$. We can write

$$\begin{aligned} \gamma = \begin{bmatrix} \tilde{\gamma}_1 \\ \tilde{\gamma}_2 \\ \vdots \\ \tilde{\gamma}_p \end{bmatrix} = \begin{bmatrix} \frac{N(\mathcal{I}_1)}{|\mathcal{I}_1|} \\ \frac{N(\mathcal{I}_2)}{|\mathcal{I}_2|} \\ \vdots \\ \frac{N(\mathcal{I}_p)}{|\mathcal{I}_p|} \end{bmatrix} = \begin{bmatrix} \sum_{k=0}^{\nu} \theta_k \tau_1^{\circ k} \\ \sum_{k=0}^{\nu} \theta_k \tau_2^{\circ k} \\ \vdots \\ \sum_{k=0}^{\nu} \theta_k \tau_p^{\circ k} \end{bmatrix} \\ = \begin{bmatrix} 1 & \tau_1^\circ & \ldots & \tau_1^{\circ \nu} \\ 1 & \tau_2^\circ & \ldots & \tau_2^{\circ \nu} \\ \vdots & \vdots & & \vdots \\ 1 & \tau_p^\circ & \ldots & \tau_p^{\circ \nu} \end{bmatrix} \begin{bmatrix} \theta_0 \\ \theta_1 \\ \vdots \\ \theta_\nu \end{bmatrix} = \tau\theta \end{aligned}$$

Because partitions $\mathcal{I}_j$ are assumed to be distinct, their centroids τ_j° are distinct, and therefore not all equal to 0 or 1, and we can therefore conclude that τ is a basis spanning the polynomial vector space P. Therefore, if $p = \dim(P)$ we can trivially solve the linear system of equations in Equation 2.6.2 to find

$$\tilde{\theta} = \tau^{-1}\gamma.$$

If $p > \dim(P)$, then we have an over-determined system, which will not necessarily be consistent. Note that this will only happen in an asymptotic context, as given finite T, we will rarely have exact linear dependence.

We can find the expectation of our estimator as follows:

$$\mathbb{E}[\tilde{\theta}] = \mathbb{E}[\tau^{-1}\gamma] = \tau^{-1}\mathbb{E}[\gamma] = \tau^{-1} \begin{bmatrix} \frac{N(\mathcal{I}_1)}{|\mathcal{I}_1|} \\ \frac{N(\mathcal{I}_2)}{|\mathcal{I}_2|} \\ \vdots \\ \frac{N(\mathcal{I}_p)}{|\mathcal{I}_p|} \end{bmatrix} = \tau^{-1} \begin{bmatrix} \gamma_1^{\text{MLE}} \\ \gamma_2^{\text{MLE}} \\ \vdots \\ \gamma_p^{\text{MLE}} \end{bmatrix} \tag{2.27}$$

Similarly, variance is calculated as

$$\mathrm{Var}(\tilde{\theta}) = \mathrm{Var}(\tau^{-1}\gamma) = \tau^{-1}\mathrm{Var}(\gamma^{\mathrm{MLE}})\tau^{-1\top}.$$

Consistency of $\tilde{\theta}$ is demonstrated via simulation in Section 2.7, and a formal proof follows from the properties of inhomogeneous Poisson processes, and the consistency of MLE, see Ogata (1978).

2.7 Simulation Study

As a proof of concept, we demonstrate that the SG estimates tend to be reasonably accurate and become increasingly accurate as T gets large for a variety of simple point processes. The `R` code for the studies can be found in Section 7.1.2. Figure 2.1 shows a simulated Cox process directed by intensity

$$\lambda(t, x, y) = e^{\alpha x} + \beta e^{y} + \gamma xy + \delta x^2 + \eta y^2 + W(x, y)$$

on $[0, 1] \times [0, 1] \times [0, 1]$, where $\theta = \{\alpha, \beta, \gamma, \delta, \eta\}$ and $W(x, y)$ is a two-dimensional Brownian sheet. The estimated intensity using the SG estimator of θ closely resembles the true intensity even though T is only 1.

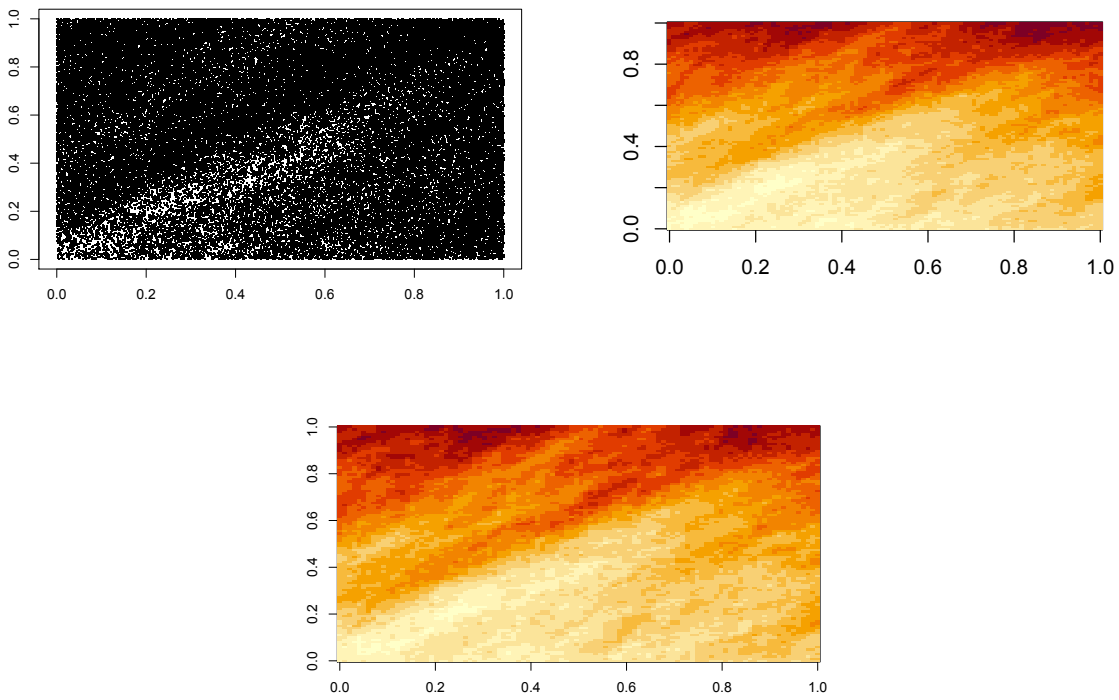


Figure 2.1: Clockwise from top left: (a) Simulated Cox process with intensity dependent on a two-dimensional Brownian sheet. (b) The true intensity

$$\lambda(t, x, y) = e^{\alpha x} + \beta e^{y} + \gamma xy + \delta x^2 + \eta y^2 + W(x, y)$$

on $[0, 1] \times [0, 1] \times [0, 1]$, where $W(x, y)$ is a two-dimensional Brownian sheet with zero drift and standard deviation $\sigma = 50$. The true parameter vector $\theta = \{\alpha, \beta, \gamma, \delta, \eta\} = \{-2, 3, 4, 5, -6\}$. (c) The estimated intensity using the SG estimator of θ.

Figure 2.2 shows a simulated Hawkes process on the unit square and in time interval $[0, 1000]$ with conditional intensity

$$\lambda(t, x, y) = \mu + \kappa \sum_{i:t_i<t} g(t - t_i)h(x - x_i, y - y_i),$$

where $g(t) = 1/\alpha$ on $[0, \alpha]$, $h(x, y) = 1/(\pi r^2)$ for $r \in [0, \beta]$. Here the parameters to be estimated are $\theta = \{\mu, \kappa, \alpha, \beta\}$ and the true values are $\{1, 0.5, 100, 0.1\}$. As with the Cox

process, the conditional intensity estimated using the SG estimator is a close approximation of the true conditional intensity for the Hawkes process.

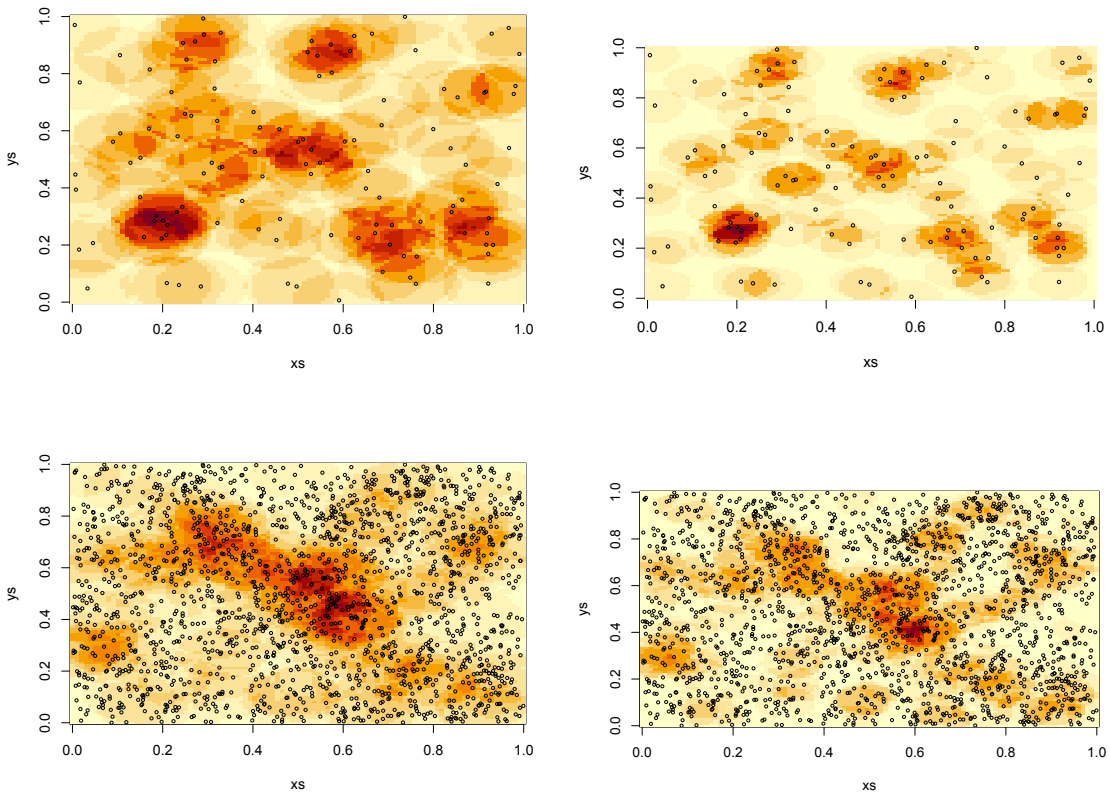


Figure 2.2: Conditional intensity of a simulated Hawkes process with

$$\lambda(t, x, y) = \mu + \kappa \sum_{i:t_i<t} g(t - t_i)h(x - x_i, y - y_i)$$

where $g(t) = 1/\alpha$ on $[0, \alpha]$ and $h(x, y) = 1/(\pi r^2)$ for $r \in [0, \beta]$ on $[0, 1] \times [0, 1] \times [0, T]$. $\theta = \{\mu, \kappa, \alpha, \beta\} = \{1, 0.5, 100, 0.1\}$. Clockwise from top left: (a) True conditional intensity at time $T = 100$. (b) Conditional intensity estimated via SG, at time $T = 100$. (c) True conditional intensity at time $T = 1000$. (d) Conditional intensity estimated via SG, at time $T = 1000$.

Figure 2.3 shows a comparison of the root mean square error (RMSE) and `R` computation time for MLE and SG estimates of the process simulated in Figure 2.2 observed on $[0,1] \times [0,1] \times [0,T]$ for various values of T. For this comparison, the integral approximation technique detailed in Schoenberg (2013) is used for MLE and $p = 4^2$ is chosen for the SG-estimator.

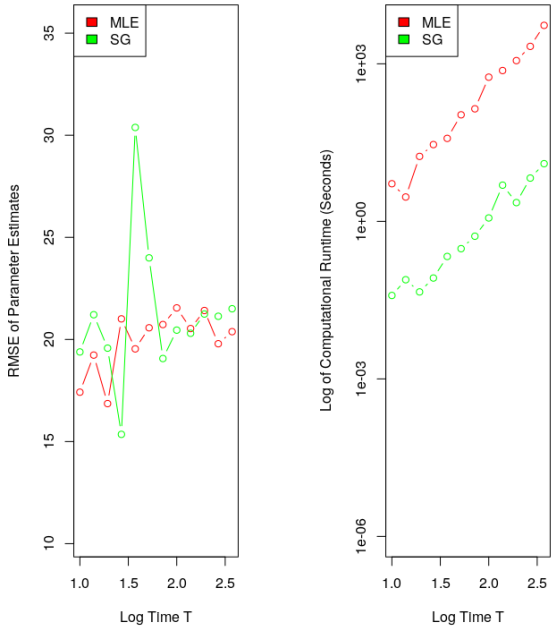


Figure 2.3: Comparison of estimate accuracy and computational (time) expense for MLE and SG-estimators. Conditional intensity of a simulated Hawkes process with

$$\lambda(t,x,y) = \mu + \kappa \sum_{i:t_i<t} g(t-t_i)h(x-x_i, y-y_i)$$

where $g(t) = 1/\alpha$ on $[0,\alpha]$ and $h(x,y) = 1/(\pi r^2)$ for $r \in [0,\beta]$ observed on $[0,1]\times[0,1]\times[0,T]$. $\theta = \{\mu,\kappa,\alpha,\beta\} = \{1, 0.5, 100, 0.1\}$. Left: root mean square error (RMSE) of parameter estimates for MLE and SG-estimates across various T. Right: computational runtime in seconds for computing MLE and SG-estimates.

Figures 2.4 and 2.5 shows the behavior of SG estimates as T increases for an inhomogenous Poisson process on $[0,T]\times[0,1]\times[0,1]$. We simulated six partitioning schemes ranging from $p=1^2$ to $p=32^2$, and various values of increasingly large T. We chose intensity

$$\lambda(t,x,y)=\alpha x^2+\beta y^2+\gamma x+\delta y+\epsilon,$$

where the vector of parameters to be estimated is

$$\theta=\{\alpha,\beta,\gamma,\delta,\epsilon\}=\{1/3,2/3,1/2,1/4,1/5\}.$$

The conditional intensity specified has t constant to avoid an explosive process, or a process where too few points are observed as T gets larger. The estimates of θ are seen to converge to θ as $T\to\infty$.

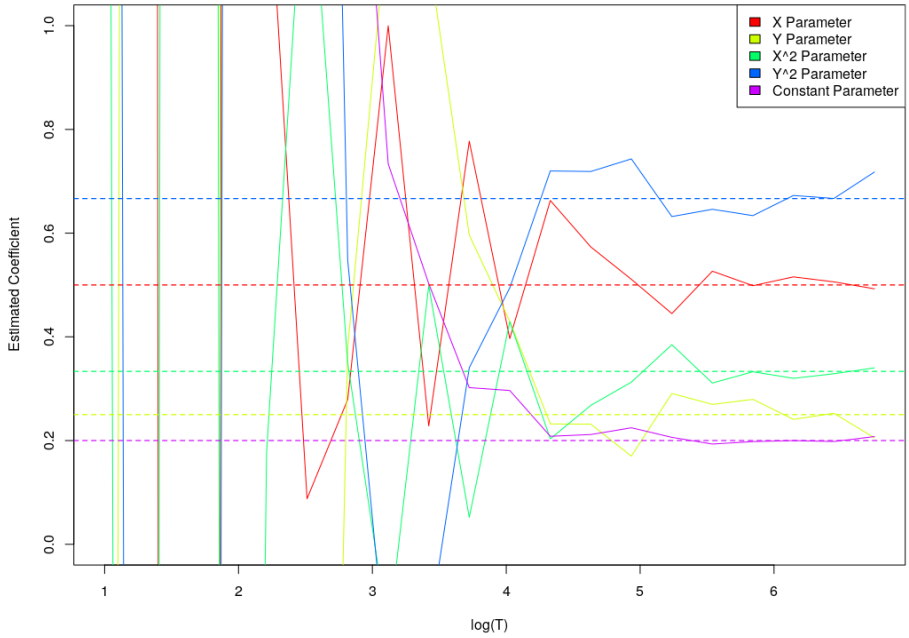


Figure 2.4: Intensity $\lambda(t,x,y)=x^2/3+(2y^2)/3+x/2+y/4+1/5$ estimated using $p=32^2$ partitions. Parameter estimates become increasingly accurate as $T\to\infty$. Horizontal dotted lines indicate true parameter values.

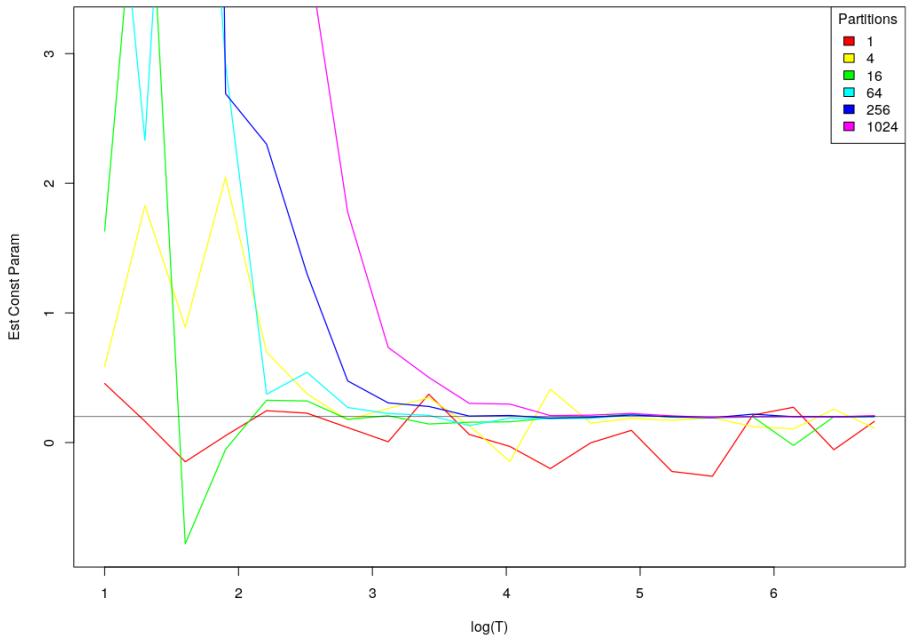


Figure 2.5: Estimates of a single parameter for a Poisson process with intensity $\lambda(t, x, y) = x^2/3 + (2y^2)/3 + x/2 + y/4 + 1/5$. Note that if $p = 1$ or $p = 4$, estimates are not accurate as Assumption A1 is violated.

Figures 2.6 and 2.7 show the behavior of aggregated SG-type estimates as T increases for an inhomogenous Poisson process on $[0, T] \times [0, 1] \times [0, 1]$. Similar to Figures 2.4 and 2.5, we simulated six partitioning schemes ranging from $p = 1^2$ to $p = 32^2$, and various values of increasingly large T. We chose the same intensity

$$\lambda(t, x, y) = \alpha x^2 + \beta y^2 + \gamma x + \delta y + \epsilon,$$

where the vector of parameters to be estimated is

$$\theta = \{\alpha, \beta, \gamma, \delta, \epsilon\} = \{1/3, 2/3, 1/2, 1/4, 1/5\}.$$

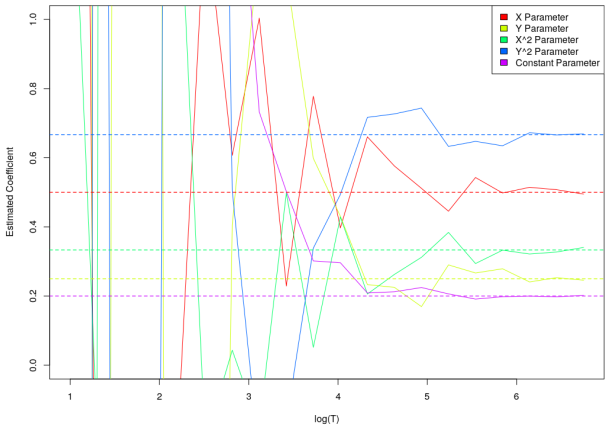


Figure 2.6: Similar to Figure 2.5, but with a aggregated SG estimator within an optimization routine, as opposed to directly solving for coefficients like in Figure 2.8. The aggregation statistic used is the mean, $\bar{\tau}_j$.

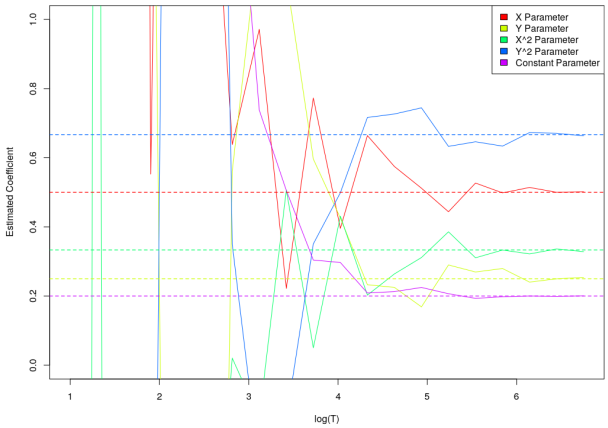


Figure 2.7: Similar to Figure 2.6, but using the centroid τ_j°.

Finally, in Figure 2.8 we estimated coefficients for the process $\lambda(t, x, y) = 3x+6y+5xy+10$ using an analytic solution for the approximated SG estimator, as discussed in Section 2.6.2. For this example, we chose the mean $\bar{\tau}_j$ as our aggregating statistic. We show a visualization of results where $p = 4$, although any $\dot{p} > p = 4$ suffices for identifiability requirements, limited only by computational precision for a floating point number needed for computation of τ^{-1}.

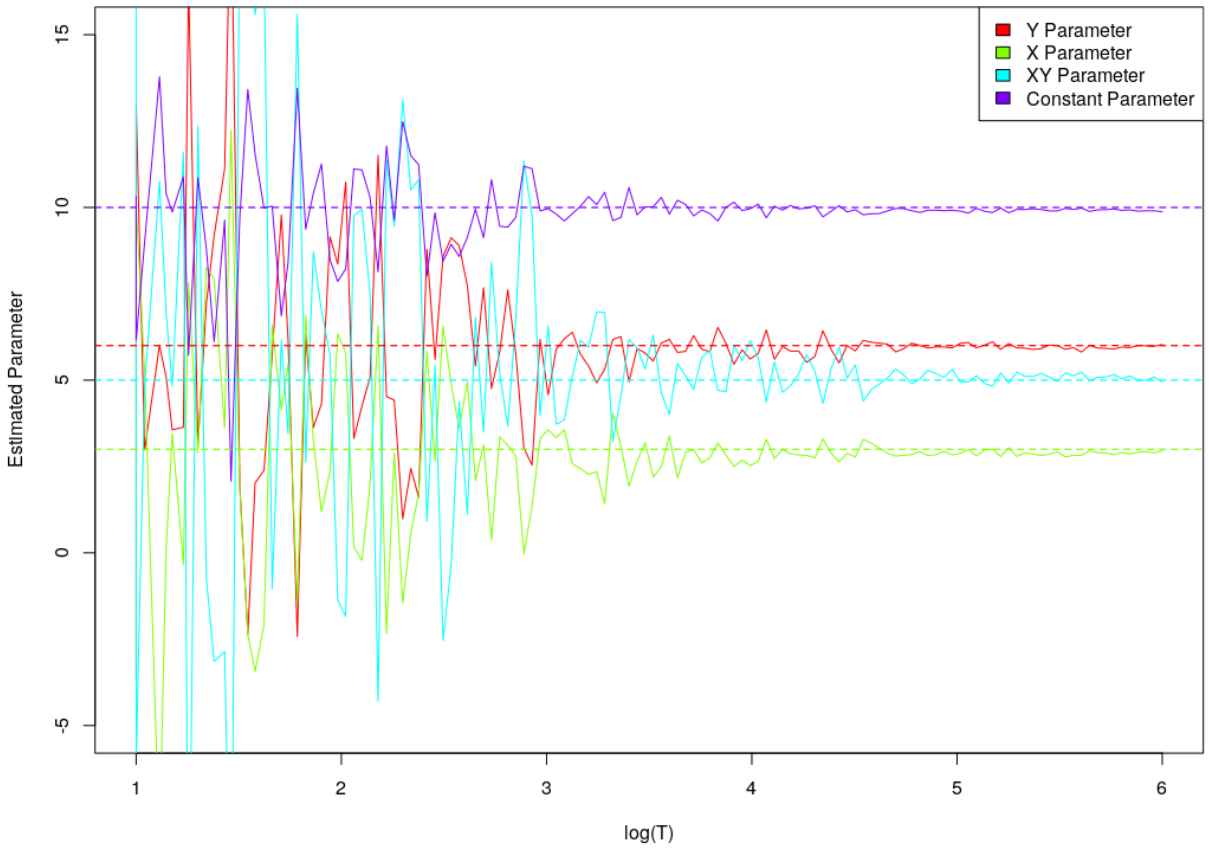


Figure 2.8: Here we see estimated coefficients for the process $\lambda(t, x, y) = 3x + 6y + 5xy + 10$ using an analytic solution for the approximated SG estimator.

2.8 Conclusion and Future Work

The SG estimator is very simple and efficient computationally and, like the MLE, is a consistent estimator for a wide class of point process models. We recommend its use as a complement to the MLE, in the many cases where the integral term in the loglikelihood is computationally burdensome to estimate accurately. This may be especially true for the rapidly emerging cases of big data where the observed number of points is very large and/or the spatial observation region is very large or complex. In situations where MLE is preferred but is sensitive to the choice of starting values in the optimization, a practical option may

be to use the SG estimator as a starting value.

Future research should focus on how best to choose the nature and number of cells in the partition when implementing SG estimation. For example, in some cases efficiency gains might be achieved via data-dependent partitioning schemes, such as Voronoi tessellations. Our preliminary investigations suggest, however, that any reasonable choice of partition will do, provided p is large enough to satisfy Assumption A1. Partitions for the case where the spatial dimension is 3 or higher are also important areas for future study.

As mentioned in Section 2.4, the SG estimator proposed here minimizes the sum of squares of the integral of the residual field over cells in a partition, but another area for future research would be to consider alternatively minimizing some other criterion, such as for example the integral of $Q^2(s)$. Such an alternative may avoid the need for choosing a rather arbitrary partition, but would replace this with the need to choose a bandwidth for the kernel smoother.

Another possibility for estimating point process parameters is via partial log-likelihood maximization (Diggle et al., 2010), and like the SG estimator, such estimators also do not require the computation or approximation of the integral term in the ordinary log-likelihood. As noted in the discussion in Diggle (2006), the partial log-likelihood estimate may be less efficient than the MLE but can be much easier and faster to compute. Future studies should investigate the advantages and disadvantages of such estimators relative to the SG estimator, both in terms of accuracy and computation speed.

2.9 Acknowledgements

Primary thanks are due to Frederic Schoenberg who proposed the Stoyan-Grabarnik statistic as a means to form an estimator of the parameters of conditional intensity functions. Further thanks to Adrian Baddeley for suggesting estimation based on kernel smoothing the scaled residual field, to Peter Diggle for suggesting the partial likelihood as an alternative way of

avoiding having to compute the integral in the MLE, to Aila Saarka for suggesting that the SG estimator could be used to obtain starting values for the MLE, and to James Molyneux for his work on selecting partitions. The contents of the chapter are largely contained within the paper *Parametric estimation of spatial temporal point processes using the Stoyan-Grabarnik Statistic* by Conor Kresin and Frederic Schoenberg, Annals of the Institute of Statistical Mathematics (2023).

2.10 Addendum: Consistency of MLE

Proof of the consistency of the MLE estimator for the conditional intensity of point processes is detailed in (Ogata, 1978). We now present a structural summary of Ogata's proof. Ogata assumes that an observed realization is from a simple and stationary[1] point process, characterized by a finite measure on a compact seperable metric space. Ogata further assumes that the the second moments exist and are finite for all $\theta \in \Theta$, the parameter space of the log likelihood.

Ogata argues consistency by leveraging properties of a predictable process in the context of the expected value of a log likelihood ratio (commonly referred to as Kullback-Leibler divergence). We now sketch out Ogata's proof, using Ogata's notation. Let ϕ be a set of points from a realization of a point process, and

$$\lambda(t,\phi) = \lim_{\delta \to 0} \mathbb{P}\left(N([t,t+\delta)) > 0 | \mathcal{H}_{-\infty,t}\right)$$
$$\lambda^*(t,\phi) = \lim_{\delta \to 0} \mathbb{P}\left(N([t,t+\delta)) > 0 | \mathcal{H}_{0,t}\right) = \mathbb{E}[\lambda(t,\phi)|\mathcal{H}_{0,t}]$$

where $\mathcal{H}_{s,t}$ is the σ-algebra generated by $\{N(u,t] \text{ s.t. } s < u \leq t\}$. We then denote $\{\lambda_\theta(t,\phi) \text{ s.t. } \theta \in \Theta \subset \mathbb{R}^d\}$ as the family of conditional intensities for simple stationary processes $\{P_\theta \text{ s.t. } \theta \in \Theta\}$ and due to simplicity and stationarity, this correspondence is unique. Ogata denotes the

[1]Together, stationarity and simplicity guarantee *orderliness*, see (Daley and Vere-Jones, 2007).

log-likelihood

$$L_T^*(\theta) = -\int_0^T \lambda^*(t,\phi)dt + \int_0^T \log \lambda^*(t,\phi)dN(t)$$

for realization ϕ on interval $[0,T]$. Finally, $\hat{\theta}_T$ denotes the maximizing $\theta \in \Theta$ of $L_T^*(\theta)$ for ϕ observed from point process P_{θ_0}, *i.e.* θ_0 is the true parameterization. For notational conciseness, Ogata uses a log-likelihood under information from the "infinite past" denoted as

$$L_T(\theta) = -\int_0^T \lambda(t,\phi)dt + \int_0^T \log \lambda(t,\phi)dN(t).$$

For the purposes of consistency of the MLE, this infinite past log-likelihood is trivially replaced with its "finite past" equivalent.

Ogata notes as a primary lemma, Lemma A[2] necessary for proof of consistency the following: if $\xi(t,\phi)$ is a stationary and predictable process with a finite second order moment, then

$$\lim_{T\to\infty} \frac{1}{T}\int_0^T \xi(t,\phi)dt = \mathbb{E}[\xi(0,\phi)] = \lim_{T\to\infty} \frac{1}{T}\int_0^T \xi(t,\phi)\frac{dN(t)}{\lambda_{\theta_0}(t,\phi)}.$$

As secondary lemma, Lemma B [3] necessary for Ogata's proof of consistency, we present the expected value of the ratio of the log likelihood:

$$\mathbb{E}[\Lambda_1(\theta_0;\theta)] = \mathbb{E}\left[\int_0^1 (\lambda_\theta(t,\phi) - \lambda_{\theta_0}(t,\phi))dt + \int_0^1 \log\left(\frac{\lambda_{\theta_0}(t,\phi)}{\lambda_\theta(t,\phi)}\right) dN(t)\right]$$

and note that by a typical log-concavity argument, $\mathbb{E}[\Lambda_1(\theta_0;\theta)] \geq 0$ with equality *iff* $\lambda_\theta(t,\phi) = \lambda_{\theta_0}(t,\phi)$ *a.s.*

Lastly, we define a *neighborhood* U of θ denoted $U = U(\theta)$ such that for all $\theta' \in U$

$$|\lambda_{\theta'}(0,\phi)| \leq \Lambda_0(\phi) \quad \text{and} \quad |\log \lambda_{\theta'}(0,\phi)| \leq \Lambda_1(\phi)$$

where $\Lambda_0(\phi), \Lambda_1(\phi)$ are random variables with finite second moments - as seen above, Λ in this context is the (log) likelihood ratio.

---

[2]Lemma 2 in (Ogata, 1978).

[3]Lemma 3 in (Ogata, 1978).

With this notation in hand, we present an annotated sketch of Ogata's proof. Statement: $\hat{\theta}_T \xrightarrow{p} \theta_0$ as $T \to \infty$. Proof (sketch): Because λ_θ is predictable for all $\theta \in \Theta$, we know that as the neighborhood $U(\theta) \to \{\theta\}$,

$$\begin{aligned}\mathbb{E}\left[\inf_{\theta' \in U} \lambda_{\theta'}(0,\phi)\right] &\to \mathbb{E}\left[\lambda_\theta(0,\phi)\right] \\ \mathbb{E}\left[\lambda_{\theta_0}(0,\phi)\log\left(\frac{\lambda_{\theta_0}(0,\phi)}{\sup_{\theta' \in U}\lambda_{\theta'}(0,\phi)}\right)\right] &\to \mathbb{E}\left[\lambda_{\theta_0}(0,\phi)\log\left(\frac{\lambda_{\theta_0}(0,\phi)}{\lambda_\theta(0,\phi)}\right)\right].\end{aligned}$$

Define an open neighborhood around the true parameter(s) θ_0, U_0. Then we know that for any $\theta \in \Theta \setminus U_0$, we can find some $\epsilon > 0$ such that $\mathbb{E}[\Lambda_1(\theta_0;\theta)] \geq 3\epsilon$ because $\lambda_{\theta_1}(0,\phi) = \lambda_{\theta_2}(0,\phi) \iff \theta_1 = \theta_2$, which in turn implies that the Kullback-Liebler divergence $\mathbb{E}[\Lambda_1(\theta_0;\theta)] > 0 \iff \theta_0 \neq \theta$.

It is therefore possible to choose U such that

$$\begin{aligned}&\mathbb{E}\left[\inf_{\theta' \in U} \lambda_{\theta'}(0,\phi) - \lambda_{\theta_0}(0,\phi) + \lambda_{\theta_0}(0,\phi)\log\left(\frac{\lambda_{\theta_0}(0,\phi)}{\sup_{\theta' \in U}\lambda_{\theta'}(0,\phi)}\right)\right] \\ \to &\mathbb{E}\left[\lambda_\theta(0,\phi) - \lambda_{\theta_0}(0,\phi) + \lambda_{\theta_0}(0,\phi)\log\left(\frac{\lambda_{\theta_0}(0,\phi)}{\lambda_\theta(0,\phi)}\right)\right] \qquad (2.28) \\ \geq &\mathbb{E}\left[\int_0^1 (\lambda_\theta(t,\phi) - \lambda_{\theta_0}(t,\phi))dt + \int_0^1 \log\left(\frac{\lambda_{\theta_0}(t,\phi)}{\lambda_\theta(t,\phi)}\right) dN(t)\right] - \epsilon \qquad (2.29) \\ = &\mathbb{E}[\Lambda_1(\theta_0;\theta)] - \epsilon.\end{aligned}$$

The final term of (2.28) simplifies in (2.29) because

$$\mathbb{E}\left[\int_0^T \xi(t,\phi)dN(t)\right] = \mathbb{E}\left[\int_0^T \xi(t,\phi)\lambda_{\theta_0}(t,\phi)dt\right]$$

which is true for any finite predictable process ξ (Meyer, 2006).

We can cover $\Theta \setminus U_0$ with a finite number of neighborhoods of θ_s, U_s for some s less than or equal to the cardinality of ϕ. Because $\inf_{\theta' \in U} \lambda_{\theta'}(0,\phi)$ and $\sup_{\theta' \in U} \lambda_{\theta'}(0,\phi)$ are both predictable processes we can see that for any $\epsilon > 0$, there exists $T_0(\epsilon) = T_0$ such that for any $T > T_0$,

$$\frac{1}{T}L_T(\theta_0) - \sup_{\theta \in U_s} L_T(\theta) \qquad (2.30)$$

$$
\begin{aligned}
=&\frac{-1}{T}\left(\int_0^T \lambda_{\theta_0}(t,\phi)dt + \int_0^T \log \lambda_{\theta_0}(t,\phi)dN(t)\right) && (2.31)\\
&+\sup_{\theta\in U_s}\frac{1}{T}\left(\int_0^T \lambda_{\theta}(t,\phi)dt + \int_0^T \log \lambda_{\theta}(t,\phi)dN(t)\right)\\
=&\frac{1}{T}\int_0^T\left(\sup_{\theta\in U_s}\lambda_\theta(t,\phi)-\lambda_{\theta_0}(t,\phi)\right)dt+\frac{1}{T}\int_0^T\log\left(\frac{\lambda_{\theta_0}(t,\phi)}{\sup_{\theta\in U_s}\lambda_\theta(t,\phi)}\right) && (2.32)\\
\geq&\frac{1}{T}\int_0^T\left(\inf_{\theta\in U_s}\lambda_\theta(t,\phi)-\lambda_{\theta_0}(t,\phi)\right)dt+\frac{1}{T}\int_0^T\log\left(\frac{\lambda_{\theta_0}(t,\phi)}{\sup_{\theta\in U_s}\lambda_\theta(t,\phi)}\right) && (2.33)\\
\geq&\left(\mathbb{E}\left[\int_0^1(\lambda_\theta(t,\phi)-\lambda_{\theta_0}(t,\phi))dt+\int_0^1\log\left(\frac{\lambda_{\theta_0}(t,\phi)}{\lambda_\theta(t,\phi)}\right)dN(t)\right]-\epsilon\right)-\epsilon && (2.34)\\
=&\mathbb{E}[\Lambda_1(\theta_0;\theta)]-2\epsilon && (2.35)\\
\geq&3\epsilon-2\epsilon && (2.36)\\
\geq&\epsilon. && (2.37)
\end{aligned}
$$

Note that T_0 is dependent on the realization ϕ. It follows that there exists $T_1 = T_1(\epsilon, U_0)$ such that for all $T > T_1$,

$$\sup_{\theta\in U_0} L_T(\theta) \geq \sup_{\theta\in\Theta\setminus U_0} L_T(\theta) + \epsilon T.$$

This implies that $\hat{\theta} \in U_0$, concluding the proof. We note (as does Ogata) that the same machinery is valid if we replace λ with * and L_T with L_T^*.

In summary, Ogata's proof relies on using a bounded neighborhood around θ_0. This bound relies on the log likelihood and associated log likelihood ratio. Ogata leverages the fact that the conditional intensity is a predictable function, and argues that the expected log likelihood ratio is greater than ϵ for any θ outside the neighborhood of θ_0, therefore deriving his result from the converse. Crucially, his result relies on two lemmas: Lemma A, the so-called martingale property applied to likelihood functions, and Lemma B, which states that the expected log likelihood ratio is equal to zero *iff* the conditional intensities are the same (Kullback-Leibler).

CHAPTER 3

Hawkes-Type Models and Their Compartmental Equivalents

3.1 Hawkes Models

The Hawkes model or self-exciting point process model is commonly used to model clustered point patterns in applications such as seismology, finance, crime, and infectious diseases (Daley and Jones, 2003; Reinhart, 2018; Ogata, 1988; Cauchemez et al., 2006). A spatial-temporal Hawkes process is specified by the model

$$\begin{aligned} \lambda(s,t|\mathcal{H}_{t-}) &= \underbrace{M(s)}_{\text{background rate}} + K \int_{t'<t} \underbrace{g(s-s',t-t')}_{\text{triggering density}} dN(s',t') \\ &= M(s) + K \sum_{(s',t'):t'<t} g(s-s',t-t'), \end{aligned} \tag{3.1}$$

for $s \in X \subseteq \mathbb{R}^2$ and $t \in [0,T)$, where $\lambda(s,t|\mathcal{H}_t)$ is the conditional rate at which points (events) are expected to accumulate around spatial-temporal location (s,t), given information on all previous events $\mathcal{H}_t$. As discussed in Chapter 1, the conditional intensity uniquely characterizes the finite-dimensional distribution of any simple point process (see Prop. 7.2.IV of (Daley and Jones, 2003)), and thus equation (3.1) fully specifies the model. The function g is typically assumed to be a density, i.e. to be nonnegative and to integrate to 1 over all time and space, and is called the triggering density. Common choices for g are the exponential or Pareto densities in time, and the Gaussian or Pareto densities in space (Reinhart, 2018). The constant K is called the productivity. Provided g is a density function, K is the expected number of points triggered directly by each point, and is thus closely connected to

the reproduction number in compartmental models such as SEIR. Each background point, associated with $\mu(s)$, is expected to generate $K + K^2 + K^3 + ... = 1/(1-K) - 1 = K/(1-K)$ triggered points. As a result, in a Hawkes process, the expected fraction of background points is $1 - K$. Extensions on the triggering function such as inhibitory or non-linear relations are discussed in (Chen et al., 2017).

Given a dataset consisting of n points within a space-time observation region B, the parameters in Hawkes processes are typically fit by maximum likelihood estimation (MLE), where one obtains parameter estimates $\hat{\theta}$ maximizing the log likelihood, see Equation 2.1. The resulting estimates have desirable properties. For instance, Ogata (1978) showed that the MLE $\hat{\theta}$, is, under standard conditions, asymptotically unbiased, consistent, asymptotically normal, and asymptotically efficient, with standard errors readily constructed using the diagonal elements of the inverse of the Hessian of L evaluated at $\hat{\theta}$ (Ogata, 1978). Further, if the fitted model is missing some relevant covariates, under general conditions the MLE will nevertheless be consistent, provided the effect of the missing covariates is small (Schoenberg, 2016). The triggering function can also be estimated non-parametrically (Marsan and Lengline, 2008), and some authors have also estimated the background rate $\mu(s)$ nonparametrically, *e.g.* (Zhuang et al., 2004; Park et al., 2019). Bayesian methods can also be used to estimate parameters and quantify uncertainty in Hawkes process models (Rasmussen, 2013; Mohler et al., 2013). Recently, Hawkes processes have been extended to accommodate a recurrent neural network (RNN) setting wherein predicted event intensities are estimated conditional upon the hidden state of the network based on past events (Mei and Eisner, 2016).

A host of other variations of the Hawkes model have been proposed (Rizoiu et al., 2018; Chiang et al., 2020). The HawkesN model, as defined in (Rizoiu et al., 2018), has a Hawkes conditional intensity scaled by the proportion of events which can still occur after time t, in order to account for the dynamic decrease in the number of susceptible individuals in a

given location (Rizoiu et al., 2018):

$$\lambda(t) = (1 - I_c(t)/N)(\mu + K \sum_{t'<t} g(t - t')). \tag{3.2}$$

In the context of a Hawkes process modeling the spread of an infectious disease, $I_c(t)$ is the cumulative number of infections that have been recorded up to time t and N is the total population size.

Hawkes models and their slight variants such as the epidemic-type aftershock sequence (ETAS) model (Ogata, 1988, 1998), HawkesN (Bertozzi et al., 2020; Rizoiu et al., 2018; Mohler et al., 2020), and the recursive model (Schoenberg et al., 2019) have been shown to be useful in modeling infectious diseases such as Ebola (Kelly et al., 2019; Park et al., 2020), chlamydia (Schoenberg, 2020), SARS (Cauchemez et al., 2006; Wallinga and Teunis, 2004), measles (Farrington et al., 2003), meningococcal disease (Meyer et al., 2014), and Rocky Mountain Spotted Fever (Schoenberg et al., 2019). Hawkes models have also been shown to be the best fitting models for forecasting seismicity in rigorous, purely prospective earthquake forecasting studies such as the Collaboratory for the Study of Earthquake Predictability (CSEP) (Clements et al., 2011, 2012; Zechar et al., 2013; Bray et al., 2014; Gordon et al., 2015; Schorlemmer et al., 2018).

3.2 Modeling Contagious Disease Spread

Various Hawkes triggering functions are equivalent to compartmental models which can be represented as systems of ordinary differential equations. For instance, the equivalence between SIR models and HawkesN with an exponential kernel is described in (Rizoiu et al., 2018, 2017). We explore this relationship in the remainder of this chapter, in the context of contagious disease spread data.

The SARS-CoV2 (Covid-19) pandemic spread from China to at least 188 countries or regions in the first six months of 2020 (Dong et al., 2020). The characteristics of the Covid-

19 virus have been estimated and forecasted by numerous researchers with highly variable results. Estimates of properties such as reproduction rate (or time-varying reproduction number), numbers of individuals infected, hospitalization rates, fatality rates, and efficacy of containment measures have varied widely (Baud et al., 2020; Ferguson et al., 2020). Accurate real-time estimates of the spread of Covid-19 are difficult to achieve without population-wide testing (Day, 2020; Bertozzi et al., 2020). Nevertheless, it is important for researchers to accurately estimate and forecast the dynamics of Covid-19 so that optimal public policy measures and other responses can be adopted.

Several different frameworks have been proposed for modeling the spread of Covid-19, including compartmental models such as the SEIR (Susceptible $\rightarrow$ Exposed $\rightarrow$ Infectious $\rightarrow$ Removed) differential equation model, and branching point process models such as the Hawkes point process model (Rizoiu et al., 2018; Jewell et al., 2020; Bertozzi et al., 2020; Chiang et al., 2020). This paper compares these two approaches for forecasting Covid-19. Relative to Hawkes models, SEIR models and their variants have been used far more widely to describe the Covid-19 pandemic (Bertsimas, 2020; for Health Metrics and , IHME; Gu, 2020b; Laboratory, 2020) as well as other infectious diseases such as Ebola (Lekone and Finkenstädt, 2006) and SARS (Dye and Gay, 2003). However, recent studies have suggested that Hawkes models may be more accurate (Yang, 2019). For general discussion of mathematical and statistical models of epidemiological phenomena, see (O'Neill, 2010; Grassly and Fraser, 2008).

The remainder of this chapter is structured as follows. Following a review of SEIR models in Section 3.2, we compare their advantages and disadvantages with that of Hawkes models, especially with respect to forecasting Covid-19 cases or deaths in Section 3.4. In Section 3.5, we detail the mathematical connection between Hawkes processes and SEIR models, and in Section 3.6 we provide concluding remarks.

3.3 The SEIR Model

SEIR models and their variants have been widely used to model and forecast the spread of many contagious diseases including Covid-19 (Vishal Tomar, 2020; Yamana et al., 2020; Bertsimas, 2020; Gu et al., 2020; Lemaitre et al., 2020; Guido Espana, 2020; Gu, 2020a; Phil Arevalo et al., 2020; Michael L. Mayo et al., 2020; Gu, 2020b). Such models employ a wide variety of modifications to the classic SEIR model, including using Bayesian inference (Michael L. Mayo et al., 2020), machine learning (Gu, 2020b), mobility networks and ensemble approaches (Guido Espana, 2020), and mixed-effects curve fitting (Vishal Tomar, 2020) to fit parameters, as well as slight compartmental variants like SuIER which account for unreported cases (Gu, 2020a). Other models explicitly define scenarios for government interventions or enforcement of public health policies in specific populations (Lemaitre et al., 2020; Bertsimas, 2020; Gu et al., 2020).

SEIR models assume that individuals within each category, or compartment (susceptible, exposed, infectious, and recovered), share pertinent characteristics, and the size of the population of interest N is equal to the total number of individuals in the compartments (Kermack and McKendrick, 1927). SEIR models are a slight extension of SIR (Susceptible $\rightarrow$ Infectious $\rightarrow$ Removed) models, generalized to account for the fact that there is an incubation time for some infectious diseases like Covid-19, during which the exposed host may be asymptomatic and thus not recorded as infected. SEIR models can be either deterministic, in which case they are comprised of a system of differential equations, or stochastic, in which case they are based on a Markov chain framework. Given large populations, sufficient initial spread, and enough time, the deterministic framework should resemble the stochastic framework in expectation, assuming properly specified models (Rizoiu et al., 2018).

Deterministic SEIR models, such as that described in Figure 3.1, can provide a reasonable approximation of the characteristics of a contagious disease such as Covid-19. There are numerous variations, but the basic idea conveyed in Figure 3.1 common to compartmental

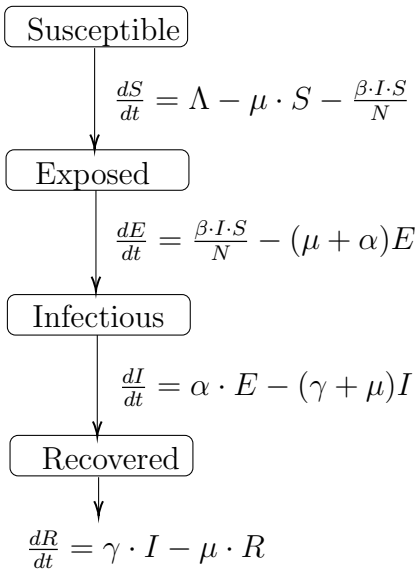


Figure 3.1: Diagram of the deterministic SEIR model. Definitions: N is a constant number of individuals in a susceptible population, $\beta \cdot I$ is equal to the force of infection, Λ equal to birth rate, μ equal to death rate, γ equal to mortality rate, α^{-1} equal to the average incubation period. Such a model has reproduction number $K = \frac{\alpha \cdot \beta}{(\mu+\alpha)(\mu+\gamma)}$.

models is that there is some rate at which people shift from one portion of the population to another, e.g. from the susceptible population to the exposed population, and these rates may be fixed or allowed to vary over time subject to certain constraints. Deterministic models such as that shown in Figure 3.1 can be extended to allow parameters governing the force of infection, number of cases by symptom onset, and death rate, with movement between compartments commonly specified as binomial random variables. Such a model has been suggested for the transmission of Ebola, for instance (Lekone and Finkenstädt, 2006). Number of cases or deaths are commonly specified as a negative binomial random variable (Levin and Andreasen, 1986; Hernández et al., 2020).

Perhaps the most common method for estimating compartmental infectious disease model parameters is by using Bayesian estimation (Ozanne et al., 2019; Clancy et al., 2008). Prior parameters are often decided on using subject matter experts (Michael L. Mayo et al., 2020) or parameters fit to prior outbreaks (Frasso and Lambert, 2016). Bayesian SEIR

models have been employed to model infectious diseases such as Ebola (Frasso and Lambert, 2016), Visceral Leishmaniasis (Ozanne et al., 2019) and Covid-19 (Michael L. Mayo et al., 2020). Prior distributions are typically specified by compartment. For instance, Frasso et al. specified number of deaths as beta-distributed, duration of incubation as normal, and observed cases as negative binomial (Frasso and Lambert, 2016). Disease characteristics such as reproduction number has been modelled within the context of SIR models with a gamma prior (Clancy et al., 2008). Joint posteriors are then solved for using a MCMC approach such as Metropolis-Hastings or Gibbs sampling (Clancy et al., 2008; Lekone and Finkenstädt, 2006).

Stochastic versions of the SIR and SEIR models allow researchers to include the effect of networks of individuals, but specification and parameter estimation can be more challenging (Artalejo et al., 2015). Various stochastic SEIR models have been developed to model Covid-19 data. A stochastic SEIR model with parameters fit using grid search, which may be viewed as a relatively agnostic machine learning approach, was implemented in (Gu, 2020b), and a stochastic SEIR model hybrid with agent-based simulation was suggested in (Laboratory, 2020). The compartmental approach of the SEIR model is slightly modified to accommodate under-detection and differentiated government intervention in the DELPHI model (Bertsimas, 2020). Their flexibility notwithstanding, the difficulty in estimating time-varying parameters in real time for stochastic SEIR models is well known, especially for large populations (Montagnon, 2019).

Parameters in SEIR models are often estimated using opinions of expert epidemiologists or using data from other locations or past epidemics (Chowell and Nishiura, 2014). This is attractive in the sense that expert opinion is integrated, but there is ample opportunity for bias as well as mis-specification, and the parameter estimates have a covariance structure that can be difficult to estimate. Further, non-identifiability is a known problem for compartmental models (Godfrey and Chapman, 1990). Although there exists algebraic approaches for testing identifiability such as exhaustive modeling (Walter and Lecourtier, 1981), such

methods are not implemented in any of the above referenced Covid-19 SEIR models. Crucially, estimated SEIR parameters in the early stages of a epidemic (before peak infection) have been shown to be structurally nonidentifiable (Sauer et al., 2020).

3.4 Comparison of Point Process and Compartmental Models

Hawkes and SEIR models both offer flexible (and somewhat complementary) frameworks for modeling infectious diseases. Hawkes models allow for nonparametric estimation of the triggering function g, as well as spatial covariates, and an intrinsic network-effect. SEIR models offer a far more physically plausible framework for describing Covid-19 relative to the Hawkes model. Specifically, SEIR models allow for specification of stochastic movement between compartments based on previous epidemics and expert opinions. The compartmental model framework allows for natural implementation of known networks within the population of interest (Montagnon, 2019). Further, quantities of interest to epidemiologists and policy makers such as infection rate within a population can be imputed using SEIR models (Lekone and Finkenstädt, 2006).

Within the context of a SEIR model, the spread or transmission of an infectious disease such as Covid-19 occurs via Markovian diffusion which, under certain regularity conditions, ultimately converges to a stationary distribution. In the context of a Hawkes model, background events trigger future events, and these trigger subsequent events, ultimately resolving due to the decay of the chosen triggering function if the productivity is less than one. A Hawkes process with $K > 1$ is not stationary (Stabile and Torrisi, 2010). The HawkesN model is a stationary process for $K > 1$.

The link between Hawkes-like and stochastic SIR models is explored in detail in (Rizoiu et al., 2018), where it is shown that an exponentially decaying triggering function chosen for a finite population Hawkes model (HawkesN) coincides in expectation with the number of individuals infected in a stochastic SIR model as it approaches stationarity. This connection

between SIR and Hawkes models was explored in particular in the context of Covid-19 (Bertozzi et al., 2020), where it was shown that the HawkesN and SIR models converge if the triggering function is exponential and the reproduction number in the SIR model is constant (Bertozzi et al., 2020). SIR and HawkesN models are shown to provide similar fit to Twitter re-tweet diffusions in (Rizoiu et al., 2018).

One may also compare features such as the doubling time for both Hawkes and SIR/-SEIR models. In the early exponential growth stage, the doubling time for SIR is $\tau = \log(2)/(\gamma(R_0 - 1))$ (Allen, 1994). The relationship between estimates of R_0 and doubling time for simulations of compartmental models is summarized in (Merler et al., 2013). The parameter K is intuitively similar to R_0, as it represents the expected number of events triggered by a previous event. The doubling time for HawkesN models as a function of K is shown next to the doubling time of a SIR model as a function of R_0 in Figure 3.2. It should be noted that doubling time for Hawkes models quickly approaches zero for $K > 1$, justifying the finite population correction present in the HawkesN in the context of modeling infectious diseases such as Covid-19 (Rizoiu et al., 2018).

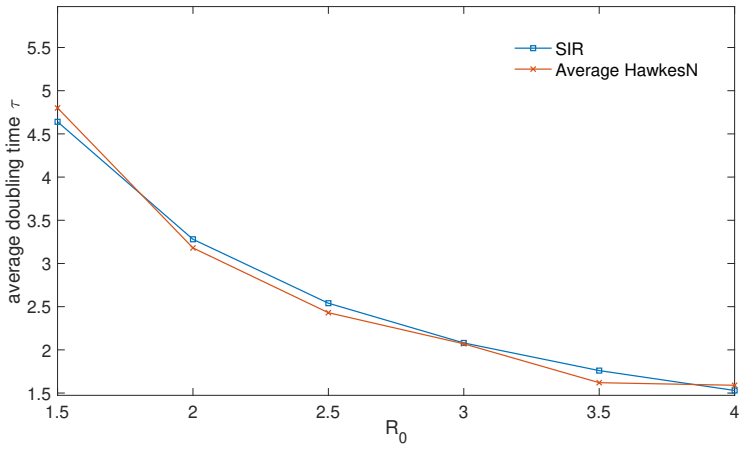


Figure 3.2: Average doubling time for HawkesN model with $\beta = \frac{1}{4}$, $I_0 = 10$, population size $N = 10^6$, and using mean intensity over 100 simulations per K (notated R_0 for SIR). Doubling time is defined as t such that $N(t) = 20$.

Hawkes models offer computationally inexpensive parametric and non-parametric estimates for important characteristics of infectious diseases such as Covid-19. Due to computational difficulty, and model-specification convenience, both SEIR and Hawkes models often make assumptions such as fixed population size, or homogeneity individuals within compartments. Despite this, the difficulty of specifying large population size stochastic SEIR models in real time is not trivial. In general, Hawkes models seem to be far simpler to implement than SEIR-type models, and in a pandemic such as the spread of Covid-19, where resources can be scarce and policies and health-allocations must be made in real-time, quick and accurate short term forecasts are highly valuable (Worden et al., 2019).

Relative to Hawkes processes, SEIR models are more natural mathematical representations of the spread of contagious diseases. However, in implementation SEIR models often require estimation of more parameters and structural modifications. As discussed above, with increased complexity, there is more opportunity for bias and random errors in parameter estimates, as well as large covariances between pairs of parameter estimates, and in some cases problems of identifiability (Evans et al., 2005; Roosa and Chowell, 2019). Even more pressingly, each component of the model is susceptible to mis-specification, which can result in highly variable estimates and large forecasting errors (Hengartner and Fenimore, 2018; Osthus et al., 2017).

Problems such as these can be particularly severe in the case of Covid-19, where available data used to fit parameters can rely can have substantial errors, due to undercounted infected populations and testing policies that vary over time and space (Kucinskas, 2020). Both Hawkes and SEIR models assume a homogenous population and do not explicitly account for testing errors, but Hawkes and HawkesN models appear to perform better than their SEIR equivalents for modeling the spread of infectious diseases (see Table 3.1 below).

3.4.1 Comparison of Existing Covid-19 Prediction Results

SEIR models appear to be far more widely used by State and Federal agencies for forecasting Covid-19 cases and deaths, with a notable exception being the State of New Jersey which is primarily using a multivariate Hawkes model (New Jersey COVID-19 Information Hub, 2020). Table 3.1 summarizes results comparing the accuracy of Hawkes models and their variants with SEIR models and their variants for forecasting infectious diseases. Point process models have been found to forecast incidence of mumps in Pennsylvania better than compartmental SVEILR models (Kaplan et al., 2020). Further, point process models have been found to improve fit and forecasting performance relative to SEIR models when applied to incidence of pertussis in (Yang, 2019). Yuan et al. find substantially improved accuracy of Hawkes models over SEIR models for forecasting Covid-19 in the European Union, California, New York, and for the United States as a whole (Yuan, 2020).

Hawkes models are directly compared to SIR and SEIR models to explain the spread of Covid-19 in California, Indiana, and New York in (Bertozzi et al., 2020). The Akaike information criterion (AIC) is used to evaluate the candidate models, and by this metric, HawkesN performs more poorly relative to its compartmental counterparts for Covid-19 death data, and with mixed results for Covid-19 case data. However, fitted parameters are found to vary materially across locations, and relative fit of parameters across models is concluded to not be strongly indicated. Rather than concluding on the merits of either type of model, the authors note the difficulty of using limited data at the beginning of an epidemic such as that of Covid-19 (Bertozzi et al., 2020).

In the context of the Covid-19 pandemic, compartmental models such as SIR and SEIR have been noted to generally have low accuracy for long-term forecasts, and machine learning models have been proposed as a superior alternative (Ardabili et al., 2020). Compartmental models also may be poorly calibrated for forecasting more than five days out: forecast numbers of Covid-19 cases in Italy six days in the future based on the SEIR model were 14%

Data	Better Fit	Worse Fit	Reduction RSME	Authors
Pertusis in NV	Recursive Hawkes	SEIR	19%	(Yang, 2019)
Mumps in PA	Recursive Hawkes	SVEILR	38%	(Kaplan et al., 2020)
	Hawkes	SVEILR	26%	
Covid-19 in CA	SEIR	Hawkes	(∗)	(Bertozzi et al., 2020)
Covid-19 in IN				
Covid-19 in NY	Hawkes	SEIR		
Covid-19 in CA	Hawkes	SEIR	63%	(Yuan, 2020)
Covid-19 in NY			21%	
Covid-19 in US			31%	
Covid-19 in EU			27%	
Covid-19 in US	Hawkes Variants	SEIR	(∗∗)	(Chiang et al., 2020)
Ebola in W. Africa	Hawkes	SEIR	38%	(Park et al., 2020)

Table 3.1: Prior results comparing the forecasting accuracy of point process and compartmental models for infectious diseases. Errors reported are the root mean squared error (RMSE) and (∗∗) mean absolute error of daily forecasts. Model selection using (∗) Akaike Information Criterion (AIC) and (∗∗) Normalized Discounted Cumulative Gain.

too low on average (Ardabili et al., 2020). Various compartmental models for forecasting Covid-19 yield slightly different projections of future cases or future deaths (Ryan Best, 2020). However, estimates of variability vary widely, with prediction interval widths often varying by a factor of 3 (Bertsimas, 2020; for Health Metrics and , IHME).

Some variation is to be expected in both mean predicted deaths and size of prediction interval between the models as each are designed differently, and with varying assumptions. Estimates of the initial reproduction number R_t for COVID-19 vary around 3.28 (1.4, 6.5)

(Pan et al., 2020). Of course, values of R_t are observed to vary substantially depending on social distance policies. In China, estimates of R_t decreased from 2 to 1 when public health measures were put in place (You et al., 2020). Estimates of R_t in Singapore correspondingly decreased over time by between 78.2% and 99.3% (Jewell et al., 2020). Similar results were observed in Europe as a result of public health measures (Flaxman et al., 2020).

SEIR forecasts of future confirmed cases or deaths depend critically on estimates of the total numbers of asymptomatic or mildly symptomatic cases, which are highly uncertain (Sood et al., 2020; Bendavid et al., 2020) and extremely difficult to estimate accurately (Lumley, 2020; Srinivasan, 2020). Jewell et al. (Jewell et al., 2020) note that more detailed and complex models may be more sensitive to assumptions regarding the incubation and infectious periods and other estimates of transmission characteristics. Further, SIR and SEIR models are highly sensitive to assumptions regarding social movement and the estimated impacts of containment policies (Pinter et al., 2020). SIR and SEIR models are known to be particularly sensitive to assumptions about the distribution of latent and infectious periods (Lloyd, 2001; Wearing et al., 2005). Further, as discussed above, nonidentifiable parameters can be an issue for compartmental models, and methods for dealing with nonidentifiability of parameters tend to work better for simpler models than for more complex compartmental models (Roosa and Chowell, 2019).

3.5 Further Connections Between Hawkes and SEIR Models

The productivity constant K in the Hawkes model is the obvious analogue of the reproduction rate R_0 in SEIR, with both interpretable as the expected number of direct transmissions per infected individual. Further, several variations of the Hawkes process in Equation 4.1 have deeper connections to SEIR-type compartmental models. The point process governed by Equation 3.2 is a continuous time analog of a discrete stochastic SIR model when $g(t)$ is specified as exponential (Rizoiu et al., 2018). When $g(t)$ is chosen to be gamma distributed,

the Hawkes process also can approximate staged compartment models, like SEIR, if the average waiting time in each compartment is equal (Lloyd, 2001). More complex parametric (or non-parametric) inter-infection time distributions $g(t)$ may be employed within the Hawkes process framework in situations where disease dynamics cannot be captured by a SIR or SEIR model. In the early exponential growth stage of an epidemic, before finite population and social distancing effects play a role, the linear Hawkes process in Equation 4.1 can readily be used to model new infections (see Figure 3.3).

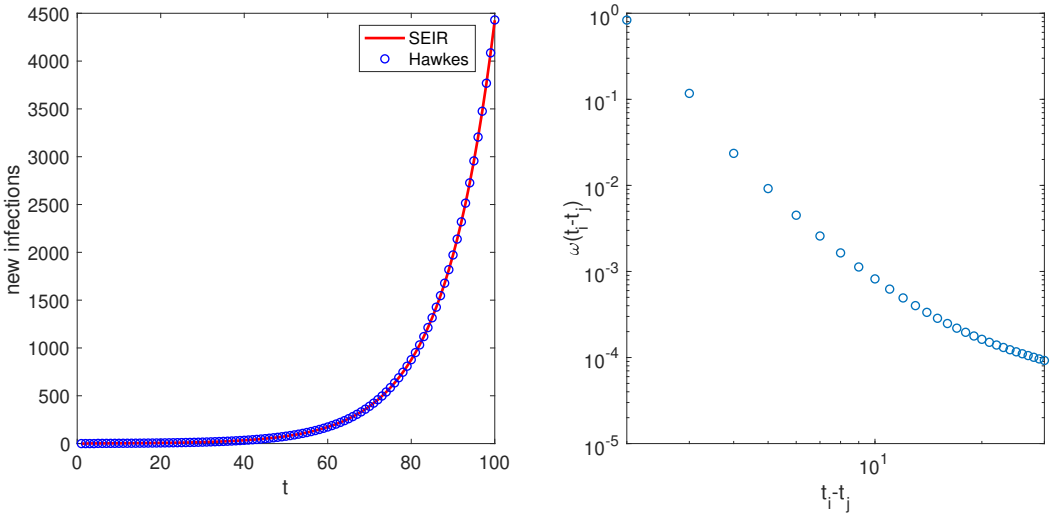


Figure 3.3: Left: (Red) SEIR differential equation $dS/dt = -\beta SI/N$, $dE/dt = \beta SI/N - \mu E$, $dI/dt = \mu E - \gamma I$, $dR/dt = \gamma I$, where $\beta = \gamma R_0$, $\gamma = .1$, $K = 2$, $\mu = 1$, and $N = 5 \cdot 10^8$. (Blue) linear Hawkes process $\lambda_t = \mu + \sum_{t>t_i} Kg(t-t_i)$ fit to the SEIR curve of new infections using non-parametric expectation-maximization (Mohler et al., 2020). Right: Non-parametric histogram estimate for $g(t)$ corresponding to the Hawkes process fit.

While the Hawkes process can approximate SEIR in some situations with an appropriately chosen kernel $g(t)$, queue-Hawkes processes (Daw and Pender, 2018) can also be used to model an exposed latent class of events. Let N be population size, N_t^E be the cumulative sum of infections (whether recovered or not) up to time t. Then we may define a hybrid model incorporating features of both SEIR and Hawkes, which we call a SEIR-Hawkes process,

where the intensity of newly exposed cases is given by

$$\lambda^E(t) = \left(1 - \frac{N_t^E}{N}\right) \sum_{t > t_j^I} K\gamma \exp\left(- \gamma(t - t_j^I)\right), \tag{3.3}$$

and the times of infection are generated via

$$P(t_j^I > t_j^E + c) = \int_c^\infty \mu \exp\left(- \mu(s - t_j^E)\right) ds. \tag{3.4}$$

Realizations of the SEIR-Hawkes process can be generated via Lewis' thinning method for simulation (Ogata, 1981; Lewis and Shedler, 1979). We first simulate an upper-bounding Hawkes process with intensity

$$\nu^E(t) = \sum_{t > s_j^I} K\gamma \exp\left(- \gamma(t - s_j^I)\right). \tag{3.5}$$

$$P(s_j^I > s_j^E + c) = \int_c^\infty \mu \exp\left(- \mu(s - s_j^E)\right) ds. \tag{3.6}$$

Because the Hawkes process in Equation 3.5 has a branching process representation (Hawkes and Oakes, 1974), the process can be simulated iteratively; for each event pair (s_j^I, s_j^E), by

1. Generating a Poisson random variable M with mean K .
2. Generating $l = 1, ..., M$ events with inter-event times $s_l^E - s_j^I$ given by an exponential random variable with parameter γ.
3. Generating $l = 1, ..., M$ events with inter-event times $s_l^I - s_l^E$ given by an exponential random variable with parameter μ.

Thinning then proceeds sequentially by accepting each event pair (s_j^I, s_j^E) with probability $\lambda^E(s_j^E)/\nu^E(s_j^E)$ where λ^E is computed using only accepted events in the history and ν^E is computed using all simulated events. In Figure 3.4 we simulate the SEIR-Hawkes process with parameters $\mu = 1$, $\gamma = .1$, $K = 2$, $N = 1000$ and $N_0^E = 10$ ($t_1^E = ...t_{10}^E = 0$) and compare to the forward-Euler approximate solution ($dt = .01$) of a SEIR differential equation $dS/dt = -\beta SI/N$, $dE/dt = \beta SI/N - \mu E$, $dI/dt = \mu E - \gamma I$, $dR/dt = \gamma I$, where $\beta = \gamma K$.

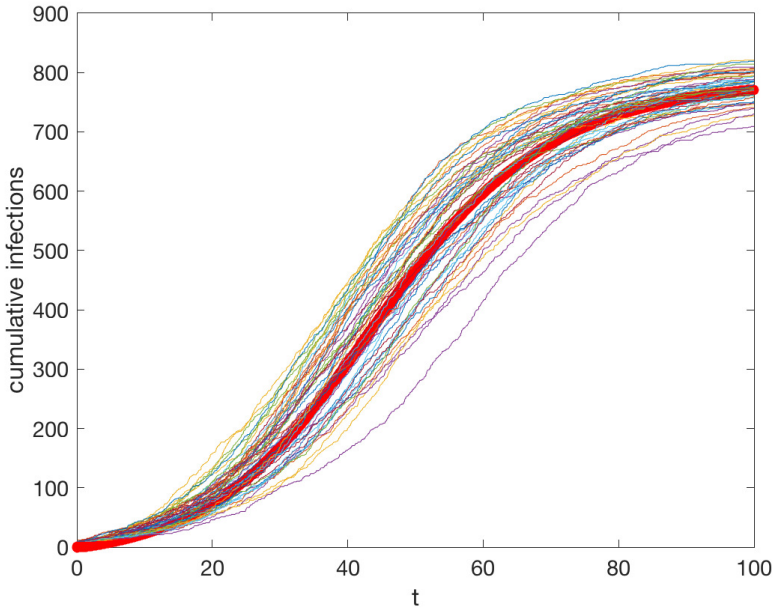


Figure 3.4: SEIR differential equation simulation (red) and 50 realizations of the SEIR-Hawkes process. Parameters for the SEIR model are $K = 2$, $\mu = 1$ for the $E \to I$ rate, $\gamma = .1$ for the $I \to R$ rate, and population size $N = 1000$.

3.6 Conclusion

The SEIR model is currently far more widely used to model epidemic diseases such as Covid-19 than the Hawkes model, and its parameterization is physically plausible, with parameters that are readily interpreted in the epidemiological community. The SEIR model also appears to forecast epidemics adequately in most cases, especially in the early spread of the disease. However, the Hawkes model seems to offer more accurate forecasts for case data, with approximately 20-30% smaller errors on average. Among the several reasons listed in Section 3.4 for this discrepancy, the most significant seem to be mis-specification in the SEIR model and its sensitivity to errors in estimates of latent quantities such as the number of asymptomatic individuals and the distribution of incubation times. In general, when maximal accuracy is desired, models for forecasting observations should typically be only as complex as necessary to represent the main features of interest in the data, with

minimal dependence on unobserved or noisy data (Kelly et al., 2019).

There are close connections between SEIR and Hawkes models, and indeed the two types of models can be constructed to be equivalent or to converge to one another in special cases. The SEIR-Hawkes model described here may provide further linkage between the two paradigms in cases where one seeks the accuracy of point process modeling without sacrificing the physical plausibility and interpretation of SEIR parameters, and the model is shown here to emulate characteristics of SEIR models closely.

Further, doubling time of cumulative cases is of interest to epidemiologists as a statistic for monitoring disease spread during a pandemic. For commonly used compartmental models such as SIR (susceptible, infectious, removed), analytical solutions for doubling time exist as a function of the underlying parameters. In particular, such a representation is dependent on the reproduction number R_0 which can be analogized to K for a Hawkes-type model (Chiang et al., 2020). Hawkes-type models are not stationary when $K \geq 1$, and therefore, doubling time for HawkesN models is of interest. See Figure 3.5.

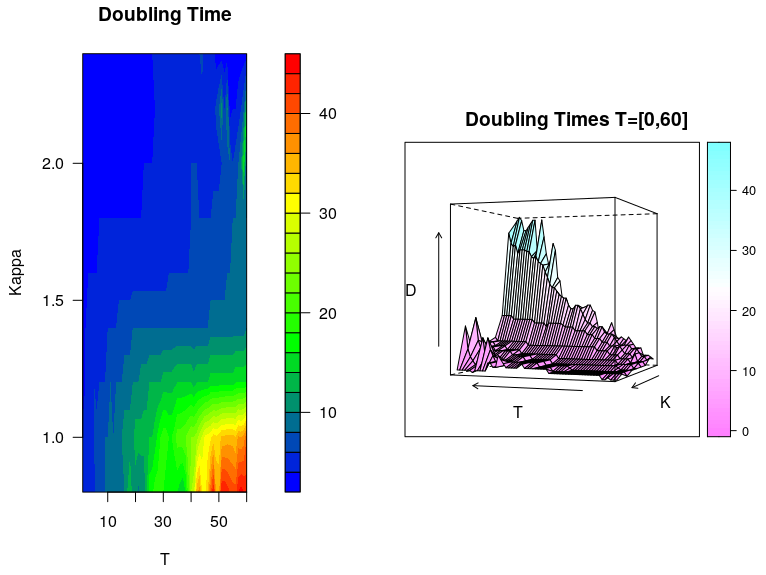


Figure 3.5: Doubling time of cumulative events as a function of K and t, for a HawkesN model with population of $5 \cdot 10^5$ and an exponential kernel with $\theta = 0.25$.

3.7 Acknowledgements

This paper owes a debt of gratitude to Frederic Schoenberg and George Mohler, who provided valuable guidance regarding Hawkes models and Covid-19 research in the first months of the pandemic. The contents of the chapter are largely contained within the paper *Comparison of Hawkes and SEIR models for the spread of COVID-19* by Conor Kresin, Frederic Schoenberg, and George Mohler, Advances and Applications in Statistics (2020).

CHAPTER 4

A Potential Outcomes Framework for Point Process Data

4.1 Introduction

A cohesive potential outcome framework for point process data is missing from the current literature. Recent attempts to use point process data (wherein points fall in continuous as opposed to discretized space) and point process models make highly restrictive modelling assumptions and ultimately rely on discretization of data, for instance, see Papadogeorgou et al. (2020). In this chapter, we present a novel framework and accompanying simulation study for general point process models in a potential outcomes framework.

4.2 A Preliminary Framework and Notation

We propose a causal interpretation for spatio-temporal point process data observed on window $\mathcal{X} = A \times B \times [0, T]$ for $A \times B \subseteq \mathbb{R}^2$ and $T \in \mathbb{R}^+$. We assume a single constant treatment $\tau \in \mathbb{R}$ was implemented at some time $t \in [0, T]$ in some subset of $A \times B$. We partition $\mathcal{X}$ into cells $\mathcal{I}_1, \ldots, \mathcal{I}_p$ such that $\cup_{i=1}^{p} \mathcal{I}_i = \mathcal{X}$ and $\mathcal{I}_j \cap \mathcal{I}_k = \emptyset \ \forall j \neq k$ and assign treatment to the cells indexed by some possibly nonrandom subset of the indices $1, \ldots, p$.

An observed data set ϕ is a realization of point process Φ. We represent Φ as the superposition of two processes Φ_0 and Φ_1, *i.e.* $\Phi = \Phi_0 \cup \Phi_1$. Φ_0 and Φ_1 represent the "control" and "treatment" processes, which do not share identical state spaces (although

the union of their state spaces is $\mathcal{X}$). We only observe Φ_0 on "control cells" and we only observe Φ_1 on "treatment cells." Therefore ϕ is an "incomplete" observation: we only observe the partial treatment and partial control process (similar to so-called fundamental problem of causal inference). In determining a treatment effect, we must therefore assume (or prove) that we have enough information about Φ_0 and Φ_1 to simulate it meaningful across $\mathcal{X}$ in the cells where each respective process was not observed.

4.3 Visualization

We now present a series of visualizations meant to clarify the above framework. For ease of visualization, we choose purely spatial processes, and for maximal simplicity, we defined both the control and treatment processes as homogenous Poisson. Point processes with more complicated intensities are discussed in Sections 4.4.

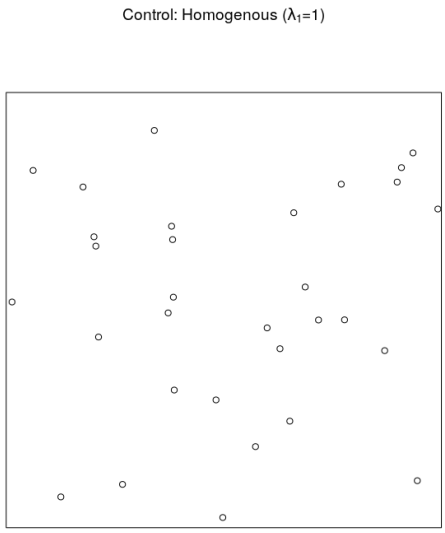


Figure 4.1: The simulated control process, a homogeneous Poisson process specified with intensity $\lambda = 1$.

Figure 4.1 shows the true unobserved and observed control process. Note that in practice, for a given data set ϕ, we would only have the observed control process. Similarly, Figure 4.2 shows the true unobserved and observed treatment process.

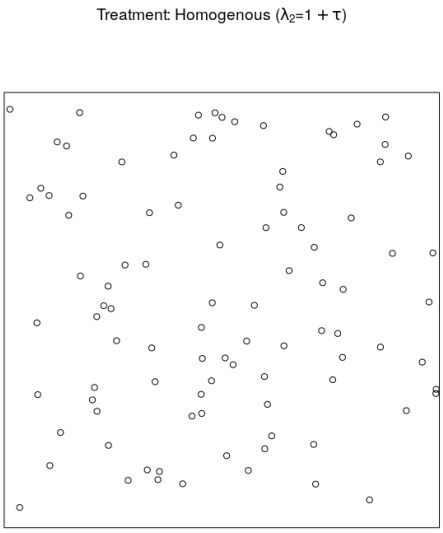


Figure 4.2: The simulated treatment process, a homogeneous Poisson process with intensity $\lambda = 4$ (τ, the treatment effect, is 3).

If we superimpose the treatment process on the control process, we can see the complete (observed and unobserved) treatment and control processes, see Figure 4.3. Note that this is not the same as ϕ.

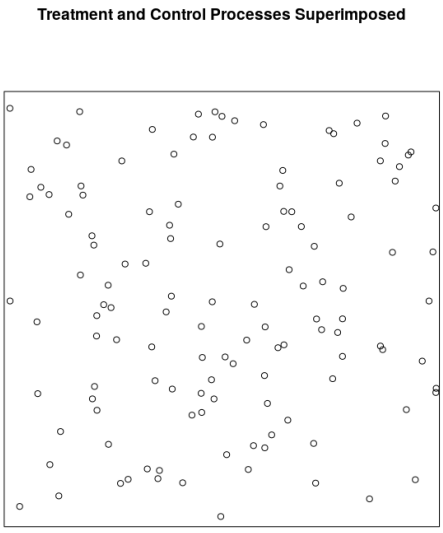


Figure 4.3: This represents what we would see if we could simultaneously observe both treatment and control.

The actual process observed, ϕ is a realization of Φ, which is dependent on how we partition our state space $\mathcal{X}$. In this example, we have arbitrarily chosen $\mathcal{X} = [0, 5] \times [0, 5]$. We can partition $\mathcal{X}$ using any scheme so long as the partitions are non overlapping and the union of all partitions is equal to $\mathcal{X}$. Figure 4.4 shows two options:

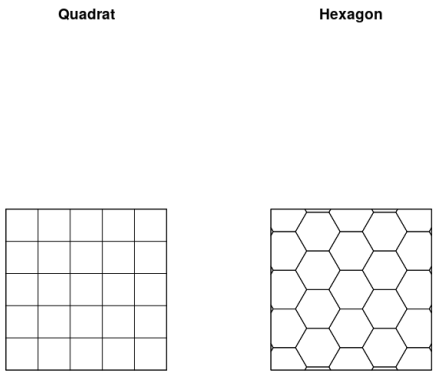


Figure 4.4: Options for partitioning $\mathcal{X}$.

We also have the option of data-dependent options, such as the Voronoi tessellation (see Figure 4.5) based on the control process seen in Figure 4.1. Note that in this case, a Voronoi

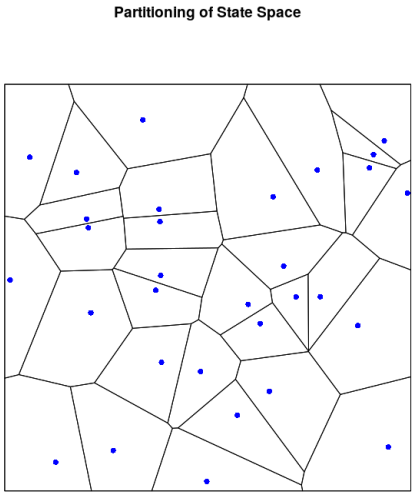


Figure 4.5: Voronoi tessellation of $\mathcal{X}$

tessellation based on entire control process (both observed and unobserved) is artificial; in

actual data sets, only the observed portion of the control process will be present. This illuminates the kind of *ouroboros* paradox inherent to our framework in the purely spatial context (spatio-temporal data escapes this dilemma – we can choose data-dependent cells on all observed data prior to treatment effect being applied). Optimal partitioning of $\mathcal{X}$

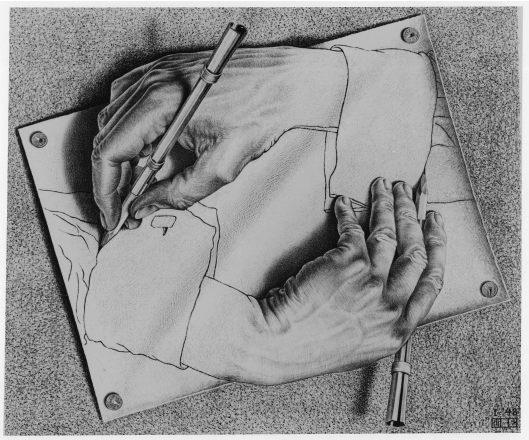


Figure 4.6: M.C. Escher's "Drawing Hands." Image credit: BYU Museum of Art.

to ensure accurate estimation of treatment effect should likely be data-dependent (with the exception of the homogenous Poisson case). Treatment assignment therefore dictates *where* we observe each process (*i.e.* in which cells we observe control and treatment, respectively). Such a scheme may make sense for some point process data: for instance if crimes are represented as marked spatio-temporal points, and some crimes are thought to trigger future crimes (perhaps modelled by a Hawkes process), then it would make sense to partition based on triggering crimes. Note that typically, points are not labelled as a background or triggered point in the point process literature, but real life data sets often do include such labels.

For this example, we continue with the Voronoi tessellation of $\mathcal{X}$ and randomly assign treatment status for each cell using Bernoulli($p = \frac{1}{2}$) assignment. Note that given sufficient data, p can approach 0 or 1, and we can still accurately estimate τ. Further note that non-randomized treatment schemes are well documented in the potential outcomes literature (Lee, 2008; Rubin, 2005).

We then keep the subset of points from the control process that fall into the control cells,

and the subset of points from the treatment process that fall into the treatment cells. The union of these two subsets is equal to ϕ, *i.e.* the observed data, see Figure 4.7 (top-left).

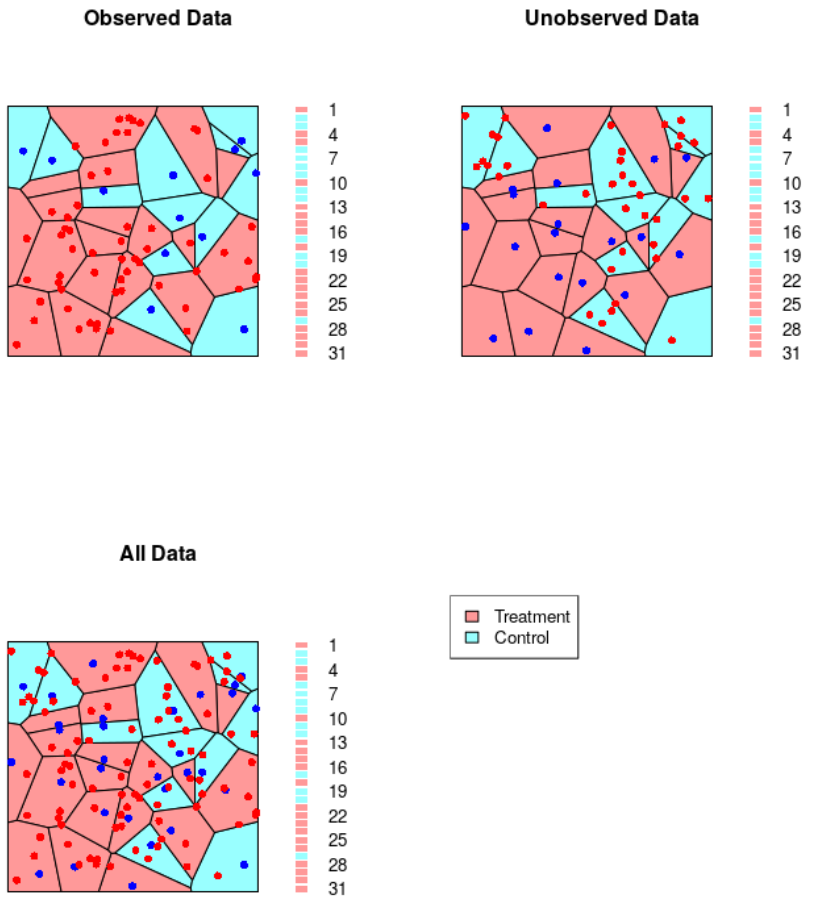


Figure 4.7: **Top left**: a single realization of ϕ (the observed data from both the control and treatment processes. **Top right**: unobserved data (these points are effectively "dropped." **Bottom left**: observed and synthetic data, superimposed.

We then need to simulate the unobserved portions of the control and treatment processes. We can do this by assuming a parametric model for each process and fitting via MLE, using the data for each corresponding process in the cells it is observed in. For instance if $\mathcal{X}$ is

partitioned into 10 cells, and the control process is observed in cells with $\{1, 2, 7, 9\}$, then using the data in cells $\{1, 2, 7, 9\}$, we can synthetically generate the control process we would expect to observe in cells $\{3, 4, 5, 6, 8, 10\}$. Note that such MLE estimates are consistent and asymptotically efficient, given correct model specification (Ogata, 1978). These simulated processes are referred to as the synthetic treatment and control. Finally, we present the full data set which we will use to estimate the treatment effect τ. The "full data" seen in

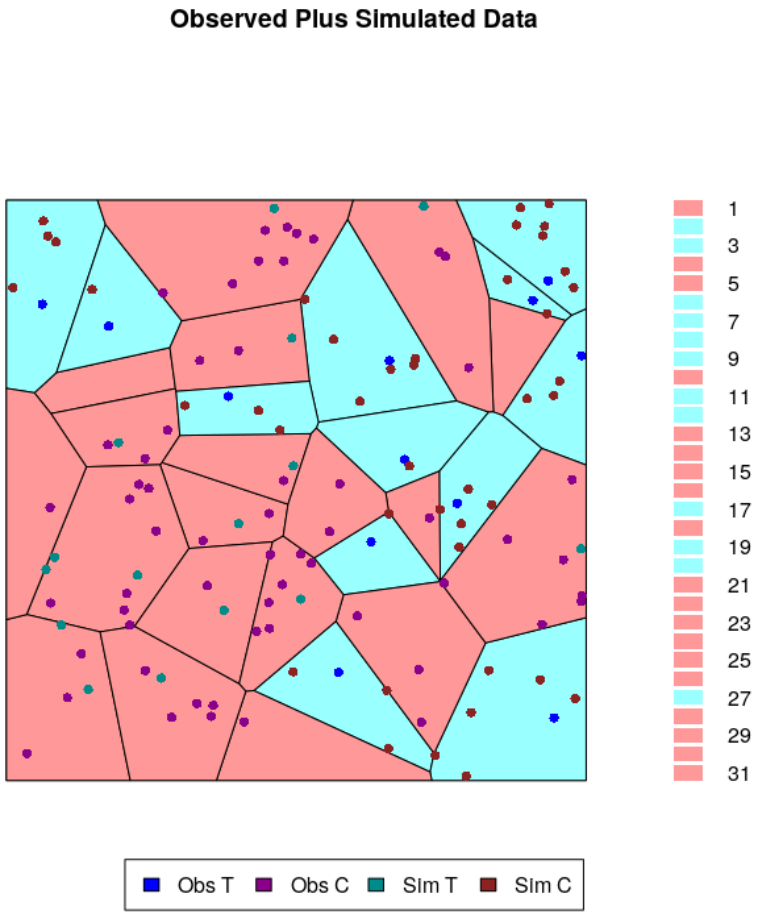


Figure 4.8: Synthetic treatment and control processes, it using MLE estimates of homogenous poisson process intensities, superimposed upon ϕ, the observed process.

Figure 4.8 is the union of observed treatment and control data with simulated treatment data (where control was assigned) and simulated control data (where treatment was assigned). We notate this full data (observed superimposed by synthetic) as $\phi \cup \hat{\phi} = \phi^*$.

4.3.1 Estimation of Treatment Effect

Given a partitioning $\mathcal{I}_1, \ldots, \mathcal{I}_p$ of $\mathcal{X}$, we can calculate the treatment effect τ as the average difference in the counting measures of partitions for observed and synthetic data. Potential outcomes are simply defined as the counting measure of a given partition.

We present the spatio-temporal case formulas, noting that if the data is purely spatial as presented in the previous section, the second time dependent integral disappears. Assuming that treatment is assigned once at a known time, we characterize the potential outcomes (after treatment at time t^*) as

$$Y_{\mathcal{I}_j, t^*} = N\left(\mathcal{I}_j \cap (t^*, T]\right) \approx \int_{\mathcal{I}_j} \int_{t^*}^{T} dN$$

which is to say the counting measure of some cell $\mathcal{I}_j$ after treatment. We then estimate the average treatment effect as the different counting measure for treated cells and synthetic control equivalent, or non treated cells and synthetic treated equivalent.

Given treatment at time t^* for a process observed up to time T, we estimate the average treatment effect τ as

$$\hat{\tau} = \frac{1}{p} \sum_{j=1}^{p} \left| N(\mathcal{I}_j \cap (t^*, T], \mathcal{I}_j \equiv \mathrm{C}) - N(\mathcal{I}_j \cap (t^*, T], \mathcal{I}_j \equiv \mathrm{T}) \right|$$

Note that C denotes "control" and T denotes "treatment." Either the first or second quantity can be easily estimated in expectation to fill in as a synthetic control: for example, if treatment occurred in cell $\mathcal{I}_j$, we can estimate the synthetic control at time l as

$$N(\mathcal{I}_j, t^*) + \int_{\mathcal{I}_j \text{ s.t. } t^* < t \leq l} \lambda_{\mathrm{C}} d\mu$$

where μ represents the Lebesgue measure and λ_{C} represents the conditional intensity function4 of the control process.

Of further note, a "spatially scaled" version of the same measure could be of some interest:

$$\hat{\tau}' = \frac{1}{k}\sum_{p=1}^{k}\frac{1}{\mu(\mathcal{I}_j \cap (t^*, T])}\Big|N(\mathcal{I}_j \cap (t^*, T], \mathcal{I}_j \equiv \mathrm{C}) - N(\mathcal{I}_j \cap (t^*, T], \mathcal{I}_j \equiv \mathrm{T})\Big|$$

Obviously, such a scalar is not necessary in the case that all cells are of equal size. This spatially scaled version is "unitized," *i.e.* it represents a unit treatment effect. For instance, if the treatment effect $\tau = 3$, then we would expect 3 more points per unit of space time in treatment partitions.

4.3.2 Interpretation

Before interpreting the above example, we note that the above framework does not allow us to tackle two related problems: The first is the problem of distinguishing inhomogeneity from clustering. This first problem is closely related to the problem of model selection within point process theory. The second is determining the "causal effect" of a single point in a given realization, as discussed in (Papadogeorgou et al., 2020). A counterfactual defining this second problem could be, "if there was a point here, given this realization, how we we expect the other points to be different." In such a framework, points themselves "cause" other points, or at least effect the location of other points.

We instead focus here on the relatively simpler problem of estimating a treatment effect using point process models and a spatially (or spatio-temporally) defined treatment mechanism. From the framework we have described above, we can make a conclusion about how the average difference of the counting measure for control and non control processes changed across partitions, given some treatment effect specified across a subset of the partitions. A counterfactual defining this problem could be "given this realization, if we were to change the partitioning of the underlying state space, how would the number of points in a given partition increase/decrease."

A major limitation of this interpretation is that point processes are typically assumed to occur in continuous space, and therefore there are infinite potential outcomes without some

level of aggregation. The remedy applied here is partitioning the underlying state space, but as mentioned above, the level and method of aggregation (*i.e.* data dependent partitioning) is not straight forward.

4.4 Simulation Study

We now present two (somewhat arbitrary) simulation studies. The `R` code for below simulated results can be found in Section 7.1.1.

4.4.1 Inhomogeneous Poisson

For each iteration, we simulate a realization ϕ of process Φ on state space $\mathcal{X} = [0, 10] \times [0, 10]$ as an inhomogenous Poisson characterized by control intensity $\lambda_{\text{C}} = 1.50x + 0.55y + 2$ and $\lambda_{\text{T}} = 1.50x + 0.55y + 2 + \tau$ where $\tau = 5$ is the prescribed constant treatment effect. Note that this parametric form was chosen arbitrarily, as were coefficients.

After simulating both processes, we tessellate $\mathcal{X}$ into a grid of quadrats (partitioning does not change across iterations), and randomly assign treatment to cells using Bernoulli assignment with p randomly drawn from a discrete uniform distribution $\text{Unif}(\{0.1, 0.2, \ldots., 0.8, 0.9\})$. We then created synthetic data for non observed cells via MLE (after dropping the unobserved subset of ϕ), and estimated τ as described above, seen in red in Figure 4.9. As a benchmark for how well we fit synthetic data, we also calculated the treatment effect based on ϕ (which includes both observed and unobserved data), seen in green in Figure 4.9.

The error shown in Figure 4.10 represents the distribution of the differences in estimating τ based on using fitted synthetic data (SATE) versus unobserved data superimposed on observed data (*i.e.* ϕ, which is never available in real life).

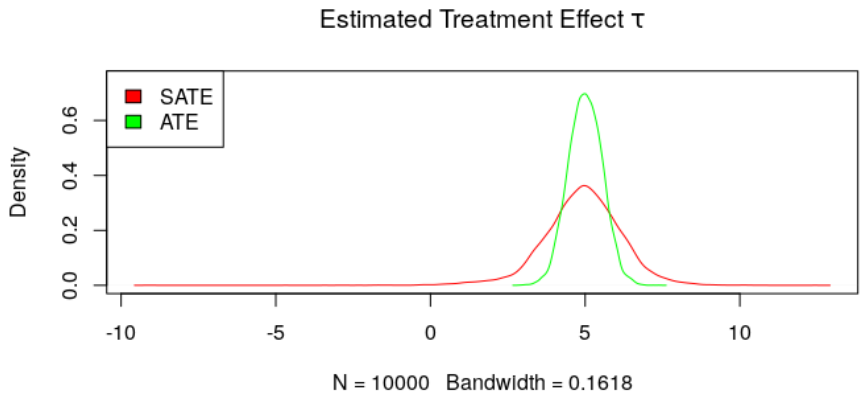


Figure 4.9: True constant treatment effect $\tau = 5$. **Red:** Treatment effect estimated using observed and fitted synthetic data, $\hat{\tau}_{\text{SATE}} = 4.911187$. **Green:** Treatment effect using observed and unobserved data, $\hat{\tau}_{\text{ATE}} = 5.005975$. All difference between ATE and true treatment effect due to stochastic variation.

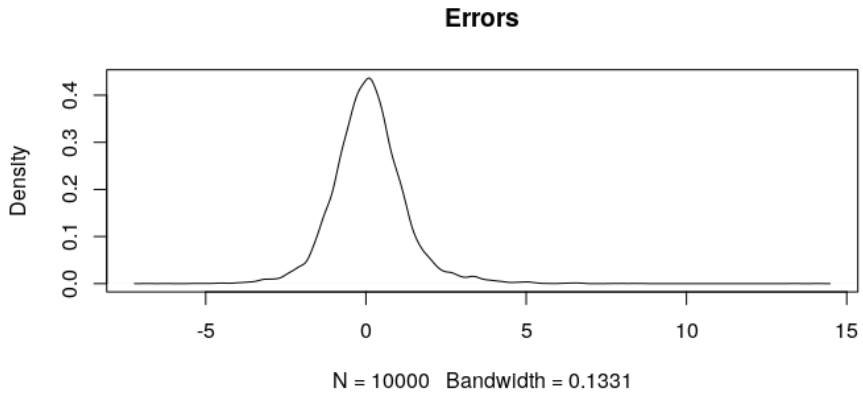


Figure 4.10: Difference between SATE and ATE estimates of τ.

4.4.2 Hawkes Process

We proceed with a simulation study of two Hawkes processes. A univariate Hawkes process is characterized by the following intensity:

$$\lambda(t|\mathcal{H}_{t-}) = \underbrace{\mu}_{\text{background rate}} + \kappa \int_{t'<t} \underbrace{g(t-t')}_{\text{triggering density}} dN(t') \tag{4.1}$$

$$= \mu + \kappa \sum_{(t'):t'<t} g(t-t'),$$

where $\lambda(t|\mathcal{H}_{t-})$ is the conditional rate at which points (events) are expected to accumulate around spatial-temporal location t, given information on all previous events $\mathcal{H}_{t-}$.

We chose an exponential kernel function for g, with control process parameters ($\mu = 1, \alpha = 3, \beta = 6$) and treatment process ($\mu = 1, \alpha = 4, \beta = 6$), *i.e.* $\alpha_{\mathrm{C}} + 1 = \alpha_{\mathrm{T}}$. Note that changing α in this way is akin to decreasing κ. Further note that all or any parameters may be changed and consistent results can still be obtained via simulation. We chose state space $\mathcal{X} = [0, 5 \cdot 10^1]$.

We then calculated the true treatment effect based on our parameters and state space. Recall from Section 2.1 that we can find our true treatment effect by taking the difference of the integrals of our treatment and control intensities with a normalization factor.

$$\begin{aligned}
\tau^{`} =& \frac{1}{|\mathcal{X}|} \left(\mathbb{E}[N(\mathcal{X})_{\mathrm{C}}] - \mathbb{E}[N(\mathcal{X})_{\mathrm{T}}] \right) \\
=& \frac{1}{|\mathcal{X}|} \left(\int_{\mathcal{X}} \lambda_{\mathrm{C}} d\mu - \int_{\mathcal{X}} \lambda_{\mathrm{T}} d\mu \right) \\
\approx& \frac{1}{|\mathcal{X}|} \left(\frac{\mu_{\mathrm{C}} \cdot |\mathcal{X}|}{1 - \frac{\alpha_{\mathrm{C}}}{\beta_{\mathrm{C}}}} - \frac{\mu_{\mathrm{T}} \cdot |\mathcal{X}|}{1 - \frac{\alpha_{\mathrm{T}}}{\beta_{\mathrm{T}}}} \right) \\
=& 0.99
\end{aligned}$$

This value is visualized as a dotted blue line in Figure 4.11. Note that we use the integral approximation detailed in (Schoenberg, 2013).

We then proceed to tesselate $\mathcal{X}$ using `kmeans` on a random sample of points where the population is the superimposed simulated control and treatment processes. Note that this population contains both observed and unobserved data, and therefore this data-dependent tessellation is somewhat artificial. Simple intervals on $\mathcal{X}$ also provided consistent results. We then randomly assigned treatment as in Section 3.1, and created synthetic data for non observed cells via MLE. We then estimated our treatment effect for 996 observations.[1] The

[1] If less than two points were observed in a given cell, we did not perform fitting via MLE and the iteration

results are summarized in Figure 4.11. The accompanying errors are summarized in Figure 4.12.

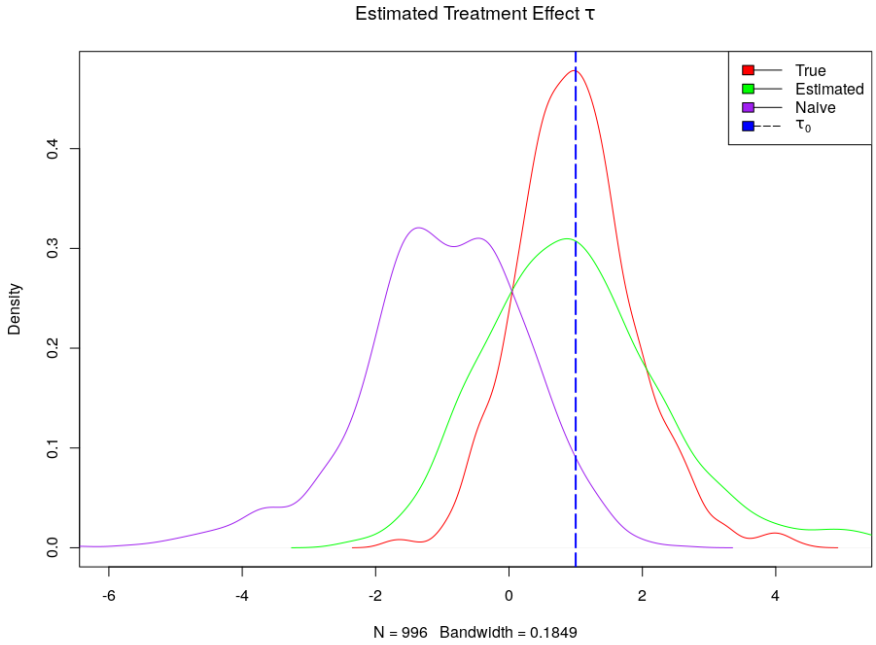


Figure 4.11: The true treatment effect is represented by the blue dotted line. ATE is represented by the red line, and as before, all difference between ATE and the true treatment effect is due to stochastic variation. SATE, what a scientist could measure in real life, is represented by the green line. Note that SATE consistently captures the treatment effect, albeit with higher variance than ATE. The purple line represents a naive approach: count the observed points for each cell, normalize counts for the size of the cells, and take the difference between control and treatment cells. This naive approach significantly under performs relative to SATE, as it fails to capture the "triggering" nature of the underlying data generating Hawkes process.

---

was dropped. This happened four times.

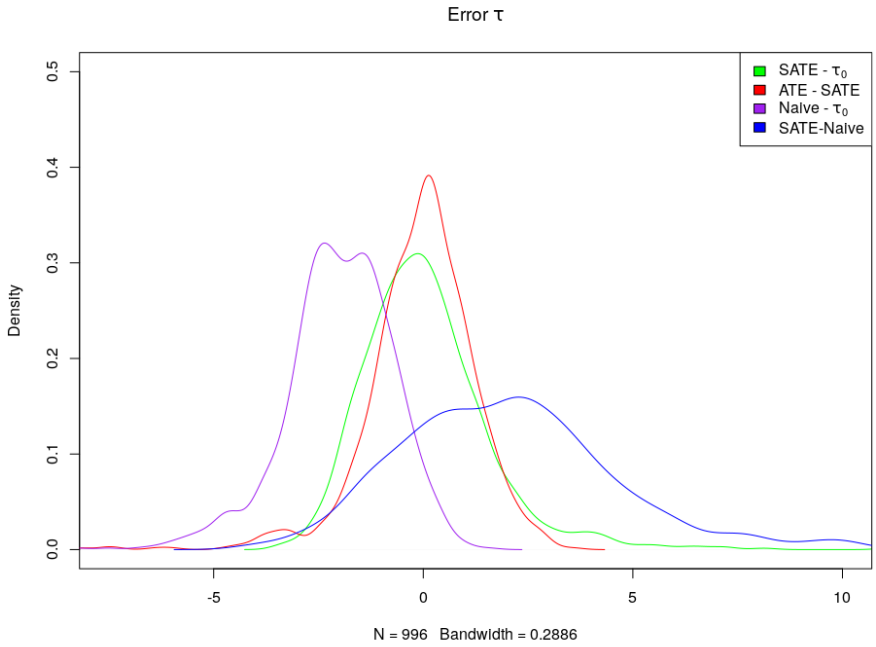


Figure 4.12: Here we see the errors for the various methods described in the above figure.

4.4.2.1 Notes

Intuitively, it seems that for most accurate measurement of treatment effect, we need to tessellate our state space such that as many points as possible that are triggered by a background point fall in to that background point's tile. Two possible remedies are as follows: (1): edge correction on a tile by tile basis, see Chapter 1, Section 4 of (Mecke and Stoyan, 2000) and (2): bound error by the expected number of points to fall outside of a given tile.

In general, it seems that if points do not have relationships with other points outside their tile, this current causal framework should provide accurate and consistent measurement of treatment effect. Essentially, we should strive for "inhomogeneous-Poisson-y-ness" between the tiles, and "anything goes" within the tiles (as long as it can be fit via MLE or some similar method). Therefore without smart data-dependent tessellation, the current method seems at least theoretically better for inhomogeneous Poisson rather than Hawkes or Gibbs

processes where the points have spatial or temporal relationships.

Drawing from the potential outcome literature, there are various quantities of interest worth labelling. As discussed above, a Hawkes process intensity can be decomposed into a background rate and a triggered rate. These two elements can be thought of as "non-interactive" and "interactive" components, analogous to direct and indirect effect, as defined in (Hudgens and Halloran, 2008), respectively. As total effect is equal to the sum of indirect and direct effect, a process such as an inhomogeneous Poisson where points do not interact with each other, implies a total effect equal to the direct effect (*i.e.* a indirect effect equal to 0). Such a direct effect (background rate, in the case of a Hawkes process) can be estimated using linearly fit covariates, as discussed in (Park et al., 2021), which can reduce interference (Hudgens and Halloran, 2008).

4.5 Conclusion and Future Work

In future work, we plan to explore how the contagion effect can be limited by data dependent tessellation of observation region. For instance, in the context of Hawkes models, with sufficiently large cells and an exponential triggering kernel with parameters such that triggered points were close by, contagion effect would be nonexistent. As an application, we plan to use retaliatory gang crime data, building on the results of Mohler et al. (2011). This application will require further theoretical augmentation of the framework laid out in this chapter, as the treatment assignment mechanism underlying this crime data is non-random.

CHAPTER 5

Measuring Complexity of Tensor Representations of Point Process Data

5.1 Point Process Representations of Tensors

Tensors are natural mathematical representations of many important data types ranging across many fields including imaging and computer vision, signal processing, linguistics, geology, and physics. Tensors are arrays of spatial dimension (termed "order" in the tensor calculus literature) $d \geq 3$. Because tensors are often large and computationally expensive objects, they are often unwieldy and require cumbersome operators and notations.

Varying approaches to approximation, decomposition, and summary of tensors exist in the literature. Section 5.3 introduces tensors, tensor notation, and basic properties of tensors and tensor operators, as well as attempts to draw analogies and contrasts between tensor and matrix algebra, before discussing a common tensor factorization. Discussion of tensor factorizations is pertinent because it is a relatively straightforward context to introduce tensor notation and operations, and because factorization is the primary method to obtain a compact representation of a complicated data object.

The properties of the rich theoretical framework developed around point processes has been under-utilized in the context of practical application to real life data science problems due to the fact that such applications commonly involve big data. As discussed in Section 2.4, commonly implemented model fitting methods for point processes such as MLE are highly computationally intensive for many large datasets or for processes with complex or volatile

structures. We propose a new methodology using point processes to represent tensors, and provide a brief literature review of existing work and proposed future work.

5.1.1 Point Process Representation of Tensors

It is our primary interest to increase the utility, decipher-ability, and information extraction of tensors. We focus on relatively simple problem of measuring the amount of information in sparse, boolean-valued tensors, although our proposed method can be generalized to real-valued tensors (as a marked point process equivalent, or where elements are thresholded, creating boolean-valued tensors). A tensor of order d can be thought of as a d-dimensional array (where each element is indexed with d-indices). In the context of a sparse boolean-valued tensor, all elements are equal to 0 or 1, and most are equal to 0. Therefore it is natural to think of elements equal to 1 as points with spatial location equal to their index.

Tensors are an inherently rich data type, and tensor representations of graphs (or graphical representations of tensors), as well as neural networks, and other areas of interest are well-discussed in the literature (Duchenne et al., 2011; Novikov et al., 2015). Why represent tensors as a point process? Doing so is a natural bridge away from often incomprehensible data to known parametric forms, and crucially, point processes retain the spatial relationships between non-zero elements. It should be noted that point process representations of tensors lose their value when $d \leq 2$ or for tensors where the total number of elements is not large. When $d \leq 2$, we can leverage the properties of matrices and vectors, and the computational burden is less and interpretability is greater. Point process representations of high-dimensional tensors has been recently explored in the context of taxis (Pang et al., 2017), workforce management (Shen et al., 2008), and neuro-imaging (Tagliazucchi et al., 2016). Theoretical properties of point process representations of tensors is recently discussed in Xu et al. (2018); Zhe and Du (2018).

We restrict ourselves to the case of sparse boolean-valued tensors of order $d \geq 3$ where the number of elements is large. We further restrict our problem of interest to determining

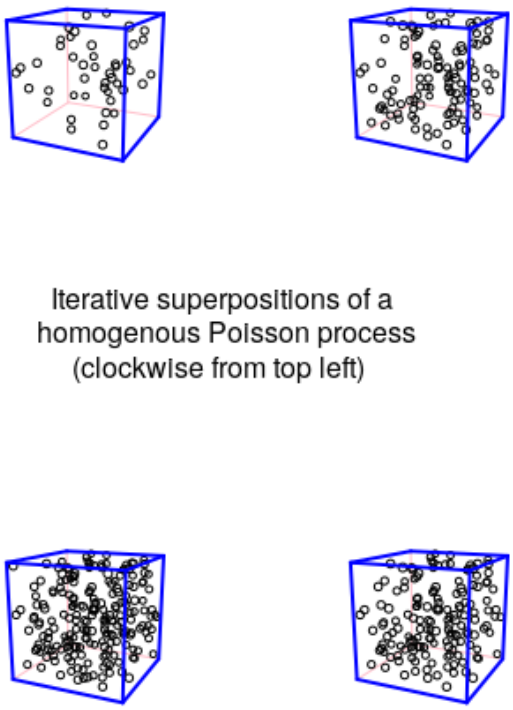


Figure 5.1: As we iteratively superpose homogenous poisson processes with a fixed intensity λ, we come closer to filling the state space with a uniform random distribution, i.e. a process with maximum entropy.

the complexity, or information content, of such tensors. To do so, we employ the following iterative process:

1. Start with sparse boolean-valued tensor $\mathcal{T}$ of order d. Approximate $\mathcal{T}$ as a realization of a simple point process Φ_1 with unknown intensity λ and state space $\mathcal{X} \subset \mathbb{R}^d$.

2. Fit a homogenous Poisson process to Φ_1 (*i.e.* fit $\hat{\lambda} \in \mathbb{R}^+$).

3. Superimpose a realization of homogenous Poisson process with intensity $\hat{\lambda}$ on Φ_1, and call the resulting process Φ_2.

4. Continue superimposing realizations of the homogeneous Poisson process, generating $\Phi_3, \ldots, \Phi_s$ until the divergence is less than some tolerance level ϵ, i.e. until $\mathfrak{D}(\Phi_s||\Psi) < \epsilon$ where Ψ is a homogenous Poisson process with intensity $(s+1)\hat{\lambda}$.

Although fitting an inhomogenous Poisson process in step 1 is attractive, it is likely computationally expensive, and more importantly, reliant on qualitative assumptions. Hypothetically, if we could parameterize an a point process process representative of $\mathcal{X}$ we could skip to step 4 and quickly find the the tensor information content. However, steps 2 and 3 are not computationally expensive, or difficult to implement, and therefore allow us to bypass an qualitative assumption about the underlying structure of $\mathcal{X}$. We propose that this workaround can be intuited as *paradoxical information loss*: to measure the content of datum without making parametric assumptions, we can back-solve by measuring how much information we can lose through entropy-increasing operations like superposition. We note that the point process representing $\mathcal{X}$ must be interpreted as a discretized approximation (due to the fact the elements of $\mathcal{X}$ have indices in $\mathbb{N}$).

5.1.2 Literature Review

"In a loose sense, each of the operations of superposition, thinning, and random translation is entropy increasing; it is not surprising then that among point processes with fixed mean rate, the Poisson process has maximum entropy..." (Daley and Vere-Jones, 2008)

The primary tools for this project rely upon the intersection of point process and information theoretic results. Literature discussing the entropy of point processes is relatively limited: first McFadden characterized point process entropy (McFadden, 1965), then Papangelou gave results for the approximation of the discrete entropy of stationary point processes (Papangelou, 1978).

Dayley and Vere Jones provided a chapter of their seminal point process theory text concerning the properties of point process entropies (Daley and Vere-Jones, 2008, Chapter 14.8). This chapter primarily busies itself the question of approximating the expected entropy of a finite simple point process from finite samples. Convergence in the L_1 norm between expected and true entropies is established based on the convergence of the pseudolikelihoods. Further, this chapter primarily characterizes the entropy of point processes as a function of their (conditional) intensities, which is natural as point process likelihoods are commonly characterized as such. However, for our current purposes, we are not interested in parameterizing intensities in the context of entropy estimation, and therefore these results remain interesting, but only useful for future work.

Ultimately, there is a dearth of recent results connecting modern information theory and point processes - so much so, that the few articles discussing the overlap of the fields mention the lack of recent publications (Koliander et al., 2018; Baccelli and Woo, 2016; Clark, 2019). We hope to exploit this relative paucity with finding some relatively accessible results.

The primary application of this project is point process representations of tensor data, an idea discussed in the context of various applications in the literature (Pang et al., 2017; Shen et al., 2008; Tagliazucchi et al., 2016; Xu et al., 2018; Zhe and Du, 2018). Point processes are a very natural way to characterize some tensor objects, and information theoretic results allow statisticians to glean valuable data from such objects even in high dimensional settings. We now begin with the notational framework for extracting the information content of point process realizations embedded in tensors (or more generally d-dimensional manifolds).

5.1.3 Characterizing Point Process Entropy

Characterizing the entropy of a point process requires some care. We therefore begin our discussion of point process entropy by building intuition about the maximum entropy of the uniform distribution, and entropy in the context of classic random variables. Entropy,

introduced by Shannon (Shannon, 2001), is defined as

$$H(X) = \mathbb{E}[I(X)] = -\sum_{i=1}^{n} \mathbb{P}(x_i) \log \mathbb{P}(x_i) \tag{5.1}$$

for random variable X and outcomes $x_1, \ldots, x_n$, where $I(X)$ is the "information content" of X. It is easy to see that entropy is maximized when $X \sim$ Unif, which we can check by taking the Lagrangian:

$$\begin{aligned}
L &= -\sum_{i=1}^{n} p_i \log p_i - \lambda \left(\sum_{i=1}^{n} p_i - 1 \right) \\
p_i &= \exp(-1 - \lambda) && \text{derivative w.r.t. } p_i \\
1 &= \sum_{i=1}^{n} p_i && \text{derivative w.r.t. } \lambda \\
\Rightarrow 1 &= n \exp(-1 - \lambda) \\
\frac{1}{n} &= \hat{p}_i
\end{aligned}$$

Plugging in $\hat{p}_i$ to Equation 5.1, we see that $H(X) = \log n$ for the uniform distribution. Similarly, for a uniform distribution on the unit hypercube in $\mathbb{R}^d$, the entropy is equal to $H(X) = \log n^d$.

For a homogenous Poisson process of n points on some bounded and finite space, the points (un-ordered) are uniformly distributed. Therefore in a spatial sense, a homogenous Poisson process represents the maximum entropy within the class of point processes with a given average rate (via their defining characteristic of *complete spatial randomness*). For more information, see Chapter 6 of Cinlar (2013) (connection between uniform distribution and Poisson processes), as well as Section 7.6 of Daley and Vere-Jones (2008) and Proposition 14.8.I of Daley and Vere-Jones (2007) (maximum entropy of Poisson processes). We strive to present a formal proof, simplifying the more general results contained in (Daley and Vere-Jones, 2007; Harremoës, 2001; Baccelli and Woo, 2016) that within the class of all Poisson point processes, homogenous Poisson processes have the maximum entropy.

In place of such a proof, we present here a simplified proof that among simple, stationary point processes which admit a Janossy density, homogenous point process have maximum entropy (Thm V.1. Baccelli and Woo (2016)). Janossy densities are defined for point processes with $N < \infty$ points in Euclidean space $\mathbb{R}^d$, and are a natural representation of general finite point processes characterized in Conditions 5.3.1 Daley and Vere-Jones (2007).[1]

A Janossy measure $J_n(\cdot)$ is a convenient and necessary measure to represent a likelihood for a stochastic un-ordered set such as a point process. We note that in general, we are not interested in finding the probability that specific *labelled* points are within some region $A \subseteq \mathcal{X}$, but rather that n points are within A. This means that any distribution $\Pi_n(\cdot)$ used to characterize a point process of interest must be symmetric for any partition $\{A_i\}$ of $\mathcal{X}$. We can then note that

$$\Pi_n^*(A_1 \times \cdots \times A_n) = \frac{1}{n!} \sum \Pi_n(A_{i_1} \times \cdots \times A_{i_n})$$

where $\Pi_n^*(\cdot)$ represents the symmetrized equivalent of $\Pi_n(\cdot)$ and the summation is across all $n!$ permutations $i_1, \ldots i_n$ of the un-ordered points $1, \ldots, n$ (each permutation is given equal weight) (Jánossy, 1950). We can then write that

$$J_n\left(\bigtimes_{i=1}^{n} A_i\right) = \frac{p_n}{n!} \sum \Pi_n(A_{i_1} \times \cdots \times A_{i_n}) = n! p_n \Pi_n^*\left(\bigtimes_{i=1}^{n} A_i\right).$$

This representation is flexible, permutation invariant, and natural to the context of point processes, but also rich in interpretation: if $\mathcal{X} = \mathbb{R}^d$, and we let $j_n(x_1, \ldots x_n)$ represent the density of $J_n(\cdot)$ with respect to the Lebesgue measure μ, the $j_n(x_1, \ldots x_n) d\mu(x_1) \cdots d\mu(x_n)$ is

[1] These coniditions can be summarized as

- Points are in complete separable metric space $\mathcal{X}$. **Seperable**: a topological space which contains countable dense subset, *e.g.* any space that is finite or countably infinite. **Complete**: Any Cauchy sequence (elements get arbitrarily close) of points in a space has a limit which is also in that space, *i.e.* there are no "holes" within or at the boundary of said space.
- A distribution $\{p_n\}$ for $n \in \mathbf{N}$ determines the total number of points, such that $\sum_n p_n = 1$.
- $\forall n \in \mathbf{N}$, $\Pi_n(\cdot)$ is a probability distribution on the Borel sets $\mathcal{X}^{(n)} = \mathcal{X} \times \ldots \times \mathcal{X}$ (Cartesian product) determining the joint distribution of the location of the points of the process given that $N = n$.

the probability that there are exactly n points in the process, one in each of the n infinitesimal regions $(x_i, x_i + d\mu(x_i))$ (Daley and Vere-Jones, 2007). Further, we can see that when $n = 0$, $j(\emptyset) = p_0$, given the convention $0! = 1$. Lastly we note that

$$\int_{\times_{i=1}^n A_i} j_n(x_1, \dots x_n) d\mu(x_1) \cdots d\mu(x_n) = n! p_n$$

and as a consequence, j_n is sufficient for backing out Π_n^* and p_n (van Lieshout, 2010). These properties allow us to represent the likelihood of a realization $x_1, \dots, x_n$ of a point process on a bounded Borel set $A \in \mathbb{R}^d$ where $n = N(A)$ and $N(\cdot)$ is the counting measure as

$$\mathcal{L}(x_1, \dots, x_n) = p_n \cdot n! \Pi_n^*(x_1, \dots, x_n) = j_n(x_1, \dots, x_n).$$

This final relation between likelihood and the Janossy density representation of a point process allows us to write down the entropy of a point process Φ on A as follows:

$$\begin{aligned}
\mathfrak{G}(\Phi) =& \frac{-1}{n!} \sum_{i=1}^{n} \int_{\times_1^n A} j_n(x_1, \dots, x_n | A) \log \left(j_n(x_1, \dots, x_n | A) \right) d\mu(x_1) \cdots d\mu(x_n) \\
=& \frac{-1}{n!} \sum_{i=1}^{n} \int_{\times_1^n A} \mathcal{L}(x_1, \dots, x_n) \log \left(\mathcal{L}(x_1, \dots, x_n) \right) d\mu(x_1) \cdots d\mu(x_n) \\
=& - E \left[\log(\mathcal{L}(x_1, \dots, x_n) \right]
\end{aligned}$$

We can the express the entropy of a point process more generally in the notation of Daley and Vere-Jones (2007) on a measure space $(\Omega, \mathcal{A}, \mu)$ and $\mathcal{P} \ll \mu$ a probability distribution on the $\mathcal{A}$. This notation is useful for leveraging the superimposition theorems in Chapter 11 of Daley et al. (2008) Daley and Vere-Jones (2008). Using this notation,

$$\begin{aligned}
\mathfrak{G}(\mathcal{P}; \mu) &= - \int_{\Omega} \Lambda(\omega) \log \Lambda(\omega) \mu(d\omega) \\
&= - \int_{\Omega} \log \Lambda(\omega) \mathcal{P}(d\omega) \\
&= - \mathbb{E}_{\mathcal{P}}[\log \Lambda]
\end{aligned}$$

represents the entropy of $\mathcal{P}$ with respect to reference measure μ and $\Lambda(\omega) = d\mathcal{P}/d\mu$ is the Radon-Nikodym derivative of $\mathcal{P}$ with respect to μ. We note that if $\mu = \mathcal{Q}$ for some

probability measure $\mathcal{Q}$ (the Lebesgue measure is not a probability measure as $\mu(\mathbb{R}^d = \infty)$, then

$$\mathfrak{G}(\mathcal{P}; \mathcal{Q}) = -\int_{\Omega} \log \frac{d\mathcal{P}}{d\mathcal{Q}} \mathcal{P}(d\omega)$$

which is the KL divergence between probability measures $\mathcal{P}$ and $\mathcal{Q}$. Such notation is unfortunately necessary to accommodate the fact that for a point process, the number of points is random, and therefore we must take the entropy with respect to the underlying counting measure. The (unnormalized) reference measure specified in Chapter 14 of Daley et al. (2008) for simple point processes is $\exp(T)\text{Poi}(\lambda, T)$ where $\text{Poi}(\lambda, T)$ is the the probability measure of a Poisson process with constant intensity λ on the interval $(0, T)$ (Daley and Vere-Jones, 2007, 2008). The larger point is that sometimes we are interested in the entropy relative to a measure commensurate with the structure of the space on which the underlying process is defined.

For the purposes of this review, we prefer the less general notation of $\mathfrak{G}(\Phi)$. We note that point process entropy $\mathfrak{G}(\Phi)$ is neither Shannon nor differential entropy as it is not fully discrete or continuous. To this point, $\mathfrak{G}$ for a single realization $\phi = \{x_1, \ldots, x_n\}$ of Φ on bounded space $A \subset \mathcal{X}$ can be decomposed into three parts: the sum of the discrete and continuous entropies, less a constant factor for the redundancy (intersection) of the two.

$$\mathfrak{G}(\Phi) = \underbrace{H(n)}_{\text{Shannon}} + \mathbb{E}[\underbrace{h(\phi|n)}_{\text{differential}}] - \mathbb{E}[\log(n!)] \tag{5.2}$$

$$\mathfrak{G}(\mathcal{P}; \mu) = -\sum_{k=0}^{\infty} p_k \log p_k - \sum_{k=1}^{\infty} p_k \int_{A^{(k)}} \Pi_k^*(\phi) \log(k! \Pi_k^*(\phi)) dx_1 \ldots dx_k \tag{5.3}$$

$$\mathfrak{G}(\mathcal{P}; \mathcal{Q}) = -\sum_{k=0}^{\infty} p_k \log \frac{p_k}{q_k} - \sum_{k=1}^{\infty} p_k \int_{A^{(k)}} \Pi_k^*(\phi) \log \left(\frac{\Pi_k^*(\phi)}{q_k^*(\phi)} \right) dx_1 \ldots dx_k \tag{5.4}$$

$$\mathfrak{G}(\mathcal{P}) = -\sum_{k=0}^{\infty} \mathbb{P}(n = k) \log \mathbb{P}(n = k) + \sum_{k=0}^{\infty} \mathbb{P}(n = k) h(\phi|n = k) \tag{5.5}$$

We provide these alternative notations in an effort to have a demonstrate the equivalences of those proposed in McFadden (1965); Papangelou (1978); Daley and Vere-Jones (2008).

These two under-bracketed components in Equation 5.2 can be intuited as the entropy commensurate with the number of points observed and the entropy commensurate with the location of said points, and are referred to as the numerical and locational entropies, respectively in the point process literature (McFadden, 1965). The $\log(n!)$ term can be intuited as the loss of information attributed to not knowing which particle is in which location (Daley and Vere-Jones, 2008). Note that $\mathfrak{G}(\Phi)$ is scale dependent due to the differential entropy term. See Proposition III.1 of Baccelli and Woo (2016) for further discussion of this decomposition, and Papangelou (1978) for in-depth discussion of point process entropies in the context of likelihoods.

For a finite inhomogenous Poisson process $\Phi \subset A \subseteq \mathbb{R}^d$, we can wrote the log likelihood as

$$\log \mathcal{L}(x_1, \ldots, x_n) = \sum_{i=1}^{n} \log \lambda(x_i) - \int_A \lambda(x) dx$$

where is the intensity of Φ (Daley and Vere-Jones, 2007). It then follows that for a homogenous Poisson process Φ_λ, wherein λ is constant,

$$\mathfrak{G}(\Phi_\lambda) = \lambda \mu(A)(1 - \log \lambda) \tag{5.6}$$

where $\mu(\cdot)$ is the Lebesgue measure. If such a process is observed on $\mathbb{R}$ on some bounded interval $(0, T]$, we can write $\mathfrak{G}(\Phi_\lambda|_{(0,T]}) = \lambda T(1 - \log \lambda)$ (of note, this formula contains an error on page 565 of (Daley and Vere-Jones, 2008), and is correctly presented in McFadden (1965)). It is directly shown by FcFadden that $\mathfrak{G}(\Phi_\lambda)$ is the maximum entropy for a given intensity λ. The proof presented below relies on a simpler method, and is presented in Baccelli and Woo (2016).

As described in Baccelli and Woo (2016), we can extend the definition for point process entropy to a global entropy rate. For a sequence of bounded convex sets $A \in \mathcal{B}(R^d)$, $A_l \subseteq A_{l+1}$ $l \in \mathbf{N}$, we can define a global entropy rate per unit volume

$$\mathfrak{G}(\Phi) = \lim_{l \to \infty} \frac{\mathfrak{G}(\Phi|_{A_l})}{\mu(A_l)}$$

although the existence of $\mathfrak{G}(\Phi)$ is not guaranteed. Conditions of existence are of interest, and to be pursued in the future. Existence is guaranteed for stationary processes (Papangelou, 1978; Daley and Vere-Jones, 2008).

We can extend the notion of a point process entropy to accommodate a point process analogue of KL divergence (Baccelli and Woo, 2016):

$$KL(\Phi_1||\Phi_2) = E_{\Phi_1}\left[\log\left(\frac{\mathcal{L}_{\Phi_1}(x_1,\ldots,x_n)}{\mathcal{L}_{\Phi_2}(x_1,\ldots,x_n)}\right)\right]$$

and find the "distance" between two point processes. Specifically, we are interested in finding the maximum entropy for simple point processes subject to the constraint that the mean intensity λ is fixed, as seen in (McFadden, 1965). We derive as follows for simple point process Φ and point process Π_n^λ with fixed mean intensity λ which share common support in $\mathcal{B}(\mathbb{R}^d)$:

$$\begin{aligned}
KL(\Phi||\Pi_n^\lambda) =& E_\Phi\left[\log\left(\frac{\mathcal{L}_\Phi(x_1,\ldots,x_n)}{\mathcal{L}_{\Pi_n^\lambda}(x_1,\ldots,x_n)}\right)\right] \\
=& \mathbb{E}[\mathcal{L}_\Phi(x_1,\ldots,x_n)] - \mathbb{E}[\mathcal{L}_{\Pi_n^\lambda}(x_1,\ldots,x_n)] \\
=& -\mathfrak{G}(\Phi) - (-\mathfrak{G}(\Pi_n^\lambda)) \\
=& \mathfrak{G}(\Pi_n^\lambda) - \mathfrak{G}(\Phi) \geq 0 \qquad (*) \\
\mathfrak{G}(\Pi_n^\lambda) \geq& \mathfrak{G}(\Phi)
\end{aligned}$$

where the $(*)$ inequality is due to the fact that the log is a concave function and Jensen's inequality (Cover, 1999). Equality in the final line is only achieved when when Φ has constant intensity λ, which is to say that the homogenous Poisson process achieves maximum entropy. Dayley et al. (2008) directly show that Equation 5.3 is maximized by (1) showing that the first term of Equation 5.3 is maximized by the Poisson distribution subject to the conditions that $\sum_{k=0}^{\infty} p_k = 1$, $\sum_{k=1}^{\infty} k p_k = \mu$ and $\forall k\ k \geq 0$ and (2) showing that conditional on k, the integrand in the second term of Equation 5.3 is maximized when the symmetric distribution is a uniform distribution on $\mathcal{X}^{(k)}$ (Daley and Vere-Jones, 2008, 14.8.2(a)).

To briefly summarize: we see in Equation 5.2 that point process entropy can be split in to two primary components - the numerical (Shannon) and locational (differential) entropies. It is intuitive that the numerical entropy, *i.e.* the number of points or the cardinality of a realization, is maximized by a Poisson distribution, and the locational entropy is maximized by a d-dimensional uniform distribution (McFadden, 1965; Daley and Vere-Jones, 2008). Therefore, we can maximize Equation 5.2 by letting the underlying point process be a homogenous Poisson point process (Prabhakar and Bambos, 1995).

In conclusion, this means that we can use a homogenous Poisson process as a *reference process* (as it is maximally entropic among the family of possible processes) and consider the change in entropy between an unknown starting process and this reference process as the *information loss* across iterative superimpostions. This is valuable because the information loss is a function of the the underlying process (whether parameterized or not), an indicator of the model's intrinsic predictability (Daley and Vere-Jones, 2007). Although not the original intent of the author, this fact could allow for model selection of the underlying unknown process, a property which we hope to explore in the future.

5.1.4 Estimation of Point Process Entropy

Why go to these lengths to show that the homogenous Poisson process achieves maximum entropy given the constraint of fixed intensity? We are interested in finding the information content of a point process representation of a tensor object. To "back-out" this information content, we compare it to the most uncertain (*i.e.* maximum entropy) member of our point process representation's assumed family.

Given that we are interested in comparing an unparameterized process (assumed to be a simple stationary point process) to a homogenous Poisson process, the natural question is how do we measure dissimilarity or divergence. Several attractive options include: KL divergence, which is easy to interpret in an information theoretic context, and the tensor-equivalent of the Frobenius norm, which is the natural operator in the tensor context for low

rank tensor approximation (defined in Section 5.3).

We discuss here the method of comparison we think simplest, which is closely related to the KL divergence: we compare the entropies of the iteratively superimposed point processes. Because the points in the superimposed process are obviously stochastic in nature, the tensor-equivalent of the Frobenius norm is only of interest to us in expectation. This presents challenges for a single realization dataset if we want to avoid making parametric assumptions. We can easily calculate the maximum entropy upper bound, as we have an analytic solution for the entropy of a homogenous Poisson process Φ^* (see Equation 5.6). However, we do not have such an expression for the (nonparameterized) process Φ_{s+1}. Therefore, we must nonparametrically estimate $\mathfrak{G}(\Phi_{s+1})$.

To nonparametriaclly estimate the entropy of a point process, we have two options. We can nonparametrically estimate the Janossy density (likelihood), or we can nonparametrically estimate the terms of Equation 5.2. Here we opt for the second choice, leaving the first to future work. But before doing so, we take a brief detour to re-introduce Shannon entropy, and the relations it shares (and lacks) with its "continuous extension," differential entropy.

Differential entropy is defined as

$$h(X) = \int_{\mathcal{S}} f(x) \log f(x) dx \tag{5.7}$$

where $\mathcal{S}$ represents the support of continuous random variable X. Introduced by Shannon (Shannon, 2001), differential entropy was put forth as the continuous analogue of discrete (often referred to as Shannon) entropy. In fact, Shannon did not derive differential entropy from first principles like he did with discrete entropy, and rather replaced the summation with an integral (Jaynes, 1957). Such an assumption is not totally unfounded, and here we sketch out the most obvious relation between differential and discrete entropy (Section 8.3 Cover (1999)). If we bin X into bins of length Δ, then by the mean value theorem, for each bin, there exists x_i such that

$$f(x_i)\Delta = \int_{i\Delta}^{(i+1)\Delta} f(x)dx = \mathbb{P}(X^{\Delta} = x_i) = p_i$$

and we can denote $\mathcal{X}^\Delta = x_i$ as a binned (quantized) equivalent of X. Then we can see

$$\begin{aligned} H(X^\Delta) &= -\sum_{i \ni x_i \in \text{supp}(X^\Delta)} p_i \log p_i \\ &= -\sum_i f(x_i)\Delta \log(f(x_i)\Delta) \\ &= -\Delta\left(\sum_i f(x_i)\log(f(x_i)) + \log\Delta \sum_i f(x_i)\right) \\ &= -\Delta\sum_i f(x_i)\log(f(x_i)) - \log\Delta \end{aligned}$$

and therefore that

$$\lim_{\Delta\to 0} H(X^\Delta) = h(X) - \log\Delta$$

providing $f(x)\log f(x)$ is Riemann-integrable. It can be shown using a similar quantizing proof that the differential entropy is closely related to the information dimension of a stochastic process (Geiger and Koch, 2019; Cover, 1999).

In our case of interest, f is often not Riemann-integrable. Luckily, a much simpler (and more point process-centric) relation between discrete and differential entropy can be found. Equation 5.7 is commonly interpreted as $h(X) = \int_{\mathcal{S}} f(x)\log f(x) d\mu(x)$ where $\mu(\cdot)$ is the Lebesgue measure. However, we can see that if instead we assume $X \in \mathcal{S}$ is discrete and denote $N(\cdot)$ as the counting measure,

$$\begin{aligned} h(X) &= -\int_{\mathcal{S}} f(x)\log f(x) dN \\ &= -\sum_{x\in\mathcal{S}} f(x)\log f(x) \\ &= H(X) \end{aligned}$$

For a continuous uniform random variable with support $[0, T]$, we see that

$$h(X) = -\int_0^T \frac{1}{T}\log\frac{1}{T} dx = \log T$$

which on the face of it is exactly what we would expect when comparing to the discrete counterpart solved for above. However, we can see that if $T < 1$, $h(X) < 0$, and therefore that the differential entropy can take negative values, in effect reducing its "information content" interpretation. In general, although tempting, we cannot think of differential entropy as the limiting expression for discrete entropy - in fact it can vary from the discrete entropy by a potentially infinite offset. Further, differential entropy is not invariant under change of variables (Cover, 1999). Jaynes suggested an alternative to differential entropy called the limiting density of discrete points (LDDP), which is constructed on the belief that continuous entropy should be derived by taking the limit of increasingly "dense" discrete distributions (Jaynes, 1957). Given that the cardinality of the support $\mathcal{S}$ of random variable X is equal to n, the LDDP is approximately $\log(N) - \log(r) + h(X)$ where r is the length of an interval where the limiting number of points is constant. The LDDP is invariant under change of variables.

Because differential entropy does not share properties of discrete entropy such as invariance to change of variables or non-negativity, it is not a good measure of absolute information, but rather relative information. From an information theoretic perspective, continuous random variables have probability of zero of taking on specific values, and therefore require an infinite number of bits to encode (they have infinitely long decimal representations and therefore convey infinite information). However, the fact that differential entropy can take negative values is a intuitive feature: if we to know the data on a level less than one unit (bin, in the discrete case), then entropy goes down as we are more informed, but if we were to make the unit size smaller, than we would once again know less and the entropy would rise (Cover, 1999). For instance, in the above example of a continuous uniform random variable, if we $T < 1$ then $h(X) < 0$, but if we scale T by some constant α such that $\alpha T \geq 1$ then $h(X) \geq 0$.

We now turn to the question of estimating the relative entropies of our homogenous Poisson reference process and the unparameterized point process (*i.e.* the point process

representation of a tensor object) which we iteratively superimpose homogenous Poisson processes upon. We note that necessary to estimate the point process entropy of our reference process (which we denote $x^{\text{ref}} = \{x_1^{\text{ref}}, \ldots, x_m^{\text{ref}}\}$) as we have a parametric form for it.

Estimating the point process entropy $\mathfrak{G}(\phi)$ requires estimating the two entropic components of Equation 5.2, *i.e.* the Shannon and differential entropy. As discussed above, the Shannon component for a single realization is constant - once observed, point process Φ has a non random number of points. Therefore we cannot estimate the Shannon component (meaningfully) without assuming some kind of parametric structure for Φ. We remedy this issue by bounding $\mathfrak{G}(\phi)$ by assuming that the number of points is a homogenous Poisson process, *i.e.* the entropy maximizing ground process. More formally, given ψ a realization of a homogeneous Poisson process,

$$\begin{aligned}\mathfrak{G}(\phi) =& H(n) + h(\phi|N(\phi) = n) - \log(n!) \\ \leq& \mathfrak{G}(\psi) + \underbrace{h(\phi|N(\phi) = n) - h(\psi|N(\psi) = n)}_{\text{difference in differential entropies}}.\end{aligned}$$

Because we have an analytic representation of $H(\psi)$, this allows us to reduce our problem of interest to estimating the differential entropy component $h(\phi|n_\phi)$.

Assuming that the first entropic component of Equation 5.2 is that of a homogenous Poisson process is a large concession in that it is cutting out a meaningful amount of information intrinsic to the process generating ϕ. An alternative is to assume that Φ has a "time dimension" upon which we can partition Φ into sub-processes $\Phi_1 \cup \ldots \cup \Phi_p$. Assuming that each partitioned process is sufficient to describe the overall behavior of counting measure of the general process (an assumption which we have no reason to believe generally holds for any Φ) we can use $N(\Phi_i)$ for all $i \in 1, \ldots, p$ to nonparametrically estimate $H(n_\phi)$ using a histogram estimator. If no apparent "time dimension" exists, we could alternatively partition across a random dimension.

If many (or even more than one) realizations of Φ are observed, the problem of estimating $H(n_\phi)$ disappears, but the general context of our current problem of interest is centered only

on a single realization. For now, we proceed with the upper bound solution (*i.e.* we plug in $H(n_\psi)$ for $H(n_\phi)$).

Nonparametric estimation of differential entropy is well-discussed in the literature, and several attractive estimations exist. Excellent summaries of nonparametric entropy estimation can be found in (Beirlant et al., 1997; Paninski, 2003). The below discussed estimates are consistent (given various and varying conditions), but all are biased. Actually, there is no unbiased estimator of differential entropy unless we know that our sample set contains at least one sample from each class (Paninski, 2003; Montgomery-Smith and Schürmann, 2014). Several so called plug-in estimates exist using kernel density estimates, histogram-type estimates, and cross-validation estimates (Joe, 1989; Beirlant et al., 1997). Recent results have demonstrated the consistency of kernel estimation for homogenous marked Poisson processes (Alonso-Ruiz and Spodarev, 2017). The properties of entropy estimates based on sample spacings, such as the m-spacing estimate are also well discussed (Hall, 1984). While spacing options have attractive asymptotic properties, they do not extend into a multi-dimensional context. Other options such as estimates based on geodesic minimal spanning trees (Costa and Hero, 2006) and Cesaro averages of longest match lengths (Kontoyiannis et al., 1998) are also well documented.

Another promising option for estimating differential entropy nonparametrically is using a nearest-neighbors approach. A nearest neighbor approach is intuitively attractive because in a uniform distribution, we would expect the distances between points and their nearest neighbors to be less variable. First introduced (in Russian) by Kozachenko and Leonenko, the nearest neighbor differential entropy estimator is commonly referred to as the KL estimator (not to be confused with Kullback-Leibler divergence) (Kozachenko and Leonenko, 1987). Kozachenko and Leonenko show consistency and asymptotic unbiasedness of their estimator. We notate the KL estimator as follows:

$$\hat{\mathfrak{G}}(\phi) = \frac{1}{n}\sum_{i=1}^{n}\log(n\rho_{n,i}) + \log(2) + C_E \tag{5.8}$$

where $\rho_{n,i} = \min_{j \neq i, j \leq n} ||x_i - x_j||$ and C_E is the Euler constant.

As a brief aside, nearest neighbor approaches are attractive from a point process standpoint because they are well explored within the point process literature and theoretical framework. We denote the d-dimensional equivalent of ρ above as

$$\mathcal{D}_x(r) = 1 - \mathbb{P}(N(B(x,r)) = 1|x) \tag{5.9}$$

where $B(x,r)$ is a d-dimensional ball centered on x with radius r. We term Equation 5.9 the nearest neighbor function. For a Poisson process Φ on $\mathbb{R}^d$ with intensity measure Λ, we can write

$$\begin{aligned} \mathcal{D}_x(r) =& 1 - \exp(-\Lambda(B(x,r))) \\ =& 1 - \exp(-\lambda\mu(B(x,r)) \qquad\qquad \text{if } \Phi \text{ homogenous} \end{aligned}$$

where $\mu(\cdot)$ represents the Lebesgue measure. Further, Equation 5.9 is closely related to the so-called F,J, and K functions which have rich theoretical properties for summarizing point processes and inhibition or clustering between points (Van Lieshout, 2011).

The KL estimator was extended to a k-nearest neighbors approach (Singh et al., 2003), and many other modifications have followed to reduce bias and increase convergence rate (Berrett, 2017; Lombardi and Pant, 2016). We propose to use a $k-$nearest neighbors differential entropy estimator to estimate $h(\phi|n_\phi)$.

5.1.5 Superposition Limit Theorems

We are interested in gaining understanding of the rate of convergence for the point process entropy of the iteratively superimposed upon Φ to the point process entropy of the reference homogenous Poisson process. This rate of convergence depends on the sparsity of the starting point process. It is our hope that gaining understanding of this convergence can yield insight into the efficacy of a complexity number as described above.

In order to approach such a result, we can first tackle the following related question:

Given starting process Φ_1, let $\mathcal{P}_s$ be the intensity measure of the iteratively superimposed process

$$\Phi = \Phi_1 \cup \left(\bigcup_{k=1}^{s} \Psi_k^{\text{pois}(\hat{\lambda})} \right). \tag{5.10}$$

Note s and Φ are dependent on tolerance ϵ. Let $\mathcal{Q}$ be the intensity measure of Ψ, a homogenous Poisson process with intensity $(s+1)\hat{\lambda}$. By construction, $\forall \epsilon > 0$, $\lim_{s\to\infty} |\mathcal{P}_s - \mathcal{Q}|_{TV} \to 0$. This only assumes that the starting tensor has finite elements. We can show this using triangular arrays (Cinlar and Agnew, 1968; Schuhmacher, 2005). Note that *strong convergence* in variation norm is defined as where $||\xi||_{\text{TV}} = ||\mu_n - \mu||_{\text{TV}}$

$$||\xi||_{\text{TV}} = \sup_{P(\mathcal{X})} \sum_{i=1}^{\#P(\mathcal{X})} |\xi(A_i)|$$

as in Daley and Vere-Jones (2008).

More recently, Wasserstein distance has been used to measure the distance between a superposition of dependent point processes and a Poisson process (Schuhmacher, 2005), and the squared Hellinger distance (which is closely related to the variation norm mentioned above) between two measures is another option.

Ultimately, we want to know if considering homogeneous Poisson process Ψ,

$$|\mathcal{P}_s - \mathcal{Q}|_{\text{TV}} \to 0 \iff |\mathfrak{G}(\mathcal{P}_s) - \mathfrak{G}(\Psi)|_1 \to 0. \tag{5.11}$$

Assuming that this is true, we are interested in the rate of convergence as our application of interest deals with a finite number of superpositions. Specifically want to bound the rate of convergence of $\mathfrak{G}(\Phi) \to \mathfrak{G}(\Psi)$ as a function of the sparsity of starting process Φ_1 and ϵ. This requires a nuanced definition for a robust measure of sparsity.

Simple results about the superposition of point processes can be found in Chiu et al. (2013), and many of the results contained therein are self-apparent. Specifically, for two homogenous Poisson processes Φ_1 and Φ_2 with intensities λ_1 and λ_2, we can define the superposition of the two processes as the set theoretic union $\Phi = \Phi_1 \cup \Phi_2$ (assuming that

$\Phi_1 \cap \Phi_2 = \emptyset$), and see that Φ is also a homogenous Poisson process with intensity $\lambda = \lambda_1 + \lambda_2$ (Daley and Vere-Jones, 2007). This is simply because the sum of independent Poisson distributions is Poisson. Less obviously, the nearest neighbor function of Φ can be expressed as

$$\mathcal{D}_\Phi(r) = 1 - \frac{\lambda_1}{\lambda}(1 - \mathcal{D}_{\Phi_1}(r))(1 - \mathcal{H}_{s,2}(r)) + \frac{\lambda_2}{\lambda}(1 - \mathcal{D}_{\Phi_2}(r))(1 - \mathcal{H}_{s,1}(r))$$

where $\mathcal{H}_{s,1}(r)$ is the spherical contact distribution function for Φ_1 and $\mathcal{H}_s(r) = 1 - (1 - \mathcal{H}_{s,1}(r))(1 - \mathcal{H}_{s,2}(r))$ (Van Lieshout and Baddeley, 1996).[2]

Further, Dayley and Vere Jones provide a chapter to limit theorems for superpositions (Daley and Vere-Jones, 2008, Chapter 11.2). In this chapter, conditions are described for the convergence of independent superpositions of point processes to a Poisson process limit. The theorems can be roughly thought of as equivalent to those of sums of independent identically distributed random variables. In particular, Theorem 11.2.III proves (and provides conditions for) convergence to a Poisson process. Example 11.2(a) describes conditions for weak convergence of superpositions to follow a Poisson process (Daley and Vere-Jones, 2008). The results in Daley et al (2008) are augmented by Cinlar's papers on the necessary and sufficient conditions for the superposition of point processes to result in a renewal process (Cinlar and Agnew, 1968) or a d-dimensional Poisson process (Çinlar, 1968). Convergence of superposed uniformly sparse point processes (to a Poisson process) is discussed in (Goldman, 1967).

In general, limit theorems for superpositions use triangular arrays, with the added assumption that such arrays are uniformly asymptotically negligible (Daley and Vere-Jones, 2008, Chapter 11.2), and rely on assumptions like the infinite divisibility of the underlying processes (Daley and Vere-Jones, 2008, Chapter 11.1),(Ripley, 1976). The results in this

[2]For some convex and compact set B in $\mathbb{R}^d$, with $\mu(B) > 0$ and $o \in B$, we define the sphereical contact distribution function as $\mathcal{H}_B(r)$ for point process Φ as

$$\mathcal{H}_B(r) = 1 - \mathbb{P}(\Phi(B(o, r) = 0)$$

which is interpreted as the distribution function of the distance from o to the nearest point of Φ (Chiu et al., 2013).

section are mentioned only as a brief preliminary survey of tools which can hopefully be applied in our future study of our problem.

5.2 Application: Damage in Ballistic-Struck Materials

We present the motivating problem which lead to our study of the entropy of embedded point processes in tensor objects and Euclidean space. Entropy has been used as a measure of complexity in images (Yu and Winkler, 2013), but the method of entropy estimation is crucial, as a simple histogram approach loses all spatial information, meaning that a very low information image can have high estimated entropy (Feldman and Crutchfield, 2003). Nearest neighbor based estimates of image entropy do not possess this disadvantage. However, in our below proposed method, we use a histogram estimator of entropy because we do not look at total-image entropy but rather fiber-wise "column-voxel" entropy. For such an approach, the spatial information is not as meaningful - rather we are just trying to estimate how variable the cracking is across layers for a projected two-dimensional location.

Entropy thresholding for images, as implemented below, has been used and studied extensively (Chang et al., 1994). Feature extraction for images using point processes is discussed in Lafarge et al. (2009), but not implemented in the below methodology as we are not as interested in feature extraction as estimation of image complexity as a proxy for amount of cracking.

Evaluation of image complexity via fractal methods is discussed in Lam et al. (2002), but not implemented here as we opted for a simpler and more interpretable method. Excellent summaries of the theory surrounding entropy of images with dimension greater than two underlying our method can be found in Larkin (2016); Thum (1984).

5.2.1 Data Description

Our team developed a statistical methodology to visualize and summarize the amount of damage in a three-dimensional material, both locally and globally.

We began with large three-dimensional images, provided in the `NIFTI` format. Initially two images were provided, each approximately three gigabytes. The first of these images, labelled "17-0789-4" will be used as the primary example in this description of our methodology. In July 2020, a second dataset of 39 images was provided, each of substantially higher resolution. These images range in size from four gigabytes to more than eleven gigabytes, substantially increasing the computational complexity of our methods; further details are provided below.

The Neuroimaging Informatics Technology Initiative ("NIFTI") file type is widely used for biomedical imaging data. It has a wide array of features for a given image-object including affine coordinate definitions relating voxel indices to their respective corresponding spatial locations; easily queried header elements and an approximated voxel-distance array; scaling for voxel size; and large amount of existing libraries and packages in `R` and `c++`. There is a large body of literature discussing tumor identification and other related studies using `NIFTI` data objects, see Bandyopadhyay and Paul (2013); Xia et al. (2016). In `R`, the `NIFTI` data objects can be unlisted in to a three-dimensional array which is approximately three gigabytes. Each image is a composite of 2000 two-dimensional "slices" of a bullet-deformed material, as seen in Figure 5.2.

5.2.2 Problem Description

From the raw image data, we

1. Identified fracture boundaries;

2. Differentiated (and labelled) vest material, background, and fracture boundaries;

3. Visualized labelled data in 2 and 3 dimensions;

4. Assessed uncertainty associated with labelling; and

5. Estimated total fracturing, i.e. damage to the material.

Various applications in seismology, radiology, dentistry, and material sciences, all document approaches to similar problems, *i.e.* the problem of quantifying irregular boundaries or fractures. Crack propagation is explored in the contexts of dental ceramics in Wang et al. (2015) and concrete foundations in Scherrer et al. (2017). Seismic acoustic events from cracking is explored in Barés et al. (2018). Fatigue crack growth rate in a machine learning context is discussed in Wang et al. (2017). So-called "statistical fractography" is discussed in the context of steel fracturing in Jamwal et al. (2013). MRI segmentation using learned Gaussian mixtures for tumor identification is discussed in Xia et al. (2016). This review revealed that although there exists related research in a wide array of applications, most results discussed are based on ad-hoc methods.

Given the literature reviewed, we opted to employ the following method to our problem of interest: canny edge detection with variable tuning parameters across slices and Lloyd's algorithm (similar to k-means) for label classification of the edge-detected images.

5.2.3 Findings

Using canny-edges and Lloyd's algorithm, we labelled three classes $\{0, 1, 2\}$ representing background, material and material-background boundary (i.e. fracture) on a slice-by-slice basis. After edge detection, we arrive at an image like Figure 5.3 for each slice. Of note, significant parameter tuning for thresholding of the gradient step and the σ parameter of the Gaussian blur step is needed to achieve relatively high quality edge detection across slices.

Once edges are detected, we label the classes with Lloyd's algorithm, resulting in an image like Figure 5.4 for each slice. After looping through the entire array and generating

an image analogous to Figure 5.4 for each slice, we form a composite visualization of the three-dimensional image in two dimensions. To do this, we need to "flatten" the voxels in a column or row, effectively projecting the image down from three to two dimensions. We experimented with various "flattening functions" including sum, average, and variance, before settling on entropy. Entropy represents the average number of bits of information per column-voxel. A column-voxel that has a higher entropy means that there are more labels, and a longer minimum unique code-length. Therefore, we propose that entropy is a meaningful proxy for damage on a given column-voxel. Similar approaches are discussed in Wang and Shen (2011).

Figures 5.5 and 5.10 are two-dimensional visualizations of the damage in the associated three-dimensional image objects. The primary goals of such visualizations is to provide accessible information for evaluating where and to what extent the material is damaged. Figure 5.6 is a perspective plot of the flattened entropy projection. We believe that a plot similar to this one could prove most valuable to a non-statistician who is attempting to attain a qualitative understand of material damage relative to the preceding plots. Figure 5.9 represents a simpler plot of the damage associated with the flattened entropy projection objects.

Lastly, we propose a simple statistic to capture the global damage of the material: the mean of the flattened voxels. For image "17-0789-4", this statistic evaluates to $\bar{x} = 1.093571$, indicating a relatively high level of damage. Table 5.2 presents various summaries of the damage to the materials for each of the 39 images obtained. Our process of using the entropy function to project into a lower dimension (i.e. flatten the three-dimensional image objects) allows us to capture and summarize the distortion caused by the projectile striking the material. Our ensemble approach appears to be both practical and novel.

5.2.4 Measuring Image Complexity

Alternatively, we employed a different methodology, as discussed in Section 5.1.1. Beginning with the respective three dimensional labeled images, we dropped (made equal to zero) all labels except those indicating probable cracking. These non-zero elements are then interpreted as representing points with corresponding spatial location equal to their respective voxel indices. This allows for a sparse representation of the labeled image, and for us to interpret the labeled image as a realization of some unparameterized point process.

Our goal is to measure the information content (as a proxy for cracking) contained in the labeled image. We can do this by first fitting a homogeneous Poisson (completely spatially random) process to our starting labeled image represented as a point process. We do this because (1) a homogeneous Poisson process is maximally entropic, and (2) an appropriately scaled homogeneous Poisson can then be iteratively superimposed on our starting process. As superposition is an entropy-increasing operation, we can measure the relative entropy across iterations, which is equivalent to the "information loss."

We perform iterative superpositions until we cannot discern our iteratively superimposed process from a true (maximally entropic and appropriately scaled) reference homogeneous Poisson process. We can measure how different our iteratively superimposed process is from a reference process using a KL analogue for point processes. We need to choose some tolerance level ϵ, where after the KL divergence is less than ϵ, we stop performing superpositions. The number of necessary iterations to achieve tolerance level ϵ, which we call s, is the resulting damage summary statistic.

We then tuned the the tolerance parameter ϵ, which regulates how many iterative superpositions are needed for a given starting image. The resulting damage summary statistic from the complexity measure approach is a positive integer s, corresponding to the number of superpositions necessary to make the KL divergence between the iteratively superimposed process and a true homogeneous Poisson process less than . Therefore, choosing smaller

results in larger S, but not necessarily more informative s.

We found that $\epsilon = 10^{-2}$ seems to work best for predicting the damage statistic obtained via the mean flattened entropy value. However, there is no best choice of ϵ that correlates best with second shot energy absorption. We believe that this is due to the fact that different images have different resolutions as well as different size plates. This problem can be minimized by finding an "optimal ϵ" based on the density of voxels. Table 5.2 summarizes the resolutions and the approximate values of which predict second shot energy absorption most accurately.

Practically, values of "optimal ϵ " are chosen as follows: $\epsilon \approx V \cdot 10^{-10}$ where V is the number of voxels per square inch. The complexity statistic is not as correlated to second shot energy absorption as the flattened entropy damage statistic, but as it uses an entirely different methodology and therefore we believe that it adds a valuable piece of predictive information.

5.2.5 Future Work

Time complexity of our ensemble methodology is dominated by the canny edge detection step. Edge detection for a two-dimensional slice with $m \times n$ pixels has $\mathcal{O}(mn \log(mn))$ time complexity. Therefore the images provided in July 2020 are significantly more time intensive to process. For instance, an eleven gigabyte image takes roughly nine times longer to process than a three gigabyte image.

Furthering the time cost, the images provided in July 2020 are taken with a different scanning procedure, resulting in new parameter tuning for the canny edge detection step, which must be performed manually. Although initial parameter tuning has allowed us to process and summarize the July 2020 images, we are interested in further parameter tuning for better edge detection. The most sensitive step to parameter tuning in our current ensemble method is the slice-by-slice Canny edge detection. Specifically, any edge detection algorithm

needs to be denoised (especially given the ultra-high resolution of the images provided). To denoise appropriately, we needed to explore the $k \times k$ Gaussian filter across k. Of even greater importance, we varied the MinThreshold and MaxThreshold parameter values which are used in the final Hysteresis step of Canny edge detection. These thresholds dictate which labelled edges are kept, specifically challenging edges between the min and max thresholds to see if they are connected to other "strong edges." To effect this parameter tuning, we need to implement a manual grid search, varying k, MinThreshold, and MaxThreshold across slices and then changing parameters based on visual inspection of the associated visualizations.

We are also interested in developing our methodological approach to summarize flattened images by fitting a point process to a thresholded image. Thresholding images in the context of general computer vision problems is discussed in Cheng et al. (2000). Practically, thresholding a flattened image entails creating a binary matrix which contains elements which are true only if the entropy value for that specific element is greater than the threshold value. We can choose a threshold value of mean(entropy value)+1 standard deviation. This allows us to represent the "points" which have an entropy value greater than the threshold value (where the points' coordinates are the indices of the matrix which have true values). Figure 5.7 demonstrates such a thresholded point process. This step allows us to re-formulate a flattened image as a point process. In the future, we are interested in fitting an inhomogenous Poisson intensity via maximum likelihood. This will allow us to create valuable summary statistics for the amount of damage done to the vest. We are interested in exploring quantities such as expected value and variance for each of the 39 images, and any future data.

5.2.6 Figures and Tables

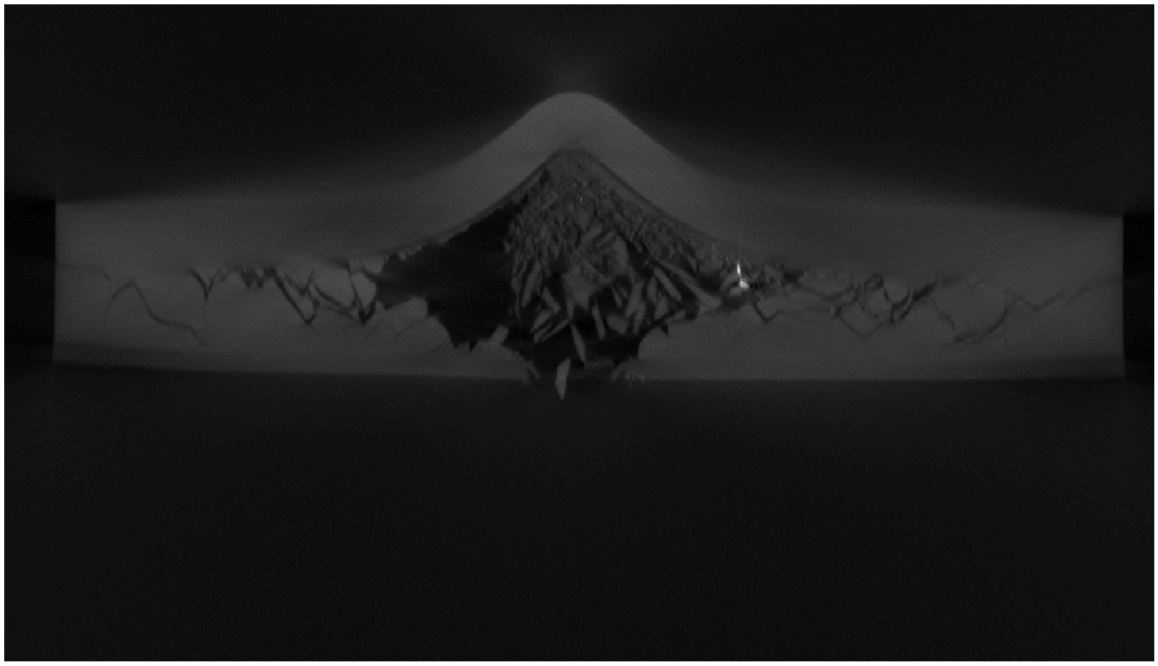


Figure 5.2: Slice 6_1400 of image "17-0789-4". Each slice is a 459×1814 grayscale image taken with a specialized CAT scan machine. A brief note on slice-naming convention: "6" refers to the image object, and "1400" refers to the slice number, 0 being the "bottom" of the material and 1999 the "top.".

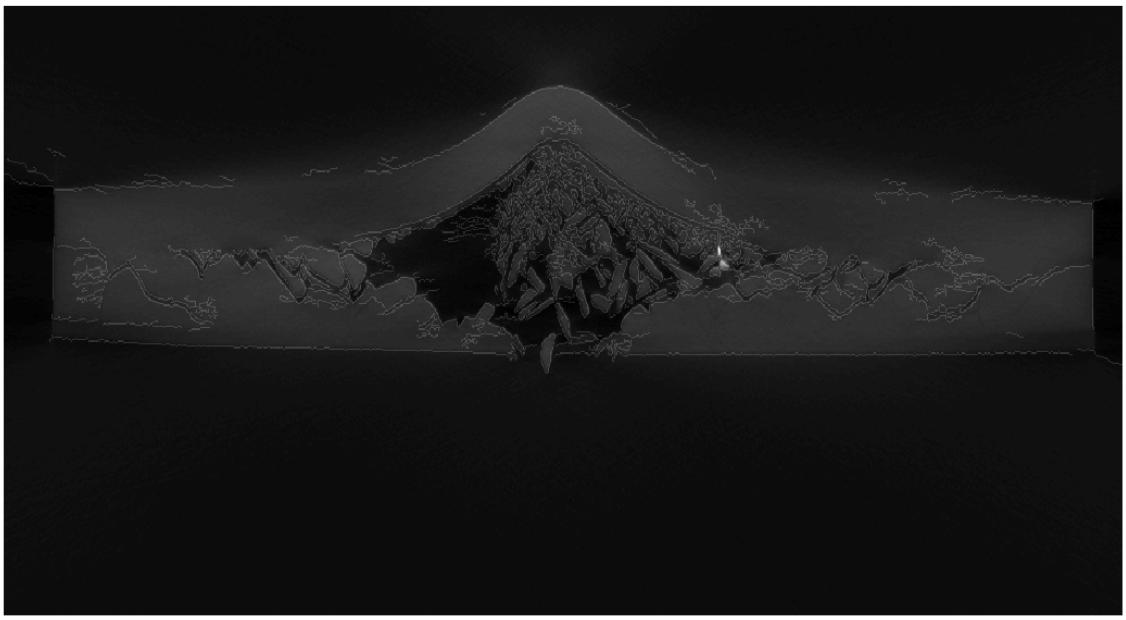


Figure 5.3: Slice 6_1400 with edges detected using canny edge detector.

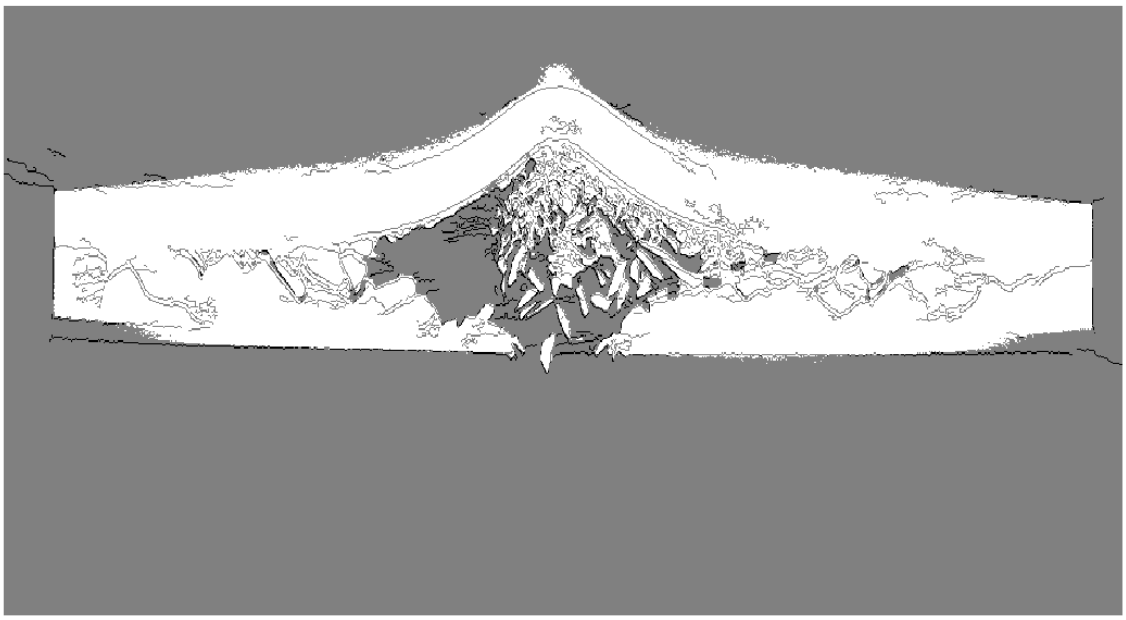


Figure 5.4: Slice 6_1400 with labels.

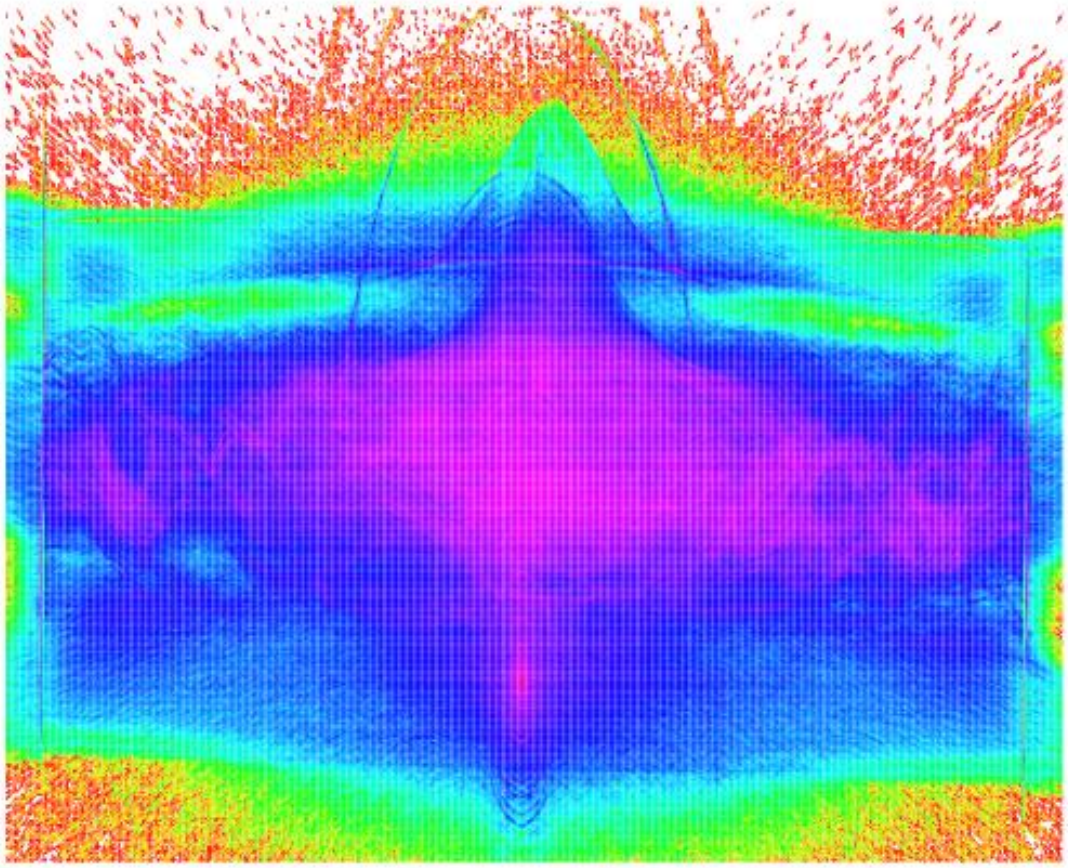


Figure 5.5: Entropy-projection of all 2000 slices of Image 6_.

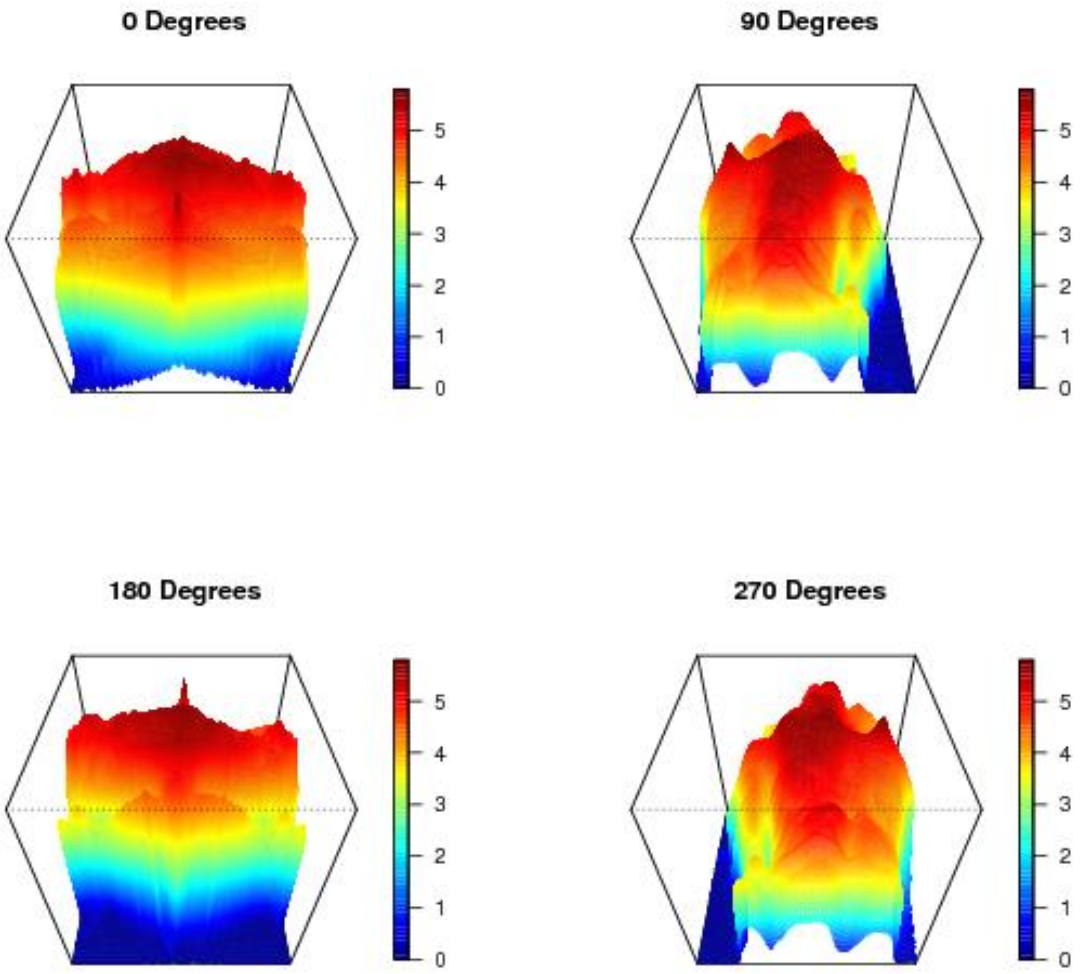


Figure 5.6: Perspective plot of flattened Image 6_. z-axis represents entropy, x and y axes represent the coordinates of the two dimensional flattened image.

Figure 5.7: Thresholded inhomogenous poisson process representing Figure 5.5 as a point process.

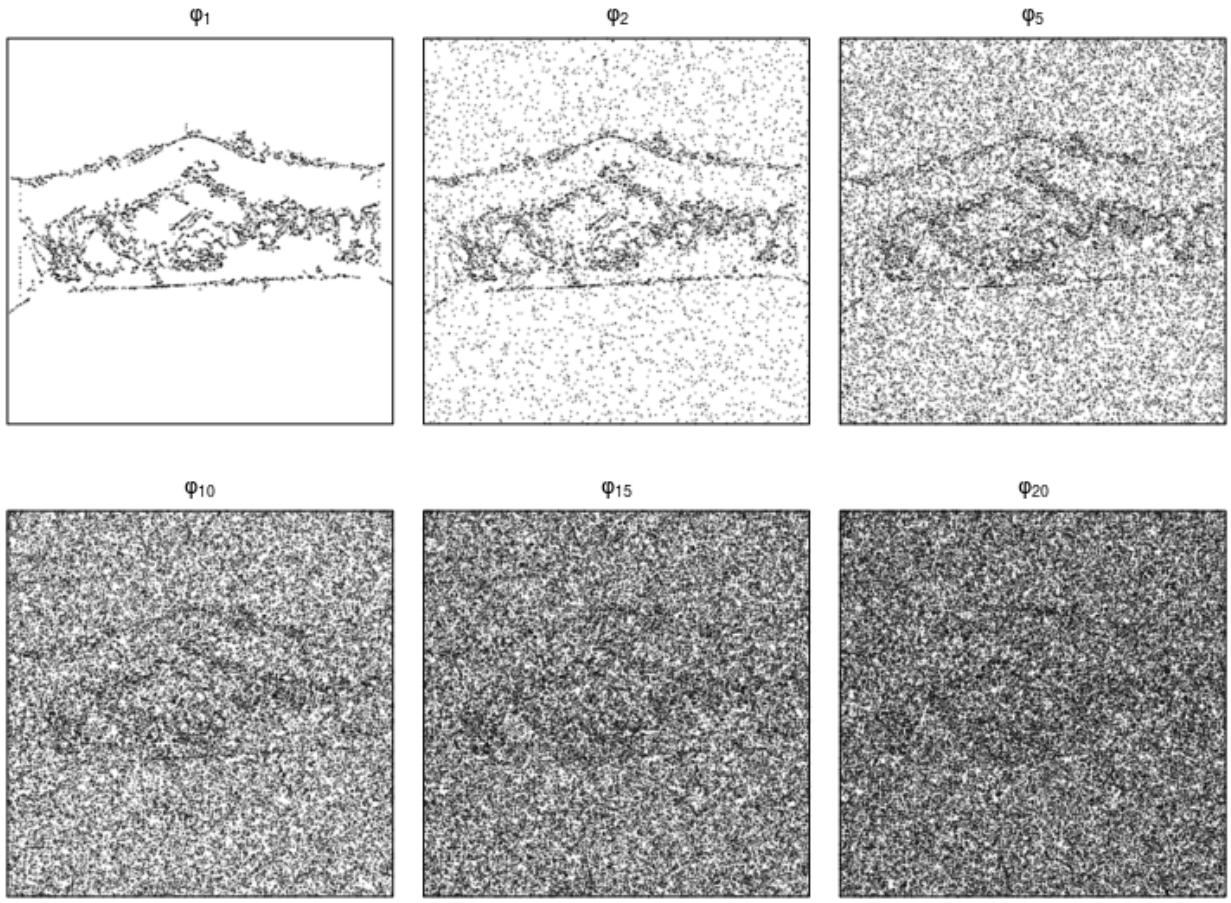


Figure 5.8: Top left: slice 6_1400 of the point process approximation of labeled image 17-0789-4 (compare to Figure 5.4). Top middle to bottom right: Iterative superpositions of an appropriately scaled homogeneous Poisson process.

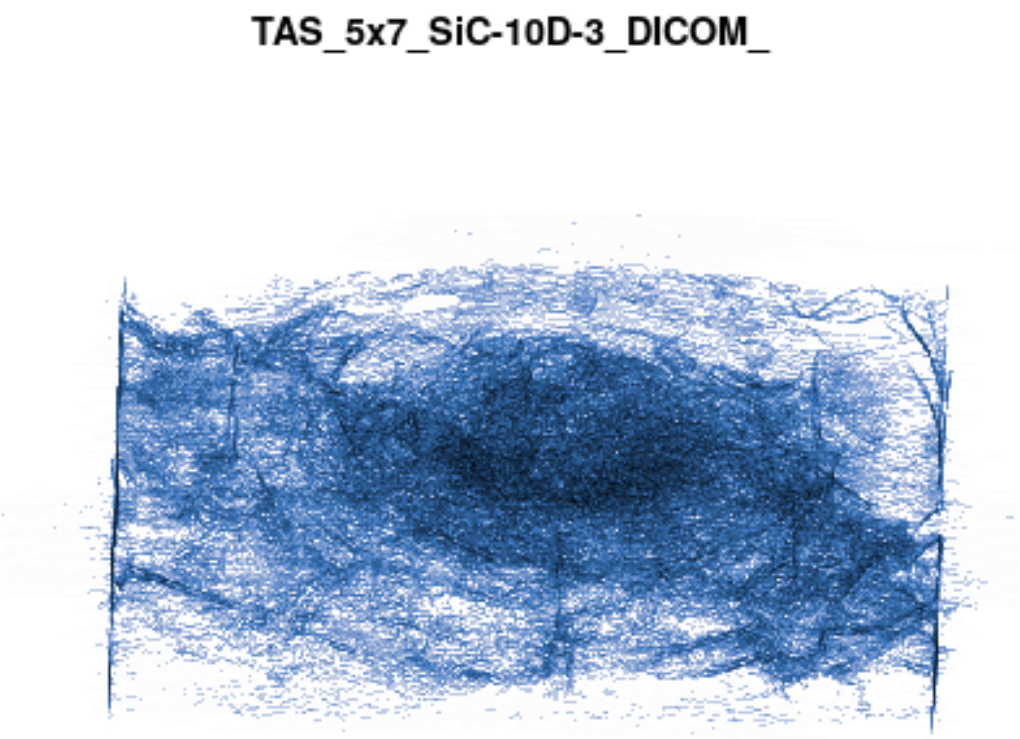


Figure 5.9: Damage plot of image TAS_5x7_SiC10D_3_DICOM.

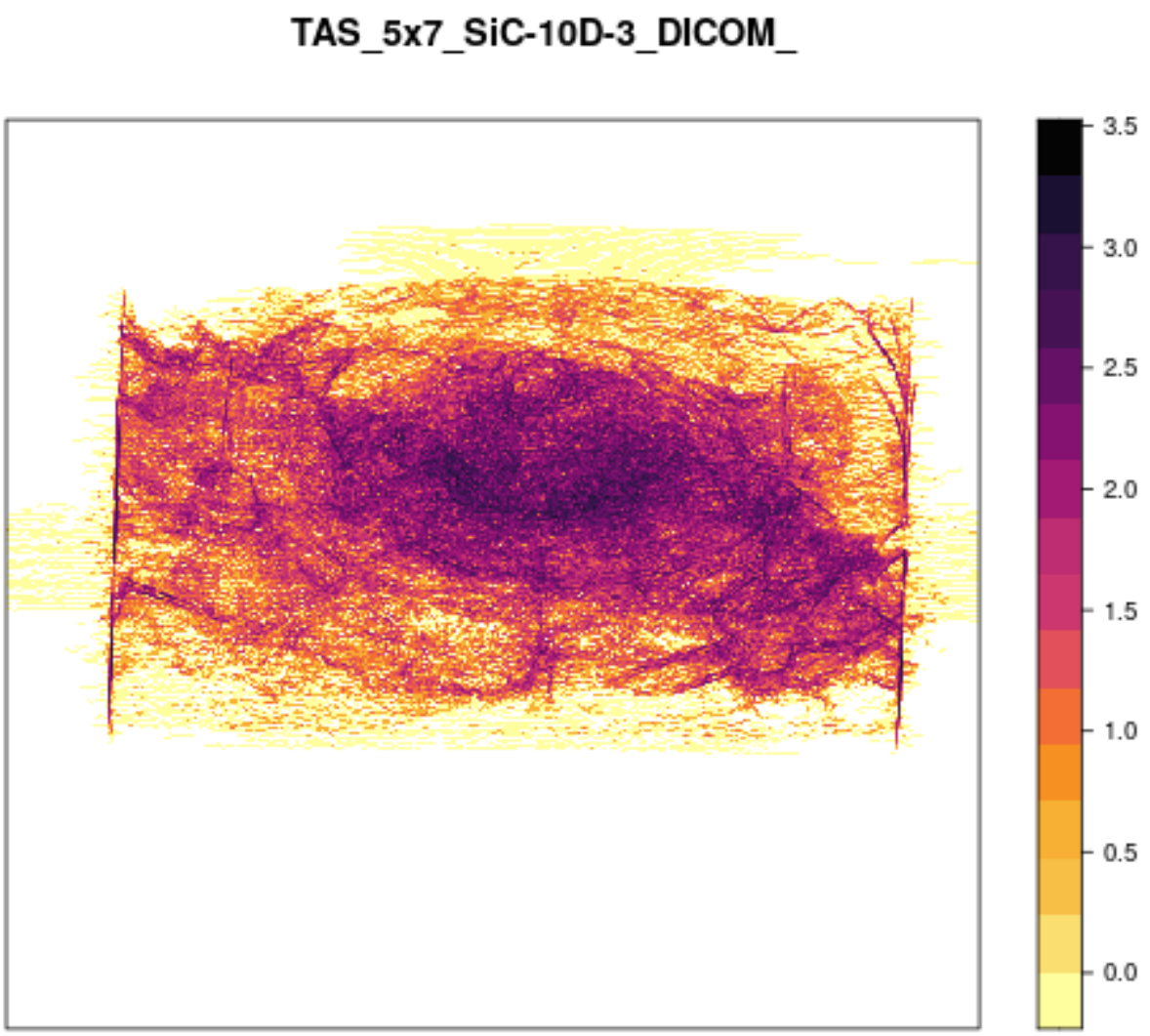


Figure 5.10: Levelplot of the flattened entropy projection of TAS_5x7_SiC10D_3_DICOM. This image has an average pixel entropy value of 1.2083, indicating a relatively high level of damage.

5.3 Addendum: Tensor Notation

Tensors are $d-$dimensional arrays, and can be thought of as a generalization of the progression from scalars to vectors and vectors to matrices. They are a convenient and natural mathematical representation for many data types arising in fields such as image processing and computer vision, signal processing, linguistics, geology and physical sciences, and topic modelling.

Tensor objects can be approached as a class which includes the higher-dimensional generalizations of the subclasses of scalars, vectors and matrices. The dimension of a tensor is referred to as the tensor order or mode throughout the literature. Tensors of order zero are scalars, tensors of order one are vectors, and tensors of order two are matrices. Practically, we can think of the order of a tensor as the minimum number of indices required to represent a higher dimensional array. If this description is intuitively unappealing, we can equivalently say that tensors of order d are generalized matrices with dimension d. Finally if this description is also unsatisfactory, we can vaguely say that tensors are array objects with order equal to their spatial dimension. Tensors of order three and greater are of primary interest throughout this paper as they possess "generalized" properties of vectors and matrices.

Most statisticians have inadvertently created and manipulated tensor objects of order three when creating lists of matrices. Tensors of order three, as opposed to a list of matrices, encapsulate the higher dimensional relationship between matrices, analogous to a matrix capturing the intra-relation between vectors, as opposed to a list of vectors. It is often valuable to operate within the tensor framework as doing so paves the way for data compression and analysis via tensor decompositions such as the Canonical Polyadic (CP) and Tucker Decompositions. Complexity-reducing decompositions are often a necessity from a computing standpoint: a tensor of order d has n^d elements, which through decomposition can be represented with $\mathcal{O}(nd)$ complexity, given certain conditions.

5.3.1 Basic Tensor Notation

A tensor has of order d has d fibers, which are defined as the subarrays or "subtensors" with one index fixed. If $d = 3$, the fibers of the tensor are columns, tubes, or rows. Similarly, a slice is defined as the subtensors where two indices are fixed, all others free. If $d = 3$, the slices of a order d tensor are matrices.

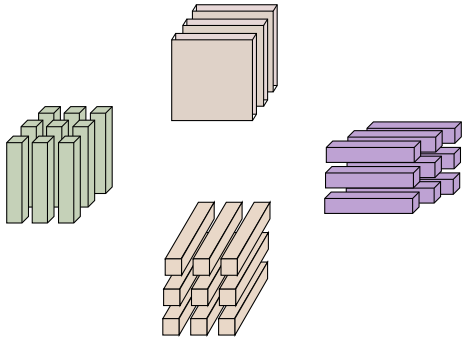


Figure 5.11: (clockwise from top) frontal slices, row fibers, tube fibers, and column fibers of a tensor of order three.

5.3.2 Outer Product Representation of Tensors

As discussed above, a tensor of order d has indices $\{1, \ldots, d\}$. For each index $i \in \{1, \ldots, d\}$, we have a sub-index set $I_i = \{1, \ldots n_i\}$, and the Cartesian product $I = I_1 \times \ldots \times I_d$. Therefore a tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times \ldots \times I_d}$ has order d and elements $x_{i_1}, \ldots, x_{i_d}$. Drawing a connection with a more familiar linear algebra representation, we note that $\mathbb{R}^I$ is equivalent to the vector space

$$\{x = (x_i)_{i \in I} \quad \ni \quad x_i \in \mathbb{R}\}$$

where we can the order of a tensor in this space as the cardinality of I. Indeed, we can see that for $\mathcal{X}, \mathcal{Y} \in \mathbb{R}^I$, $\alpha\mathcal{X} \in \mathbb{R}^I$ for $\alpha \in \mathbb{R}$, $0 \in \mathbb{R}^I$, and $\mathcal{X} + \mathcal{Y}$ commutes. Therefore, we can see that the set of tensors over the reals is in fact a vector space, and that the many wonderful properties of vector spaces apply.

This vector space representation of a tensor can be explicitly notated as the outer product

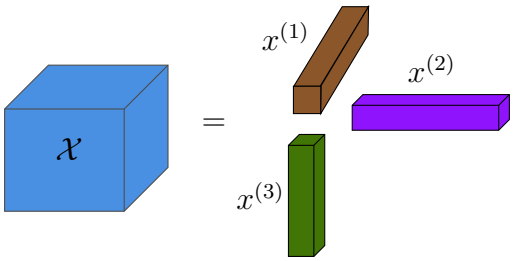
or tensor product $\mathcal{X} := x^{(1)} \otimes \ldots \otimes x^{(d)}$ or more compactly as

$$\mathcal{X} := \bigotimes_{i=1}^{d} x^{(i)}.$$

Further, we can define the tensor space $\mathcal{T}$ as the tensor product of indices over the reals, *i.e.*

$$\mathcal{T} = \bigotimes_{i=1}^{d} \mathbb{R}^{I_i} = \text{span}\{x^{(1)} \otimes \ldots \otimes x^{(d)} \quad \ni \quad x^{(i)} \in \mathbb{R}^{I_i}\} = \mathbb{R}^{I}.$$

r0.5



Using this notation, we define

$$x^{(1)} \otimes \ldots \otimes x^{(d)}$$

as rank one or elementary tensor. Rank one tensors can be thought of as the DNA, or "atomic" building blocks, of all tensor objects. Abusing notation, we can analogize rank one tensors as the baseis for the larger tensor space $\mathcal{T}$. Similarly, we can think of rank one tensors as analogous to prime numbers, which by the fundamental theorem of algebra make up all natural numbers. Crucially, not all tensors in $\mathcal{X} \in \mathcal{T}$ can be written as an rank one tensor, but all can be represented as a multilinear combination (sum) of rank one tensors in $\mathcal{T}$. If $x^{(i)} \in \mathbb{R}^{I_1 \times \ldots \times I_d} = \mathbb{R}^{I}$ for $i = 1, ..., d$, and

$$\mathcal{X} = x^{(1)} \otimes x^{(2)} \otimes ... \otimes x^{(d)},$$

where $\mathcal{X}_{i_1,\ldots,i_d} = x^{(1)}_{i_1} \cdot x^{(2)}_{i_2} ... x^{(d)}_{i_d}$, then we say that $\mathcal{X}$ is a *rank-1 tensor*.

Stepping back, we explicitly define the outer product operator, as it is the crucial operator for the below mentioned tensor decompositions. The outer product of two vectors v, w is a

matrix M such that $\text{rank}(M) = 1 \; \forall v, w$. Explicitly,

$$v \otimes w = vw^\top = \begin{bmatrix} v_1 w_1 & v_1 w_2 & \cdots & v_1 w_m \\ v_2 w_1 & v_2 w_2 & \cdots & v_2 w_m \\ \vdots & \vdots & \ddots & \vdots \\ v_n w_1 & v_n w_2 & \cdots & v_n w_m \end{bmatrix}$$

which is obvious of rank equal to one. This follows immediately from the fact that

$$\text{rank}(vw^\top) \leq \min\{\text{rank}(v), \text{rank}(w^\top)\} \leq 1.$$

To ease the transition between these the rank one matrix and outerproduct representation, it can be shown that there is an isomorphism between matrices and tensors of order two, see Hackbusch (2012).

Expanding upon this example, we can say that a tensor is the tensor product (identical to the outer product, save for context) for two vector spaces V, W, with basis

$$v = \begin{bmatrix} v_1 & v_2 & \cdots & v_n \end{bmatrix}^\top \quad \text{and} \quad w = \begin{bmatrix} w_1 & w_2 & \cdots & w_m \end{bmatrix}^\top.$$

A tensor, then, is by definition a map $T(v, w) = vTw^\top$ and a rank one tensor formed from v, w has the form $vw^\top$, where the multilinear transformation T is the identity.

In conclusion, the tensor product operator allows us to extend the notion of underlying vector spaces to tensor objects, using a multilinear transformation. With the tensor product representation, we have introduced the concept of rank one tensors, which form the underlying building blocks for tensors. Crucially, all tensors can be expressed as a multilinear combination of rank one tensors.

5.3.3 Rank Extension, the Frobenius Analog, and Low Rank Approximation

A natural question arises after the above description of tensor products and rank one tensors: what is a rank r tensor? A rank r tensor is defined as a tensor which is the multilinear

combination (sum) of at least r rank one tensors. This definition illuminates the fact that rank one tensors are the higher dimensional generalization of vector baseis in linear algebra. Formally, a tensor $\mathcal{X}$ of rank r and order d can be expressed as

$$\mathcal{X} = \underbrace{\sum_{i=1}^{r}}_{\substack{\text{multilinear} \\ \text{combination}}} \underbrace{\bigotimes_{\nu=1}^{d} x_{\nu}^{(i)}}_{\substack{\text{rank one} \\ \text{tensors}}}$$

where $x_{\nu}^{(i)} \in X^{(i)}$, and $X^{(i)}$ is the corresponding vector space for $x^{(i)}$. Obviously, a tensor can also be expressed by any larger representation rank tensor (by adding zeros, or terms which telescope out).

The above definition of tensor rank naturally appeals to the notion of a low rank approximation to tensor $\mathcal{X}$, i.e. finding a tensor $\mathcal{Y}$ of rank $k < r$ such that $||\mathcal{X} - \mathcal{Y}||$ is minimized. To approach this problem, we define the tensor analog for the Frobenius norm:

$$||\mathcal{X}||_F = \left(\sum_{i \in I} x_{i_1 \dots i_d}^2 \right)^{\frac{1}{2}} = \left(\sum_{i_1=1}^{I_1} \cdots \sum_{i_d=1}^{I_d} x_{i_1 \dots i_d}^2 \right)^{\frac{1}{2}}.$$

Unfortunately this problem is ill-posed, as discussed in De Silva and Lim (2008). Unlike the singular value decomposition in linear algebra, in which the best rank $r+1$ approximation of a matrix could be found by using the best rank r approximation, plus some rank one matrix, it is not generally true that the best rank $r+1$ approximation of a tensor is the best rank r approximation, plus some rank one tensor. Further, the best rank $r+1$ tensor approximation (as found by a CP decomposition, discussed below) is not guaranteed to be strictly better than (or for that matter worse or the same) than the best rank r approximation.

Given that tensors of higher order often require large amounts of memory, and operations on such tensors are computationally intensive if not intractible, it is important to utilize tensor decompositions which allow for reduction in complexity. Analogous to a matrix rank decomposition, the Canonical Polyadic (CP) decomposition factorizes a tensor into a multilinear combination (*i.e.* sum) of rank one tensors. For instance, a tensor of order

three

$$\mathcal{X} = \sum_{i=1}^{r} \bigotimes_{\nu=1}^{3} x_{\nu}^{(i)}$$

has the follow rank r CP decomposition:

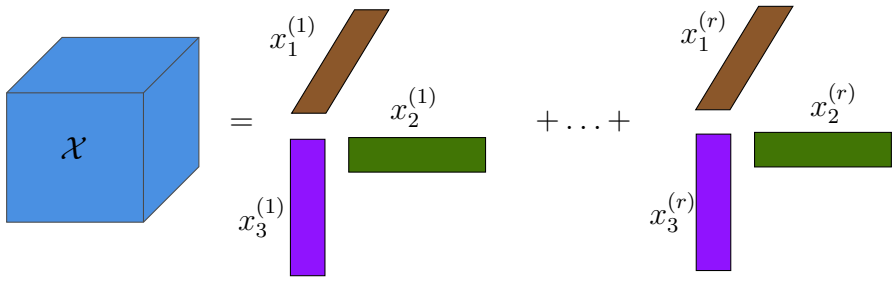


Figure 5.12: a tensor $\mathcal{X}$ of order three, decomposed into r rank one tensors.

In practice, determining r is NP-hard problem (see Hillar and Lim (2013)), and as discussed above, the best rank k approximation is not guaranteed to be better than $k-1$ approximation. In fact, best rank-k approximations may not always exist. Such tensors are termed degenerate. If r is unknown, the equality in Figure 5.12 should be replaced with $\approx$. An exact CP decomposition of a d-order tensor $\mathcal{X}$ such that $\text{rank}(\mathcal{X}) = r$ can be expressed as $\mathcal{X} = \sum_{i=1}^{r} \bigotimes_{\nu=1}^{d} x_{\nu}^{(i)}$, or more commonly, as $\mathcal{X} = \sum_{i=1}^{r} \lambda_i \bigotimes_{\nu=1}^{d} x_{\nu}^{(i)}$ where the $X^{(i)}$'s have been normalized to length one ($\lambda \in \mathbb{R}^r$).

We define $X_{(j)} = [x_j^{(1)} \cdots x_j^{(r)}]$ as the factor matrices for each $j \in \{1, \ldots, d\}$. Referencing Figure 5.12, we can say that the $X_{(2)}$ factor matrix would be have columns equal to the pink vectors, $X_{(2)}$ would be have columns equal to the blue vectors, and $X_{(3)}$ would be have columns equal to the purple vectors. We compactly notate

$$\mathcal{X} = [[\lambda; X_{(1)}, \ldots, X_{(d)}]]$$

which is termed throughout the literature as the Kruskal form.

Matrix decompositions are not unique (without constraints), but CP decompositions of tensors often are. If the $k-$rank (defined as the maximum k s.t. any k columns are linearly independent) of the factor matrices achieves the following sufficient condition, we know our

CP decomposition of rank r is unique:

$$\sum_{i=1}^{d} \mathcal{K}(X_{(i)}) \leq 2r + (d-1)$$

where $\mathcal{K} : \boldsymbol{X} \mapsto k$ is the function that returns the $k-$rank of matrix $\boldsymbol{X}$.

5.4 Acknowledgements

Thank you to Shane Bartus and his colleagues at the Army Research Office who have provided the tomographic images discussed in this chapter. Further thanks to Frederic Schoenberg who has contributed many ideas to this chapter. Some of the content in Section 5.3 was also submitted to Oscar Padilla in a paper written with Melody Huang, although all content was originally presented in Section 5.3 was solely authored by Conor Kresin.

Image	Mean	Squared Sum	Variance	Maximum	X-Dim	Y-Dim	Z-Dim
TAS_5x7_B4C_10D-1_DICOM	1.2717	308831.7604	0.8326	3.4657	1273	1202	1517
TAS_5x7_B4C_10D-2_DICOM	1.2465	309521.7814	0.9091	3.4340	1268	1172	1496
TAS_5x7_B4C_10D-4_Top	1.1791	275083.4999	0.9104	3.4012	1189	1301	1519
TAS_5x7_B4C_10D-5_DICOM	1.2706	294517.4122	0.8200	3.3673	1282	1159	1506
TAS_5x7_B4C_10D-6_DICOM	1.2276	294210.0818	0.8682	3.3673	1319	1215	1504
TAS_5x7_SiC-10D-1_DICOM	1.2501	286855.1963	0.7700	3.2958	1168	1304	1479
TAS_5x7_SiC-6D-1_DICOM	0.8355	109455.3097	0.4613	3.2189	1286	1129	1474
TAS_5x7_SiC-6D-2_Top	1.2251	284519.1009	0.7701	3.3322	1302	1155	1460
TAS_5x7_SiC-6D-3_DICOM	1.2042	253087.7758	0.7327	3.3322	1284	1209	1458
TAS_5x7_SiC-6D-4_DICOM	1.2103	256687.7021	0.7275	3.1781	1338	1199	1449
TAS_5x7_SiC-HB8-10D-2_DICOM	0.9342	141329.3633	0.5709	3.1781	1310	1151	1615
TAS_5x7_SiC-HB8-6D-4_DICOM	0.6113	62359.3976	0.4504	3.1781	1368	1228	1678
TAS_5x7_SiC-HB80-10D-5	0.8629	116436.9566	0.6571	3.3673	1118	1298	1615
TAS_5x7_SiC-HB8-6D-2_DICOM	0.7361	75713.0710	0.5884	3.5553	1304	1161	1718
TAS_5x7_SiC-HB8-6D-3_DICOM	0.9141	149154.1019	0.6569	3.2581	1299	1194	1661
TAS_5x7_SiC-HB8-6D-5_DICOM	0.8755	94622.8471	0.6536	3.1781	1294	1119	1526
TAS_5x7_SiC-10D-2_DICOM	1.2726	293903.5913	0.7216	3.2189	1266	1172	1450
TAS_5x7_SiC-10D-3_DICOM	1.2073	223611.2049	0.7099	3.2958	1289	1202	1526
TAS_5x7_SiC-10D-4_DICOM	1.2669	270607.6889	0.7025	3.2189	1268	1220	1522
TAS_5x7_SiC-10D-5_DICOM	1.1365	226341.8570	0.7447	3.2189	1316	1228	1512
TAS_5x7_SiC-6D-5_DICOM	1.2083	166873.1911	0.5746	3.1781	1309	1281	1603
TAS_5x7_SiC-HB80-10D-1_DICOM	0.9272	125924.2789	0.7192	3.2581	1285	1151	1582
TAS_5x7_SiC-HB80-10D-3_DICOM	0.7524	89638.1580	0.5506	3.2189	1311	1251	1791
TAS_5x7_SiC-HB80_6D-1_DICOM	0.9569	201393.7600	0.5307	3.4965	1372	1216	1927
TAS_6x10_B4C-10D-1_DICOM	1.2217	218548.9832	0.7942	4.2341	1684	1846	1807
TAS_6x10_B4C-10D-2_DICOM	1.3381	238789.5118	0.7266	4.7707	1534	1834	1772
TAS_6x10_B4C-10D-3_DICOM	1.2709	252697.3287	0.7134	3.9703	1501	1805	1697
TAS_6x10_B4C_10D-4_DICOM	1.3319	262375.8576	0.7583	4.0775	1496	1778	1686
TAS_6x10_B4C-10D-5_DICOM	1.2485	215223.8321	0.8514	4.6634	1516	1830	1780
TAS_6x10_B4C-15D-1_DICOM	1.2264	223967.0777	0.7826	4.3944	1512	1899	1838
TAS_6x10_B4C-15D-2_DICOM	1.2170	215029.5090	0.7616	4.2485	1505	1899	1725
TAS_6x10_B4C-15D-3_DICOM	1.1820	212881.4869	0.7246	4.1897	1525	1894	1735
TAS_6x10_B4C-15D-4_DICOM	1.1490	179095.1215	0.7110	4.4886	1506	1866	1726
TAS_6x10_B4C-15D-5_DICOM	1.1406	187337.5070	0.7082	4.3438	1515	1845	1719
TAS_6x10_B4C-20D-1_DICOM	1.1207	166895.8585	0.7695	4.5218	1646	1971	1800
TAS_6x10_B4C-20D-2_DICOM	1.2045	206768.4530	0.7685	4.1589	1500	1930	1797
TAS_6x10_B4C-20D-3_DICOM	1.1588	184323.0376	0.7504	4.3944	1505	1859	1751
TAS_6x10_B4C-20D-4_DICOM	1.2813	222837.2316	0.7811	4.3438	1516	1834	1720
TAS_6x10_B4C-20D-5_DICOM	1.3989	307085.0710	0.8880	4.5109	1512	1872	1910

Table 5.1: Statistical summary of the 39 provided images. Summary statistics provided are average, squared sum, variance and maximum of flattened entropy pixel values. The X, Y and Z-dimensions are the width and height of each slice, and the number of slices, respectively.

Image Name	$\bar{e}$	**X-Dim**	**Y-Dim**	**Z-Dim**	**Voxels**	**Ceramic Size**	V	**Optimal** ϵ^*	$s\|\epsilon^*$
TAS_5x7_SiC-HB8-6D-4_DICOM	0.6113	1368	1228	1678	2.8189E+09	5x7	8.05E+07	0.00805	13
TAS_5x7_SiC-HB8-6D-5_DICOM	0.8755	1294	1119	1526	2.2096E+09	5x7	6.31E+07	0.00631	36
TAS_5x7_SiC-HB80-10D-5	0.8629	1118	1298	1615	2.3436E+09	5x7	6.70E+07	0.00670	37
TAS_5x7_SiC-6D-1_DICOM	0.8355	1286	1129	1474	2.1401E+09	5x7	6.11E+07	0.00612	46
TAS_5x7_SiC-HB8-6D-3_DICOM	0.9141	1299	1194	1661	2.5762E+09	5x7	7.36E+07	0.00736	49
TAS_6x10_B4C-20D-3_DICOM	1.1588	1505	1859	1751	4.8989E+09	6x10	8.16E+07	0.00817	62
TAS_5x7_B4C_10D-6_DICOM	1.2276	1319	1215	1504	2.4103E+09	5x7	6.89E+07	0.00689	62
TAS_5x7_SiC-HB80-10D-3_DICOM	0.7524	1311	1251	1791	2.9373E+09	5x7	8.39E+07	0.00839	70
TAS_5x7_SiC-HB8-6D-2_DICOM	0.7361	1304	1161	1718	2.6010E+09	5x7	7.43E+07	0.00743	73
TAS_5x7_SiC-10D-4_DICOM	1.2669	1268	1220	1522	2.3545E+09	5x7	6.73E+07	0.00673	78
TAS_5x7_B4C_10D-2_DICOM	1.2465	1268	1172	1496	2.2232E+09	5x7	6.35E+07	0.00635	79
TAS_5x7_SiC-HB8-10D-2_DICOM	0.9342	1310	1151	1615	2.4351E+09	5x7	6.96E+07	0.00696	81
TAS_6x10_B4C-10D-2_DICOM	1.3381	1534	1834	1772	4.9853E+09	6x10	8.31E+07	0.00831	81
TAS_5x7_SiC-6D-5_DICOM	1.2083	1309	1281	1603	2.6880E+09	5x7	7.68E+07	0.00768	88
TAS_6x10_B4C-20D-4_DICOM	1.2813	1516	1834	1720	4.7822E+09	6x10	7.97E+07	0.00797	88
TAS_5x7_SiC-6D-3_DICOM	1.2042	1284	1209	1458	2.2633E+09	5x7	6.47E+07	0.00647	89
TAS_6x10_B4C-15D-2_DICOM	1.217	1505	1899	1725	4.9300E+09	6x10	8.22E+07	0.00822	90
TAS_6x10_B4C-15D-1_DICOM	1.2264	1512	1899	1838	5.2774E+09	6x10	8.80E+07	0.00880	91
TAS_6x10_B4C-20D-5_DICOM	1.3989	1512	1872	1910	5.4062E+09	6x10	9.01E+07	0.00901	91
TAS_6x10_B4C-10D-5_DICOM	1.2485	1516	1830	1780	4.9382E+09	6x10	8.23E+07	0.00823	92
TAS_6x10_B4C-15D-3_DICOM	1.182	1525	1894	1735	5.0113E+09	6x10	8.35E+07	0.00835	94
TAS_5x7_SiC-HB80-10D-1_DICOM	0.9272	1285	1151	1582	2.3398E+09	5x7	6.69E+07	0.00669	95
TAS_5x7_SiC-HB80_6D-1_DICOM	0.9569	1372	1216	1927	3.2149E+09	5x7	9.19E+07	0.00919	95
TAS_5x7_SiC-10D-5_DICOM	1.1365	1316	1228	1512	2.4435E+09	5x7	6.98E+07	0.00698	107
TAS_6x10_B4C-15D-4_DICOM	1.149	1506	1866	1726	4.8504E+09	6x10	8.08E+07	0.00808	107
TAS_6x10_B4C-20D-2_DICOM	1.2045	1500	1930	1797	5.2023E+09	6x10	8.67E+07	0.00867	107
TAS_6x10_B4C-20D-1_DICOM	1.1207	1646	1971	1800	5.8397E+09	6x10	9.73E+07	0.00973	109
TAS_5x7_SiC-6D-4_DICOM	1.2103	1338	1199	1449	2.3246E+09	5x7	6.64E+07	0.00664	110
TAS_6x10_B4C-15D-5_DICOM	1.1406	1515	1845	1719	4.8049E+09	6x10	8.01E+07	0.00801	111
TAS_6x10_B4C-10D-1_DICOM	1.2217	1684	1846	1807	5.6174E+09	6x10	9.36E+07	0.00936	114
TAS_5x7_B4C_10D-4_Top	1.1791	1189	1301	1519	2.3497E+09	5x7	6.71E+07	0.00671	118
TAS_5x7_SiC-10D-3_DICOM	1.2073	1289	1202	1526	2.3644E+09	5x7	6.76E+07	0.00676	119
TAS_5x7_B4C_10D-5_DICOM	1.2706	1282	1159	1506	2.2377E+09	5x7	6.39E+07	0.00639	127
TAS_6x10_B4C-10D-3_DICOM	1.2709	1501	1805	1697	4.5977E+09	6x10	7.66E+07	0.00766	129
TAS_6x10_B4C_10D-4_DICOM	1.3319	1496	1778	1686	4.4846E+09	6x10	7.47E+07	0.00747	130
TAS_5x7_B4C_10D-1_DICOM	1.2717	1273	1202	1517	2.3212E+09	5x7	6.63E+07	0.00663	131
TAS_5x7_SiC-10D-2_DICOM	1.2726	1266	1172	1450	2.1514E+09	5x7	6.15E+07	0.00615	131
TAS_5x7_SiC-6D-2_Top	1.2251	1302	1155	1460	2.1956E+09	5x7	6.27E+07	0.00627	134
TAS_5x7_SiC-10D-1_DICOM	1.2501	1168	1304	1479	2.2526E+09	5x7	6.44E+07	0.00644	138

Table 5.2: From left to right: $\bar{e}$ denotes the mean flattened entropy pixel value, Voxels represents the total number of voxels (elements) in a given tomographic image, Ceramic Size is the size of the silicon material in inches, V is the number of voxels per square inch, ϵ^* is the optimal stopping criteria threshold given V, and $x|\epsilon^*$ is the number of iterative superpositions to reach convergence to homogeneous Poisson. s is the complexity number, and an alternative statistic to $\bar{e}$ for estimating damage.

CHAPTER 6

Concluding Remarks

Many of the canonical datasets in the point process literature are necessarily small due to data collection limitations, or model fitting restrictions. For instance, perhaps the most famous point process data set, `finpines`, is comprised of observations of trees which required the data collector to walk around in a forest, manually measuring height and trunk circumference statistics for each tree (Stoyan and Penttinen, 2000).

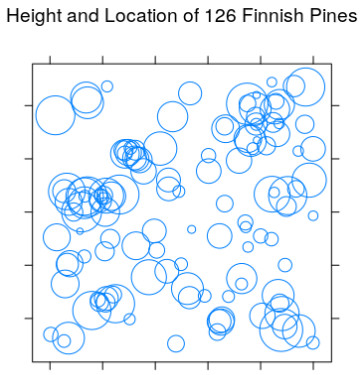


Figure 6.1: A canonical point process dataset (Stoyan and Penttinen, 2000). Trees in 10m $\times$ 10m window. Circles represent trees with radius proportional to tree height. Accessed via the `spatstat` package (Baddeley et al., 2004).

In the so-called age of big data, many point process datasets can be found embedded in location tracking data, satellite imagery data, etc, and we believe that point processes offer a valuable option as a highly compressed mathematical representation of data. There are important connections between point processes and information theory and causality that are underdeveloped in the current literature. Computationally efficient estimators like the SG estimator allow us to model large data and complex probabilistic structures while leveraging the rich theoretical framework of point processes. Ultimately, we want to model the forest - not the trees.

CHAPTER 7

Appendix

7.1 Code

7.1.1 Causal Simulation Study

The below code simulates our causal study with spatial inhomogenous Poisson processes.

```
library(spatstat)
library(tidyverse)
library(parallel)

statespaceOmega<-c(0,10,0,10)
partitionsOmega<-quadrats(X=statespaceOmega, nx = statespaceOmega[2])

condInten1<-function(x,y) {1.5*x+0.55*y+2}
TE<-5
condInten2<-function(x,y) {1.5*x+0.55*y+2+TE}

run_iter<-function(i, statespace, partitions, partitionsDynamic,
    condInten_treatment, condInten_control){
  simProcess1<-rpoispp(condInten1,win = statespace)
  simProcess2<-rpoispp(condInten2,win = statespace)

  if(partitionsDynamic==T){
    superProcess<-superimpose(simProcess1,simProcess2)
    dataTess<-sample(1:superProcess$n,size=superProcess$n/(TE*statespace[2]))
    partitions<-dirichlet(superProcess[dataTess])
  }
  treatprob<-sample(seq(0.1,0.9,length.out = 9),size=1)
  treatmentInd<-sample(c(0,1), replace=TRUE, prob=c(treatprob,1-treatprob),
      size=partitions$n)
```

```
tilesControl<-cut(simProcess1,partitions)
pointTilesControl<-as.numeric(tilesControl$marks)
marksTilesControl<-c()
for(i in 1:length(pointTilesControl)){
  marksTilesControl[i]<-pointTilesControl[i]
  *(1-treatmentInd[pointTilesControl[i]])
}
marks(simProcess1)<-marksTilesControl
observedProcess1<-subset(simProcess1,marks!=0)

tilesTreatment<-cut(simProcess2,partitions)
pointTilesTreatment<-as.numeric(tilesTreatment$marks)
marksTilesTreatment<-c()
for(i in 1:length(pointTilesTreatment)){
  marksTilesTreatment[i]<-pointTilesTreatment[i]*
  treatmentInd[pointTilesTreatment[i]]
}
marks(simProcess2)<-marksTilesTreatment
observedProcess2<-subset(simProcess2,marks!=0)


statespaceControl<-as.owin(partitions[marksTilesControl])
statespaceTreatment<-as.owin(partitions[marksTilesTreatment])
process1<-ppm(unmark(observedProcess1),~condInten_control,
clipwin=statespaceControl)
process2<-ppm(unmark(observedProcess2),~condInten_treatment,
clipwin=statespaceTreatment)
simfitted1<-rmh(process1,w=statespaceTreatment)
```

```
  simfitted2<-rmh(process2,w=statespaceControl)
  obPlusSim1<-cut(superimpose(unmark(observedProcess1),simfitted1),
  partitions)
  obPlusSim2<-cut(superimpose(unmark(observedProcess2),simfitted2),
  partitions)

  tileCountsSate<-c()
  for(i in 1:partitions$n){
    tileCountsSate[i]<-sum(as.numeric(obPlusSim2$marks)==i)-
    sum(as.numeric(obPlusSim1$marks)==i)
  }
  SATE<-(partitions$n/((statespace[2]-statespace[1])*
  (statespace[4]-statespace[3])))
        *mean(tileCountsSate)

  tileCounts<-c()
  for(i in 1:partitions$n){
    tileCounts[i]<-sum(pointTilesTreatment==i)-sum(pointTilesControl==i)
  }
  ATE<-(partitions$n/((statespace[2]-statespace[1])*(statespace[4]-
  statespace[3])))
        *mean(tileCounts)

  errorTE<-ATE-SATE
  return(data.frame(ATE, SATE,errorTE))
}

set.seed(0)
results<-mclapply(1:10^5, run_iter, statespace=statespaceOmega,
```

```
    partitions=partitionsOmega, partitionsDynamic=F,
    condInten_treatment=condInten2, condInten_control=condInten1,
    mc.cores=detectCores())
  %>% bind_rows()
```

The below code simulates a Hawkes process via thinning Ogata (1981) in the function `simulate_uni_hawkes()`, allows for a Hawkes process to be fitted via MLE in the function `loglhawkLewis()`, and then conducts the simulation study.

```
simulate_uni_hawkes <- function(mu, alpha, beta, t_max) {
  arrivals <- c()
  s <- 0
  t <- 0
  lambda_star <- mu
  s <- s - log(runif(1)) / lambda_star
  t <- s
  dlambda <- alpha
  arrivals <- c(arrivals, t)
  while (s < t_max) {
    U <- runif(1)
    s <- s - log(U) / lambda_star
    u <- runif(1)
    if (u <= (mu + dlambda * exp(-beta * (s - t))) / lambda_star) {
      dlambda <- alpha + dlambda * exp(-beta * (s - t))
      lambda_star <- lambda_star + alpha
      t <- s
      arrivals <- c(arrivals, t)
    }
  }
```

```
  return(arrivals)
}

loglhawkLewis <- function(theta, historyZ, BigT) {
  mu <- theta[1]
  K <- theta[2]
  beta <- theta[3]
  eps <- 10 ^ -7
  if ((min(mu, K, beta) < eps) | (K > (1 - eps))) {
    return(Inf)
  }
  sumlog <- log(mu)

  intlam <- mu * BigT + K * length(historyZ)
  const <- K * beta
  for (j in 2:length(historyZ)) {
    sumterm <- 0
    for (i in 1:(j - 1)) {
      sumterm <- sumterm + exp(-beta * (historyZ[j] - historyZ[i]))
    }
    lamj <- mu + const * sumterm
    if (is.na(lamj) | (lamj < 0)) {
      return(Inf)
    }
    sumlog <- sumlog + log(lamj)
  }
  loglik <- sumlog - intlam
  return(-1.0 * loglik)
}
```

```
statespaceOmega <- c(0, 5 * 10 ^ 1)
condInten1 <- list(mu = 1, alpha = 3, beta = 6)
TE <- 1
condInten2 <- list(mu = condInten1$mu, alpha = condInten1$alpha + TE, beta =
    condInten1$beta)

trueEff <-((condInten2$mu * statespaceOmega[2]) / (1 - condInten2$alpha /
    condInten2$beta) -
     (condInten1$mu * statespaceOmega[2]) / (1 - condInten1$alpha /
         condInten1$beta)) / (statespaceOmega[2] - statespaceOmega[1])

numCells <- 2.5 * 10 ^ 1
partitions <- seq(statespaceOmega[1], statespaceOmega[2], length.out = numCells
    + 1)
partitionsDynamic <- T

nIters <- 1 * 10 ^ 3
effx <- list()
set.seed(1)

for (iter in 1:nIters) {
  simProcess1 <- simulate_uni_hawkes(mu = condInten1$mu, alpha =
      condInten1$alpha, beta = condInten1$beta, t_max = statespaceOmega[2])
  simProcess1<-simProcess1[simProcess1<statespaceOmega[2] &
      simProcess1>statespaceOmega[1]]
  simProcess2 <- simulate_uni_hawkes(mu = condInten2$mu, alpha =
      condInten2$alpha, beta = condInten2$beta, t_max = statespaceOmega[2])
  simProcess2<-simProcess2[simProcess2<statespaceOmega[2] &
```

```
    simProcess2>statespaceOmega[1]]

treatprob <- sample(seq(0.2, 0.8, length.out = 10), size = 1)

if (partitionsDynamic == T) {
  superProcess <- sort(c(simProcess1, simProcess2))
  dataTess <- sample(1:length(superProcess), size = length(superProcess) *
      treatprob)
  numCenters <- min(numCells, length(superProcess[dataTess]-1))
  clusts <- kmeans(superProcess[dataTess], numCenters)

  cellDef <- c()
  for (i in 1:(length(clusts$centers) - 1)) {
    cellDef[i] <- (sort(clusts$centers)[i + 1] - sort(clusts$centers)[i]) / 2 +
        sort(clusts$centers)[i]
  }
  partitions <- c(statespaceOmega[1], cellDef, statespaceOmega[2])
}

treatmentInd <- 0
while(((sum(treatmentInd) == 0) | (sum(treatmentInd) == (length(partitions) -
    1)))){
  treatmentInd <- sample(c(0, 1), replace = TRUE, prob = c(treatprob, 1 -
      treatprob), size = length(partitions) - 1)
}

pointTilesControl <- as.numeric(cut(simProcess1, partitions))
marksTilesControl <- c()
for (i in 1:length(pointTilesControl)) {
```

```
  marksTilesControl[i] <- pointTilesControl[i] * (1 -
      treatmentInd[pointTilesControl[i]])
}
allows
observedProcess1 <- simProcess1[marksTilesControl != 0]

pointTilesTreatment <- as.numeric(cut(simProcess2, partitions))
marksTilesTreatment <- c()
for (i in 1:length(pointTilesTreatment)) {
  marksTilesTreatment[i] <- pointTilesTreatment[i] *
      treatmentInd[pointTilesTreatment[i]]
}
observedProcess2 <- simProcess2[marksTilesTreatment != 0]

controlStatespace <- list()
treatmentStatespace <- list()
parsC <- 1
parsT <- 1
for (i in 1:length(partitions)) {
  bottomEdgeC <- sort(unique(partitions[marksTilesControl]))[i]
  for (j in 1:length(partitions)) {
    if (bottomEdgeC == partitions[j] & !is.na(bottomEdgeC)) {
      controlStatespace[[parsC]] <- c(bottomEdgeC, partitions[j + 1])
      parsC <- parsC + 1
    }
  }
  bottomEdgeT <- sort(unique(partitions[marksTilesTreatment]))[i]
  for (j in 1:length(partitions)) {
    if (bottomEdgeT == partitions[j] & !is.na(bottomEdgeT)) {
```

```
      treatmentStatespace[[parsT]] <- c(bottomEdgeT, partitions[j + 1])
      parsT <- parsT + 1
    }
  }
}

canFit <- length(observedProcess1) > 1 & length(observedProcess2) > 1

if (canFit == T) {
  controlIntervals <- do.call(rbind.data.frame, controlStatespace)
  controlT <- sum(controlIntervals[, 2] - controlIntervals[, 1])
  treatmentIntervals <- do.call(rbind.data.frame, treatmentStatespace)
  treatmentT <- sum(treatmentIntervals[, 2] - treatmentIntervals[, 1])

  mleObs1 <- optim(runif(3), loglhawkLewis, historyZ = observedProcess1,
  BigT = controlT)
  fitObs1 <- mleObs1$par
  mleObs2 <- optim(runif(3), loglhawkLewis, historyZ = observedProcess2,
  BigT = treatmentT)
  fitObs2 <- mleObs2$par
} else {
  fitObs1 <- NA
  fitObs2 <- NA
}

if(sum(is.na(c(fitObs1, fitObs2)))==0) {
  simfitted1 <- simulate_uni_hawkes(mu = fitObs1[1], alpha = fitObs1[2] *
  fitObs1[3], beta = fitObs1[3], t_max = statespaceOmega[2])
  simfitted1<-simfitted1[simfitted1<statespaceOmega[2]]
```

```
if(length(simfitted1)==0) {
  simfitted1 <- c(0)
}
simfitted1Subset <- c()
for (i in 1:length(simfitted1)) {
  for (j in 1:length(treatmentStatespace)) {
    if (simfitted1[i] < treatmentStatespace[[j]][2] & simfitted1[i] >
    treatmentStatespace[[j]][1])
      simfitted1Subset <- c(simfitted1Subset, simfitted1[i])
  }
}

simfitted2 <- simulate_uni_hawkes(mu = fitObs2[1], alpha = fitObs2[2] *
fitObs2[3], beta = fitObs2[3], t_max = statespaceOmega[2])
simfitted2<-simfitted2[simfitted2<statespaceOmega[2]]
if (length(simfitted2) == 0) {
  simfitted2 <- c(0)
}
simfitted2Subset <- c()
for (i in 1:length(simfitted2)) {
  for (j in 1:length(controlStatespace)) {
    if (simfitted2[i] < controlStatespace[[j]][2] & simfitted2[i] >
    controlStatespace[[j]][1])
      simfitted2Subset <- c(simfitted2Subset, simfitted2[i])
  }
}

full1 <- sort(c(observedProcess1, simfitted1Subset))
full2 <- sort(c(observedProcess2, simfitted2Subset))
```

```
  allintervals <- c(controlStatespace, treatmentStatespace)
  tileCountsSate <- c()
  tileCounts <- c()
  for (i in 1:length(allintervals)) {
    tileInt <- allintervals[[i]]
    tileCountsSate[i] <- (sum((full2 < tileInt[2]) & (full2 > tileInt[1]))
    - sum((full1 < tileInt[2]) & (full1 > tileInt[1])))
    tileCounts[i] <- sum(pointTilesTreatment == i) -
    sum(pointTilesControl == i)
  }

  SATE <- mean(tileCountsSate) / (statespaceOmega[2] - statespaceOmega[1])
  * (length(partitions) - 1)
  ATE <- mean(tileCounts) / (statespaceOmega[2] - statespaceOmega[1]) *
  (length(partitions) - 1)

  estError <- ATE - SATE
  trueError <- SATE - trueEff

  naive <- length(observedProcess1) * (1 / controlT) -
  length(observedProcess2) * (1 / treatmentT)

  naiveError <- naive - trueEff

  effx[[iter]] <- c(SATE, ATE, naive, estError, trueError, naiveError)
}
else{effx[[iter]] <- rep(NA,6)}
if (iter %% 10 == 0) {
```

```
    print(iter)
  }
}

estEFX <- do.call(rbind.data.frame, effx)
estEFX <- estEFX[complete.cases(estEFX),]
```

7.1.2 Stoyan Grabarnik

The below code fits coefficients to simulated Poisson processes with polynomial intensities using the Stoyan Grabarnik estimator.

```
library(spatstat)
set.seed(1)

#Hyperparameters
statespaceWindow <- c(0, 1, 0, 1)

#Intensity function
condIntenSim <- function(x, y, params) {
  params[[3]] * params[[1]] * x + params[[3]] * params[[4]] * x ^ 2 +
  params[[3]] * params[[2]] * y + params[[3]] * params[[5]] * y ^ 2 +
  params[[3]] * params[[6]]
}

condInten <- function(x, y, params) {
  params[[1]] * x + params[[4]] * x ^ 2 + params[[2]] * y +
  params[[5]] * y ^ 2 + params[[6]]
}
```

```
#Function for Stoyan Grabarnik Estimation
SGest <- function(theta, processForFit, partitionScheme, stateSpace, timeT) {
  argParams <- as.list(theta)
  #argParams[[3]] <- timeT #CHANGED THIS HERE
  #SHOULD BE SPACE TIME OF CELL
  oneRect <- ((stateSpace[2] - stateSpace[1]) * (stateSpace[4] -
  stateSpace[3]) / partitionScheme$n ) * timeT
  lMeasures <- rep(oneRect, partitionScheme$n)
  pointInverseIntensities <- c()
  for (i in 1:processForFit$n) {
    pointInverseIntensities[i] <- 1 / condInten(x = processForFit$x[i],
    y = processForFit$y[i], params = argParams) #CHANGED FUNCTION HERE
  }
  tileInverseIntensities <- c()
  for (j in 1:length(lMeasures)) {
    tileInverseIntensities[j] <- sum(pointInverseIntensities[processForFit$marks
        == j])
  }
  SG <- sum((tileInverseIntensities - lMeasures) ^ 2)
}

###########SIMULATE###########
#Vector of T's
#bigT <- floor(10^seq(0,6,length.out = 10))
bigT <- floor(10^seq(1,6.75,length.out = 20))

superprocesses <- vector(mode='list', length=length(bigT))
nproc<-c()
```

```
for(i in 1:length(bigT)){
  start.time<-Sys.time()
  params0 <- list(xparam = 1/2, yparam = 1/4, tparam=bigT[i],
  x2param = 1/3, y2param = 2/3, constparam = 1/5)
  superprocesses[[i]]<-rpoispp(condIntenSim, params = params0,
  win = statespaceWindow)
  nproc[i]<-superprocesses[[i]]$n
}

############FIT############
#For partitioning with finer or coarser grid
#(each value equal to sqrt(number cells))

partitionSchemes<-c(1,2,4,8,16,32)

#Storage container (list of lists) for simulation values
estimatedParams <- vector(mode='list', length=length(partitionSchemes))

scheme<-1
for(p in partitionSchemes){

  partitions<-quadrats(X=statespaceWindow, nx = p)

  j<-1
  for (k in bigT) {
    simProcess <- superprocesses[[j]]
    tiles <- cut(simProcess, partitions)
    marks(simProcess) <- as.numeric(tiles$marks)
```

```
    SGfit <- optim(
      runif(length(params0)),
      SGest,
      processForFit = simProcess,
      partitionScheme = partitions,
      stateSpace = statespaceWindow,
      timeT=k
    )
    estimatedParams[[scheme]][[j]] <- c(SGfit$par)
    print(c(scheme,j))
    j<-j+1
  }
  scheme<-scheme+1
}

############VISUALIZE############
sgEsts<-vector(mode='list', length=length(partitionSchemes))
for(i in 1:length(sgEsts)){
  sgEsts[[i]]<-do.call(rbind.data.frame, estimatedParams[[i]])
}


#par(mfrow=c(1,1))

sgEstDf1<-sgEsts[[1]][,1]
sgEstDf2<-sgEsts[[1]][,2]
sgEstDf3<-sgEsts[[1]][,4]
sgEstDf4<-sgEsts[[1]][,5]
sgEstDf5<-sgEsts[[1]][,6]
```

```
if(length(partitionSchemes)>1){
   for(i in 1:(length(partitionSchemes)-1)){
  #for(i in 1:9){
    sgEstDf1<-cbind(sgEstDf1,sgEsts[[i+1]][,1])
    sgEstDf2<-cbind(sgEstDf2,sgEsts[[i+1]][,2])
    sgEstDf3<-cbind(sgEstDf3,sgEsts[[i+1]][,4])
    sgEstDf4<-cbind(sgEstDf4,sgEsts[[i+1]][,5])
    sgEstDf5<-cbind(sgEstDf5,sgEsts[[i+1]][,6])
  }
}

#library(RColorBrewer)
#MyCol <- brewer.pal(n=length(partitionSchemes[subcols]),name="RdBu")
MyCol <- rainbow(length(partitionSchemes))

MyLab <- c(paste(floor(partitionSchemes)))
par(mfrow=c(2,3))

matplot(x=log10(bigT),y=sgEstDf1,type='l',lty=1,lwd=1.25,col=MyCol,
        ylab="Est X Param, deg=1",xlab="log(T)",ylim=c(-1,1)+params0$xparam)
abline(h=params0[[1]],lty=1,lwd=0.5)

matplot(x=log10(bigT),y=sgEstDf2,type='l',lty=1,lwd=1.25,col=MyCol,
        ylab="Est Y Param, deg=1",xlab="log(T)",ylim=c(-1,1)+params0$yparam)
abline(h=params0[[2]],lty=1,lwd=0.5)

matplot(x=log10(bigT),y=sgEstDf3,type='l',lty=1,lwd=1.25,col=MyCol,
        ylab="Est X Param, deg=2",xlab="log(T)",ylim=c(-1,1)+params0$x2param)
abline(h=params0[[4]],lty=1,lwd=0.5)
```

```
matplot(x=log10(bigT),y=sgEstDf4,type='l',lty=1,lwd=1.25,col=MyCol,
       ylab="Est Y Param, deg=2",xlab="log(T)",ylim=c(-1,1)+params0$y2param)
abline(h=params0[[5]],lty=1,lwd=0.5)

matplot(x=log10(bigT),y=sgEstDf5,type='l',lty=1,lwd=1.25,col=MyCol,
       ylab="Est Const Param",xlab="log(T)",ylim=c(-1,1)+params0$constparam)
abline(h=params0[[6]],lty=1,lwd=0.5)



plot.new()
legend("center",MyLab,fill=c(MyCol),ncol=1,title="Partitions",cex=2)
```

7.1.3 Stoyan Grabarnik (Centroid and Histogram)

The below code fits coefficients to simulated Poisson processes with polynomial intensities using the histogram and centroid-based equivalents of the Stoyan Grabarnik estimator.

```
SGestHist <- function(theta, processForFit, partitionScheme,
stateSpace, timeT) {
  argParams <- as.list(theta)
  #SHOULD BE SPACE TIME OF CELL
  oneRect <- ((stateSpace[2] - stateSpace[1]) * (stateSpace[4] -
  stateSpace[3]) / partitionScheme$n ) * timeT
  lMeasures <- rep(oneRect, partitionScheme$n)
  pointInverseIntensities <- c()

  for (i in 1:partitionScheme$n) {
```

```
    partitionProcess <- processForFit[processForFit$marks == i]
    if(partitionProcess$n!=0){
    #move mean calculation outside of function to optimize
      pointInverseIntensities[i] <- (1 / condInten(x =
      mean(partitionProcess$x), y = mean(partitionProcess$y),
      params = argParams))*partitionProcess$n
    }else{
      pointInverseIntensities[i] <- 0
    }
  }

  SG <- sum((pointInverseIntensities - lMeasures) ^ 2)
}

SGestCentroid <- function(theta, processForFit, partitionScheme,
stateSpace, timeT,centroidProcess) {
  argParams <- as.list(theta)
  oneRect <- ((stateSpace[2] - stateSpace[1]) * (stateSpace[4] -
  stateSpace[3]) / partitionScheme$n ) * timeT
  lMeasures <- rep(oneRect, partitionScheme$n)
  pointInverseIntensities <- c()

  for (i in 1:partitionScheme$n) {
    partitionProcess <- processForFit[processForFit$marks == i]
    xcentval<-centroidProcess$x[centroidProcess$marks == i]
    ycentval<-centroidProcess$y[centroidProcess$marks == i]
    if(partitionProcess$n!=0){
      pointInverseIntensities[i] <- (1 / condInten(x = xcentval,
      y = ycentval, params = argParams))*partitionProcess$n
```

```
    }else{
      pointInverseIntensities[i] <- 0
    }
  }

  SG <- sum((pointInverseIntensities - lMeasures) ^ 2)
}
```

7.1.4 Stoyan Grabarnik (Analytical Solution)

The below code fits coefficients to simulated Poisson processes with polynomial intensities using an approximation and analytical solution detailed in Section 2.6.2.

```
condIntenSim <- function(x, y, params) {
  params0$tparam * params0$xparam * x +
  params0$tparam * params0$x2param * x ^ 2 +
    params0$tparam * params0$yparam * y +
    params0$tparam * params0$y2param * y ^ 2 +
    params0$tparam * params0$yxparam * y * x +
    #params0$tparam * params0$yx2param * y * x ^ 2 +
    #params0$tparam * params0$y2xparam * y ^ 2 * x +
    params0$tparam * params0$y2x2param * y ^ 2 * x ^ 2 +
    params0$tparam * params0$constparam
}


bigT <- floor(10^seq(1,6.5,length.out = 50))
```

```
superprocesses <- vector(mode='list', length=length(bigT))
nproc<-c()

for(i in 1:length(bigT)){
  start.time<-Sys.time()
  params0 <- list(xparam = 3,
                  yparam = 6,
                  tparam=bigT[i],
                  x2param = 4,
                  y2param = 7,
                  yxparam = 5,
                  #yx2param = 2,#DEGREE 3
                  #y2xparam = 4, #DEGREE 3
                  #y2x2param = 1.5,#DEGREE 4
                  constparam = 10)
  superprocesses[[i]]<-rpoispp(condIntenSim, params = params0,
  win = statespaceWindow)
  nproc[i]<-superprocesses[[i]]$n
  print(i)
}

thetasxy<-vector(mode='list', length=length(bigT))

for(j in 1:length(bigT)){
  partitions<-quadrats(X=statespaceWindow, nx =3, ny=3)
  simProcess <- superprocesses[[j]]
  tiles <- cut(simProcess, partitions)
  marks(simProcess) <- as.numeric(tiles$marks)
```

```
integrandsP<-c()
volP<-c()
meansPx<-c()
meansPy<-c()

for(i in 1:partitions$n){
  partitionProcess <- simProcess[simProcess$marks == i]
  integrandsP[i]<-partitionProcess$n
  volP[i]<- (area(statespaceWindow) / partitions$n ) * bigT[j]
  meansPx[i]<-mean(partitionProcess$x)
  meansPy[i]<-mean(partitionProcess$y)
}

gammasP<-integrandsP/volP

MeanMatrix<-matrix(nrow=partitions$n, ncol=partitions$n)
MeanMatrix[,1]<-meansPy^1
MeanMatrix[,2]<-meansPy^2
MeanMatrix[,3]<-meansPx^1
MeanMatrix[,4]<-meansPx^2

MeanMatrix[,5]<-meansPy^1 * meansPx^1 #xy
MeanMatrix[,6]<-meansPy^2 * meansPx^1 #xy^2
MeanMatrix[,7]<-meansPy^1 * meansPx^2 #x^2 y
MeanMatrix[,8]<-meansPy^2 * meansPx^2 #x^2 y^2
MeanMatrix[,9]<-rep(1,length(meansPx))

thetasxy[[j]]<-c(solve(MeanMatrix,tol = 1e-50)%*%gammasP)
```

```
  print(j)
}

thetaSolvedxy<-do.call(rbind.data.frame, thetasxy)


scaleRlim<-10

par(mfrow=c(3,3))
plot(x=log10(bigT), y=thetaSolvedxy[,2],main="Y^2",
ylab="Solved Coef",ylim=params0$y2param+scaleRlim*c(-1,1))
abline(h=params0$y2param)

plot(x=log10(bigT), y=thetaSolvedxy[,1],main="Y^1",
ylab="Solved Coef",ylim=params0$yparam+scaleRlim*c(-1,1))
#,ylim=10*c(1,-1)+params0$x3param)
abline(h=params0$yparam)

plot(x=log10(bigT), y=thetaSolvedxy[,4],main="X^2",
ylab="SolvedCoef",ylim=params0$x2param+scaleRlim*c(-1,1))#,ylim=c(-100,100))
abline(h=params0$x2param)

plot(x=log10(bigT), y=thetaSolvedxy[,3],main="X^1",
ylab="Solved Coef",ylim=params0$xparam+scaleRlim*c(-1,1))#,ylim=c(0,1))
abline(h=params0$xparam)


plot(x=log10(bigT), y=thetaSolvedxy[,5],main="XY",
```

```
ylab="Solved Coef",ylim=params0$yxparam+scaleRlim*c(-1,1))
abline(h=params0$yxparam)

plot(x=log10(bigT), y=thetaSolvedxy[,6],main="X Y^2",
ylab="Solved Coef",ylim=0+scaleRlim*c(-1,1))#,ylim=10*c(1,-1)+params0$x3param)
abline(h=0)

plot(x=log10(bigT), y=thetaSolvedxy[,7],main="X^2 Y",
ylab="Solved Coef",ylim=0+scaleRlim*c(-1,1))#,ylim=c(-100,100))
abline(h=0)

plot(x=log10(bigT), y=thetaSolvedxy[,8],main="X^2 Y^2",
ylab="Solved Coef",ylim=0+scaleRlim*c(-1,1))#,ylim=c(0,1))
abline(h=0)

plot(x=log10(bigT), y=thetaSolvedxy[,9],main="Const",
ylab="Solved Coef",ylim=params0$constparam+scaleRlim*c(-1,1))#,ylim=c(0,1))
abline(h=params0$constparam)


mtext(bquote("SG Estimation of Inhomogeneous Poisson " ~
lambda==3*x+6*y+4*x^2+7*y^2+5*x*y+10), side = 3, line = -2, outer = TRUE)
```

7.1.5 Stoyan Grabarnik: `C` code for Hawkes Process Intensity

The below code is a function that can be used in `optim()` for quick parameter estimation of Hawkes processes. Credit to Frederic Schoenberg.

```
#include <R.h>
```

```
#include <Rmath.h>

void sgc (double *lon, double *lat, double *t, int *n,
     double *T, double *theta, int *grid, int *m, double *result){
    int i,j,w;
    double sum1, r2, mu, K, a, b, lam, area;
    mu = theta[0]; K = theta[1]; a = theta[2]; b = theta[3];
    *result = 0.0;
    area = *T / *m;

    for(i=0; i < *m; i++){
       sum1 = 0.0;
       for(j=0; j < *n; j++){
           if(grid[j] == i){
      lam = mu;
      for(w = 0; w < j; w++){
         r2 = (lon[j]-lon[w])*(lon[j]-lon[w]) +
                   (lat[j]-lat[w])*(lat[j]-lat[w]);
         lam += K * b * exp(-1.0 * b * (t[j]-t[w])) * a / 3.141593 *
                   exp(-1.0 * a * r2);
         }
      sum1 += 1/lam;
           }
           }
       }
    *result += (sum1 - area)*(sum1-area);
}
```

Bibliography

Allen, L. J. (1994). Some discrete-time SI, SIR, and SIS epidemic models. *Mathematical Biosciences*, 124(1):83–105.

Alonso-Ruiz, P. and Spodarev, E. (2017). Estimation of entropy for poisson marked point processes. *Advances in Applied Probability*, 49(1):258–278.

Ardabili, S. F., Mosavi, A., Ghamisi, P., Ferdinand, F., Varkonyi-Koczy, A. R., Reuter, U., Rabczuk, T., and Atkinson, P. M. (2020). COVID-19 outbreak prediction with machine learning. *Available at SSRN 3580188*.

Artalejo, J. R., Economou, A., and Lopez-Herrero, M. J. (2015). The stochastic SEIR model before extinction: computational approaches. *Applied Mathematics and Computation*, 265:1026–1043.

Baccelli, F. and Woo, J. O. (2016). On the entropy and mutual information of point processes. In *2016 IEEE International Symposium on Information Theory (ISIT)*, pages 695–699. IEEE.

Baddeley, A., Turner, R., Møller, J., and Hazelton, M. (2005). Residual analysis for spatial point processes (with discussion). *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 67(5):617–666.

Baddeley, A. J., Turner, R., et al. (2004). Spatstat: An R package for analyzing spatial point pattens.

Bandyopadhyay, S. K. and Paul, T. U. (2013). Segmentation of brain tumour from mri image analysis of k-means and dbscan clustering. *International Journal of Research in Engineering and Science*, 1(1):48–57.

Barés, J., Dubois, A., Hattali, L., Dalmas, D., and Bonamy, D. (2018). Aftershock sequences and seismic-like organization of acoustic events produced by a single propagating crack. *Nature communications*, 9(1):1253.

Baud, D., Qi, X., Nielsen-Saines, K., Musso, D., Pomar, L., and Favre, G. (2020). Real estimates of mortality following COVID-19 infection. *The Lancet Infectious Diseases*.

Beirlant, J., Dudewicz, E. J., Györfi, L., and Van der Meulen, E. C. (1997). Nonparametric entropy estimation: An overview. *International Journal of Mathematical and Statistical Sciences*, 6(1):17–39.

Bendavid, E., Mulaney, B., Sood, N., Shah, S., Ling, E., Bromley-Dulfano, R., Lai, C., Weissberg, Z., Saavedra, R., Tedrow, J., et al. (2020). COVID-19 antibody seroprevalence in Santa Clara County, California. *MedRxiv*.

Berman, M. and Turner, T. R. (1992). Approximating point process likelihoods with glim. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 41(1):31–38.

Berrett, T. B. (2017). *Modern k-Nearest Neighbour Methods in Entropy Estimation, Independence Testing and Classification*. PhD thesis, University of Cambridge.

Bertozzi, A. L., Franco, E., Mohler, G., Short, M. B., and Sledge, D. (2020). The challenges of modeling and forecasting the spread of COVID-19. *Proceedings of the National Academy of Sciences*, 117(29):16732–16738.

Bertsimas, D. (2020). MIT COVID analytics. covidanalytics.io [Online; accessed 24-May-2020].

Błaszczyszyn, B. (2017). Lecture notes on random geometric models—random graphs, point processes and stochastic geometry.

Błaszczyszyn, B. and Schott, R. (2003). Approximate decomposition of some modulated-Poisson Voronoi tessellations. *Advances in Applied Probability*, 35(4):847–862.

Błaszczyszyn, B. and Schott, R. (2005). Approximations of functionals of some modulated-Poisson Voronoi tessellations with applications to modeling of communication networks. *Japan Journal of Industrial and Applied Mathematics*, 22(2):179–204.

Bray, A., Wong, K., Barr, C. D., Schoenberg, F. P., et al. (2014). Voronoi residual analysis of spatial point process models with applications to california earthquake forecasts. *The Annals of Applied Statistics*, 8(4):2247–2267.

Cauchemez, S., Boëlle, P.-Y., Donnelly, C. A., Ferguson, N. M., Thomas, G., Leung, G. M., Hedley, A. J., Anderson, R. M., and Valleron, A.-J. (2006). Real-time estimates in early detection of SARS. *Emerging Infectious Diseases*, 12(1):110.

Chang, C.-I., Chen, K., Wang, J., and Althouse, M. L. (1994). A relative entropy-based approach to image thresholding. *Pattern recognition*, 27(9):1275–1289.

Chen, S., Shojaie, A., Shea-Brown, E., and Witten, D. (2017). The multivariate hawkes process in high dimensions: Beyond mutual excitation. *arXiv preprint arXiv:1707.04928*.

Cheng, H., Chen, Y., and Jiang, X. (2000). Thresholding using two-dimensional histogram and fuzzy entropy principle. *IEEE Transactions on Image Processing*, 9(4):732–735.

Chiang, W.-H., Liu, X., and Mohler, G. (2020). Hawkes process modeling of covid-19 with mobility leading indicators and spatial covariates. *medRxiv*.

Chiu, S. N., Stoyan, D., Kendall, W. S., and Mecke, J. (2013). *Stochastic geometry and its applications*. John Wiley & Sons.

Chowell, G. and Nishiura, H. (2014). Transmission dynamics and control of Ebola virus disease (EVD): a review. *BMC Medicine*, 12(1):196.

Çinlar, E. (1968). On the superposition of m-dimensional point processes. *Journal of Applied Probability*, pages 169–176.

Cinlar, E. (2013). *Introduction to stochastic processes*. Courier Corporation.

Cinlar, E. and Agnew, R. (1968). On the superposition of point processes. *Journal of the Royal Statistical Society: Series B (Methodological)*, 30(3):576–581.

Clancy, D., O'Neill, P. D., et al. (2008). Bayesian estimation of the basic reproduction number in stochastic epidemic models. *Bayesian Analysis*, 3(4):737–757.

Clark, D. E. (2019). Local entropy statistics for point processes. *IEEE Transactions on Information Theory*, 66(2):1155–1163.

Clements, R. A., Schoenberg, F. P., and Schorlemmer, D. (2011). Residual analysis methods for space-time point processes with applications to earthquake forecast models in California. *The Annals of Applied Statistics*, pages 2549–2571.

Clements, R. A., Schoenberg, F. P., and Veen, A. (2012). Evaluation of space–time point process models using super-thinning. *Environmetrics*, 23(7):606–616.

Costa, J. A. and Hero, A. O. (2006). Determining intrinsic dimension and entropy of high-dimensional shape spaces. In *Statistics and Analysis of Shapes*, pages 231–252. Springer.

Cover, T. M. (1999). *Elements of information theory*. John Wiley & Sons.

Cronie, O. and Van Lieshout, M. N. M. (2018). A non-model-based approach to bandwidth selection for kernel estimators of spatial intensity functions. *Biometrika*, 105(2):455–462.

Daley, D. J. and Jones, D. V. (2003). *An Introduction to the Theory of Point Processes: Elementary Theory of Point Processes*. Springer.

Daley, D. J. and Vere-Jones, D. (2007). *An Introduction to the Theory of Point Processes: General Theory and Structure*. Springer Science & Business Media.

Daley, D. J. and Vere-Jones, D. (2008). *An Introduction to the Theory of Point Processes. Volume II: General Theory and Structure*. Springer.

Daw, A. and Pender, J. (2018). The queue-Hawkes process: Ephemeral self-excitement. *arXiv preprint arXiv:1811.04282*.

Day, M. (2020). COVID-19: identifying and isolating asymptomatic people helped eliminate virus in Italian village. *British Medical Journal*, 368:m1165.

De Silva, V. and Lim, L.-H. (2008). Tensor rank and the ill-posedness of the best low-rank approximation problem. *SIAM Journal on Matrix Analysis and Applications*, 30(3):1084–1127.

Diggle, P. J. (2006). Spatio-temporal point processes, partial likelihood, foot and mouth disease. *Statistical Methods in Medical Research*, 15(4):325–336.

Diggle, P. J., Kaimi, I., and Abellana, R. (2010). Partial-likelihood analysis of spatio-temporal point-process data. *Biometrics*, 66(2):347–354.

Dong, E., Du, H., and Gardner, L. (2020). An interactive web-based dashboard to track COVID-19 in real time. *The Lancet Infectious Diseases*, 20(5):533–534.

Duchenne, O., Bach, F., Kweon, I.-S., and Ponce, J. (2011). A tensor-based algorithm for high-order graph matching. *IEEE transactions on pattern analysis and machine intelligence*, 33(12):2383–2395.

Dye, C. and Gay, N. (2003). Modeling the SARS epidemic. *Science*, 300(5627):1884–1885.

Evans, N. D., White, L. J., Chapman, M. J., Godfrey, K. R., and Chappell, M. J. (2005). The structural identifiability of the susceptible infected recovered model with seasonal forcing. *Mathematical Biosciences*, 194(2):175–197.

Farrington, C., Kanaan, M., and Gay, N. (2003). Branching process models for surveillance of infectious diseases controlled by mass vaccination. *Biostatistics*, 4(2):279–295.

Feldman, D. P. and Crutchfield, J. P. (2003). Structural information in two-dimensional patterns: Entropy convergence and excess entropy. *Physical Review E*, 67(5):051104.

Ferguson, N., Laydon, D., Nedjati Gilani, G., Imai, N., Ainslie, K., Baguelin, M., Bhatia, S., Boonyasiri, A., Cucunuba Perez, Z., Cuomo-Dannenburg, G., et al. (2020). Report 9: impact of non-pharmaceutical interventions (NPIs) to reduce COVID-19 mortality and healthcare demand.

Flaxman, S., Mishra, S., Gandy, A., Unwin, H. J. T., Mellan, T. A., Coupland, H., Whittaker, C., Zhu, H., Berah, T., Eaton, J. W., et al. (2020). Estimating the effects of non-pharmaceutical interventions on COVID-19 in Europe. *Nature*, 584(7820):257–261.

for Health Metrics, I. and (IHME), E. (2020). IMHE COVID-19 predictions. covid19.healthdata.org [Online; accessed 24-May-2020].

Frasso, G. and Lambert, P. (2016). Bayesian inference in an extended SEIR model with nonparametric disease transmission rate: an application to the Ebola epidemic in Sierra Leone. *Biostatistics*, 17(4):779–792.

Geiger, B. C. and Koch, T. (2019). On the information dimension of stochastic processes. *IEEE transactions on information theory*, 65(10):6496–6518.

Godfrey, K. R. and Chapman, M. J. (1990). Identifiability and indistinguishability of linear compartmental models. *Mathematics and Computers in Simulation*, 32(3):273–295.

Goldman, J. R. (1967). Stochastic point processes: limit theorems. *The Annals of Mathematical Statistics*, 38(3):771–779.

Gordon, J. S., Clements, R. A., Schoenberg, F. P., and Schorlemmer, D. (2015). Voronoi residuals and other residual analyses applied to CSEP earthquake forecasts. *Spatial Statistics*, 14:133–150.

Grassly, N. C. and Fraser, C. (2008). Mathematical models of infectious disease transmission. *Nature Reviews Microbiology*, 6(6):477–487.

Gu, J., Yan, H., Huang, Y., Zhu, Y., Sun, H., Zhang, X., Wang, Y., Qiu, Y., and Chen, S. (2020). Better strategies for containing COVID-19 epidemics—a study of 25 countries via an extended varying coefficient SEIR model. *medRxiv*.

Gu, Q. (2020a). UCLA statistical machine learning lab. covid19.uclaml.org [Online; accessed 19-June-2020].

Gu, Y. (2020b). COVID-19 projections using machine learning. covid19-projections.com [Online; accessed 24-May-2020].

Guido Espana, A. P. (2020). Notre Dame mobility. github.com/TAlexPerkins/covid19_NDmobility_forecasting [Online; accessed 19-June-2020].

Hackbusch, W. (2012). *Tensor spaces and numerical tensor calculus*, volume 42. Springer.

Hall, P. (1984). Limit theorems for sums of general functions of m-spacings. In *Mathematical Proceedings of the Cambridge Philosophical Society*, volume 96, pages 517–532. Cambridge University Press.

Harremoës, P. (2001). Binomial and poisson distributions as maximum entropy distributions. *IEEE Transactions on Information Theory*, 47(5):2039–2041.

Harte, D. (2010). Ptprocess: An R package for modelling marked point processes indexed by time. *Journal of Statistical Software*, 35:1–32.

Hawkes, A. G. and Oakes, D. (1974). A cluster process representation of a self-exciting process. *Journal of Applied Probability*, 11(3):493–503.

Hengartner, N. and Fenimore, P. (2018). Quantifying model form uncertainty of epidemic forecasting models from incidence data. *Online Journal of Public Health Informatics*, 10(1).

Hernández, P., Pena, C., Ramos, A., and Gómez-Cadenas, J. (2020). A simple formulation of non-Markovian SEIR. *arXiv preprint arXiv:2005.09975*.

Hillar, C. J. and Lim, L.-H. (2013). Most tensor problems are np-hard. *Journal of the ACM (JACM)*, 60(6):1–39.

Hudgens, M. G. and Halloran, M. E. (2008). Toward causal inference with interference. *Journal of the American Statistical Association*, 103(482):832–842.

Jamwal, R., Gokhale, A., and Bhat, S. (2013). Quantitative fractographic analysis of variability in the tensile ductility of a high strength dual-phase steel. *Metallography, Microstructure, and Analysis*, 2(1):30–34.

Jánossy, L. (1950). On the absorption of a nucleon cascade. In *Proceedings of the Royal Irish Academy. Section A: Mathematical and Physical Sciences*, volume 53, pages 181–188. JSTOR.

Jaynes, E. T. (1957). Information theory and statistical mechanics. *Physical review*, 106(4):620.

Jewell, N. P., Lewnard, J. A., and Jewell, B. L. (2020). Predictive mathematical models of the COVID-19 pandemic: underlying principles and value of projections. *The Journal of the American Medical Association*, 323(19):1893–1894.

Joe, H. (1989). Estimation of entropy and other functionals of a multivariate density. *Annals of the Institute of Statistical Mathematics*, 41(4):683–697.

Kaplan, A., Park, J., Kresin, C., and F., S. (2020). Nonparametric estimation of recursive point processes with application to Mumps in Pennsylvania. *Biometrics*. Submitted May 20, 2020.

Kelly, J. D., Park, J., Harrigan, R. J., Hoff, N. A., Lee, S. D., Wannier, R., Selo, B., Mossoko, M., Njoloko, B., Okitolonda-Wemakoy, E., et al. (2019). Real-time predictions of the 2018–

2019 Ebola virus disease outbreak in the Democratic Republic of the Congo using Hawkes point process models. *Epidemics*, 28:100354.

Kermack, W. O. and McKendrick, A. G. (1927). A contribution to the mathematical theory of epidemics. *Proceedings of the Royal Society of London. Series A, Containing Papers of a Mathematical and Physical Character*, 115(772):700–721.

Koliander, G., Schuhmacher, D., and Hlawatsch, F. (2018). Rate-distortion theory of finite point processes. *IEEE Transactions on Information Theory*, 64(8):5832–5861.

Kontoyiannis, I., Algoet, P. H., Suhov, Y. M., and Wyner, A. J. (1998). Nonparametric entropy estimation for stationary processes and random fields, with applications to english text. *IEEE Transactions on Information Theory*, 44(3):1319–1327.

Kozachenko, L. and Leonenko, N. N. (1987). Sample estimate of the entropy of a random vector. *Problemy Peredachi Informatsii*, 23(2):9–16.

Krickeberg, K. (1982). Processus ponctuels en statistique. In *Ecole d'Eté de Probabilités de Saint-Flour X-1980*, pages 205–313. Springer.

Kucinskas, S. (2020). Tracking R of COVID-19. *Available at SSRN 3581633*.

Laboratory, L. A. N. (2020). COVID-19 confirmed and forecasted case data. covid-19.bsvgateway.org [Online; accessed 24-May-2020].

Lafarge, F., Gimel'Farb, G., and Descombes, X. (2009). Geometric feature extraction by a multimarked point process. *IEEE transactions on pattern analysis and machine intelligence*, 32(9):1597–1609.

Lam, N. S.-N., Qiu, H.-l., Quattrochi, D. A., and Emerson, C. W. (2002). An evaluation of fractal methods for characterizing image complexity. *Cartography and Geographic Information Science*, 29(1):25–35.

Larkin, K. G. (2016). Reflections on shannon information: In search of a natural information-entropy for images. *arXiv preprint arXiv:1609.01117*.

Lee, D. S. (2008). Randomized experiments from non-random selection in us house elections. *Journal of Econometrics*, 142(2):675–697.

Lekone, P. E. and Finkenstädt, B. F. (2006). Statistical inference in a stochastic epidemic SEIR model with control intervention: Ebola as a case study. *Biometrics*, 62(4):1170–1177.

Lemaitre, J. C., Grantz, K. H., Kaminsky, J., Meredith, H. R., Truelove, S. A., Lauer, S. A., Keegan, L. T., Shah, S., Wills, J., Kaminsky, K., et al. (2020). A scenario modeling pipeline for COVID-19 emergency planning. *medRxiv*.

Levin, S. A. and Andreasen, V. (1986). Mathematical models of infectious diseases. *Forefronts*, 2(8):4–6.

Lewis, P. W. and Shedler, G. S. (1979). Simulation of nonhomogeneous Poisson processes by thinning. *Naval Research Logistics Quarterly*, 26(3):403–413.

Lloyd, A. L. (2001). Destabilization of epidemic models with the inclusion of realistic distributions of infectious periods. *Proceedings of the Royal Society of London. Series B: Biological Sciences*, 268(1470):985–993.

Lombardi, D. and Pant, S. (2016). Nonparametric k-nearest-neighbor entropy estimator. *Physical Review E*, 93(1):013310.

Lumley, T. (2020). Counting rare things is hard. statschat.org.nz [Online; accessed 16-May-2020].

Marsan, D. and Lengline, O. (2008). Extending earthquakes' reach through cascading. *Science*, 319(5866):1076–1079.

McFadden, J. (1965). The entropy of a point process. *Journal of the society for industrial and applied mathematics*, 13(4):988–994.

Mecke, K. R. and Stoyan, D. (2000). *Statistical physics and spatial statistics: the art of analyzing and modeling spatial structures and pattern formation*, volume 554. Springer Science & Business Media.

Mei, H. and Eisner, J. (2016). The neural Hawkes process: a neurally self-modulating multivariate point process. *arXiv preprint arXiv:1612.09328*.

Merler, S., Ajelli, M., Fumanelli, L., and Vespignani, A. (2013). Containing the accidental laboratory escape of potential pandemic influenza viruses. *BMC Medicine*, 11(1):252.

Meyer, P.-A. (2006). *Martingales and stochastic integrals I*, volume 284. Springer.

Meyer, S., Held, L., and Höhle, M. (2014). Spatio-temporal analysis of epidemic phenomena using the R package surveillance. *arXiv preprint arXiv:1411.0416*.

Michael L. Mayo, M. A. R. et al. (2020). US army engineer research and development center. github.com/reichlab/covid19-forecast-hub [Online; accessed 19-June-2020].

Mohler, G. et al. (2013). Modeling and estimation of multi-source clustering in crime and security data. *The Annals of Applied Statistics*, 7(3):1525–1539.

Mohler, G., Short, M. B., Schoenberg, F., and Sledge, D. (2020). Analyzing the impacts of public policy on COVID-19 transmission: A case study of the role of model and dataset selection using data from Indiana. *Statistics and Public Policy*, pages 1–17.

Mohler, G. O., Short, M. B., Brantingham, P. J., Schoenberg, F. P., and Tita, G. E. (2011). Self-exciting point process modeling of crime. *Journal of the American Statistical Association*, 106(493):100–108.

Moller, J. and Waagepetersen, R. P. (2003). *Statistical inference and simulation for spatial point processes*. CRC Press.

Montagnon, P. (2019). A stochastic SIR model on a graph with epidemiological and population dynamics occurring over the same time scale. *Journal of Mathematical Biology*, 79(1):31–62.

Montgomery-Smith, S. and Schürmann, T. (2014). Unbiased estimators for entropy and class number. *arXiv preprint arXiv:1410.5002*.

New Jersey COVID-19 Information Hub (2020). How is the state using data to make decisions and slow the spread of COVID-19. covid19.nj.gov [Online; accessed 16-May-2020].

Novikov, A., Podoprikhin, D., Osokin, A., and Vetrov, D. (2015). Tensorizing neural networks. *arXiv preprint arXiv:1509.06569*.

Ogata, Y. (1978). The asymptotic behaviour of maximum likelihood estimators for stationary point processes. *Annals of the Institute of Statistical Mathematics*, 30(2):243–261.

Ogata, Y. (1981). On Lewis' simulation method for point processes. *IEEE Transactions on Information Theory*, 27(1):23–31.

Ogata, Y. (1988). Statistical models for earthquake occurrences and residual analysis for point processes. *Journal of the American Statistical Association*, 83(401):9–27.

Ogata, Y. (1998). Space-time point-process models for earthquake occurrences. *Annals of the Institute of Statistical Mathematics*, 50(2):379–402.

Ogata, Y. and Katsura, K. (1988). Likelihood analysis of spatial inhomogeneity for marked point patterns. *Annals of the Institute of Statistical Mathematics*, 40(1):29–39.

O'Neill, P. D. (2010). Introduction and snapshot review: relating infectious disease transmission models to data. *Statistics in Medicine*, 29(20):2069–2077.

Osthus, D., Hickmann, K. S., Caragea, P. C., Higdon, D., and Del Valle, S. Y. (2017). Forecasting seasonal influenza with a state-space SIR model. *The Annals of Applied Statistics*, 11(1):202.

Ozanne, M. V., Brown, G. D., Oleson, J. J., Lima, I. D., Queiroz, J. W., Jeronimo, S. M., Petersen, C. A., and Wilson, M. E. (2019). Bayesian compartmental model for an infectious disease with dynamic states of infection. *Journal of Applied Statistics*, 46(6):1043–1065.

Pan, A., Liu, L., Wang, C., Guo, H., Hao, X., Wang, Q., Huang, J., He, N., Yu, H., Lin, X., et al. (2020). Association of public health interventions with the epidemiology of the COVID-19 outbreak in Wuhan, China. *The Journal of the American Medical Association*.

Pang, J., Huang, J., Yang, X., Wang, Z., Yu, H., Huang, Q., and Yin, B. (2017). Discovering fine-grained spatial pattern from taxi trips: Where point process meets matrix decomposition and factorization. *IEEE Transactions on Intelligent Transportation Systems*, 19(10):3208–3219.

Paninski, L. (2003). Estimation of entropy and mutual information. *Neural computation*, 15(6):1191–1253.

Papadogeorgou, G., Imai, K., Lyall, J., and Li, F. (2020). Causal inference with spatio-temporal data: Estimating the effects of airstrikes on insurgent violence in iraq. *arXiv preprint arXiv:2003.13555*.

Papangelou, F. (1978). On the entropy rate of stationary point processes and its discrete approximation. *Zeitschrift für Wahrscheinlichkeitstheorie und Verwandte Gebiete*, 44(3):191–211.

Park, J., Chaffee, A. W., Harrigan, R. J., and Schoenberg, F. P. (2020). A non-parametric hawkes model of the spread of ebola in west africa. *Journal of Applied Statistics*, pages 1–17.

Park, J., Schoenberg, F. P., Bertozzi, A. L., and Brantingham, P. J. (2019). Investigating clustering and violence interruption in gang-related violent crime data using spatial-temporal point processes with covariates.

Park, J., Schoenberg, F. P., Bertozzi, A. L., and Brantingham, P. J. (2021). Investigating clustering and violence interruption in gang-related violent crime data using spatial–temporal point processes with covariates. *Journal of the American Statistical Association*, pages 1–14.

Phil Arevalo, E. B. et al. (2020). Forecasting SARS-CoV-2 dynamics for the state of Illinois. github.com/cobeylab/covid_IL [Online; accessed 19-June-2020].

Pinter, G., Felde, I., Mosavi, A., Ghamisi, P., and Gloaguen, R. (2020). COVID-19 pandemic prediction for hungary; a hybrid machine learning approach. *A Hybrid Machine Learning Approach (May 2, 2020)*.

Prabhakar, B. and Bambos, N. (1995). The entropy and delay of traffic processes in atm networks. In *Proceedings of the Conference on Information Science and Systems (CISS)*, pages 448–453.

Rasmussen, J. G. (2013). Bayesian inference for Hawkes processes. *Methodology and Computing in Applied Probability*, 15(3):623–642.

Reinhart, A. (2018). A review of self-exciting spatio-temporal point processes and their applications. *Statistical Science*, 33(3):299–318.

Ripley, B. (1976). On stationarity and superposition of point processes. *The Annals of Probability*, pages 999–1005.

Rizoiu, M.-A., Lee, Y., Mishra, S., and Xie, L. (2017). A tutorial on hawkes processes for events in social media. *arXiv preprint arXiv:1708.06401*.

Rizoiu, M.-A., Mishra, S., Kong, Q., Carman, M., and Xie, L. (2018). SIR-Hawkes: linking epidemic models and Hawkes processes to model diffusions in finite populations. In *Proceedings of the 2018 World Wide Web Conference*, pages 419–428.

Roosa, K. and Chowell, G. (2019). Assessing parameter identifiability in compartmental dynamic models using a computational approach: application to infectious disease transmission models. *Theoretical Biology and Medical Modelling*, 16(1):1.

Rubin, D. B. (2005). Causal inference using potential outcomes: Design, modeling, decisions. *Journal of the American Statistical Association*, 100(469):322–331.

Ryan Best, J. B. (2020). Where the latest COVID-19 models think we're headed — and why they disagree. fivethirtyeight.com [Online; posted 24-May-2020].

Sauer, T., Berry, T., Ebeigbe, D., Norton, M. M., Whalen, A., and Schiff, S. J. (2020). Identifiability of infection model parameters early in an epidemic. *medRxiv*.

Scherrer, S. S., Lohbauer, U., Della Bona, A., Vichi, A., Tholey, M. J., Kelly, J. R., van Noort, R., and Cesar, P. F. (2017). Adm guidance—ceramics: guidance to the use of fractography in failure analysis of brittle materials. *Dental Materials*, 33(6):599–620.

Schoenberg, F. P. (2013). Facilitated estimation of ETAS. *Bulletin of the Seismological Society of America*, 103(1):601–605.

Schoenberg, F. P. (2016). A note on the consistent estimation of spatial-temporal point process parameters. *Statistica Sinica*, pages 861–879.

Schoenberg, F. P. (2020). Nonparametric estimation of variable productivity Hawkes processes. *arXiv preprint arXiv:2003.08858*.

Schoenberg, F. P., Hoffmann, M., and Harrigan, R. J. (2019). A recursive point process model for infectious diseases. *Annals of the Institute of Statistical Mathematics*, 71(5):1271–1287.

Schorlemmer, D., Werner, M. J., Marzocchi, W., Jordan, T. H., Ogata, Y., Jackson, D. D., Mak, S., Rhoades, D. A., Gerstenberger, M. C., Hirata, N., et al. (2018). The collaboratory for the study of earthquake predictability: achievements and priorities. *Seismological Research Letters*, 89(4):1305–1313.

Schuhmacher, D. (2005). Distance estimates for dependent superpositions of point processes. *Stochastic processes and their applications*, 115(11):1819–1837.

Shannon, C. E. (2001). A mathematical theory of communication. *ACM SIGMOBILE mobile computing and communications review*, 5(1):3–55.

Shen, H., Huang, J. Z., et al. (2008). Forecasting time series of inhomogeneous poisson processes with application to call center workforce management. *The Annals of Applied Statistics*, 2(2):601–623.

Singh, H., Misra, N., Hnizdo, V., Fedorowicz, A., and Demchuk, E. (2003). Nearest neighbor estimates of entropy. *American journal of mathematical and management sciences*, 23(3-4):301–321.

Sood, N., Simon, P., Ebner, P., Eichner, D., Reynolds, J., Bendavid, E., and Bhattacharya, J. (2020). Seroprevalence of SARS-CoV-2–specific antibodies among adults in Los Angeles County, California, on April 10-11, 2020. *Journal of the American Medical Association*.

Srinivasan, B. (2020). Peer review of "COVID-19 antibody seroprevalence in Santa Clara County, California". medium.com/@balajis [Online; accessed 16-May-2020].

Stabile, G. and Torrisi, G. L. (2010). Risk processes with non-stationary Hawkes claims arrivals. *Methodology and Computing in Applied Probability*, 12(3):415–429.

Stoyan, D. and Grabarnik, P. (1991). Second-order characteristics for stochastic structures connected with Gibbs point processes. *Mathematische Nachrichten*, 151(1):95–100.

Stoyan, D. and Penttinen, A. (2000). Recent applications of point process methods in forestry statistics. *Statistical science*, pages 61–78.

Tagliazucchi, E., Siniatchkin, M., Laufs, H., and Chialvo, D. R. (2016). The voxel-wise functional connectome can be efficiently derived from co-activations in a sparse spatio-temporal point-process. *Frontiers in neuroscience*, 10:381.

Thum, C. (1984). Measurement of the entropy of an image with application to image focusing. *Optica Acta: International Journal of Optics*, 31(2):203–211.

Van Lieshout, M. (2011). A j–function for inhomogeneous point processes. *Statistica Neerlandica*, 65(2):183–201.

Van Lieshout, M. and Baddeley, A. (1996). A nonparametric measure of spatial interaction in point patterns. *Statistica Neerlandica*, 50(3):344–361.

van Lieshout, M.-C. (2010). Spatial point process theory. *Handbook of Spatial Statistics*, pages 263–82.

Vishal Tomar, C. J. (2020). Auquan data science. covid19-infection-model.auquan.com [Online; accessed 19-June-2020].

Wallinga, J. and Teunis, P. (2004). Different epidemic curves for severe acute respiratory syndrome reveal similar impacts of control measures. *American Journal of epidemiology*, 160(6):509–516.

Walter, E. and Lecourtier, Y. (1981). Unidentifiable compartmental models: what to do? *Mathematical Biosciences*, 56(1-2):1–25.

Wang, C. and Shen, H.-W. (2011). Information theory in scientific visualization. *Entropy*, 13(1):254–273.

Wang, G., Lu, W., Zhou, C., and Zhou, W. (2015). The influence of initial cracks on the crack propagation process of concrete gravity dam-reservoir-foundation systems. *Journal of Earthquake Engineering*, 19(6):991–1011.

Wang, H., Zhang, W., Sun, F., and Zhang, W. (2017). A comparison study of machine learning based algorithms for fatigue crack growth calculation. *Materials*, 10(5):543.

Wearing, H. J., Rohani, P., and Keeling, M. J. (2005). Appropriate models for the management of infectious diseases. *PLoS Medicine*, 2(7).

Worden, L., Wannier, R., Hoff, N. A., Musene, K., Selo, B., Mossoko, M., Okitolonda-Wemakoy, E., Tamfum, J. J. M., Rutherford, G. W., Lietman, T. M., et al. (2019). Projections of epidemic transmission and estimation of vaccination impact during an ongoing Ebola virus disease outbreak in Northeastern Democratic Republic of Congo, as of Feb. 25, 2019. *PLoS Neglected Tropical Diseases*, 13(8):e0007512.

Xia, Y., Ji, Z., and Zhang, Y. (2016). Brain mri image segmentation based on learning local variational gaussian mixture models. *Neurocomputing*, 204:189–197.

Xu, H., Luo, D., and Carin, L. (2018). Online continuous-time tensor factorization based on pairwise interactive point processes. In *IJCAI*, pages 2905–2911.

Yamana, T., Pei, S., and Shaman, J. (2020). Projection of COVID-19 cases and deaths in the us as individual states re-open may 4, 2020. *medRxiv*.

Yang, A. S. (2019). *Modeling the Transmission Dynamics of Pertussis Using Recursive Point Process and SEIR model*. PhD thesis, UCLA.

You, C., Deng, Y., Hu, W., Sun, J., Lin, Q., Zhou, F., Pang, C. H., Zhang, Y., Chen, Z., and Zhou, X.-H. (2020). Estimation of the time-varying reproduction number of COVID-19 outbreak in China. *International Journal of Hygiene and Environmental Health*, page 113555.

Yu, H. and Winkler, S. (2013). Image complexity and spatial information. In *2013 Fifth International Workshop on Quality of Multimedia Experience (QoMEX)*, pages 12–17. IEEE.

Yuan, B. (2020). Multivariate Hawkes processes for real-time COVID-19 death forecasting. Preprint.

Zechar, J. D., Schorlemmer, D., Werner, M. J., Gerstenberger, M. C., Rhoades, D. A., and

Jordan, T. H. (2013). Regional earthquake likelihood models I: First-order results. *Bulletin of the Seismological Society of America*, 103(2A):787–798.

Zhe, S. and Du, Y. (2018). Stochastic nonparametric event-tensor decomposition. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, pages 6857–6867.

Zhuang, J., Ogata, Y., and Vere-Jones, D. (2004). Analyzing earthquake clustering features by using stochastic reconstruction. *Journal of Geophysical Research: Solid Earth*, 109(B5).