XGBoostPP: Tree-based Estimation of Point Process Intensity Functions

Changqing Lu
Centrum Wiskunde & Informatica, Amsterdam, Netherlands
and
Yongtao Guan*
The Chinese University of Hong Kong, Shenzhen, China
and
Marie-Colette van Lieshout*
Centrum Wiskunde & Informatica, Amsterdam, Netherlands
University of Twente, Enschede, Netherlands
and
Ganggang Xu*
University of Miami, Coral Gables, FL, USA

June 11, 2025

Abstract

Medium-sized point pattern data arises in many applications, however, their analyses have been overlooked in the machine learning community. In this paper, we propose a novel tree-based ensemble method, named XGBoostPP, to nonparametrically estimate the intensity of a point process as a function of covariates. It extends the use of gradient-boosted regression trees (Chen and Guestrin, 2016) to the point process literature via two carefully designed loss functions. The first loss is based on the Poisson likelihood and works for general point processes. The second loss derives from a weighted likelihood, where spatially dependent weights are dynamically computed and incorporated to further improve the estimation efficiency for clustered point processes. An efficient learning algorithm and an associated validation procedure are developed for model estimation, and the effectiveness of the proposed method is demonstrated through extensive simulation studies and two real data analyses. In particular, we report that XGBoostPP achieves superior performance to state-of-the-art approaches, showcasing the advantages of using tree ensembles to estimate complex intensity functions for medium-sized point patterns.

Keywords: Medium-sized point pattern, Nonparametric intensity estimation, Tree-based ensemble method

*The last three authors are listed in alphabetical order.

1 Introduction

Point process models are increasingly popular for analyzing point pattern data across diverse fields, including criminology, epidemiology, and sociology. The study of classic medium-sized point patterns is an important branch, where data sets contain merely hundreds or thousands of events in an observation window and replicates are generally unavailable. A crucial aspect of analyzing such a point process is the estimation of its intensity function which defines the likelihood of event occurrences over the observation window. Existing literature explores two primary scenarios: one based on the Cartesian coordinates of events (Diggle, 1985) and the other involving spatial covariates (Guan, 2008; Baddeley et al., 2012). This work concentrates on the latter because, theoretically, modelling the intensity as a function of covariates can achieve consistency (Guan, 2008) and, practically, the intensities at nearby locations may vary due to differences in their covariates whereas distant locations might exhibit similar intensities owing to shared covariates.

Relating spatial covariates to the intensity of a point process has a well-established history. A particularly useful approach is to assume a parametric form for the intensity function, e.g. a log-linear model that formulates the log-intensity as a linear combination of accessible covariates. Extensive studies have been conducted on the estimation accuracy and asymptotic properties of the resulting estimators (Schoenberg, 2005; Waagepetersen and Guan, 2009; Guan et al., 2015; Hessellund et al., 2022a,b; Xu et al., 2023). Although a parametric model is easy to estimate and interpret, it may not be flexible enough for certain applications due to the inherent rigidity of the parametric form.

In contrast, nonparametric approaches impose fewer and less unrealistic assumptions and have gained considerable attention recently. Examples include kernel intensity estimators (KIE) (Guan, 2008; Baddeley et al., 2012), Gaussian Cox process approaches (GCP) (Cunningham et al., 2008; Kim et al., 2022) and Bayesian nonparametric models (Yin et al., 2022). KIEs apply the standard kernel smoother in the covariate space and are compu-

tationally efficient (Baddeley et al., 2015). However, due to the high local variability and bandwidth selection problems, their estimation accuracy may not be as good as others. GCPs model the latent intensity as a function of a Gaussian process and require more computational resources to approximate the intractable integration of the Gaussian process, either via Markov chain Monto Carlo algorithms (Møller and Waagepetersen, 2004) or domain discretization (Rue et al., 2009; Illian et al., 2012). Recent studies have proposed various approximation schemes to reduce the computational demand associated with this integration (e.g. Walder and Bishop, 2017; Aglietti et al., 2019), although the computational cost remains a significant challenge.

Another important limitation of existing nonparametric approaches is the constraint to consider only a small number of covariates, primarily due to the well-known 'curse of dimensionality' phenomenon. For instance, KIEs typically accommodate only one or two spatial covariates without imposing additional, potentially unrealistic restrictions, such as isotropy. GCPs, on the other hand, usually treat the intensity as a function of spatial coordinates, thereby lacking the capability to incorporate additional covariates. A noteworthy exception is the augmented permanental process (APP) (Kim et al., 2022), which extends the work of Flaxman et al. (2017). In APP, the square root-intensity is assumed to be a Gaussian process defined on a multi-dimensional covariate domain. Although, theoretically, there is no limit on the dimension of the covariate space, our numerical results suggest a significant deterioration in the performance of APP as the covariate dimension increases.

With the rapid development of data-collection technologies, an expanding array of covariates has become accessible for point pattern analysis (e.g. Lu et al., 2023). This surge underscores the necessity for an approach adept at handling the dimensionality of covariates in intensity estimation. Neural network-based point processes (NNPP) (Okawa et al., 2019; Zhang et al., 2023) have been explored to learn the deep dependence of covariates on event occurrences. However, they require large data sets, containing millions of events, to obtain

predictive performance and may not be suitable for classic medium-sized point patterns. Tree-based models have proven to excel in predicting the relationship between numerous covariates and the response (Grinsztajn et al., 2022) on medium-sized data. Moreover, ensemble methods, such as random forests (Breiman, 2001) and gradient boosting machines (Natekin and Knoll, 2013), leverage the combination of multiple weak learners to build a robust learner, behaving highly effective across various tasks. Surprisingly, to the best of our knowledge, no existing work in the point process literature has employed tree-based models for intensity estimation.

This work fills the gap discussed above. Specifically, our contributions are three-fold: (i) we propose a novel tree-based approach, XGBoostPP, for intensity estimation of medium-sized point pattern data, particularly over a high-dimensional covariate space, and design the associated learning algorithm and validation procedure; (ii) we derive a dynamic weighted loss function, tailored for our tree-based ensemble model whose variable space varies considerably during the fitting process, to improve the estimation efficiency for clustered point patterns; (iii) we demonstrate the superiority of the proposed approach to the state-of-the-art via numerical and real data experiments: XGBoostPP achieves comparable performance to the best baseline, APP, when the covariate dimension is small, while it outperforms all existing methods when the dimension increases.

Although XGBoostPP has proven effective in estimating intensities for medium-sized point pattern data, it presents computational challenges when applied to larger data sets due to the combination of the greedy search algorithm and the cross-validation procedure. In addition, statistical inference remains challenging for tree-based models in general. Asymptotic uncertainty quantification for intensity estimates is currently unavailable.

The remainder of this paper is organized as follows. Section 2 introduces the background and literature. Section 3 presents the proposed model, along with the design of the loss functions, the learning algorithm, and the procedure for hyperparameter selection.

Section 4 displays a simulation study to evaluate the performance of XGBoostPP and compares it with the state-of-the-art approaches. In Section 5, we apply the method to real-world data sets to further demonstrate its practical utility. Finally, the paper concludes with a discussion of the findings, limitations and directions for future research.

2 Background and Literature

2.1 Point Process Theory

Consider a point process X defined on a bounded domain $\mathcal{S} \subset \mathbb{R}^2$. For any Borel set $\mathcal{B} \subset \mathcal{S}$, let $|\mathcal{B}|$ and $N(\mathcal{B})$ denote the area of $\mathcal{B}$ and the number of events falling in $\mathcal{B}$. Let $\mathrm{d}\boldsymbol{s}$ denote an infinitesimal ball centred at $\boldsymbol{s} \in \mathcal{S}$. Following Diggle (2003), we define the intensity function of X as $\lambda(\boldsymbol{s}) = \lim_{|\mathrm{d}s|\to 0} \mathbb{E}\left[\frac{N(\mathrm{d}\boldsymbol{s})}{|\mathrm{d}s|}\right]$ and the second-order intensity function as $\lambda^{(2)}(\boldsymbol{x},\boldsymbol{y}) = \lim_{|\mathrm{d}\boldsymbol{x}|,|\mathrm{d}\boldsymbol{y}|\to 0} \mathbb{E}\left[\frac{N(\mathrm{d}\boldsymbol{x})N(\mathrm{d}\boldsymbol{y})}{|\mathrm{d}\boldsymbol{x}||\mathrm{d}\boldsymbol{y}|}\right]$. By scaling, one obtains the pair correlation function as $g(\boldsymbol{x},\boldsymbol{y}) = \lambda^{(2)}(\boldsymbol{x},\boldsymbol{y})/[\lambda(\boldsymbol{x})\lambda(\boldsymbol{y})]$. Assuming second-order intensity-reweighted stationarity (Baddeley et al., 2000), it can be simplified to $g(r)$ at distance $r = \|\boldsymbol{x}-\boldsymbol{y}\|$.

Poisson process. A Poisson process is defined by: (i) for any bounded set $\mathcal{B}$, the number of events falling in it is Poisson distributed with mean $\int_{\mathcal{B}} \lambda(\boldsymbol{s})\mathrm{d}\boldsymbol{s}$; (ii) for disjoint bounded sets, the numbers of events falling in them are independent. Under these assumptions, the likelihood of a realization from a Poisson process X reads $\prod_{\boldsymbol{x}\in X} \lambda(\boldsymbol{x}) \exp\left[-\int_{\mathcal{S}} \lambda(\boldsymbol{s})\mathrm{d}\boldsymbol{s}\right]$. For any Poisson process, $g(r) \equiv 1$.

Log-Gaussian Cox process. A Cox process is a Poisson process with a random intensity function. In particular, we consider the log-Gaussian Cox process (LGCP) whose log-intensity reads $\log[\Lambda(\boldsymbol{s})] = \log[\tilde{\lambda}(\boldsymbol{s})] + Y(\boldsymbol{s})$ where $\tilde{\lambda}(\cdot)$ is a deterministic function of spatial covariates and $Y(\cdot)$ is a zero-mean Gaussian process with a covariance kernel $\rho(\cdot,\cdot)$. The kernel can take, but is not limited to, the exponential form $\rho(\boldsymbol{x},\boldsymbol{y}) = \tau^2 \exp\left(-\|\boldsymbol{x}-\boldsymbol{y}\|/\sigma\right)$

with τ^2 as the variance and σ as the scaling parameter. The intensity function of this process is given by $\lambda(\boldsymbol{s}) = \mathbb{E}[\Lambda(\boldsymbol{s})] = \tilde{\lambda}(\boldsymbol{s})\exp(\tau^2/2)$, and the pair correlation function reads $g(r) = \exp[\tau^2\exp(-r/\sigma)]$ which quickly approaches one as r grows large.

Neyman-Scott process. A Neyman-Scott process (NSP) is generated in two stages. First, a set of parent points are produced from a homogeneous Poisson process of intensity κ. Then, for each parent $\boldsymbol{a}$, a set of offspring points are generated independently from a Poisson process with an intensity function $\lambda_{\boldsymbol{a}}(\boldsymbol{s}) = k(\boldsymbol{s}-\boldsymbol{a};\sigma)\tilde{\lambda}(\boldsymbol{s})$ where $k(\cdot;\sigma)$ is a density function parameterized by a scalar σ, quantifying the scattering distance. In particular, the Thomas process assumes $k(\cdot;\sigma)$ to be an isotropic Gaussian kernel with a standard deviation σ. Its intensity function is $\lambda(\boldsymbol{s}) = \kappa\tilde{\lambda}(\boldsymbol{s})$, and the pair correlation function reads $g(r) = 1 + \exp(-r^2/4\sigma^2)/(4\pi\kappa\sigma^2)$ which also quickly approaches one as r goes large.

2.2 Intensity Estimation Approaches

Kernel intensity estimators. KIEs (Diggle, 1985; Guan, 2008; Baddeley et al., 2012) are extensions of kernel density estimators (Silverman, 2018) to point pattern data. Similar to kernel density estimators, estimation can be challenging for KIEs when the dimension of the covariate space is high (Cronie and Lieshout, 2018). To address this challenge, several dimension-reduction tools have been developed (e.g. Guan and Wang, 2010). Implementations of KIEs are available in the R-package *spatstat* (Baddeley et al., 2015), however, only data with at most two covariates is supported.

Gaussian Cox processes. GCPs (Cunningham et al., 2008; Adams et al., 2009; Gunter et al., 2014; Samo and Roberts, 2014; Lloyd et al., 2015; Walder and Bishop, 2017; Donner and Opper, 2018; John and Hensman, 2018; Aglietti et al., 2019; Kim et al., 2022) can be viewed as Bayesian alternatives to KIEs. GCPs model the intensity by a Gaussian process with a positive link function and maximize the posterior probability. Different

link functions lead to variants of GCPs, such as log-Gaussian Cox processes (exponential link, Møller et al., 1998), permanental processes (quadratic link, McCullagh and Møller, 2006) and sigmoidal Gaussian Cox processes (sigmoid link, Adams et al., 2009). To our knowledge, most GCPs estimate the intensity as a function of spatial coordinates, with the exception of APP (Kim et al., 2022). It assumes that the square root-intensity is generated from a Gaussian process defined on the covariate domain and tackles the intractable integration via path integral formulation, achieving great computational improvements. However, it is not immune to the curse of dimensionality. In our numerical experiments, a significant deterioration in its performance is observed as the dimension of the covariate space increases.

Neural network-based point processes. NNPPs (Du et al., 2016; Mei and Eisner, 2017; Zuo et al., 2020; Zhu et al., 2022; Okawa et al., 2019; Zhang et al., 2023) employ deep neural networks to capture the complex occurrence features of point patterns. Related to our work, the deep mixture point process (DMPP) (Okawa et al., 2019) models the intensity as a mixture of spatial kernels centred at finite representative points in the observation window, and the mixture weight of a representative point is estimated by a multi-layer perceptron over its covariates. The deep kernel mixture point process (DKMPP) (Zhang et al., 2023) extends DMPP by adopting another neural network to project spatial coordinates to a hyper-plane and learns the similarity between the representation point and the estimated point using kernels there. Although, methodologically, both DMPP and DKMPP can be directly applied for classic medium-sized point patterns, their performance needs to be evaluated since deep structures usually require plenty of data for model training to obtain predictive performance.

3 Methodology

3.1 The XGBoostPP Model

Consider a point process X defined on an observation window $\mathcal{S}$. Assume that its intensity λ is a function of a p-dimensional covariate vector $\boldsymbol{z}(\boldsymbol{s})$ with $\boldsymbol{s} \in \mathcal{S}$. To estimate the mapping $\lambda[\boldsymbol{z}(\boldsymbol{s})] : \mathbb{R}^p \to \mathbb{R}^+$, we propose a tree-based ensemble model, XGBoostPP, following the classic scalable tree boosting system (Chen and Guestrin, 2016). The standard XGBoost was designed for general regression or classification tasks and is not directly applicable to point pattern data, especially with specific point correlations. Hence, we propose two carefully designed likelihood-based loss functions, an efficient learning algorithm and a validation procedure to adapt to the point process context.

Formally, our XGBoostPP model estimates the log-intensity of X by K additive trees:

$$\log\left\{\lambda\left[\boldsymbol{z}(\boldsymbol{s})\right]\right\} = \sum_{k=1}^{K} f_k\left[\boldsymbol{z}(\boldsymbol{s})\right] \tag{1}$$

with $f_k \in \mathcal{F}$, where $\mathcal{F} = \{f[\boldsymbol{z}(\boldsymbol{s})] = \boldsymbol{\theta}_{q[\boldsymbol{z}(\boldsymbol{s})]}\}$ is the space of regression trees with $q[\boldsymbol{z}(\boldsymbol{s})] : \mathbb{R}^p \to \{1, \ldots, N_q\}$ and $\boldsymbol{\theta}_{q[\boldsymbol{z}(\boldsymbol{s})]} \in \mathbb{R}^{N_q}$. Here, q represents a tree structure that maps the covariate information at a location $\boldsymbol{s}$ to a corresponding leaf index, N_q is the number of leaves in this tree, and $\boldsymbol{\theta}_q$ is the vector of leaf scores. Each tree f_k is uniquely defined by an independent tree structure q_k and the associated leaf score vector $\boldsymbol{\theta}_{q_k}$. We denote the estimated intensity function by $\lambda(\boldsymbol{x}; \boldsymbol{f}_K)$ with $\boldsymbol{f}_K = (f_1, \cdots, f_K)$ and write $\boldsymbol{\theta}_k$ for $\boldsymbol{\theta}_{q_k}$, with $\theta_{k,v}$ representing the element of $\boldsymbol{\theta}_k$ on leaf v. Given a location $\boldsymbol{s}$, the decision rules of the trees in $\boldsymbol{f}_K$ will first classify $\boldsymbol{z}(\boldsymbol{s})$ into K leaves and then calculate the estimated log-intensity by adding up the scores on all corresponding leaves.

3.2 Likelihood-based Loss Functions

3.2.1 Poisson likelihood loss

It is well-known that a parametric intensity function defined as $\lambda(\boldsymbol{s};\boldsymbol{\theta}) = \exp[\boldsymbol{\theta}^\top \boldsymbol{z}(\boldsymbol{s})]$ can be consistently estimated by maximizing the Poisson likelihood (Schoenberg, 2005), even when the underlying point process is not Poisson. Motivated by this phenomenon, we propose to estimate the optimal tree predictors and leaf scores in model (1) by minimizing the following penalized loss function

$$L(\boldsymbol{f}_K) = \sum_{k=1}^{K} \Omega(f_k) - \sum_{\boldsymbol{x}\in X} \phi(\boldsymbol{x};\boldsymbol{f}_K) + \int_{\mathcal{S}} \exp\left[\phi(\boldsymbol{s};\boldsymbol{f}_K)\right] \mathrm{d}\boldsymbol{s}, \tag{2}$$

where $\phi(\boldsymbol{s};\boldsymbol{f}_K) = \log[\lambda(\boldsymbol{x};\boldsymbol{f}_K)]$ and $\Omega(f_k) = \gamma \sum_v |\theta_{k,v}|$ with some $\gamma > 0$. The last two terms in (2) compute the negative Poisson log-likelihood and the first term penalizes the complexity of $\boldsymbol{f}_K$ to avoid overfitting.

Note that one needs to minimize (2) with respect to both tree structures $\{q_k\}$ and leaf scores $\{\boldsymbol{\theta}_k\}$ simultaneously. However, it is impractical to exhaust all possible tree structures. For feasible computation, we develop a greedy search algorithm, which will be illustrated in Section 3.3, to obtain the optimal intensity estimates.

3.2.2 Dynamic weighted likelihood loss

When the underlying point process deviates from a Poisson process, e.g. LGCP and NSP, maximizing the Poisson likelihood, as discussed in Schoenberg (2005), still results in a consistent estimator for a parametric intensity function. However, the estimation efficiency can be poor due to spatial dependence (Guan and Shen, 2010). Such a limitation also applies to XGBoostPP when minimizing (2). To improve the estimation efficiency for clustered processes, we propose a dynamic weighted likelihood loss, inspired by the quasi-likelihood method developed in Guan et al. (2015).

To motivate our proposal, we first assume that the optimal tree structures under model (1) exist, denote them by $\boldsymbol{q}^{\mathcal{O}}$, and examine the 'oracle' estimator for the associated optimal

leaf scores that minimizes $\mathbb{E}[L(\boldsymbol{f}_K)]$ in (2). In this case, all values of $\theta_{k,v}$ that do not correspond to a 'terminal' node in $\boldsymbol{q}^{\mathcal{O}}$ are precisely set to zero. Our focus will then shift to the estimation efficiency of the scores on the remaining nodes, denoted by $\boldsymbol{\theta}_{\mathcal{A}}$ with $\mathcal{A}$ indicating the set of true supporting leaves.

It is important to note that, with the knowledge of the optimal tree structures $\boldsymbol{q}^{\mathcal{O}}$, (1) can be interpreted as a generalized linear model over a vector of transformed covariates $\boldsymbol{Z}_{\mathcal{A}}(\boldsymbol{s})$ that denote the membership identities of a location $\boldsymbol{s}$ with respect to the supporting tree leaves. Specifically, $\lambda(\boldsymbol{s};\boldsymbol{\theta}_{\mathcal{A}}) = \exp[\boldsymbol{\theta}_{\mathcal{A}}^{\top}\boldsymbol{Z}_{\mathcal{A}}(\boldsymbol{s})]$. In this setting, the estimated intensity function is always positive and differentiable with respect to $\boldsymbol{\theta}_{\mathcal{A}}$.

By the Karush–Kuhn–Tucker optimality condition, the estimate of $\boldsymbol{\theta}_{\mathcal{A}}$ obtained by minimizing (2) must satisfy the estimating equation

$$\boldsymbol{e}_{\boldsymbol{\phi}}(\boldsymbol{\theta}_{\mathcal{A}}) = \gamma \mathrm{sgn}(\boldsymbol{\theta}_{\mathcal{A}}) + \sum_{\boldsymbol{x}\in X} \frac{\partial \phi(\boldsymbol{x};\boldsymbol{\theta}_{\mathcal{A}})}{\partial \boldsymbol{\theta}_{\mathcal{A}}} - \int_{\mathcal{S}} \frac{\partial \phi(\boldsymbol{x};\boldsymbol{\theta}_{\mathcal{A}})}{\partial \boldsymbol{\theta}_{\mathcal{A}}} \lambda(\boldsymbol{s};\boldsymbol{\theta}_{\mathcal{A}}) \mathrm{d}\boldsymbol{s} = \boldsymbol{0}.$$

According to Guan et al. (2015), a direct generalization of $\boldsymbol{e}_{\boldsymbol{\phi}}(\boldsymbol{\theta}_{\mathcal{A}}) = \boldsymbol{0}$ is to consider the following more general form

$$\boldsymbol{e}_{\boldsymbol{h}}(\boldsymbol{\theta}_{\mathcal{A}}) = \gamma \mathrm{sgn}(\boldsymbol{\theta}_{\mathcal{A}}) + \sum_{\boldsymbol{x}\in X} \boldsymbol{h}(\boldsymbol{x};\boldsymbol{\theta}_{\mathcal{A}}) - \int_{\mathcal{S}} \boldsymbol{h}(\boldsymbol{s};\boldsymbol{\theta}_{\mathcal{A}}) \lambda(\boldsymbol{s};\boldsymbol{\theta}_{\mathcal{A}}) \mathrm{d}\boldsymbol{s} = \boldsymbol{0}, \tag{3}$$

where $\boldsymbol{h}(\boldsymbol{s};\boldsymbol{\theta}_{\mathcal{A}})$ can be any measurable vector function defined on $\mathcal{S}$ and of the same dimension as $\boldsymbol{\theta}_{\mathcal{A}}$. Following Guan et al. (2015), we call (3) a first-order penalized estimating function. To improve the estimation efficiency of $\boldsymbol{\theta}_{\mathcal{A}}$ for clustered point processes, we analyze the estimation variance of (3). Recall that $\boldsymbol{\theta}_{\mathcal{A}} \neq \boldsymbol{0}$, and denote the sensitivity matrix

$$-\mathbb{E}\left\{\frac{\partial \boldsymbol{e}_{\boldsymbol{h}}(\boldsymbol{\theta}_{\mathcal{A}})}{\partial \boldsymbol{\theta}_{\mathcal{A}}^{\top}}\right\} = \int_{\mathcal{S}} \frac{\partial \lambda(\boldsymbol{s};\boldsymbol{\theta}_{\mathcal{A}})}{\partial \boldsymbol{\theta}_{\mathcal{A}}} \boldsymbol{h}^{\top}(\boldsymbol{s};\boldsymbol{\theta}_{\mathcal{A}}) \mathrm{d}\boldsymbol{s}$$

by $\boldsymbol{S}_{\boldsymbol{h}}$ and the variance matrix

$$\begin{aligned}\mathrm{var}\left\{\boldsymbol{e}_{\boldsymbol{h}}(\boldsymbol{\theta}_{\mathcal{A}})\right\} = &\int_{\mathcal{S}} \boldsymbol{h}(\boldsymbol{s};\boldsymbol{\theta}_{\mathcal{A}}) \boldsymbol{h}^{\top}(\boldsymbol{s};\boldsymbol{\theta}_{\mathcal{A}}) \lambda(\boldsymbol{s};\boldsymbol{\theta}_{\mathcal{A}}) \mathrm{d}\boldsymbol{s} \\ &+ \int_{\mathcal{S}}\int_{\mathcal{S}} \boldsymbol{h}(\boldsymbol{u};\boldsymbol{\theta}_{\mathcal{A}}) \boldsymbol{h}^{\top}(\boldsymbol{s};\boldsymbol{\theta}_{\mathcal{A}}) \lambda(\boldsymbol{u};\boldsymbol{\theta}_{\mathcal{A}}) \lambda(\boldsymbol{s};\boldsymbol{\theta}_{\mathcal{A}}) \left[g(\boldsymbol{s},\boldsymbol{u}) - 1\right] \mathrm{d}\boldsymbol{u}\mathrm{d}\boldsymbol{s}\end{aligned}$$

by $\boldsymbol{\Sigma}_{\boldsymbol{h}}$. Under a proper asymptotic scheme, the estimation covariance matrix of (3) reads $\boldsymbol{S}_{\boldsymbol{h}}^{-1}\boldsymbol{\Sigma}_{\boldsymbol{h}}(\boldsymbol{S}_{\boldsymbol{h}}^{-1})^{\top}$, and its inverse, $\boldsymbol{G}_{\boldsymbol{h}} = \boldsymbol{S}_{\boldsymbol{h}}^{\top}\boldsymbol{\Sigma}_{\boldsymbol{h}}^{-1}\boldsymbol{S}_{\boldsymbol{h}}$, is known as the Godambe information.

The optimal $\boldsymbol{h}^*(\boldsymbol{s};\boldsymbol{\theta}_{\mathcal{A}})$ that minimizes the inverse Godambe information reduces the estimation variance for $\boldsymbol{\theta}_{\mathcal{A}}$ the most. By Guan et al. (2015), it must satisfy a Fredholm integral equation of the second kind, that is, for any $\boldsymbol{s} \in \mathcal{S}$,

$$\int_{\mathcal{S}} \boldsymbol{h}^*(\boldsymbol{u};\boldsymbol{\theta}_{\mathcal{A}})\lambda(\boldsymbol{u};\boldsymbol{\theta}_{\mathcal{A}})\left[g(\boldsymbol{s}-\boldsymbol{u})-1\right]\mathrm{d}\boldsymbol{u} + \boldsymbol{h}^*(\boldsymbol{s};\boldsymbol{\theta}_{\mathcal{A}}) = \frac{\partial\lambda(\boldsymbol{s};\boldsymbol{\theta}_{\mathcal{A}})}{\partial\boldsymbol{\theta}_{\mathcal{A}}}/\lambda(\boldsymbol{s};\boldsymbol{\theta}_{\mathcal{A}}). \tag{4}$$

Solving (4) for $\boldsymbol{h}^*(\boldsymbol{s};\boldsymbol{\theta}_{\mathcal{A}})$ requires either numerical quadrature approximation (Nyström, 1930) or stochastic approximation (Xu et al., 2019); both are computationally expensive. Alternatively, a special case (Guan and Shen, 2010; Chu et al., 2022) can be considered:

$$\int_{\mathcal{S}} \boldsymbol{h}^*(\boldsymbol{u};\boldsymbol{\theta}_{\mathcal{A}})\lambda(\boldsymbol{u};\boldsymbol{\theta}_{\mathcal{A}})\left[g(\boldsymbol{s}-\boldsymbol{u})-1\right]\mathrm{d}\boldsymbol{u} \simeq \boldsymbol{h}^*(\boldsymbol{s};\boldsymbol{\theta}_{\mathcal{A}})\lambda(\boldsymbol{s};\boldsymbol{\theta}_{\mathcal{A}})\int_{\mathcal{S}}\left[g(\boldsymbol{s}-\boldsymbol{u})-1\right]\mathrm{d}\boldsymbol{u},$$

assuming that $\boldsymbol{h}^*(\boldsymbol{u};\boldsymbol{\theta}_{\mathcal{A}})\lambda(\boldsymbol{u};\boldsymbol{\theta}_{\mathcal{A}}) \simeq \boldsymbol{h}^*(\boldsymbol{s};\boldsymbol{\theta}_{\mathcal{A}})\lambda(\boldsymbol{s};\boldsymbol{\theta}_{\mathcal{A}})$ for proximate pairs of locations and that $g(\boldsymbol{s}-\boldsymbol{u}) \simeq 1$ for distant pairs, which yields a closed-form solution $\boldsymbol{h}^*(\boldsymbol{s};\boldsymbol{\theta}_{\mathcal{A}}) = w^*(\boldsymbol{s};\boldsymbol{\theta}_{\mathcal{A}})\frac{\partial\lambda(\boldsymbol{s};\boldsymbol{\theta}_{\mathcal{A}})}{\partial\boldsymbol{\theta}_{\mathcal{A}}}/\lambda(\boldsymbol{s};\boldsymbol{\theta}_{\mathcal{A}})$ with a spatially dependent weight function

$$w^*(\boldsymbol{s};\boldsymbol{\theta}_{\mathcal{A}}) = \frac{1}{1+\lambda(\boldsymbol{s};\boldsymbol{\theta}_{\mathcal{A}})\int_{\mathcal{S}}\left[g(\boldsymbol{s}-\boldsymbol{u})-1\right]\mathrm{d}\boldsymbol{u}}. \tag{5}$$

It is thus straightforward to show that solving (4) with $\boldsymbol{h}^*(\boldsymbol{s};\boldsymbol{\theta}_{\mathcal{A}})$ is equivalent to minimizing the weighted negative log-likelihood:

$$\gamma|\boldsymbol{\theta}_{\mathcal{A}}| - \sum_{\boldsymbol{x}\in X} w^*(\boldsymbol{s};\boldsymbol{\theta}_{\mathcal{A}})\log\left[\lambda(\boldsymbol{x};\boldsymbol{\theta}_{\mathcal{A}})\right] + \int_{\mathcal{S}} w^*(\boldsymbol{s};\boldsymbol{\theta}_{\mathcal{A}})\lambda(\boldsymbol{s};\boldsymbol{\theta}_{\mathcal{A}})\mathrm{d}\boldsymbol{s},$$

where $|\cdot|$ indicates the L_1-norm of a vector. However, this weighted negative log-likelihood, in practice, is unobtainable as the optimal tree structures $\boldsymbol{q}^{\mathcal{O}}$ are inaccessible.

For the parametric model with a fixed covariate space in Guan and Shen (2010), the intensity term in $w^*(\boldsymbol{s};\boldsymbol{\theta}_{\mathcal{A}})$ needs to be estimated in advance by maximizing the Poisson likelihood and set as fixed afterwards, while the integral $\int_{\mathcal{S}}\left[g(\boldsymbol{s}-\boldsymbol{u})-1\right]\mathrm{d}\boldsymbol{u}$ is approximated by $\int_0^m 2\pi r[g(r)-1]\mathrm{d}r$ for some sufficiently large distance m.

Such an estimation procedure does not suit XGBoostPP as the latter has a considerably varying variable space. The intensity estimated by minimizing (2) may not properly approximate $\lambda(\boldsymbol{s};\boldsymbol{\theta}_{\mathcal{A}})$. Instead, we decide to compute the weights dynamically at each iteration of the model fitting process and propose the following penalized weighted loss function

$$L_w(\boldsymbol{f}_K) = \sum_{k=1}^{K}\Omega(f_k) - \sum_{\boldsymbol{x}\in X} w(\boldsymbol{x};\boldsymbol{f}_K)\phi(\boldsymbol{x};\boldsymbol{f}_K) + \int_{\mathcal{S}} w(\boldsymbol{s};\boldsymbol{f}_K)\exp\left[\phi(\boldsymbol{s};\boldsymbol{f}_K)\right]\mathrm{d}\boldsymbol{s}. \tag{6}$$

It is worth emphasizing that the weight function $w(\boldsymbol{x};\boldsymbol{f}_K)$ depends on $\boldsymbol{f}_K$ and varies based on updated tree structures, which incorporates the approximation of $w^*(\boldsymbol{s};\boldsymbol{\theta}_{\mathcal{A}})$ into the estimation of the optimal $\boldsymbol{f}_K$, simultaneously. Indeed, the advanced performance of the dynamic weighted likelihood loss against the fixed one in Guan and Shen (2010), when applied to the XGBoostPP model, has been observed in our preliminary experiments.

3.3 Additive Learning Algorithm

3.3.1 Adding regression trees

We elaborate on the computation details of practically minimizing (4) between iterations. Let us denote the estimated log-intensity over the $k-1$ trees $\hat{\boldsymbol{f}}_{k-1} = (f_1,\cdots,f_{k-1})$ by $\hat{\phi}(\boldsymbol{s};\hat{\boldsymbol{f}}_{k-1})$. At the k-th iteration, we add a tree predictor $\hat{f}_k$ to minimize

$$\begin{aligned} L_w^{(k)}(f_k) = \Omega(f_k) - &\sum_{\boldsymbol{x}\in X}\hat{w}_k(\boldsymbol{x})\left[\hat{\phi}(\boldsymbol{x};\hat{\boldsymbol{f}}_{k-1}) + f_k(\boldsymbol{x})\right] \\ &+ \int_{\mathcal{S}}\hat{w}_k(\boldsymbol{s})\exp\left[\hat{\phi}(\boldsymbol{s};\hat{\boldsymbol{f}}_{k-1}) + f_k(\boldsymbol{s})\right]\mathrm{d}\boldsymbol{s}, \end{aligned} \tag{7}$$

where $\hat{w}_k(\boldsymbol{s}) := w(\boldsymbol{s};\hat{\boldsymbol{f}}_k)$. Without the knowledge of f_k and for computational efficiency, we approximate $w(\boldsymbol{s};\hat{\boldsymbol{f}}_k)$ based on $\hat{\boldsymbol{f}}_{k-1}$:

$$w(\boldsymbol{s};\hat{\boldsymbol{f}}_k) \simeq \frac{\omega_k}{1+\exp\left[\hat{\phi}(\boldsymbol{s};\hat{\boldsymbol{f}}_{k-1})\right]\left[\hat{K}(m)-\pi m^2\right]}. \tag{8}$$

The normalizing scalar ω_k is chosen such that $\int_{\mathcal{S}} w(\boldsymbol{s};\hat{\boldsymbol{f}}_k)\mathrm{d}\boldsymbol{s} = 1$, and $\hat{K}(m)-\pi m^2$ estimates $\int_0^m 2\pi r[g(r)-1]\mathrm{d}r$ using Ripley's K-function which can be computed by the following

standard nonparametric estimator (Møller and Waagepetersen, 2004)

$$\hat{K}(m) = \sum_{\boldsymbol{x} \in X} \sum_{\boldsymbol{y} \in X, \boldsymbol{y} \neq \boldsymbol{x}} \frac{1\{\|\boldsymbol{x} - \boldsymbol{y}\| \leq m\}}{\hat{\lambda}(\boldsymbol{x}; \hat{\boldsymbol{f}}) \hat{\lambda}(\boldsymbol{y}; \hat{\boldsymbol{f}}) \, |\mathcal{S} \cap \mathcal{S}_{\boldsymbol{x}-\boldsymbol{y}}|}.$$

Here, $\boldsymbol{x}, \boldsymbol{y}$ denote pairs of distinct points, $\|\cdot\|$ is the L_2-norm of a vector, $\hat{\lambda}(\cdot; \hat{\boldsymbol{f}})$ is the estimated intensity that may be replaced with the one fitted using (2) and $1/|\mathcal{S} \cap \mathcal{S}_{\boldsymbol{x}-\boldsymbol{y}}|$ is the translation edge correction factor (Ohser and Stoyan, 1981). We manually set $\hat{K}(m) - \pi m^2$ to zero when it is negative.

By applying a quadratic approximation to (7) and removing the constant terms, the loss to be minimized at this iteration can be simplified to

$$\begin{aligned} \tilde{L}_w^{(k)}(f_k) =& \Omega(f_k) - \sum_{\boldsymbol{x} \in X} \hat{w}_k(\boldsymbol{x}) f_k(\boldsymbol{x}) \\ &+ \int_{\mathcal{S}} \hat{w}_k(\boldsymbol{s}) \exp\left[\hat{\phi}(\boldsymbol{s}; \hat{\boldsymbol{f}}_{k-1})\right] \left[f_k(\boldsymbol{s}) + \frac{1}{2} f_k^2(\boldsymbol{s})\right] \mathrm{d}\boldsymbol{s}. \end{aligned} \tag{9}$$

To compute the integrals throughout the derivation, we employ a numerical quadrature approximation. Suppose that the observation window $\mathcal{S}$ can be divided into m grid cells $\mathcal{T}_1, \cdots, \mathcal{T}_m$, and each cell is centered at $\boldsymbol{t}_i$ and has volume $|\mathcal{T}_i|$. Then, the following integral can be approximated as $\int_{\mathcal{S}} \hat{w}_k(\boldsymbol{s}) h(\boldsymbol{s}) \mathrm{d}\boldsymbol{s} = \sum_{i=1}^{m} \hat{w}_k(\boldsymbol{t}_i) h(\boldsymbol{t}_i) |\mathcal{T}_i|$, where $h(\boldsymbol{s})$ is any measurable function defined on $\mathcal{S}$ such that $\hat{w}_k(\boldsymbol{s}) h(\boldsymbol{s})$ is absolutely integrable.

3.3.2 Growing an added tree

To grow an added tree f_k, we adopt a top-down greedy search algorithm. Starting from the root node, we iteratively identify the optimal split among various candidates until a predefined convergence criterion is reached. The key to developing a computationally feasible algorithm is to efficiently compute the loss reduction upon any node split.

To achieve this goal, we denote the set of locations whose $\boldsymbol{z}(\boldsymbol{s})$ belongs to leaf v by $I_{k,v} = \{\boldsymbol{s} : q_k(\boldsymbol{s}) = v\}$. Then, we can extract the contribution of this leaf to (9), expand

$\Omega(f_k)$, and replace f_k by $\theta_{k,v}$ to obtain

$$\begin{aligned}\tilde{L}_{w,v}^{(k)} =& \gamma|\theta_{k,v}| - \sum_{\boldsymbol{x}\in X} \hat{w}_k(\boldsymbol{x}) 1\{\boldsymbol{x} \in I_{k,v}\}\theta_{k,v} \\ &+ \int_{\mathcal{S}} \hat{w}_k(\boldsymbol{s}) \exp\left[\hat{\phi}(\boldsymbol{s}; \hat{\boldsymbol{f}}_{k-1})\right] 1\{\boldsymbol{s} \in I_{k,v}\} \left(\theta_{k,v} + \frac{1}{2}\theta_{k,v}^2\right) \mathrm{d}\boldsymbol{s}.\end{aligned} \tag{10}$$

For any given tree structure q_k, the optimal score $\hat{\theta}_{k,v}$ on a leaf v, minimizing (10), has the following closed-form expression

$$\hat{\theta}_{k,v} = \frac{\operatorname{sgn}(R_{k,v} - T_{k,v}) \max\left(|R_{k,v} - T_{k,v}| - \gamma, 0\right)}{T_{k,v}},$$

where $R_{k,v} = \sum_{\boldsymbol{x}\in X} \hat{w}_k(\boldsymbol{x}) 1\{\boldsymbol{x} \in I_{k,v}\}$ and $T_{k,v} = \int_{\mathcal{S}} \hat{w}_k(\boldsymbol{s}) \exp[\hat{\phi}(\boldsymbol{s}; \hat{\boldsymbol{f}}_{k-1})] 1\{\boldsymbol{s} \in I_{k,v}\} \mathrm{d}\boldsymbol{s}$. It is important to mention that, when $\hat{\theta}_{k,v} = 0$, the leaf will be eliminated from the tree. Hence, the L_1-penalized loss always tends to produce a smaller tree.

Consequently, for any q_k, one can compute a performance score by plugging in all optimal leaf scores as above into the loss (9). Utilizing these performance scores, one can find the best split that minimizes (9) among various candidate splits for each node and grow the tree sequentially. To exhaust possible candidate splits at a node, we use Chen and Guestrin (2016, Algorithm 1).

3.3.3 More stable learning

To achieve a more stable learning performance during the fitting of the XGBoostPP, we also incorporate the shrinkage and subsampling techniques of the standard XGBoost (Chen and Guestrin, 2016). The former shrinks the impacts of the newly added tree by a parameter $\eta > 0$, which is well-known as the learning rate in stochastic optimization problems. By replacing f_k with ηf_k, one leaves more room for future trees to improve model performance. The latter randomly subsamples covariates as the candidate dependent variables for each node split, leading to a boosted random forest.

3.4 Hyperparameter Selection

Selecting proper hyperparameters, including the number of tree predictors K, the learning rate η and the penalty scalar γ, is crucial for a well-performing XGBoostPP. To this end, we propose a two-fold cross-validation method.

The validation procedure can be approximately regarded as model fitting on two thinned point processes, both of which have around a half intensity and the same second-order information of the original process (Cronie et al., 2023). Specifically, we randomly split data points into two parts and, at each time, use one part to train the model and compute the Poisson log-likelihood based on the estimated intensities of the other test part. We summarize the Poisson log-likelihoods on two test parts as the performance measure. To further reduce the randomness in the data splitting, we repeat such a procedure three times and report the averaged test Poisson log-likelihood as the final evaluation metric to choose the optimal combination of hyperparameters.

Note that higher-fold cross validation can also be applied to specific data sets. However, the Poisson log-likelihood on the test parts must be carefully computed using scaled estimated intensities obtained from the training part. Our unreported experiments suggest that higher-fold cross validation produces similar hyperparameter selections in the simulation study.

4 Numerical Study

To evaluate XGBoostPP, we conduct a numerical study on synthetic data of the point process models introduced in Section 2.1. The intensity is governed by a set of covariates that are independently generated from an isotropic Gaussian process with a covariance function $\exp(-10r)$. For comparison, we adopt KIEs with the ratio (ra) method (Baddeley et al., 2012). It is again worth noting that kernel intensity estimators implemented in the R-package *spatstat* only support data with at most two covariates. For GCPs, we implement

APP with the naive kernel (naive) and the degenerate kernel of the random Fourier map (rfm) (Kim et al., 2022). The naive kernel also encounters computational problems when applied to a high-dimensional covariate space. For NNPPs, we reproduce the DKMPP with the radial basis function kernel (rbf) (Zhang et al., 2023).

We use two evaluation metrics to measure the performance of all approaches: the Poisson log-likelihood on new randomly generated test data under the same model (reported in the main text) and the integrated absolute error from true intensities (reported in the supplementary material). For each method to be tested, we report the averaged values and include the standard deviations of the two metrics across 500 simulations. In experiments, we employ the subscripts 'p' and 'w' to represent the XGBoostPP models fitted using the Poisson likelihood loss and the dynamic weighted likelihood loss, respectively. The hyper-parameters – K, γ and η – are chosen from the candidate sets $\{1, \ldots, 600\}$, $\{10, 30, 50\}$ and $\{0.1, 0.05, 0.01\}$. Moreover, we employ ten parallel trees for each training iteration of XGBoostPP and set the proportion of the subsampled covariates at a node split to 1/3.

4.1 Low-dimensional Covariate Space

The intensity function in this scenario is determined by two covariates, denoted by $z_1(\boldsymbol{s})$ and $z_2(\boldsymbol{s})$. For the Poisson process, we set $\lambda(\boldsymbol{s}) = \alpha \exp\{\beta[z_1(\boldsymbol{s}) + z_2(\boldsymbol{s})]\}$, where $\beta = 0.5$ or 1.0 and α is chosen such that the expected number of events in $\mathcal{S} = [0, 1]^2$ is 400. For LGCP, we simulate data from a Poisson process with the random intensity $\Lambda(\boldsymbol{s}) = \alpha \exp\{\beta[z_1(\boldsymbol{s}) + z_2(\boldsymbol{s})] + Y(\boldsymbol{s})\}$, where $Y(\boldsymbol{s})$ is a zero-mean isotropic Gaussian random field with a covariance function $\tau^2 \exp(-r/\sigma)$ and $\tau^2 = 1.0$ or 2.0 and $\sigma = 0.02$ or 0.04. For NSP, we generate parent points from a homogeneous Poisson process with intensity $\kappa = 100$ or 200 and, for each parent, simulate a Poisson number of offspring points, $\max_{\boldsymbol{s} \in \mathcal{S}}[\alpha \exp\{\beta[z_1(\boldsymbol{s}) + z_2(\boldsymbol{s})]\}]$, where the location of each relative to its parent follows a Gaussian distribution with a standard deviation $\sigma = 0.02$ or 0.04. We then keep an offspring point with a probability

$\exp\{\beta[z_1(\boldsymbol{s})+\beta z_2(\boldsymbol{s})]\}/\max_{\boldsymbol{s}\in\mathcal{S}}[\exp\{\beta[z_1(\boldsymbol{s})+z_2(\boldsymbol{s})]\}]$. The distance m used to approximate the weights (8) is set to 0.06 for the Poisson process and to 3σ for the other two processes.

We report the Poisson log-likelihood results on Poisson test data in Table 1 and those on LGCP and NSP test data in Tables 2 and 3. In general, XGBoostPP achieves competitive performance on all types of data under various parameter settings, although APP mostly exhibits the best behavior. KIE performs poorly as spatial heterogeneity increases, as it tends to produce overly smooth estimates. DKMPP seems unsuitable for this estimation task, which may be due to the lack of data for sufficient training. For all approaches, intensity estimation is challenging when observed point patterns are more spatially varying (i.e. large β) and clustered (i.e. large τ^2 and σ for LGCP and small κ and σ for NSP). Moreover, comparing $\text{XGBoostPP}_\text{p}$ with $\text{XGBoostPP}_\text{w}$, it is evident that the performance improves on clustered processes while it remains almost unchanged on Poisson processes. This indicates that the proposed dynamic weighted likelihood loss and the designed learning algorithm work well when the clustering feature of a point pattern is unknown. Furthermore, these improvements are greater when point patterns are more spatially heterogeneous and clustered, aligning with the findings in Guan and Shen (2010).

Table 1: Averaged Poisson log-likelihoods (standard deviations) of different intensity estimation approaches on Poisson test data.

Poisson	**Covs**	$\beta=0.5$	$\beta=1.0$	**Covs**	$\beta=0.5$	$\beta=1.0$	**Covs**	$\beta=0.2$	$\beta=0.4$
True (1800+)		289.4	490.3		289.4	490.3		248.1	360.9
KIE_ra		272.9(3.2)	426.6(5.9)		-	-		-	-
APP_naive		277.7(3.9)	473.6(4.3)		-	-		-	-
APP_rfm	2	277.8(3.9)	473.5(4.2)	10	143.8(15.0)	327.3(14.4)	10	57.9(41.2)	162.2(18.0)
DKMPP_rbf	(§4.1)	255.2(12.5)	395.5(46.9)	(§4.3)	250.7(5.3)	380.9(30.8)	(§4.2)	215.0(4.9)	288.7(8.9)
$\text{XGBoostPP}_\text{p}$		273.2(3.0)	461.4(5.0)		262.3(4.5)	445.4(5.5)		225.9(3.8)	315.9(6.3)
$\text{XGBoostPP}_\text{w}$		273.2(3.0)	461.6(5.3)		262.4(4.5)	445.4(5.5)		225.9(3.8)	315.9(6.3)

Table 2: Averaged Poisson log-likelihoods (standard deviations) of different intensity estimation approaches on LGCP test data.

LGCP	**Covs**	$\tau^2 = 1$		$\tau^2 = 2$		$\tau^2 = 1$		$\tau^2 = 2$	
		$\sigma = 0.02$	$\sigma = 0.04$	$\sigma = 0.02$	$\sigma = 0.04$	$\sigma = 0.02$	$\sigma = 0.04$	$\sigma = 0.02$	$\sigma = 0.04$
		$\beta = 0.5$				$\beta = 1.0$			
True (1800+)		270.2	280.5	289.7	302.5	504.8	469.1	456.0	483.2
KIE_{ra}		252.7(5.4)	260.5(7.8)	271.5(7.1)	277.4(12.8)	438.8(11.4)	402.3(15.7)	388.2(15.1)	411.1(22.8)
$\text{APP}_{\text{naive}}$		257.2(6.6)	264.3(9.3)	274.5(8.2)	275.7(17.1)	484.7(8.2)	444.0(12.6)	429.6(13.4)	443.4(27.7)
APP_{rfm}	2	257.3(6.6)	264.6(9.2)	274.8(8.2)	276.2(16.7)	484.7(8.1)	444.0(12.5)	429.9(12.8)	444.4(25.9)
$\text{DKMPP}_{\text{rbf}}$	(§4.1)	229.4(11.3)	221.0(12.8)	229.9(11.8)	195.7(17.2)	366.2(45.2)	326.3(43.6)	343.1(43.0)	294.6(43.5)
$\text{XGBoostPP}_{\text{p}}$		251.8(5.0)	259.0(8.0)	269.0(6.9)	274.0(14.7)	472.2(8.0)	431.8(11.8)	419.4(11.6)	434.5(21.6)
$\text{XGBoostPP}_{\text{w}}$		252.6(5.7)	260.9(7.7)	272.1(6.8)	278.9(12.9)	475.8(8.7)	435.5(12.9)	428.8(9.6)	444.9(18.6)
APP_{rfm}		109.3(21.7)	101.5(28.9)	107.1(28.3)	72.4(42.0)	316.5(22.6)	262.9(30.7)	238.5(31.7)	221.0(51.3)
$\text{DKMPP}_{\text{rbf}}$	10	222.5(9.6)	207.7(14.5)	218.7(13.4)	173.1(22.4)	360.2(30.1)	320.7(29.8)	340.0(29.8)	277.3(31.2)
$\text{XGBoostPP}_{\text{p}}$	(§4.3)	242.5(6.5)	249.4(8.8)	259.0(11.0)	254.5(25.0)	456.4(8.3)	414.6(12.6)	399.3(14.7)	407.3(32.8)
$\text{XGBoostPP}_{\text{w}}$		244.2(6.9)	251.6(9.0)	262.7(8.4)	261.5(16.3)	458.8(9.1)	415.7(13.0)	408.5(12.2)	415.1(23.5)
		$\beta = 0.2$				$\beta = 0.4$			
True (1600+)		432.9	466.7	480.8	480.7	534.6	581.9	553.3	571.0
APP_{rfm}		248.3(31.6)	271.8(30.8)	279.9(34.4)	238.5(42.0)	323.8(24.5)	344.8(30.6)	316.3(30.3)	287.6(45.7)
$\text{DKMPP}_{\text{rbf}}$	10	383.7(9.1)	361.7(14.0)	359.2(12.1)	303.7(21.5)	460.9(9.9)	429.6(15.5)	415.8(13.4)	393.6(23.5)
$\text{XGBoostPP}_{\text{p}}$	(§4.2)	411.5(4.3)	441.7(7.0)	454.1(11.1)	440.5(19.0)	492.0(7.8)	530.7(12.4)	506.2(12.4)	505.4(24.6)
$\text{XGBoostPP}_{\text{w}}$		412.8(4.2)	442.4(7.0)	456.0(7.8)	442.3(15.0)	496.0(8.7)	533.5(11.8)	512.6(10.8)	510.6(20.3)

4.2 Higher-dimensional Covariate Space

In this scenario, we conduct similar experiments but increase the number of accessible covariates to ten. Denote the covariates by $z_1(\boldsymbol{s}), \ldots, z_{10}(\boldsymbol{s})$. The intensity function is now defined as $\lambda(\boldsymbol{s}) = \alpha \exp(\beta\{z_1(\boldsymbol{s}) + z_2(\boldsymbol{s})z_3(\boldsymbol{s})/2 + \exp[z_4(\boldsymbol{s})]/6 + z_5(\boldsymbol{s})^2/2 + 3\sin[z_6(\boldsymbol{s})]\})$. Note that $z_7(\boldsymbol{s}), \ldots, z_{10}(\boldsymbol{s})$ are included as nuisance variables, which exist often in real-world applications and may affect the effectiveness of an intensity estimation approach. We set σ, τ^2, κ to the same values as in Section 4.1 while $\beta = 0.2$ or 0.4 to balance the spatial heterogeneity to a reasonable scale. Considering that the detection of clustering patterns

Table 3: Averaged Poisson log-likelihoods (standard deviations) of different intensity estimation approaches on NSP test data.

NSP	**Covs**	$\kappa = 100$		$\kappa = 200$		$\kappa = 100$		$\kappa = 200$	
		$\sigma = 0.02$	$\sigma = 0.04$	$\sigma = 0.02$	$\sigma = 0.04$	$\sigma = 0.02$	$\sigma = 0.04$	$\sigma = 0.02$	$\sigma = 0.04$
		$\beta = 0.5$				$\beta = 1.0$			
True (1800+)		290.9	274.8	291.2	271.7	463.1	455.6	472.5	506.9
KIE_{ra}		271.1(8.0)	255.2(6.9)	272.2(6.2)	253.4(5.1)	395.3(17.5)	390.0(14.2)	406.2(13.5)	438.9(10.8)
$\text{APP}_{\text{naive}}$		273.3(10.4)	260.2(8.0)	277.5(7.2)	259.0(5.9)	433.6(16.0)	433.5(11.8)	448.9(11.7)	485.6(8.1)
APP_{rfm}	2	273.6(10.2)	260.4(8.0)	277.6(7.2)	259.1(5.8)	434.4(15.1)	433.6(11.6)	448.9(11.9)	485.8(8.1)
$\text{DKMPP}_{\text{rbf}}$	(§4.1)	243.6(12.4)	230.1(12.4)	247.0(11.2)	231.7(12.1)	325.9(44.9)	327.7(44.6)	345.6(44.6)	381.7(47.8)
$\text{XGBoostPP}_{\text{p}}$		268.0(8.0)	254.6(6.9)	270.7(6.1)	252.8(4.8)	422.1(13.9)	419.9(10.8)	436.4(10.6)	473.2(7.7)
$\text{XGBoostPP}_{\text{w}}$		271.2(8.1)	256.2(7.7)	272.6(6.7)	253.1(5.0)	434.1(12.6)	423.7(11.2)	446.2(9.5)	474.2(8.4)
APP_{rfm}		90.8(34.2)	105.0(26.5)	119.6(24.6)	112.1(21.0)	222.2(39.8)	254.3(31.8)	263.8(31.7)	320.1(24.9)
$\text{DKMPP}_{\text{rbf}}$	10	222.1(17.8)	214.2(14.8)	235.3(13.2)	221.5(10.6)	307.0(31.5)	317.7(29.3)	338.8(27.3)	381.7(32.2)
$\text{XGBoostPP}_{\text{p}}$	(§4.3)	257.6(11.5)	244.4(8.4)	261.4(7.3)	243.0(6.2)	402.1(17.2)	403.3(11.8)	420.9(11.4)	457.2(8.5)
$\text{XGBoostPP}_{\text{w}}$		260.9(9.8)	246.4(8.3)	264.0(7.7)	243.2(6.3)	411.7(19.9)	404.6(11.9)	428.7(11.9)	457.2(8.5)
		$\beta = 0.2$				$\beta = 0.4$			
True(1600+)		377.8	363.7	470.6	447.9	627.0	614.4	528.7	575.2
APP_{rfm}	10	177.1(33.2)	186.8(27.5)	284.5(27.5)	265.6(29.5)	355.6(39.4)	380.7(32.3)	316.2(30.2)	361.8(24.3)
$\text{DKMPP}_{\text{rbf}}$	(§4.2)	316.1(17.8)	311.1(13.7)	423.0(11.9)	405.0(9.6)	513.3(29.6)	514.4(17.4)	405.0(9.6)	490.2(11.1)
$\text{XGBoostPP}_{\text{p}}$		350.2(10.3)	338.4(6.3)	448.1(5.1)	425.8(4.8)	573.2(13.0)	563.9(12.0)	487.4(8.1)	528.3(8.4)
$\text{XGBoostPP}_{\text{w}}$		352.1(8.1)	339.8(6.1)	449.2(5.4)	426.1(4.7)	576.6(16.0)	566.0(11.8)	489.0(8.4)	528.4(8.4)

over complex covariate relationships is difficult, we change m to 0.04 for the Poisson process and to 2σ for the other two models.

The relevant Poisson log-likelihood results are also reported in Tables 1, 2 and 3. In this scenario, the performance of APP deteriorates considerably while the DKMPP outperforms it, showcasing the advantage of deep neural networks in approximating complex reponse-covariate relationships. XGBoostPP achieves the best performance and the standard deviations are much smaller than those of APP and DKMPP, indicating a more stable estimation accuracy of XGBoostPP. Comparing $\text{XGBoostPP}_{\text{p}}$ with $\text{XGBoostPP}_{\text{w}}$, the latter again shows better performance and the improvements are, overall, slightly higher for

spatially heterogeneous and clustered point patterns. However, compared to those in the lower-dimensional case, such improvements become less distinguished.

4.3 Simple Intensity with Many Nuisance Variables

As an extensive experiment, we simulate data of the same models with the simple intensity function $\lambda(\boldsymbol{s}) = \alpha \exp\{\beta[z_1(\boldsymbol{s}) + z_2(\boldsymbol{s})]\}$ as in Section 4.1, however, input the ten covariates from Section 4.2, $z_1(\boldsymbol{s}), \ldots, z_{10}(\boldsymbol{s})$, for intensity estimation, thus with a large number of nuisance variables. Such a scenario occurs widely in practice when numerous covariates are available while researchers are uncertain about which ones are most relevant and should be included.

We again evaluate all approaches with the two performance metrics and report the results. From Tables 1, 2 and 3, it is obvious that XGBoostPP exhibits more robust behavior compared to APP and DKMPP, which can be evidenced by the significantly smaller reductions in the Poisson log-likelihoods when comparing the results to those in Section 4.2. Such robustness suggests great potentials of XGBoostPP in practical applications of classic medium-sized point pattern data. In addition, the reductions of DKMPP is less than APP, revealing that, while the true intensity function is simple, APP is less robust when there is considerable uncertainty in selecting the most relevant covariates.

5 Real Data Analyses

To demonstrate the utility of XGBoostPP in practice, we apply it to two real data sets: tropical forest data on Barro Colorado Island in Panama and kitchen fire data in the Twente region in The Netherlands. Details of the covariates in the two data sets are given in the supplementary material. For every data set, we conduct a four-fold cross-validation to evaluate different intensity estimation approaches based on the test Poisson log-likelihood (i.e., we randomly split the data points into four subsets, assign one for test and the others

for training). Based on the four-fold cross-validation, the test Poisson log-likelihood over the data of one of the four subsets $\mathbf{x}_i$, with $X = \{\mathbf{x}_1, \ldots, \mathbf{x}_4\}$, reads

$$PL_{\text{test},i} = \sum_{\boldsymbol{x} \in \mathbf{x}_i} \log \left[\frac{1}{3} \hat{\lambda}_i(\boldsymbol{x}) \right] - \frac{1}{3} \int_{\mathcal{S}} \hat{\lambda}_i(\boldsymbol{s}) \mathrm{d}\boldsymbol{s},$$

where $\hat{\lambda}_i(\boldsymbol{s})$ is the estimated intensity function for fold i using the other three folds of training data. We sum up this test Poisson log-likelihood over all four subsets of $\mathbf{x}_i$ as the cross-validated test Poisson log-likelihood

$$PL_{\text{test}} = \sum_{i=1}^{4} PL_{\text{test},i} = \sum_{i=1}^{4} \left\{ \sum_{\boldsymbol{x} \in \mathbf{x}_i} \log \left[\frac{1}{3} \hat{\lambda}_i(\boldsymbol{x}) \right] - \frac{1}{3} \int_{\mathcal{S}} \hat{\lambda}_i(\boldsymbol{s}) \mathrm{d}\boldsymbol{s} \right\}.$$

5.1 Tropical Rain Forest Data

We select two tree species in the tropical rain forest data to study: the Beilschmiedia pendula (Bei) and the Capparis frondosa (Capp), containing 3604 and 3299 tree locations, respectively. We investigate eight covariates that may explain their spatial distributions, including terrain elevation and slope, four soil nutrients and solar and wetness indices. Proper intensity estimation for tree species helps forestry scientists research their suitable living environments. We report the cross-validated test Poisson log-likelihoods in Table 4. It indicates that, on both Bei and the Capp data, XGBoostPP outperforms APP and DKMPP. Since Bei appears more clustered compared to Capp (Guan and Shen, 2010; Yue and Loh, 2011), XGBoostPP$_{\text{w}}$ improves the performance on the former relatively more. Figure 1 displays the estimated log-intensities by APP and XGBoostPP, showing that, in comparison to the former, XGBoostPP produces significantly different estimates for the areas with a small number of event observations.

5.2 Kitchen Fire Data

The kitchen fire data (Fire) comprises 699 kitchen fire incidents in Twente from 2004 to 2020. The covariates of interest include building information, urbanity degrees, population

Table 4: Cross-validated (4-fold) test Poisson log-likelihoods of different intensity estimation approaches on tropical forestry and kitchen fire data.

Data set	Bei	Capp	Fire		Bei	Capp	Fire
Covs	8	8	29		8	8	29
APP_{rfm}	-24955.4	-24295.8	-	$\text{XGBoostPP}_{\text{p}}$	-24557.6	-24097.3	-10773.2
$\text{DKMPP}_{\text{rbf}}$	-26505.6	-24487.8	-	$\text{XGBoostPP}_{\text{w}}$	-24376.1	-24080.5	-10519.7

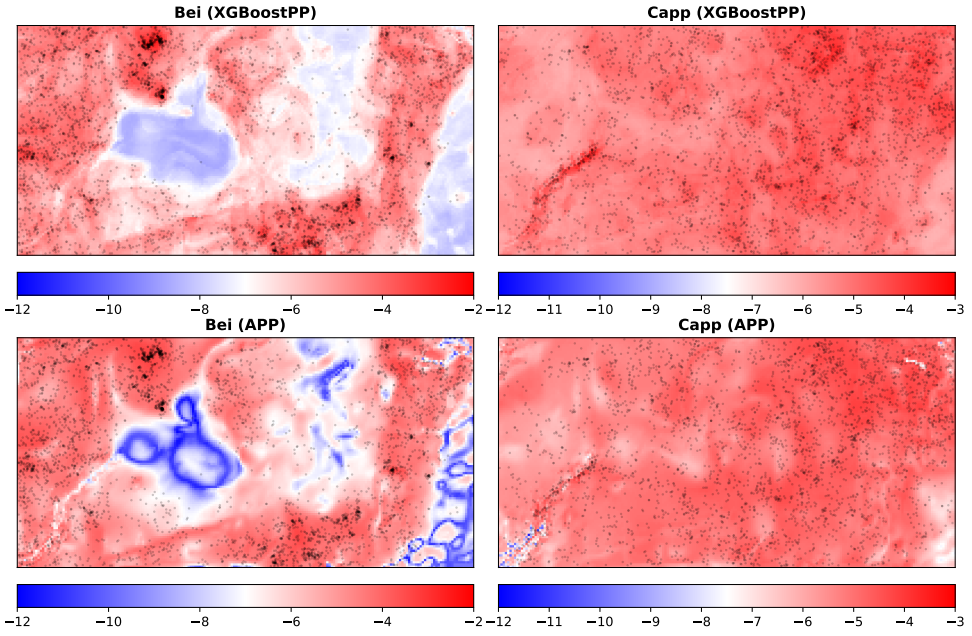

Figure 1: Observed tree locations of Bei and Capp and estimated log-intensities by $\text{XGBoostPP}_{\text{w}}$ and APP.

components and energy consumption, leading to 29 covariates in total. Identifying the most relevant covariates for fire occurrences enables firefighters to organize public campaigns for fire prevention more effectively. Table 4 provides the cross-validated test results for XGBoostPP only, as the other two approaches encounter computational problems and produce significantly less accurate estimates. We also plot the fire data and the estimated log-intensities by XGBoostPP in Figure 2. On this data set, XGBoostPP demonstrates its strong capacity to handle covariate spaces of high dimensions given very limited training data. As expected, $\text{XGBoostPP}_{\text{w}}$ again yields a higher test Poisson log-likelihood due to

the apparent spatial dependence observed in the raw data.

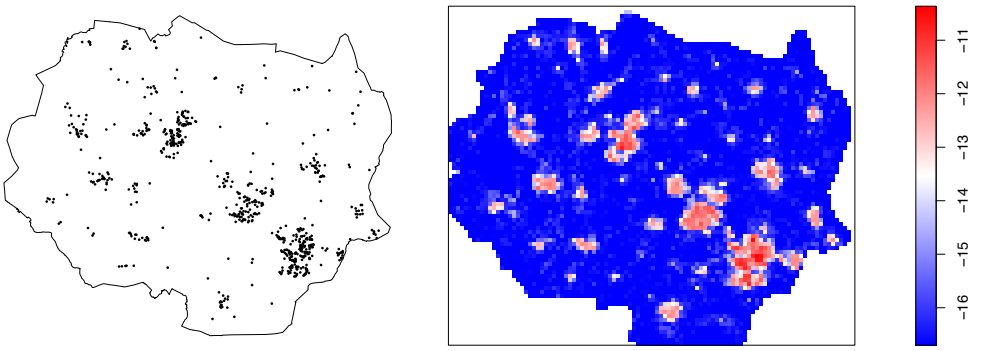

Figure 2: Kitchen fire locations and estimated log-intensities by XGBoostPP$_{\text{w}}$.

5.3 Practical Utility

Intuitively, the estimated intensities produced by XGBoostPP highlight regions with higher tree species densities and fire occurrence risks, providing valuable insights for tropical forestry research and fire management. Estimation uncertainty can be visualized using the bootstrapping method, by randomly sampling a proportion of events without replacement to fit XGBoostPP at a time. For instance, we report the standard deviations of bootstrapping estimates for tropical forestry data in Figure 3. As expected, locations with fewer tree species exhibit higher estimation errors. Additionally, very clustered coordinates also show large estimation errors due to the complexity of second-order information. It is worth noting that such uncertainty results do not directly represent confidence intervals but provide a map of approximated estimation error bounds. Furthermore, leveraging the feature importance capability of standard XGBoost, the relationship between intensity and covariates can be analyzed. Figure 4 plots the covariate importance for the tropical forestry data. The results show that, for both Bei and Capp, Cu nutrient, terrain elevation and slope are significant variables, whereas the wetness index is the least important. To

illustrate these findings, we visualize the four covariates in Figure 5 in comparison to the estimated log-intensities in Figure 1. The results indicate that locations with higher Cu nutrient content, terrain elevation and slope tend to support more Bei and Capp trees. While the wetness index contributes to Bei estimates, it appears quite noisy, which may disrupt model predictions and is less used. Such knowledge encourages further investigation into specific influences of individual covariates on tree species growth. A similar analysis can be conducted on kitchen fire data. Figure 6 presents the covariate importance and highlights the most significant variable – the total number of houses in an area. Notably, the distribution of kitchen fires in Figure 2 aligns closely with the building density. Additional insights from other covariates can help identify key factors driving fire occurrences, providing valuable guidance for fire prevention campaigns.

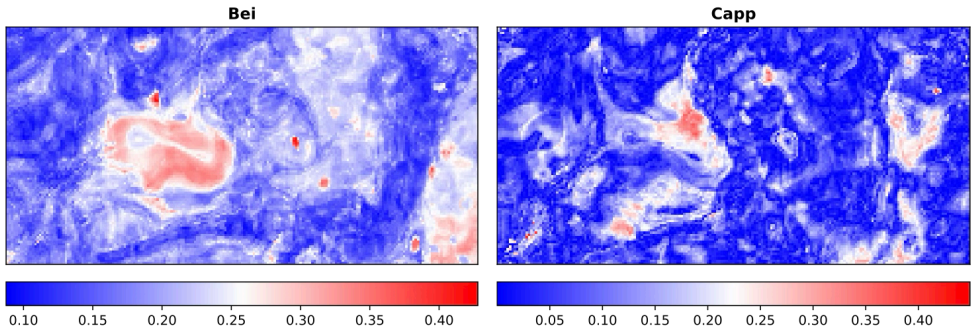

Figure 3: Bootstrapping errors for the estimated log-intensities of Bei and Capp by XGBoostPP$_{\text{w}}$.

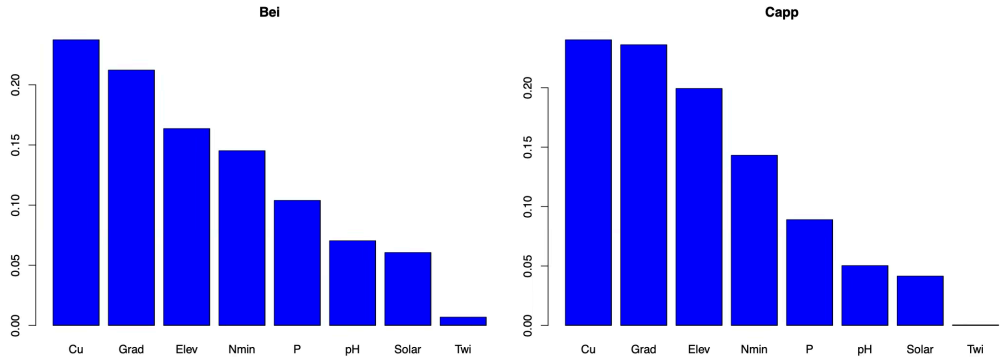

Figure 4: Covariate importance for Bei and Capp identified by XGBoostPP$_{\mathrm{w}}$.

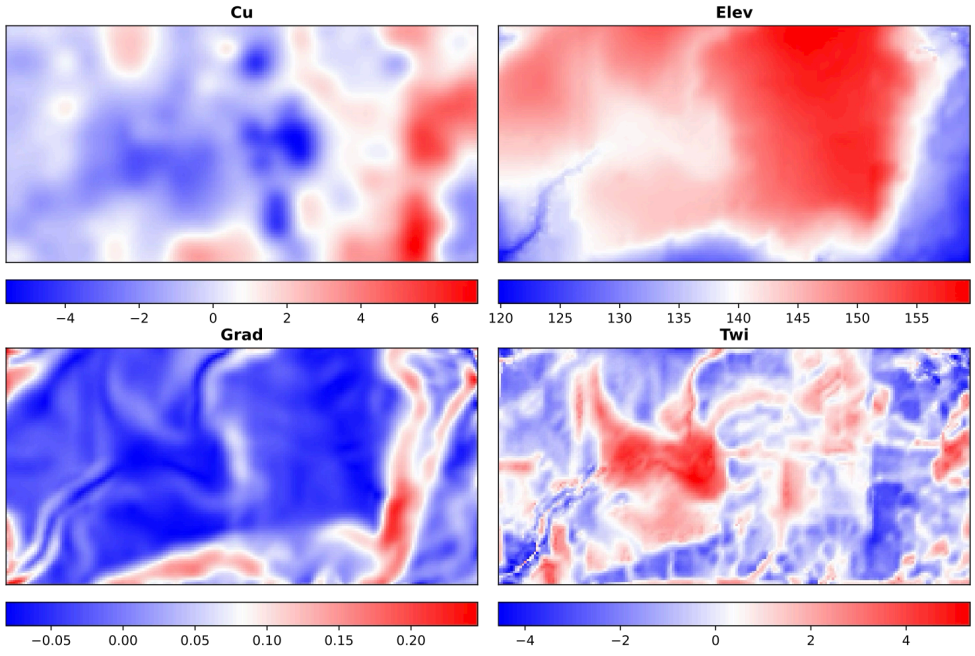

Figure 5: The covariate plots for Cu nutrient (Cu), terrain elevation (Elev) and slope (Grad), and the wetness index (Twi) in tropical forestry data.

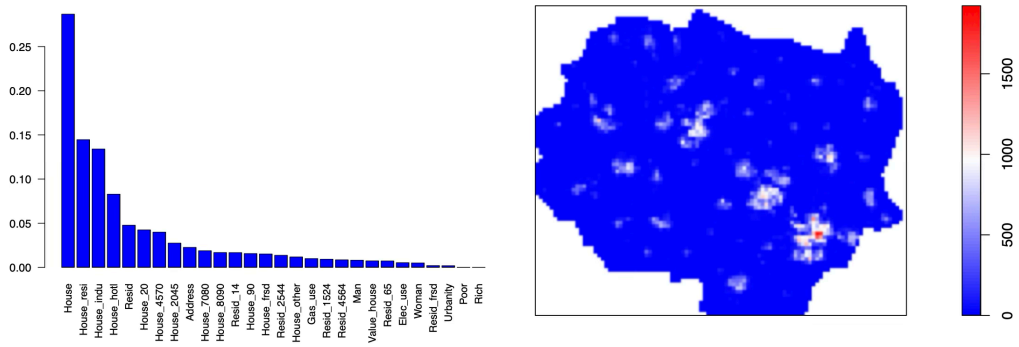

Figure 6: Covariate importance for Fire identified by $\text{XGBoostPP}_{\text{w}}$ and the covariate plot for the total number of houses in kitchen fire data.

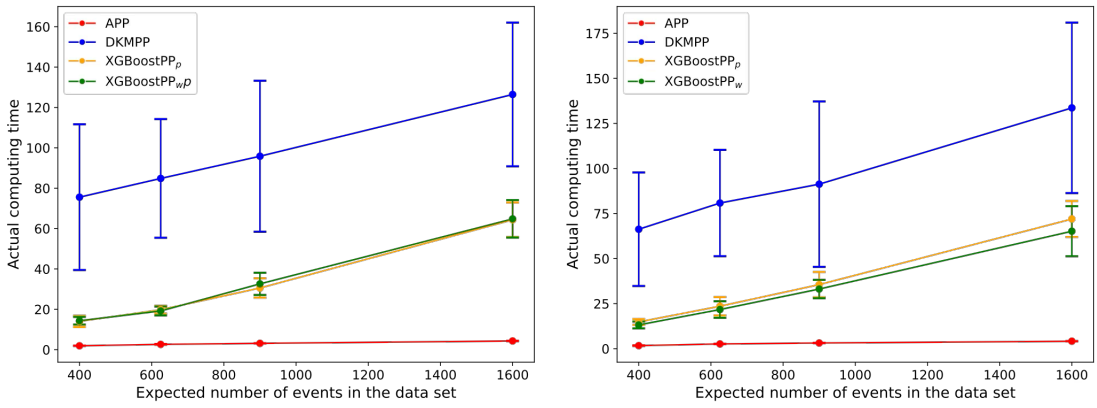

Figure 7: Averaged actual computing times and standard deviations of different intensity estimation approaches obtained from 50 simulations on LGCP (left) and NSP (right) test data in the scenario of higher-dimensional covariate spaces.

6 Conclusions

In this paper, we proposed a novel tree-based ensemble method, XGBoostPP, to nonparametrically estimate the intensity of a point process as a function of covariates, working extremely well for classic medium-sized data. Two loss functions were carefully designed for model estimation, the second of which was specially derived for our tree-based ensemble model to improve the estimation efficiency for clustered processes. We also developed an efficient learning algorithm for model fitting and proposed an associated validation procedure to select hyperparameters. Numerical and real data analyses show that XGBoostPP achieves superior performance to state-of-the-art approaches, suggesting it a useful and flexible tool for analyzing unknown, complex medium-sized point patterns in practice.

Limitations. One limitation of XGBoostPP is the computational demand of hyperparameter selection due to the combination of the greedy search algorithm and the cross-validation procedure (if one has many hyperparameter candidates). In experiments, by using parallel computing tools to tune hyperparameters, XGBoostPP has a larger running time than APP and a smaller time than DKMPP. To illustrate this, Figure 8 presents the averaged values and standard deviations of actual computing times for different intensity estimation approaches, obtained from 50 simulations on LGCP ($\tau^2 = 1, \sigma = 0.02, \beta = 0.2$) and NSP ($\kappa = 100, \sigma = 0.02, \beta = 0.2$) test data in the scenario of higher-dimensional covariate spaces (cf., Section 4.2). All experiments were conducted on a MacBook Pro (M4, 2024), with CPU time measured in seconds. To further examine how computational cost scales with the data size, we extended the experiment by increasing the observation windows from $\mathcal{S} = [0, 1]^2$ to $[0, 1.25]^2$, $[0, 1.5]^2$ and $[0, 2]^2$. The expected number of events in the data sets are correspondingly 400, 625, 900 and 1600. The results show that, the computing time of XGBoostPP increases approximately linearly with the size, which is faster than that of APP, indicating that its implementation to very large data sets would

require more computational demand than APP. However, it remains faster than DKMPP.

Another limitation of XGBoostPP is the lack of asymptotic uncertainty quantification and hypothesis testing for its intensity estimates, as statistical inference remains a general challenge for tree-based models in the literature.

Future Work. The first direction would be to improve the estimation of Ripley's K-function for point patterns with high-dimensional covariate spaces, so that the approximated dynamic weighted likelihood loss could be improved. Moreover, it would also be interesting to extend the proposed method to estimating second-order intensities for non-stationary point pattern data.

SUPPLEMENTARY MATERIAL

Supplementary evaluation results for the simulation study: We report here the results of the integrated absolute error in the numerical study in Section 4. The integrated absolute error reads $\int_{\mathcal{S}} |\lambda(\boldsymbol{s}) - \hat{\lambda}(\boldsymbol{s})| \mathrm{d}\boldsymbol{s}$, where $\lambda(\boldsymbol{s})$ and $\hat{\lambda}(\boldsymbol{s})$ denote the theoretical and estimated intensities, respectively. We report the results on Poisson test data in Table 5 and those on LGCP and NSP test data in Tables 6 and 7. The findings convey similar messages to those reflected in Tables 1, 2 and 3. Specifically, in the scenario of Section 4.1 where there are a small number of covariates, XGBoostPP obtains competitive performance against existing approaches on all data types under various parameter settings. It outperforms KIE and DKMPP, while APP behaves the best. In the scenario of Section 4.2 where there are a larger number of covariates, XGBoostPP significantly outperforms APP whose estimation error is about twice that of XGBoostPP. DKMPP also behaves better than APP in this setting, showing the capacity of neural networks in approximating complex response-covariate relationships. Comparing the estimation errors of $\text{XGBoostPP}_{\text{p}}$ to $\text{XGBoostPP}_{\text{w}}$, the latter are substantially reduced on clustered processes while remain almost unchanged on Poisson processes. These reductions are greater when point patterns are more spatially varying and clustered. In addition, comparing the scenario of Section 4.3 to Section 4.1, we find that, due to the presence of many nuisance variables, the integrated absolute errors of all approaches increase. However, the increases of XGBoostPP and DKMPP are smaller than that of APP. This again indicates that, while the true intensity function is simple, APP is less robust when there is considerable uncertainty in selecting the most relevant covariates.

Table 5: Averaged Integrated Absolute Errors (standard derivations) of different intensity estimation approaches on Poisson test data.

Poisson	**Covs**	$\beta = 0.5$	$\beta = 1.0$	**Covs**	$\beta = 0.5$	$\beta = 1.0$	**Covs**	$\beta = 0.2$	$\beta = 0.4$
KIE_{ra}		91.1(10.2)	170.0(8.9)		-	-		-	-
$\text{APP}_{\text{naive}}$		59.8(11.0)	75.8(11.1)		-	-		-	-
APP_{rfm}	2	60.7(10.6)	77.7(10.6)	10	191.7(7.2)	218.4(7.7)	10	210.0(15.0)	226.5(7.9)
$\text{DKMPP}_{\text{rbf}}$	(§4.1)	125.6(25.9)	236.1(52.4)	(§4.3)	134.8(9.8)	222.5(35.0)	(§4.2)	124.2(9.4)	178.3(12.3)
$\text{XGBoostPP}_{\text{p}}$		86.0(10.3)	103.9(12.0)		116.9(12.0)	137.0(11.3)		109.3(10.2)	149.8(11.9)
$\text{XGBoostPP}_{\text{w}}$		86.1(10.3)	103.7(12.2)		116.9(12.0)	137.0(11.3)		109.3(10.2)	149.8(11.8)

Table 6: Averaged Integrated Absolute Errors (standard derivations) of different intensity estimation approaches on LGCP test data.

LGCP	**Covs**	$\tau^2 = 1$		$\tau^2 = 2$		$\tau^2 = 1$		$\tau^2 = 2$	
		$\sigma = 0.02$	$\sigma = 0.04$	$\sigma = 0.02$	$\sigma = 0.04$	$\sigma = 0.02$	$\sigma = 0.04$	$\sigma = 0.02$	$\sigma = 0.04$
		$\beta = 0.5$				$\beta = 1.0$			
KIE_{ra}		93.7(14.7)	99.5(19.6)	96.8(18.0)	110.8(29.8)	171.5(12.3)	175.7(16.6)	174.0(15.9)	182.0(24.4)
$\text{APP}_{\text{naive}}$		65.9(15.4)	78.0(23.0)	74.9(20.0)	101.7(38.2)	84.0(15.4)	98.7(26.8)	97.2(27.0)	125.7(52.2)
APP_{rfm}	2	66.3(15.0)	77.9(22.9)	74.8(19.7)	100.8(37.7)	85.3(15.1)	99.2(26.5)	97.6(26.3)	124.7(50.5)
$\text{DKMPP}_{\text{rbf}}$	(§4.1)	138.3(23.2)	147.3(23.7)	145.2(21.7)	158.0(25.5)	237.0(47.9)	238.8(44.6)	238.0(44.6)	240.7(39.4)
$\text{XGBoostPP}_{\text{p}}$		94.1(14.9)	102.6(20.3)	100.5(18.4)	120.0(31.2)	113.3(17.3)	127.1(25.4)	123.8(23.1)	146.2(40.7)
$\text{XGBoostPP}_{\text{w}}$		92.6(17.0)	97.7(21.1)	92.1(19.2)	108.4(29.2)	108.6(18.2)	121.8(26.8)	110.0(18.9)	131.8(33.6)
APP_{rfm}		200.2(9.9)	213.2(17.5)	213.5(18.0)	243.9(36.0)	232.0(12.6)	247.4(23.0)	250.8(25.5)	280.1(50.7)
$\text{DKMPP}_{\text{rbf}}$	10	147.9(13.5)	165.1(20.4)	159.6(18.5)	184.2(30.8)	236.8(32.7)	249.3(33.8)	246.9(32.8)	263.2(35.3)
$\text{XGBoostPP}_{\text{p}}$	(§4.3)	118.8(16.1)	126.9(19.8)	125.8(20.3)	155.0(36.6)	146.1(14.8)	162.2(22.2)	164.4(22.3)	193.8(39.8)
$\text{XGBoostPP}_{\text{w}}$		115.1(16.8)	121.2(19.8)	116.0(17.0)	141.5(26.2)	142.2(15.9)	160.2(22.7)	147.5(19.1)	179.1(31.6)
		$\beta = 0.2$				$\beta = 0.4$			
APP_{rfm}		208.0(12.2)	214.2(14.2)	214.2(14.9)	240.9(32.6)	234.4(10.2)	246.1(17.5)	246.5(17.0)	272.7(40.1)
$\text{DKMPP}_{\text{rbf}}$	10	135.9(13.5)	155.6(21.1)	148.7(18.2)	176.3(31.0)	189.2(14.2)	203.1(21.0)	199.7(18.8)	222.1(29.2)
$\text{XGBoostPP}_{\text{p}}$	(§4.2)	109.2(12.0)	116.9(17.7)	117.6(20.6)	142.0(34.3)	149.5(14.3)	160.1(20.0)	156.0(20.5)	180.4(35.8)
$\text{XGBoostPP}_{\text{w}}$		104.9(11.8)	112.9(16.8)	110.5(16.4)	135.6(26.4)	140.9(16.7)	154.3(20.3)	139.7(17.3)	166.9(29.8)

Table 7: Averaged Integrated Absolute Errors (standard derivations) of different intensity estimation approaches on NSP test data.

NSP	**Covs**	$\kappa = 100$		$\kappa = 200$		$\kappa = 100$		$\kappa = 200$	
		$\sigma = 0.02$	$\sigma = 0.04$	$\sigma = 0.02$	$\sigma = 0.04$	$\sigma = 0.02$	$\sigma = 0.04$	$\sigma = 0.02$	$\sigma = 0.04$
		$\beta = 0.5$				$\beta = 1.0$			
KIE_{ra}		98.5(19.4)	97.8(18.1)	95.4(15.7)	94.0(14.4)	175.3(17.4)	174.5(14.6)	173.6(13.3)	172.3(11.7)
$\text{APP}_{\text{naive}}$		79.6(22.7)	72.8(20.2)	69.2(17.2)	66.3(14.8)	104.9(27.0)	92.6(23.2)	90.1(19.3)	84.8(16.4)
APP_{rfm}	2	79.3(22.5)	72.9(20.0)	69.3(17.1)	66.8(14.5)	104.9(26.8)	93.6(22.7)	91.2(18.8)	85.9(16.1)
$\text{DKMPP}_{\text{rbf}}$	(§4.1)	148.3(21.6)	144.5(23.2)	142.5(21.8)	137.5(24.8	239.2(43.5)	234.4(46.4)	233.6(48.2)	230.9(50.6)
$\text{XGBoostPP}_{\text{p}}$		103.9(19.3)	99.9(19.0)	96.8(16.1)	94.1(14.6)	129.7(25.4)	122.5(21.9)	120.6(19.7)	114.8(17.3)
$\text{XGBoostPP}_{\text{w}}$		95.3(20.1)	95.4(20.5)	92.4(18.4)	93.6(15.5)	112.0(22.3)	117.5(22.7)	105.0(17.6)	113.7(18.1)
APP_{rfm}		223.3(16.0)	209.6(13.5)	206.8(11.3)	199.5(9.8)	263.4(20.8)	242.6(16.5)	240.7(14.3)	230.0(11.0)
$\text{DKMPP}_{\text{rbf}}$	10	173.3(19.8)	166.0(19.3)	156.8(15.4)	152.1(14.7)	260.4(28.1)	249.4(30.7)	244.4(28.8)	235.4(34.1)
$\text{XGBoostPP}_{\text{p}}$	(§4.3)	129.6(20.6)	124.6(19.2)	121.2(16.7)	119.9(15.3)	172.2(22.3)	155.9(20.1)	153.1(17.1)	146.1(15.8)
$\text{XGBoostPP}_{\text{w}}$		119.5(17.5)	119.9(18.6)	114.7(17.2)	119.4(15.5)	148.7(22.2)	153.7(20.1)	138.8(17.9)	146.1(15.8)
		$\beta = 0.2$				$\beta = 0.4$			
APP_{rfm}	10	221.1(13.8)	211.0(11.6)	209.0(10.6)	207.3(11.4)	253.9(14.7)	241.1(12.0)	238.7(11.1)	232.7(9.2)
$\text{DKMPP}_{\text{rbf}}$	(§4.2)	164.7(22.2)	155.4(19.4)	146.4(15.5)	141.6(14.4)	212.9(18.2)	204.3(18.6)	196.8(16.6)	191.9(15.4)
$\text{XGBoostPP}_{\text{p}}$		120.0(19.4)	114.8(16.2)	110.0(13.3)	110.9(13.4)	161.3(17.7)	158.1(16.8)	143.5(13.4)	153.3(14.2)
$\text{XGBoostPP}_{\text{w}}$		113.9(17.0)	111.2(15.3)	104.9(13.1)	109.9(13.1)	143.6(17.5)	153.0(17.8)	138.4(12.9)	153.2(14.2)

Poisson toy examples: In addition to the main simulation study, we test XGBoostPP on three Poisson toy examples to intuitively visualize its flexibility in modelling various nonlinear relationships between covariates and the intensity function. The examples are created on the unit window and take the x, y coordinates as covariates. The first example has the intensity function $\lambda(x, y) = \exp(3 + 2x + 7y)$, which is simply log-linear. The second example employs the intensity $\lambda(x, y) = \exp[8 + 16(x - 0.5)^2]$, which aims to show the automatic variable selection capability of XGBoostPP. The third example considers the intensity function $\lambda(x, y) = \exp[2+4\sin(16x)+6\sin(16y)]$, which intends to show the capacity for modelling extraordinary covariate responses. For all examples, we generate a point pattern based on the designed intensity function

and fit XGBoostPP using the Poisson likelihood loss to estimate the intensity as they are all Poisson processes. To test the flexibility, we only input x, y as the spatial covariates to XGBoostPP and force it to learn the covariate relationships and the intensity trends on its own. We plot point patterns, true intensities and the estimated intensities by XGBoostPP in Figure 8. The plots clearly show that XGBoostPP can detect different covariate relationships and perform an automatic variable selection.

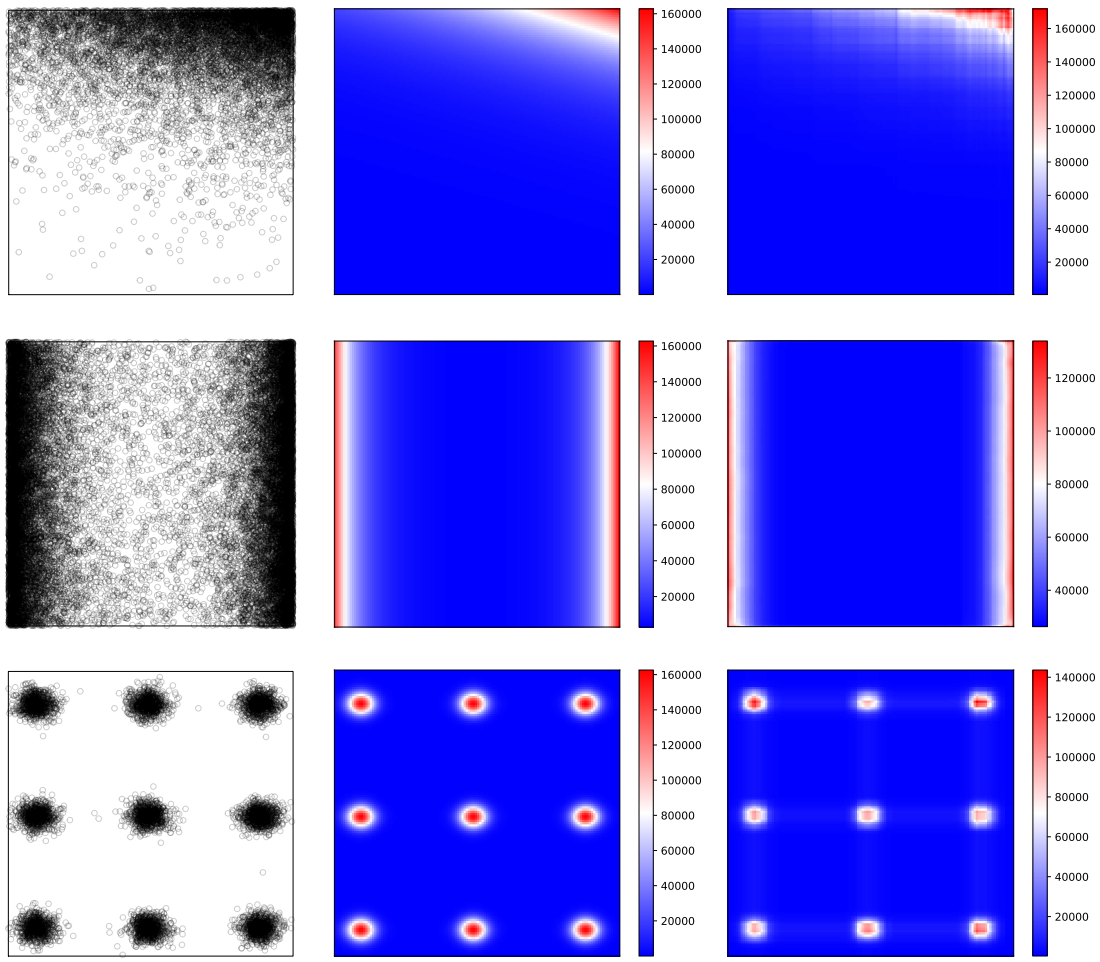

Figure 8: Point patterns, true intensities and estimated intensities by $\text{XGboostPP}_\text{p}$ for the three Poisson toy examples.

Covariates information in the two real data sets: The covariates considered in the tropical rain forest data set (Bei, Capp) and the kitchen fire data set (Fire), along with their descriptions, are listed in Tables 8 and 9, respectively.

Table 8: The covariates considered in the tropical forestry data set.

Data	**Covariate**	**Description**
	Elev	The terrain elevation
	Grad	The terrain slope
	Cu	The content of Cu
Bei	Nmin	The content of Nmin
Capp	P	The content of P
	pH	The pH value of soil
	Solar	The solar index
	Twi	The wetness index

Table 9: The covariates considered in the kitchen fire data set.

Data	Covariate	Description
Fire	House	The total number of houses
	House_indu	The number of houses with an industrial function
	House_hotl	The number of houses with a hotel function
	House_resi	The number of houses with a residential function
	House_20	The number of houses constructed before 1920
	House_2045	The number of houses constructed in $[1920, 1945)$
	House_4570	The number of houses constructed in $[1945, 1970)$
	House_7080	The number of houses constructed in $[1970, 1980)$
	House_8090	The number of houses constructed in $[1980, 1990)$
	House_90	The number of houses constructed after 1990
	House_frsd	The number of free standing houses
	House_other	The number of other houses
	Resid	The number of residents
	Resid_14	The number of residents with an age in $[0, 14)$
	Resid_1524	The number of residents with an age in $[15, 24)$
	Resid_2544	The number of residents with an age in $[25, 44)$
	Resid_4564	The number of residents with an age in $[45, 64)$
	Resid_65	The number of residents with an age over 65
	Man	The number of male residents
	Woman	The number of female residents
	Resid_frsd	The number of residents living in free standing houses
	Address	The density of addresses in the block
	Urbanity	The urbanity of the block
	Town	Boolean variable indicating the presence of a town
	Poor	The percentage of poor residents (income $0 - 20$ percent)
	Rich	The percentage of rich residents (income $80 - 100$ percent)
	Value_house	The average value of the houses in the block
	Gas_use	The average gas use in m^3 in the block
	Elec_use	The average electricity use in kWh in the block

Data sources: The tropical rain forest data was made possible by National Science Foundation, the John D. and Catherine T. MacArthur Foundation, the Mellon Foundation, the Celera Foundation, and numerous private individuals, and through the hard work

of over 100 people from 10 countries over the past several decades. The kitchen fire data was provided by the Twente Fire Brigade, and was cleaned by Paul Visscher and Niels Peters.

Acknowledgments: This work was carried out when Changqing Lu was at the Department of Applied Mathematics at the University of Twente and during a research visit to the Chinese University of Hong Kong. The research was funded by the Dutch Research Council (NWO) for the project 'Data Driven Risk Management for Fire Services' (18004).

References

Adams, R. P., Murray, I., and MacKay, D. J. C. (2009), "Tractable nonparametric Bayesian inference in Poisson processes with Gaussian process intensities," in *ICML 2009*.

Aglietti, V., Bonilla, E. V., Damoulas, T., and Cripps, S. (2019), "Structured variational inference in continuous Cox process models," in *NeurIPS 2019*.

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

Baddeley, A., Rubak, E., and Turner, R. (2015), *Spatial Point Patterns: Methodology and Applications with R*, CRC Press.

Baddeley, A. J., Chang, Y., Song, Y., and Turner, R. (2012), "Nonparametric estimation of the dependence of a spatial point process on spatial covariates," *Stat. Interface*, 5, 221–236.

Breiman, L. (2001), "Random forests," *Mach. Learn.*, 45, 5–32.

Chen, T. and Guestrin, C. (2016), "XGBoost: A scalable tree boosting system," in *KDD 2016*.

Chu, T., Guan, Y., Waagepetersen, R., and Xu, G. (2022), "Quasi-likelihood for multivariate spatial point processes with semiparametric intensity functions," *Spatial Stat.*, 50, 100605.

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

Cronie, O., Moradi, M., and Biscio, C. A. N. (2023), "A cross-validation-based statistical theory for point processes," *Biometrika*, 111, 625–641.

Cunningham, J. P., Shenoy, K. V., and Sahani, M. (2008), "Fast Gaussian process methods for point process intensity estimation," in *ICML 2008*.

Diggle, P. (1985), "A kernel method for smoothing point process data," *J. R. Stat. Soc. C*, 34, 138–147.

Diggle, P. J. (2003), *Statistical Analysis of Spatial Point Patterns*, Oxford University Press.

Donner, C. and Opper, M. (2018), "Efficient Bayesian inference of sigmoidal Gaussian Cox processes," *J. Mach. Learn. Res.*, 19, 1–34.

Du, N., Dai, H., Trivedi, R., Upadhyay, U., Gomez-Rodriguez, M., and Song, L. (2016), "Recurrent marked temporal point processes: Embedding event history to vector." in *KDD 2016*.

Flaxman, S., Teh, Y. W., and Sejdinovic, D. (2017), "Poisson intensity estimation with reproducing kernels," in *AISTATS 2017*.

Grinsztajn, L., Oyallon, E., and Varoquaux, G. (2022), "Why do tree-based models still outperform deep learning on typical tabular data?" in *NeurIPS 2022*.

Guan, Y. (2008), "On consistent nonparametric intensity estimation for inhomogeneous spatial point processes," *J. Am. Stat. Assoc.*, 103, 1238–1247.

Guan, Y., Jalilian, A., and Waagepetersen, R. (2015), "Quasi-likelihood for spatial point processes," *J. R. Stat. Soc. B*, 77, 677–697.

Guan, Y. and Shen, Y. (2010), "A weighted estimating equation approach for inhomogeneous spatial point processes," *Biometrika*, 97, 867–880.

Guan, Y. and Wang, H. (2010), "Sufficient dimension reduction for spatial point processes directed by Gaussian random fields," *J. R. Stat. Soc. B*, 72, 367–387.

Gunter, T., Lloyd, C., Osborne, M. A., and Roberts, S. J. (2014), "Efficient Bayesian nonparametric modelling of structured point processes," in *UAI 2014*.

Hessellund, K. B., Xu, G., Guan, Y., and Waagepetersen, R. (2022a), "Second-order semiparametric inference for multivariate log Gaussian Cox processes," *J. R. Stat. Soc. C*, 71, 244–268.

— (2022b), "Semiparametric multinomial logistic regression for multivariate point pattern data," *J. Am. Stat. Assoc.*, 117, 1500–1515.

Illian, J. B., Sørbye, S. H., and Rue, H. (2012), "A toolbox for fitting complex spatial point process models using integrated nested Laplace approximation (INLA)," *Ann. Appl. Stat.*, 6, 1499–1530.

John, S. and Hensman, J. (2018), "Large-scale Cox process inference using variational Fourier features," in *ICML 2018*.

Kim, H., Asami, T., and Toda, H. (2022), "Fast Bayesian estimation of point process intensity as function of covariates," in *NeurIPS 2022*.

Lloyd, C., Gunter, T., Osborne, M. A., and Roberts, S. J. (2015), "Variational inference for Gaussian process modulated Poisson processes," in *ICML 2015*.

Lu, C., Lieshout, M. N. M., Graaf, M., and Visscher, P. (2023), "Data-driven chimney fire risk prediction using machine learning and point process tools," *Ann. Appl. Stat.*, 17, 3088–3111.

McCullagh, P. and Møller, J. (2006), "The permanental process," *Adv. Appl. Probab.*, 38, 873–888.

Mei, H. and Eisner, J. (2017), "The neural Hawkes process: A neurally self-modulating multivariate point process." in *NeurIPS 2017*.

Møller, J., Syversveen, A. R., and Waagepetersen, R. P. (1998), "Log Gaussian Cox processes," *Scand. J. Stat.*, 25, 451–482.

Møller, J. and Waagepetersen, R. (2004), *Statistical Inference and Simulation for Spatial Point Processes*, Taylor & Francis.

Natekin, A. and Knoll, A. (2013), "Gradient boosting machines, a tutorial," *Front. Neurorob.*, 7, 21.

Nyström, E. J. (1930), "Über die Prakische Auflösung von Integralgleichungen mit Anwendungen auf Randwertaufgaben," *Acta Math.*, 54, 185–204.

Ohser, J. and Stoyan, D. (1981), "On the second-order and orientation analysis of planar stationary point processes," *Biom. J.*, 23, 523–533.

Okawa, M., Iwata, T., Kurashima, T., Tanaka, Y., Toda, H., and Ueda, N. (2019), "Deep mixture point processes: Spatio-temporal event prediction with rich contextual information," in *KDD 2019*.

Rue, H., Martino, S., and Chopin, N. (2009), "Approximate Bayesian inference for latent Gaussian models by using integrated nested Laplace approximations," *J. R. Stat. Soc. B*, 71, 319–392.

Samo, Y. K. and Roberts, S. (2014), "Scalable nonparametric Bayesian inference on point processes with Gaussian processes," in *ICML 2014*.

Schoenberg, F. (2005), "Consistent parametric estimation of the intensity of a spatial-temporal point process," *J. Stat. Plann. Inference*, 128, 79–93.

Silverman, B. W. (2018), *Density Estimation for Statistics and Data Analysis*, CRC Press.

Waagepetersen, R. and Guan, Y. (2009), "Two-step estimation for inhomogeneous spatial point processes," *J. R. Stat. Soc. B*, 71, 685–702.

Walder, C. J. and Bishop, A. N. (2017), "Fast Bayesian intensity estimation for the permanental process," in *ICML 2017*.

Xu, G., Liang, C., Waagepetersen, R., and Guan, Y. (2023), "Semiparametric goodness-of-fit test for clustered point processes with a shape-constrained pair correlation function," *J. Am. Stat. Assoc.*, 118, 2072–2087.

Xu, G., Waagepetersen, R., and Guan, Y. (2019), "Stochastic quasi-likelihood for case-control point pattern data," *J. Am. Stat. Assoc.*, 114, 631–644.

Yin, F., Jiao, J., Yan, J., and Hu, G. (2022), "Bayesian nonparametric learning for point process with spatial homogeneity: A spatial analysis of NBA shot locations," in *ICML 2022*.

Yue, Y. R. and Loh, J. M. (2011), "Bayesian semiparametric intensity estimation for inhomogeneous spatial point processes," *Biometrics*, 67, 937–946.

Zhang, Y., Kong, Q., and Zhou, F. (2023), "Integration-free training for spatio-temporal multimodal covariate deep kernel point processes," in *NeurIPS 2023*.

Zhu, S., Wang, H., Dong, Z., Cheng, X., and Xie, Y. (2022), "Neural spectral marked point processes." in *ICLR 2022*.

Zuo, S., Jiang, H., Li, Z., Zhao, T., and Zha, H. (2020), "Transformer Hawkes process." in *ICML 2020*.