

XGBoostPP: Tree-based Estimation of Point Process Intensity Functions

Changqing Lu¹ Yongtao Guan^{2*} Marie-Colette van Lieshout^{3,1} Ganggang Xu⁴

¹University of Twente, Enschede, The Netherlands

²The Chinese University of Hong Kong, Shenzhen, China

³Centrum Wiskunde & Informatica, Amsterdam, The Netherlands

⁴Univeristy of Miami, Coral Gables, FL, USA

Abstract

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 & Guestrin, 2016) to the point process literature via two carefully designed loss functions. The first loss is based on the Poisson likelihood, working for general point processes. The second loss is based on the weighted Poisson likelihood, where spatially dependent weights are introduced to further improve the estimation efficiency for clustered processes. An efficient greedy search algorithm is 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 existing approaches when the dimension of the covariate space is high, revealing the advantages of tree-based ensemble methods in estimating complex intensity functions.

1 Introduction

Point process models are increasingly popular for analyzing point pattern data across diverse disciplines, including tourism (e.g., D’Angelo et al., 2022), sociology (e.g., Xu et al., 2017; Zhang et al., 2023), criminology (e.g., Yin & Sang, 2021; Zhu & Xie, 2022) and epidemiology (e.g., Dong et al., 2023; Schoenberg, 2023). A crucial aspect of point process analysis is the estimation of the intensity, which defines the likelihood of event occurrences over the observation window. Existing literature explores two primary scenarios for point process intensity estimation: one based on the Cartesian coordinates of events (e.g., Diggle, 1985) and the other involving spatial covariates (e.g., Guan, 2008; Baddeley et al., 2012). This paper concentrates on the latter, emphasizing the significance of spatial covariates in explaining intensity variation.

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 (see, e.g., Schoenberg, 2005; Waagepetersen & Guan, 2009; Guan et al., 2015; Hessellund et al., 2022; 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

*Corresponding Author. guanyongtao@cuhk.edu.cn

in the covariate space and are computationally efficient (Baddeley et al., 2015). However, due to high local variability, the estimation accuracy of KIEs may not be as good as other nonparametric estimators. 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 through MCMC algorithms (Møller & 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 & Bishop, 2017; Aglietti et al., 2019), although the computational cost remains a significant challenge.

Another 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, typically 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) proposed in 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 the 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 (see, e.g., Lu et al., 2023). This surge underscores the necessity for an approach adept at handling the dimensionality of covariates in nonparametric intensity estimation. Tree-based models have proven to excel in predicting complex relationships between numerous covariates and the response variable (see, e.g., Grinsztajn et al., 2022). Moreover, ensemble methods, such as random forests (Breiman, 2001) and gradient boosting machines (Natekin & 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.

To fill the gap, in this study, we introduce a novel tree-based approach, XGBoostPP, tailored for nonparametrically estimating the intensity of a point process over a high-dimensional covariate space. Numerical experiments show that our approach demonstrates comparable performance to the existing methods when only a small number of covariates are available. However, as the covariate number increases, XGBoostPP significantly outperforms other approaches that may either encounter prohibitive computational challenges or produce significantly less accurate estimates. Finally, we apply XGBoostPP to forestry and fire data, further emphasizing its effectiveness in real-world scenarios.

2 Background and Literature

In this section, we provide a brief overview of point process theory, along with a summary of nonparametric intensity estimation approaches in the literature.

2.1 Point Process Theory

Consider a point process X defined on a bounded domain $\mathcal{S} \subset \mathbb{R}^d$. 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 ds denote an infinitesimal ball centred at $s \in \mathcal{S}$. Following Diggle (2003), we define the intensity function of X as $\lambda(s) = \lim_{|ds| \rightarrow 0} \mathbb{E}[N(ds)]/|ds|$ and the second-order intensity function as

$$\lambda^{(2)}(\mathbf{x}, \mathbf{y}) = \lim_{|\mathbf{dx}|, |\mathbf{dy}| \rightarrow 0} \frac{\mathbb{E}[N(\mathbf{dx})N(\mathbf{dy})]}{|\mathbf{dx}||\mathbf{dy}|}.$$

By scaling, one obtains the pair correlation function $g(\mathbf{x}, \mathbf{y}) = \lambda^{(2)}(\mathbf{x}, \mathbf{y})/[\lambda(\mathbf{x})\lambda(\mathbf{y})]$. Assuming second-order intensity-reweighted stationarity (Baddeley et al., 2000), it can be simplified to $g(r)$ at distance $r = \|\mathbf{x} - \mathbf{y}\|$.

Poisson process. The Poisson process is defined as the following: (i) for any bounded set \mathcal{B} , the number of events falling in it is Poisson distributed with mean $\int_{\mathcal{B}} \lambda(s) ds$; (ii) for disjoint bounded

sets, the numbers of events falling in them are independent. Under these assumptions, the likelihood of a realization \mathbf{x} from a Poisson process is of the form

$$\prod_{\mathbf{x} \in \mathbf{x}} \lambda(\mathbf{x}) \exp \left[- \int_{\mathcal{S}} \lambda(\mathbf{s}) d\mathbf{s} \right].$$

We remark that, for a 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 whose log-intensity is of the form $\log[\Lambda(\mathbf{s})] = \log[\tilde{\lambda}(\mathbf{s})] + Y(\mathbf{s})$, where $\tilde{\lambda}(\cdot)$ is a deterministic function of spatial covariates and $Y(\cdot)$ is a zero-mean Gaussian process with a covariance kernel $\rho(\cdot, \cdot)$. For instance, $\rho(\cdot, \cdot)$ can take, but is not limited to, the exponential form $\rho(\mathbf{x}, \mathbf{y}) = \tau^2 \exp(-\|\mathbf{x} - \mathbf{y}\|/\sigma)$ with τ^2 as the variance and σ as the scaling parameter. The intensity of such a process is given by $\lambda(\mathbf{s}) = \mathbb{E}[\Lambda(\mathbf{s})] = \tilde{\lambda}(\mathbf{s}) \exp(\tau^2/2)$, and the pair correlation function reads $g(r) = \exp[\tau^2 \exp(-r/\sigma)]$ which quickly approaches one as r becomes large.

Neyman-Scott Process. A Neyman-Scott process is generated in two stages. First, a set of parents, denoted by \mathbf{c} , are produced from a homogeneous Poisson process of intensity κ . Then, for each parent $\mathbf{a} \in \mathbf{c}$, a set of offspring points, denoted by $X_{\mathbf{a}}$, are generated independently from a Poisson process with an intensity function $\lambda_{\mathbf{a}}(\mathbf{s}) = k(\mathbf{s} - \mathbf{a}; \sigma) \tilde{\lambda}(\mathbf{s})$, where $k(\cdot; \sigma)$ is a density function parameterized by a spreading scalar σ . In particular, the Thomas process assumes $k(\cdot; \sigma)$ to be an isotropic Gaussian kernel with a standard deviation σ . As a result, the overall intensity function of such a process is given by $\lambda(\mathbf{s}) = \kappa \tilde{\lambda}(\mathbf{s})$, 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 grows large.

2.2 Nonparametric Estimation Approaches

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

Gaussian Cox Process Approaches. Gaussian Cox process approaches (Cunningham et al., 2008; Adams et al., 2009; Gunter et al., 2014; Samo & Roberts, 2014; Lloyd et al., 2015; Walder & Bishop, 2017; Donner & Oppen, 2018; John & 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 & 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 the augmented permanental process approach (Kim et al., 2022). APP 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.

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 spatial covariate vector $\mathbf{z}(\mathbf{s})$ with $\mathbf{s} \in \mathcal{S}$. To estimate the complex mapping $\lambda[\mathbf{z}(\mathbf{s})] : \mathbb{R}^p \rightarrow \mathbb{R}^+$, we propose a tree-based ensemble model, XGBoostPP, following the classic scalable tree boosting system – XGBoost (Chen & Guestrin, 2016). The standard XGBoost

was designed for general regression or classification tasks and is not directly applicable to point pattern data with specific spatial structures. Hence, we propose two carefully designed likelihood-based loss functions and an efficient learning algorithm to adapt to the point process context.

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

$$\log \{ \lambda [z(s)] \} = \sum_{k=1}^K f_k [z(s)], \quad f_k \in \mathcal{F}, \quad (1)$$

where $\mathcal{F} = \{f[z(s)] = \theta_{q[z(s)]}\}$ is the space of regression trees with $q[z(s)] : \mathbb{R}^p \rightarrow \{1, \dots, N_q\}$ and $\theta_{q[z(s)]} \in \mathbb{R}^{N_q}$. Here, q represents a tree structure working as a function that maps the covariate information at a location s to a corresponding leaf index, N_q is the number of leaves in this tree and θ_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 θ_{q_k} . In the remainder, we denote the estimated intensity function as $\lambda(x; \mathbf{f})$ with $\mathbf{f} = (f_1, \dots, f_K)$ and write θ_k for θ_{q_k} with θ_{kv} representing the element of θ_k on leaf v . Given a location s , the decision rules of the trees in \mathbf{f} first classify $z(s)$ 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

To estimate the structures of the K tree predictors in the XGBoostPP model, as well as their leaf scores, we propose two likelihood-based loss functions: the Poisson likelihood loss and the weighted Poisson likelihood loss.

3.2.1 Poisson Likelihood Loss

It is well-known that a parametric intensity function defined as $\lambda(s; \beta) = \exp[\beta^\top z(s)]$ can be consistently estimated by maximizing the Poisson log-likelihood (Schoenberg, 2005)

$$\sum_{x \in \mathbf{x}} \log [\lambda(x; \beta)] - \int_S \lambda(s; \beta) ds,$$

even when the underlying point process is not Poisson. Motivated by this phenomenon, we propose to approximate the optimal tree predictors \mathbf{f} in model (1) by minimizing the following penalized loss function

$$L(\mathbf{f}) = \sum_{k=1}^K \Omega(f_k) - \sum_{x \in \mathbf{x}} \phi(x; \mathbf{f}) + \int_S \exp [\phi(s; \mathbf{f})] ds, \quad (2)$$

where $\phi(s; \mathbf{f}) = \sum_{k=1}^K f_k [z(s)]$ and $\Omega(f_k) = \gamma \sum_v |\theta_{kv}|$ with some $\gamma > 0$. The last two terms of (2) correspond to the negative Poisson log-likelihood and the first term penalizes the complexity of \mathbf{f} to avoid overfitting.

Note that one needs to minimize (2) with respect to both tree structures, i.e. $\{q_k\}_{k=1}^K$, and leaf scores, i.e. $\{\theta_k\}_{k=1}^K$, simultaneously. However, it is impractical to exhaust all possible tree structures. Thus, for feasible computation, we develop a greedy additive search algorithm, as illustrated in Sec. 3.3, for the optimal intensity estimate.

3.2.2 Weighted Poisson Likelihood Loss

When the underlying point process deviates from a Poisson process, e.g. the log-Gaussian Cox process and the Neyman-Scott process, maximizing the Poisson log-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 & Shen, 2010). Such a limitation also applies to the estimation of our XGBoostPP model when minimizing (2). To improve the estimation efficiency, we propose a weighted Poisson likelihood loss, inspired by the quasi-likelihood developed in Guan et al. (2015).

To motivate our proposal, we assume that the optimal tree structures under model (1) exist, denoted by $q^\mathcal{O}$, and examine the ‘oracle’ estimator for the associated optimal leaf scores that minimizes $\mathbb{E}[L(\mathbf{f})]$ in (2). In this case, all values of θ_{kv} that do not correspond to a ‘terminal’ node in $q^\mathcal{O}$ are

precisely set to zero. Our focus then shifts to the estimation efficiency of the scores on the remaining nodes, denoted by $\theta_{\mathcal{A}}$ with \mathcal{A} indicating the set of the true supporting leaves.

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

$$e_{\phi}(\theta_{\mathcal{A}}) = \gamma \text{sgn}(\theta_{\mathcal{A}}) + \sum_{\mathbf{x} \in \mathbf{x}} \frac{\partial \phi(\mathbf{x}; \theta_{\mathcal{A}})}{\partial \theta_{\mathcal{A}}} - \int_{\mathcal{S}} \frac{\partial \phi(\mathbf{x}; \theta_{\mathcal{A}})}{\partial \theta_{\mathcal{A}}} \lambda(\mathbf{s}; \theta_{\mathcal{A}}) d\mathbf{s} = \mathbf{0}.$$

Note that we change $\phi(\mathbf{x}; \mathbf{f})$ to $\phi(\mathbf{x}; \theta_{\mathcal{A}})$ due to the knowledge of $\mathbf{q}^{\mathcal{O}}$. According to Guan & Shen (2010), a direct generalization of $e_{\phi}(\theta_{\mathcal{A}}) = \mathbf{0}$ is to consider the following more general estimating equation

$$e_h(\theta_{\mathcal{A}}) = \gamma \text{sgn}(\theta_{\mathcal{A}}) + \sum_{\mathbf{x} \in \mathbf{x}} h(\mathbf{x}; \theta_{\mathcal{A}}) - \int_{\mathcal{S}} h(\mathbf{s}; \theta_{\mathcal{A}}) \lambda(\mathbf{s}; \theta_{\mathcal{A}}) d\mathbf{s} = \mathbf{0}, \quad (3)$$

where $h(\mathbf{s}; \theta_{\mathcal{A}})$ can be any measurable function defined on \mathcal{S} of the same dimension as $\theta_{\mathcal{A}}$. Guan et al. (2015) show that the optimal $h^*(\mathbf{s}; \theta_{\mathcal{A}})$ that minimizes the inverse Godambe information of the resulting estimator must satisfy the Fredholm integral equation of the second kind (Hackbusch, 1995), that is, for any $\mathbf{s} \in \mathcal{S}$,

$$\int_{\mathcal{S}} h^*(\mathbf{u}; \theta_{\mathcal{A}}) \lambda(\mathbf{u}; \theta_{\mathcal{A}}) [g(\mathbf{s} - \mathbf{u}) - 1] d\mathbf{u} + h^*(\mathbf{s}; \theta_{\mathcal{A}}) = \nabla \lambda(\mathbf{s}; \theta_{\mathcal{A}}) / \lambda(\mathbf{s}; \theta_{\mathcal{A}}),$$

where $\nabla \lambda(\mathbf{s}; \theta_{\mathcal{A}}) = \partial \lambda(\mathbf{s}; \theta_{\mathcal{A}}) / \partial \theta_{\mathcal{A}}$. However, solving for $h^*(\mathbf{s}; \theta_{\mathcal{A}})$ requires either numerical quadrature approximation (Nyström, 1930) or stochastic approximation (Xu et al., 2019); both are computationally expensive.

Alternatively, Guan & Shen (2010) consider the special case when

$$\int_{\mathcal{S}} h^*(\mathbf{u}; \theta_{\mathcal{A}}) \lambda(\mathbf{u}; \theta_{\mathcal{A}}) [g(\mathbf{s} - \mathbf{u}) - 1] d\mathbf{u} \simeq h^*(\mathbf{s}; \theta_{\mathcal{A}}) \lambda(\mathbf{s}; \theta_{\mathcal{A}}) \int_{\mathcal{S}} [g(\mathbf{s} - \mathbf{u}) - 1] d\mathbf{u},$$

assuming that $h^*(\mathbf{u}; \theta_{\mathcal{A}}) \lambda(\mathbf{u}; \theta_{\mathcal{A}}) \simeq h^*(\mathbf{s}; \theta_{\mathcal{A}}) \lambda(\mathbf{s}; \theta_{\mathcal{A}})$ for proximate pairs of locations and $g(\mathbf{s} - \mathbf{u}) \simeq 1$ for distant pairs, which thus yields a solution $h^*(\mathbf{s}; \theta_{\mathcal{A}}) = w^*(\mathbf{s}; \theta_{\mathcal{A}}) \nabla \lambda(\mathbf{s}; \theta_{\mathcal{A}}) / \lambda(\mathbf{s}; \theta_{\mathcal{A}})$ with a weight function

$$w^*(\mathbf{s}; \theta_{\mathcal{A}}) = \frac{1}{1 + \lambda(\mathbf{s}; \theta_{\mathcal{A}}) \int_{\mathcal{S}} [g(\mathbf{s} - \mathbf{u}) - 1] d\mathbf{u}}.$$

It is straightforward to show that solving (3) with $h^*(\mathbf{s}; \theta_{\mathcal{A}})$ is equivalent to minimizing the penalized weighted negative Poisson log-likelihood

$$\gamma \|\theta_{\mathcal{A}}\|_1 - \sum_{\mathbf{x} \in \mathbf{x}} w^*(\mathbf{s}; \theta_{\mathcal{A}}) \log [\lambda(\mathbf{x}; \theta_{\mathcal{A}})] + \int_{\mathcal{S}} w^*(\mathbf{s}; \theta_{\mathcal{A}}) \lambda(\mathbf{s}; \theta_{\mathcal{A}}) d\mathbf{s},$$

where $\|\cdot\|_1$ indicates the L^1 -norm of a vector.

The optimal tree structures $\mathbf{q}^{\mathcal{O}}$ are, of course, inaccessible in practice, but the derivation above provides useful insights into reducing the estimation variance of the leaf scores in general. Consequently, we can define the following penalized weighted loss function for our XGBoostPP model

$$L_w(\mathbf{f}) = \sum_{k=1}^K \Omega(f_k) - \sum_{\mathbf{x} \in \mathbf{x}} w(\mathbf{x}; \mathbf{f}) \phi(\mathbf{x}; \mathbf{f}) + \int_{\mathcal{S}} w(\mathbf{s}; \mathbf{f}) \exp [\phi(\mathbf{s}; \mathbf{f})] d\mathbf{s}, \quad (4)$$

where we set

$$w(\mathbf{s}; \mathbf{f}) = \frac{1}{1 + \exp [\phi(\mathbf{s}; \mathbf{f})] \int_{\mathcal{S}} [g(\mathbf{s} - \mathbf{u}) - 1] d\mathbf{u}}. \quad (5)$$

In practice, we may approximate $\int_{\mathcal{S}} [g(\mathbf{s} - \mathbf{u}) - 1] d\mathbf{u} \simeq \int_0^m 2\pi r [g(r) - 1] dr$ for some sufficiently large distance m . The latter equals $K(m) - \pi m^2$ where $K(\cdot)$ is Ripley's K-function (see, e.g., Lieshout, 2019).

3.3 Additive Training Algorithm

Given the impracticality of exhausting all possible tree structures in the minimization of equations (2) and (4), we adopt a sequential exploration in the space of \mathbf{f} , following the methodology of the standard XGBoost in Chen & Guestrin (2016). To maintain generality, we focus on the minimization of (4) in the subsequent demonstration.

3.3.1 Adding Regression Trees

Denote the estimated log-intensity over $k-1$ trees $\hat{\mathbf{f}}_{k-1} = \{\hat{f}_1, \dots, \hat{f}_{k-1}\}$ by $\hat{\phi}(\mathbf{s}; \hat{\mathbf{f}}_{k-1})$ with $\mathbf{s} \in \mathcal{S}$. At the k -th iteration, we add a tree predictor \hat{f}_k to minimize

$$L_w^{(k)}(f_k) = \Omega(f_k) - \sum_{\mathbf{x} \in \mathbf{x}} w_k(\mathbf{x}) \left[\hat{\phi}(\mathbf{x}; \hat{\mathbf{f}}_{k-1}) + f_k(\mathbf{x}) \right] + \int_{\mathcal{S}} w_k(\mathbf{s}) \exp \left[\hat{\phi}(\mathbf{s}; \hat{\mathbf{f}}_{k-1}) + f_k(\mathbf{s}) \right] d\mathbf{s},$$

where we iteratively update the spatial weight $w_k(\mathbf{s})$ to better approximate (5) according to

$$\hat{w}_k(\mathbf{s}) = \frac{\omega_k}{1 + \exp \left[\hat{\phi}(\mathbf{s}; \hat{\mathbf{f}}_{k-1}) \right] \left[\hat{K}(m) - \pi m^2 \right]}. \quad (6)$$

In practice, the normalizing scalar ω_k is chosen such that $\int_{\mathcal{S}} \hat{w}_k(\mathbf{s}) d\mathbf{s} = 1$ and Ripley's K-function is estimated by a standard nonparametric estimator (see, e.g., Møller & Waagepetersen, 2004)

$$\hat{K}(m) = \sum_{\mathbf{x}_1, \mathbf{x}_2 \in \mathbf{x}}^{\neq} \frac{1\{\|\mathbf{x}_1 - \mathbf{x}_2\|_2 \leq m\}}{\hat{\lambda}(\mathbf{x}_1; \hat{\mathbf{f}}^{\mathcal{P}}) \hat{\lambda}(\mathbf{x}_2; \hat{\mathbf{f}}^{\mathcal{P}}) |\mathcal{S} \cap \mathcal{S}_{\mathbf{x}_1 - \mathbf{x}_2}|},$$

where $\mathbf{x}_1, \mathbf{x}_2$ denote pairs of distinct points, $\|\cdot\|_2$ is the L_2 norm of a vector, $\hat{\lambda}(\cdot; \hat{\mathbf{f}}^{\mathcal{P}})$ represents the estimated intensity by XGBoostPP fitted using the Poisson likelihood loss (2) and $1/|\mathcal{S} \cap \mathcal{S}_{\mathbf{x}_1 - \mathbf{x}_2}|$ is the translation edge correction factor (Ohser & Stoyan, 1981). Note that, we manually set $\hat{K}(m) - \pi m^2$ to zero when it is negative.

By quickly applying a quadratic approximation to $L_w^{(k)}(f_k)$ and removing the constant terms involving $\hat{\phi}(\mathbf{s}; \hat{\mathbf{f}}_{k-1})$, the loss function at the k -th iteration can be simplified to

$$\tilde{L}_w^{(k)}(f_k) = \Omega(f_k) - \sum_{\mathbf{x} \in \mathbf{x}} \hat{w}_k(\mathbf{x}) f_k(\mathbf{x}) + \int_{\mathcal{S}} \hat{w}_k(\mathbf{s}) \times \exp \left[\hat{\phi}(\mathbf{s}; \hat{\mathbf{f}}_{k-1}) \right] \left[f_k(\mathbf{s}) + \frac{1}{2} f_k^2(\mathbf{s}) \right] d\mathbf{s}. \quad (7)$$

To approximate the above integral, we use a numerical quadrature approximation; see Appendix B for details.

3.3.2 Growing an Added Tree

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

To achieve this goal, we show in Appendix B that, for any given tree structure q_k , the optimal score $\hat{\theta}_{kv}$ on a leaf v , minimizing (7), has a closed-form expression

$$\hat{\theta}_{kv} = \frac{\text{sgn}(R_{kv} - T_{kv}) \max(|R_{kv} - T_{kv}| - \gamma, 0)}{T_{kv}},$$

where we use $I_{kv} = \{\mathbf{s} : q_k(\mathbf{s}) = v\}$ to denote the set of locations where $\mathbf{z}(\mathbf{s})$ belongs to leaf v and

$$R_{kv} = \sum_{\mathbf{x} \in \mathbf{x}} \hat{w}_k(\mathbf{x}) 1\{\mathbf{x} \in I_{kv}\}, \quad T_{kv} = \int_{\mathcal{S}} \hat{w}_k(\mathbf{s}) \exp \left[\hat{\phi}(\mathbf{s}; \hat{\mathbf{f}}_{k-1}) \right] 1\{\mathbf{s} \in I_{kv}\} d\mathbf{s}.$$

It is important to note that, when $\hat{\theta}_{kv} = 0$, this leaf will be eliminated from the tree. Hence, compared to the original weighted Poisson likelihood loss, the L_1 -penalized loss (4) tends to produce a smaller tree.

Consequently, for any given tree structure q_k , one can compute a performance score by plugging in all optimal leaf scores as above into the loss function (7). Utilizing these performance scores, one can find the best split that minimizes (7) among various candidate splits for each node and grow the tree sequentially. To exhaust the possible candidate splits at a node, we use the exact greedy algorithm illustrated in Algorithm 1 of Chen & Guestrin (2016).

3.3.3 More Stable Learning

To achieve a more stable learning performance, we also incorporate the shrinkage and subsampling techniques of the standard XGBoost (Chen & Guestrin, 2016). The former shrinks the impacts of the newly added tree by a parameter $\eta > 0$, which is 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 spatial 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. We randomly split data points into two parts and, at each time, use one part to train the model and then compute the Poisson log-likelihood based on the estimated intensities on the other test part. We summarize the Poisson likelihoods on two test parts as the performance measure. To further reduce the randomness in data splitting, we repeat such a procedure three times and report the averaged test Poisson log-likelihoods as the final evaluation metric for choosing the optimal combination of hyperparameters.

4 Simulation Study

To evaluate the performance of XGBoostPP, we conduct a simulation study on synthetic data generated from the point process models introduced in Sec. 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) and the reweight (re) methods (Baddeley et al., 2012) and select the bandwidths by Silverman’s rule of thumb (Silverman, 2018). It is again worth noting that KIEs 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). We find that the naive kernel also encounters computation problems when estimating the intensity function over a high-dimensional covariate space.

We use two evaluation metrics to quantitatively measure the performance of XGBoostPP and other approaches: the Poisson log-likelihood on new randomly generated test data under the same point process model (reported in the main text) and the integrated absolute error from true intensities (reported in Appendix C). For each scenario, we report the averaged values and include the standard deviations of the two metrics across 500 simulation runs. Note that, in all tables throughout this paper, the ‘XGBoostPP’ terms with subscripts ‘p’ and ‘wp’ represent the models fitted using the Poisson likelihood loss function and the weighted Poisson likelihood loss function, respectively.

In all simulation runs, the hyperparameters of the XGBoostPP model – K , γ and η – are chosen through the two-fold cross-validation method in Sec. 3.4 from the candidate sets $\{1, \dots, 600\}$, $\{10, 30, 50\}$ 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

In this scenario, the intensity function of the point process is determined by two covariates, denoted by $z_1(\mathbf{s})$ and $z_2(\mathbf{s})$. For the Poisson process, we set $\lambda(\mathbf{s}) = \alpha \exp[\beta z_1(\mathbf{s}) + \beta z_2(\mathbf{s})]$, where $\beta = 0.5$ or 1.0 and α is chosen such that the expected number of events in $S = [0, 1]^2$ is 400. For the log-Gaussian Cox process, we simulate point pattern data from a Poisson process with the random intensity $\Lambda(\mathbf{s}) = \alpha \exp[\beta z_1(\mathbf{s}) + \beta z_2(\mathbf{s}) + Y(\mathbf{s})]$, where $Y(\mathbf{s})$ 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 .

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

COVS	POISSON PROCESS	$\beta = 0.5$	$\beta = 1.0$
2 (§4.1)	True	2089.4	2290.3
	KIE _{ra}	2072.9(3.2)	2226.6(5.9)
	KIE _{re}	2049.6(3.9)	2184.6(6.6)
	APP _{naive}	2077.7(3.9)	2273.6 (4.3)
	APP _{rfm}	2077.8 (3.9)	2273.5(4.2)
	XGBoostPP _p	2073.2(3.0)	2261.4(5.0)
	XGBoostPP _{wp}	2073.2(3.0)	2261.6(5.3)
10 (§4.3)	APP _{rfm}	1943.8(15.0)	2127.3(14.4)
	XGBoostPP _p	2062.3(4.5)	2245.4 (5.5)
	XGBoostPP _{wp}	2062.4 (4.5)	2245.4 (5.5)
COVS	POISSON PROCESS	$\beta = 0.2$	$\beta = 0.4$
10 (§4.2)	True	2048.1	2160.9
	APP _{rfm}	1857.9(41.2)	1962.2(18.0)
	XGBoostPP _p	2025.9 (3.8)	2115.9 (6.3)
	XGBoostPP _{wp}	2025.9 (3.8)	2115.9 (6.3)

For the Neyman-Scott process, we generate the parents from a homogeneous Poisson process with intensity $\kappa = 100$ or 200 and, for each parent, simulate the offspring points using a Gaussian kernel with a standard deviation $\sigma = 0.02$ or 0.04 . Following Guan & Shen (2010), the distance m used to approximate the weights (6) is set to 0.06 for the Poisson process and to 3σ for the other two processes.

We report the averaged Poisson log-likelihood results on Poisson test data in Tab. 1 and those on log-Gaussian Cox and Neyman-Scott test data in Tab. 2 and 3. In general, when there are only two covariates, XGBoostPP achieves competitive performance on all types of data under various parameter settings, although APP mostly exhibits the best behavior. 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 the log-Gaussian Cox process and small κ and σ for the Neyman-Scott process). Moreover, comparing XGBoostPP_p with XGBoostPP_{wp}, it is evident that the performance improves on clustered processes while it remains almost unchanged on Poisson processes. This indicates that the practical training algorithm designed in Sec. 3.3 works well when the clustering feature of a point pattern is unknown. More specifically, these improvements are greater when point patterns are more spatially heterogeneous and clustered, aligning with the findings in Guan & Shen (2010).

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 $z_1(\mathbf{s}), \dots, z_{10}(\mathbf{s})$. The intensity function of the point process is now defined as $\lambda(\mathbf{s}) = \alpha \exp(\beta\{z_1(\mathbf{s}) + \frac{1}{2}z_2(\mathbf{s})z_3(\mathbf{s}) + \frac{1}{6}\exp[z_4(\mathbf{s})] + \frac{1}{2}z_5(\mathbf{s})^2 + 3\sin[z_6(\mathbf{s})]\})$. Note that $z_7(\mathbf{s}), \dots, z_{10}(\mathbf{s})$ are employed as nuisance variables which indeed exist often in real-world applications. We set σ, τ^2, κ to the same values as in Sec. 4.1 while $\beta = 0.2$ or 0.4 to balance the spatial heterogeneity of point patterns to a reasonable scale. Considering that the detection of clustering patterns over complex covariate relationships is difficult, we change m to 0.04 for the Poisson process and to 2σ for the log-Gaussian Cox and the Neyman-Scott processes.

The relevant evaluation results are also reported in Tab. 1, 2 and 3. When there are ten covariates, the performance of APP, with the ‘rfm’ kernel, deteriorates considerably, while XGBoostPP maintains good performance. Moreover, the standard deviations given by APP tend to be larger than those by XGBoostPP, indicating a more stable estimation accuracy for XGBoostPP. Comparing XGBoostPP_p with XGBoostPP_{wp}, the latter again shows better performance and the improvements are, overall, slightly higher for more 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 Covariates

In this section, we simulate data of the same models as those in Sec. 4.1, using the simple function $\lambda(\mathbf{s}) = \alpha \exp[\beta z_1(\mathbf{s}) + \beta z_2(\mathbf{s})]$. However, we input the ten covariates from Sec. 4.2,

Table 2: Averaged Poisson log-likelihoods (standard deviations) of different approaches on log-Gaussian Cox test data.

COVS	LOG-GAUSSIAN COX PROCESS	$\tau^2 = 1$		$\tau^2 = 2$	
		$\sigma = 0.02$	$\sigma = 0.04$	$\sigma = 0.02$	$\sigma = 0.04$
$\beta = 0.5$					
2 (\$4.1)	True	2070.2	2080.5	2089.7	2102.5
	KIE _{ra}	2052.7(5.4)	2060.5(7.8)	2071.5(7.1)	2077.4(12.8)
	KIE _{re}	2029.4(6.5)	2037.1(8.7)	2049.0(8.2)	2051.8(14.1)
	APP _{naive}	2057.2(6.6)	2064.3(9.3)	2074.5(8.2)	2075.7(17.1)
	APP _{rfm}	2057.3 (6.6)	2064.6 (9.2)	2074.8 (8.2)	2076.2(16.7)
	XGBoostPP _p	2051.8(5.0)	2059.0(8.0)	2069.0(6.9)	2074.0(14.7)
10 (\$4.3)	XGBoostPP _{wp}	2052.6(5.7)	2060.9(7.7)	2072.1(6.8)	2078.9 (12.9)
	APP _{rfm}	1909.3(21.7)	1901.5(28.9)	1907.1(28.3)	1872.4(42.0)
	XGBoostPP _p	2042.5(6.5)	2049.4(8.8)	2059.0(11.0)	2054.5(25.0)
$\beta = 1.0$					
2 (\$4.1)	True	2304.8	2269.1	2256.0	2283.2
	KIE _{ra}	2238.8(11.4)	2202.3(15.7)	2188.2(15.1)	2211.1(22.8)
	KIE _{re}	2196.2(12.5)	2159.4(16.2)	2144.2(16.5)	2166.7(23.5)
	APP _{naive}	2284.7(8.2)	2244.0(12.6)	2229.6(13.4)	2243.4(27.7)
	APP _{rfm}	2284.7 (8.1)	2244.0 (12.5)	2229.9 (12.8)	2244.4(25.9)
	XGBoostPP _p	2272.2(8.0)	2231.8(11.8)	2219.4(11.6)	2234.5(21.6)
10 (\$4.3)	XGBoostPP _{wp}	2275.8(8.7)	2235.5(12.9)	2228.8(9.6)	2244.9 (18.6)
	APP _{rfm}	2116.5(22.6)	2062.9(30.7)	2038.5(31.7)	2021.0(51.3)
	XGBoostPP _p	2256.4(8.3)	2214.6(12.6)	2199.3(14.7)	2207.3(32.8)
$\beta = 0.2$					
10 (\$4.2)	True	2032.9	2066.7	2080.8	2080.7
	APP _{rfm}	1848.3(31.6)	1871.8(30.8)	1879.9(34.4)	1838.5(42.0)
	XGBoostPP _p	2011.5(4.3)	2041.7(7.0)	2054.1(11.1)	2040.5(19.0)
	XGBoostPP _{wp}	2012.8 (4.2)	2042.4 (7.0)	2056.0 (7.8)	2042.3 (15.0)
$\beta = 0.4$					
10 (\$4.2)	True	2134.6	2181.9	2153.3	2171.0
	APP _{rfm}	1923.8(24.5)	1944.8(30.6)	1916.3(30.3)	1887.6(45.7)
	XGBoostPP _p	2092.0(7.8)	2130.7(12.4)	2106.2(12.4)	2105.4(24.6)
	XGBoostPP _{wp}	2096.0 (8.7)	2133.5 (11.8)	2112.6 (10.8)	2110.6 (20.3)

$z_1(\mathbf{s}), \dots, z_{10}(\mathbf{s})$, to estimate the intensity, thus with eight nuisance covariates. Such a scenario is common in practice when numerous covariates are available for point process intensity estimation while researchers are uncertain about which ones are most relevant. We again evaluate XGBoostPP and APP with the two performance metrics, the Poisson log-likelihood on new generated data and the integrated absolute error, and report the relevant results. From Tab. 1, 2 and 3, it is obvious that XGBoostPP exhibits much more robust behavior in the presence of many nuisance covariates compared to APP, which can be evidenced by the significantly smaller reductions in the test Poisson log-likelihoods. Such robustness suggests great non-parametric potential of XGBoostPP in practical applications.

4.4 Poisson Toy Examples

In addition to the main simulation study, we test XGBoostPP on three Poisson toy examples in Appendix D to intuitively visualize its flexibility in modeling various nonlinear relationships between covariates and the intensity function.

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 (BCI) in Panama and kitchen fire data in the Twente region in The Netherlands. 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., randomly splitting the data points into four subsets, assigning one for test and the others for training). Details on the covariate

Table 3: Averaged Poisson log-likelihoods (standard deviations) of different approaches on Neyman-Scott test data.

COVS	NEYMAN-SCOTT PROCESS	$\kappa = 100$		$\kappa = 200$	
		$\sigma = 0.02$	$\sigma = 0.04$	$\sigma = 0.02$	$\sigma = 0.04$
$\beta = 0.5$					
2 (\$4.1)	True	2090.9	2074.8	2091.2	2071.7
	KIE _{ra}	2071.1(8.0)	2055.2(6.9)	2072.2(6.2)	2053.4(5.1)
	KIE _{re}	2047.6(10.0)	2031.0(8.0)	2049.7(7.4)	2030.4(6.0)
	APP _{naive}	2073.3(10.4)	2060.2(8.0)	2077.5(7.2)	2059.0(5.9)
	APP _{rfm}	2073.6 (10.2)	2060.4 (8.0)	2077.6 (7.2)	2059.1 (5.8)
	XGBoostPP _p	2068.0(8.0)	2054.6(6.9)	2070.7(6.1)	2052.8(4.8)
10 (\$4.3)	XGBoostPP _{wp}	2071.2(8.1)	2056.2(7.7)	2072.6(6.7)	2053.1(5.0)
	APP _{rfm}	1890.8(34.2)	1905.0(26.5)	1919.6(24.6)	1912.1(21.0)
	XGBoostPP _p	2057.6(11.5)	2044.4(8.4)	2061.4(7.3)	2043.0(6.2)
	XGBoostPP _{wp}	2060.9 (9.8)	2046.4 (8.3)	2064.0 (7.7)	2043.2 (6.3)
$\beta = 1.0$					
2 (\$4.1)	True	2263.1	2255.6	2272.5	2306.9
	KIE _{ra}	2195.3(17.5)	2190.0(14.2)	2206.2(13.5)	2238.9(10.8)
	KIE _{re}	2151.6(19.5)	2148.3(14.8)	2162.8(14.8)	2194.9(10.9)
	APP _{naive}	2233.6(16.0)	2233.5(11.8)	2248.9 (11.7)	2285.6(8.1)
	APP _{rfm}	2234.4 (15.1)	2233.6 (11.6)	2248.9(11.9)	2285.8 (8.1)
	XGBoostPP _p	2222.1(13.9)	2219.9(10.8)	2236.4(10.6)	2273.2(7.7)
10 (\$4.3)	XGBoostPP _{wp}	2234.1(12.6)	2223.7(11.2)	2246.2(9.5)	2274.2(8.4)
	APP _{rfm}	2022.2(39.8)	2054.3(31.8)	2063.8(31.7)	2120.1(24.9)
	XGBoostPP _p	2202.1(17.2)	2203.3(11.8)	2220.9(11.4)	2257.2(8.5)
	XGBoostPP _{wp}	2211.7 (19.9)	2204.6 (11.9)	2228.7 (11.9)	2257.2 (8.5)
$\beta = 0.2$					
10 (\$4.2)	True	1977.8	1963.7	2070.6	2047.9
	APP _{rfm}	1777.1(33.2)	1786.8(27.5)	1884.5(27.5)	1865.6(29.5)
	XGBoostPP _p	1950.2(10.3)	1938.4(6.3)	2048.1(5.1)	2025.8(4.8)
	XGBoostPP _{wp}	1952.1 (8.1)	1939.8 (6.1)	2049.2 (5.4)	2026.1 (4.7)
$\beta = 0.4$					
10 (\$4.2)	True	2227.0	2214.4	2128.7	2175.2
	APP _{rfm}	1955.6(39.4)	1980.7(32.3)	1916.2(30.2)	1961.8(24.3)
	XGBoostPP _p	2173.2(13.0)	2163.9(12.0)	2087.4(8.1)	2128.3(8.4)
	XGBoostPP _{wp}	2176.6 (16.0)	2166.0 (11.8)	2089.0 (8.4)	2128.4 (8.4)

information in the two data sets and the calculation of the test Poisson log-likelihood metric used here are given in Appendix E.1 and E.2.

Table 4: Cross-validated (four folds) test Poisson log-likelihoods of different approaches on ‘Bei’, ‘Capp’ and ‘Fire’ data.

DATA SET	BEI	CAPP	FIRE
COVS	8	8	29
APP _{rfm}	−24955.4	−27900.1	−
XGBoostPP _p	−24557.6	−27701.7	−10773.2
XGBoostPP _{wp}	−24376.1	−27684.9	−10519.7

BCI Forestry Data. From the BCI forestry data, we select two tree species to study: Beilschmiedia pendula (Bei) and Capparis frondosa (Capp), containing 3604 and 3299 tree locations, respectively. We investigate eight covariates that may contribute to explaining 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. Tab.4 indicates that, on both ‘Bei’ and ‘Capp’ data, XGBoostPP outperforms APP. Since ‘Bei’ appears more clustered compared to ‘Capp’ (see, Guan & Shen, 2010; Yue & Loh, 2011), XGBoostPP_{wp} improves the performance on the former relatively more. Fig.1 displays the estimated log-intensities, showing that, in comparison to APP, XGBoostPP produces significantly different estimates for the areas with a small number of event observations.

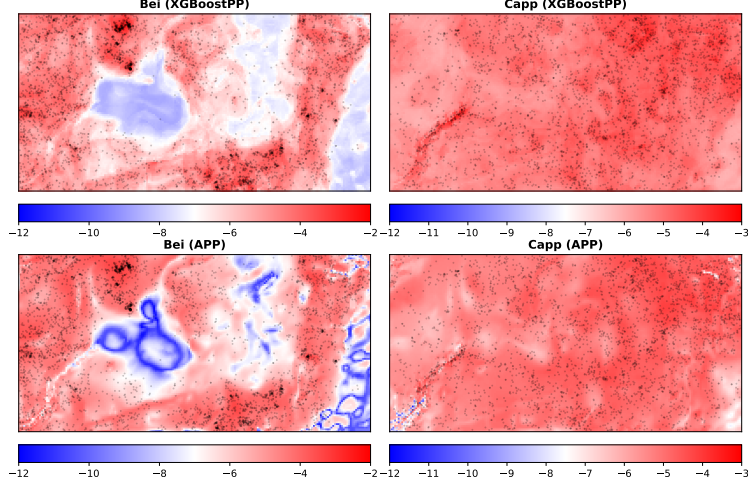


Figure 1: Observed tree locations and estimated log-intensities by $\text{XGBoostPP}_{\text{wp}}$ and APP over 8 covariates.

Kitchen Fire Data. The fire data (Fire) comprises 699 kitchen fire locations in Twente from 2004 to 2020. The covariates of interest include building information, urbanity degrees, population components, and energy consumption (29 covariates in total). Identifying the most relevant covariates for fire occurrences enables firefighters to organize public campaigns for fire prevention more effectively. Tab.4 provides estimation results for XGBoostPP only, as APP reports computational errors when fitted with 29 covariates. On ‘Fire’ data, XGBoostPP demonstrates its strong capacity to handle covariate spaces of high dimensions. As expected, $\text{XGBoostPP}_{\text{wp}}$ again yields a higher test Poisson log-likelihood due to the apparent spatial dependence observed in the raw data. Finally, the estimated log-intensity for ‘Fire’ by XGBoostPP is presented in Fig.2.

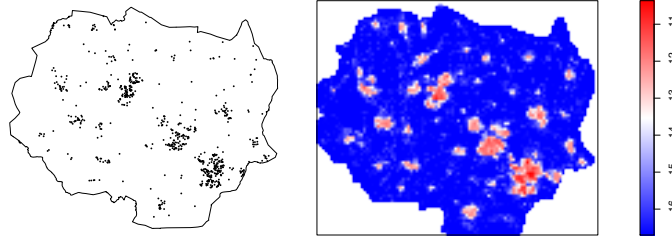


Figure 2: Kitchen fire locations and estimated log-intensity by $\text{XGBoostPP}_{\text{wp}}$ over 29 covariates.

6 Conclusions

In this paper, we proposed a novel tree-based ensemble method, named XGBoostPP, to nonparametrically estimate the intensity of a point process as a function of spatial covariates. Two loss functions were carefully designed for model estimation. The first one is the Poisson likelihood loss, working for general point processes. However, when the underlying process is not Poisson, this loss can be inefficient. The second one is the weighted Poisson likelihood loss, where a weight function is introduced to account for potential spatial dependence in order to further improve the estimation efficiency for clustered processes. We developed an efficient greedy search algorithm for model fitting and proposed a two-fold cross-validation procedure to select hyperparameters.

To show the effectiveness of XGBoostPP, we performed extensive simulation studies on data over low- and higher-dimensional covariate spaces and demonstrated its advantages against existing approaches. In particular, XGBoostPP achieves superior performance when the dimension of the covariate space

is large. The applications on the two real-world data sets reveal that a well-constructed and -tuned XGBoostPP can flexibly analyze unknown, complex point patterns in practice.

For 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 weighted Poisson 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.

References

- Adams, R. P., Murray, I., and MacKay, D. J. C. Tractable nonparametric Bayesian inference in Poisson processes with Gaussian process intensities. In *Proceedings of the 26th International Conference on Machine Learning*, pp. 9–16, Montreal, Canada, 2009.
- Aglietti, V., Bonilla, E. V., Damoulas, T., and Cripps, S. Structured variational inference in continuous Cox process models. In *Advances in Neural Information Processing Systems 32*, pp. 12458–12468, Vancouver, Canada, 2019.
- Baddeley, A., Møller, J., and Waagepetersen, R. Non and semi-parametric estimation of interaction in inhomogeneous point patterns. *Statistica Neerlandica*, 54:329–350, 2000.
- Baddeley, A., Rubak, E., and Turner, R. *Spatial Point Patterns: Methodology and Applications with R*. Chapman and Hall/CRC, Boca Raton, 2015.
- Baddeley, A. J., Chang, Y., Song, Y., and Turner, R. Nonparametric estimation of the dependence of a spatial point process on spatial covariates. *Statistics and Its Interface*, 5:221–236, 2012.
- Breiman, L. Random forests. *Machine Learning*, 45:5–32, 2001.
- Chen, T. and Guestrin, C. XGBoost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 785–794, New York, NY, 2016.
- Cronie, O. and Van Lieshout, M. N. M. A non-model-based approach to bandwidth selection for kernel estimators of spatial intensity functions. *Biometrika*, 105:455–462, 2018.
- Cunningham, J. P., Shenoy, K. V., and Sahani, M. Fast Gaussian process methods for point process intensity estimation. In *Proceedings of the 25th International Conference on Machine Learning*, pp. 192–199, Helsinki, Finland, 2008.
- Diggle, P. A kernel method for smoothing point process data. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 34:138–147, 1985.
- Diggle, P. J. *Statistical Analysis of Spatial Point Patterns*. Oxford University Press, New York, 2003.
- Dong, Z., Zhu, S., Xie, Y., Mateu, J., and Rodríguez-Cortés, F. J. Non-stationary spatio-temporal point process modeling for high-resolution COVID-19 data. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 72:368–386, 2023.
- Donner, C. and Opper, M. Efficient Bayesian inference of sigmoidal Gaussian Cox processes. *Journal of Machine Learning Research*, 19:1–34, 2018.
- D’Angelo, N., Adelfio, G., Abbruzzo, A., and Mateu, J. Inhomogeneous spatio-temporal point processes on linear networks for visitors’ stops data. *Annals of Applied Statistics*, 16:791–815, 2022.
- Flaxman, S., Teh, Y. W., and Sejdinovic, D. Poisson intensity estimation with reproducing kernels. In *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, pp. 270–279, Fort Lauderdale, FL, 2017.
- Grinsztajn, L., Oyallon, E., and Varoquaux, G. Why do tree-based models still outperform deep learning on typical tabular data? In *Advances in Neural Information Processing Systems 35*, pp. 507–520, New Orleans, LA, 2022.

- Guan, Y. On consistent nonparametric intensity estimation for inhomogeneous spatial point processes. *Journal of the American Statistical Association*, 103:1238–1247, 2008.
- Guan, Y. and Shen, Y. A weighted estimating equation approach for inhomogeneous spatial point processes. *Biometrika*, 97:867–880, 2010.
- Guan, Y. and Wang, H. Sufficient dimension reduction for spatial point processes directed by Gaussian random fields. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 72: 367–387, 2010.
- Guan, Y., Jalilian, A., and Waagepetersen, R. Quasi-likelihood for spatial point processes. *Journal of the Royal Statistical Society. Series B (Statistical Methodology)*, 77:677–697, 2015.
- Gunter, T., Lloyd, C., Osborne, M. A., and Roberts, S. J. Efficient Bayesian nonparametric modelling of structured point processes. In *Proceedings of the 30th Conference on Uncertainty in Artificial Intelligence*, pp. 310–319, Quebec, Canada, 2014.
- Hackbusch, W. *Integral Equations: Theory and Numerical Treatment*. Birkhauser Verlag, Basel, 1995.
- Hessellund, K. B., Xu, G., Guan, Y., and Waagepetersen, R. Semiparametric multinomial logistic regression for multivariate point pattern data. *Journal of the American Statistical Association*, 117: 1500–1515, 2022.
- Illian, J. B., Sørbye, S. H., and Rue, H. A toolbox for fitting complex spatial point process models using integrated nested Laplace approximation (INLA). *Annals of Applied Statistics*, 6:1499–1530, 2012.
- John, S. and Hensman, J. Large-scale Cox process inference using variational Fourier features. In *Proceedings of the 35th International Conference on Machine Learning*, pp. 2367–2375, Stockholm, Sweden, 2018.
- Kim, H., Asami, T., and Toda, H. Fast Bayesian estimation of point process intensity as function of covariates. In *Advances in Neural Information Processing Systems 35*, pp. 25711–25724, New Orleans, LA, 2022.
- Lieshout, M. N. M. *Theory of Spatial Statistics: A Concise Introduction*. Chapman and Hall/CRC, Boca Raton, 2019.
- Lloyd, C., Gunter, T., Osborne, M. A., and Roberts, S. J. Variational inference for Gaussian process modulated Poisson processes. In *Proceedings of the 32th International Conference on Machine Learning*, pp. 1814–1822, Lille, France, 2015.
- Lu, C., Lieshout, M. N. M., Graaf, M., and Visscher, P. Data-driven chimney fire risk prediction using machine learning and point process tools. *Annals of Applied Statistics*, 17:3088–3111, 2023.
- McCullagh, P. and Møller, J. The permanental process. *Advances in Applied Probability*, 38:873–888, 2006.
- Møller, J. and Waagepetersen, R. *Statistical Inference and Simulation for Spatial Point Processes*. Chapman and Hall/CRC, Boca Raton, 2004.
- Møller, J., Syversveen, A. R., and Waagepetersen, R. P. Log Gaussian Cox processes. *Scandinavian Journal of Statistics*, 25:451–482, 1998.
- Natekin, A. and Knoll, A. Gradient boosting machines, a tutorial. *Frontiers in neurorobotics*, 7:21, 2013.
- Nyström, E. J. Über die Praktische Auflösung von Integralgleichungen mit Anwendungen auf Randwertaufgaben. *Acta Mathematica*, 54:185–204, 1930.
- Ohser, J. and Stoyan, D. On the second-order and orientation analysis of planar stationary point processes. *Biometrical Journal*, 23:523–533, 1981.

- Rue, H., Martino, S., and Chopin, N. Approximate Bayesian inference for latent Gaussian models by using integrated nested Laplace approximations. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 71:319–392, 2009.
- Samo, Y. K. and Roberts, S. Scalable nonparametric Bayesian inference on point processes with Gaussian processes. In *Proceedings of the 31st International Conference on Machine Learning*, pp. 2227–2236, Beijing, China, 2014.
- Schoenberg, F. Consistent parametric estimation of the intensity of a spatial-temporal point process. *Journal of Statistical Planning and Inference*, 128:79–93, 2005.
- Schoenberg, F. Some statistical problems involved in forecasting and estimating the spread of SARS-CoV-2 using Hawkes point processes and SEIR models. *Environmental and Ecological Statistics*, 30:851–862, 2023.
- Silverman, B. W. *Density Estimation for Statistics and Data Analysis*. Routledge, New York, 2018.
- Waagepetersen, R. and Guan, Y. Two-step estimation for inhomogeneous spatial point processes. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 71:685–702, 2009.
- Walder, C. J. and Bishop, A. N. Fast Bayesian intensity estimation for the permanental process. In *Proceedings of the 34th International Conference on Machine Learning*, pp. 3579–3588, Sydney, Australia, 2017.
- Xu, G., Waagepetersen, R., and Guan, Y. Stochastic quasi-likelihood for case-control point pattern data. *Journal of the American Statistical Association*, 114:631–644, 2019.
- Xu, G., Liang, C., Waagepetersen, R., and Guan, Y. Semiparametric goodness-of-fit test for clustered point processes with a shape-constrained pair correlation function. *Journal of the American Statistical Association*, 118:2072–2087, 2023.
- Xu, H., Luo, D., and Zha, H. Learning Hawkes processes from short doubly-censored event sequences. In *Proceedings of the 34th International Conference on Machine Learning*, pp. 3831–3840, Sydney, Australia, 2017.
- Yin, F., Jiao, J., Yan, J., and Hu, G. Bayesian nonparametric learning for point process with spatial homogeneity: a spatial analysis of NBA shot locations. In *Proceedings of the 39th International Conference on Machine Learning*, pp. 25523–25551, Baltimore, MD, 2022.
- Yin, L. and Sang, H. Fused spatial point process intensity estimation with varying coefficients on complex constrained domains. *Spatial Statistics*, 46:100547, 2021.
- Yue, Y. R. and Loh, J. M. Bayesian semiparametric intensity estimation for inhomogeneous spatial point processes. *Biometrics*, 67:937–946, 2011.
- Zhang, J., Cai, B., X, Z., Wang, H., Xu, G., and Guan, Y. Learning human activity patterns using clustered point processes with active and inactive states. *Journal of Business & Economic Statistics*, 41:388–398, 2023.
- Zhu, S. and Xie, Y. Spatiotemporal-textual point processes for crime linkage detection. *Annals of Applied Statistics*, 16:1151–1170, 2022.

A Motivation for the Weighted Poisson Likelihood Loss in Sec. 3.2

We motivate the weighted Poisson likelihood loss by carefully analyzing the estimation efficiency of the scores on the non-‘terminal’ node (true supporting leaves) under the ‘oracle’ tree structures of XGBoostPP.

Note that, with the knowledge of the optimal tree structures $\mathbf{q}^\mathcal{O}$, the XGBoostPP model can be interpreted as a generalized linear model over a vector of transformed covariates $\mathbf{Z}_\mathcal{A}(\mathbf{s})$ that denote the membership identities of a location \mathbf{s} with respect to the supporting tree leaves. Specifically,

$$\lambda(\mathbf{s}; \boldsymbol{\theta}_\mathcal{A}) = \exp[\boldsymbol{\theta}_\mathcal{A}^\top \mathbf{Z}_\mathcal{A}(\mathbf{s})].$$

In this setting, the estimated intensity function is 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

$$L(\mathbf{f}) = \sum_{k=1}^K \Omega(f_k) - \sum_{\mathbf{x} \in \mathbf{x}} \log [\lambda(\mathbf{x}; \mathbf{f})] + \int_{\mathcal{S}} \lambda(\mathbf{s}; \mathbf{f}) d\mathbf{s},$$

must satisfy the estimating equation

$$\mathbf{e}_\lambda(\boldsymbol{\theta}_\mathcal{A}) = \gamma \text{sgn}(\boldsymbol{\theta}_\mathcal{A}) + \sum_{\mathbf{x} \in \mathbf{x}} \mathbf{Z}_\mathcal{A}(\mathbf{s}) - \int_{\mathcal{S}} \lambda(\mathbf{s}; \boldsymbol{\theta}_\mathcal{A}) \mathbf{Z}_\mathcal{A}(\mathbf{s}) d\mathbf{s} = \mathbf{0},$$

which, according to Guan & Shen (2010), in a more general form is

$$\mathbf{e}_h(\boldsymbol{\theta}_\mathcal{A}) = \gamma \text{sgn}(\boldsymbol{\theta}_\mathcal{A}) + \sum_{\mathbf{x} \in \mathbf{x}} \mathbf{h}(\mathbf{s}; \boldsymbol{\theta}_\mathcal{A}) - \int_{\mathcal{S}} \lambda(\mathbf{s}; \boldsymbol{\theta}_\mathcal{A}) \mathbf{h}(\mathbf{s}; \boldsymbol{\theta}_\mathcal{A}) d\mathbf{s} = \mathbf{0}. \quad (\text{A1})$$

Here, $\mathbf{h}(\mathbf{s}; \boldsymbol{\theta}_\mathcal{A})$ can be any measurable vector function of the same dimension as $\boldsymbol{\theta}_\mathcal{A}$. Following Guan et al. (2015), we call (A1) a first-order penalized estimating function.

To improve the estimation efficiency of $\boldsymbol{\theta}_\mathcal{A}$ for general point processes, we analyze the estimation variance of (A1). Recall that $\boldsymbol{\theta}_\mathcal{A} \neq \mathbf{0}$, and denote the sensitivity matrix

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

by \mathbf{S}_h and the variance matrix

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

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

The optimal $\mathbf{h}^*(\mathbf{s}; \boldsymbol{\theta}_\mathcal{A})$ 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 the following Fredholm integral equation of the second kind

$$\int_{\mathcal{S}} \mathbf{h}^*(\mathbf{u}; \boldsymbol{\theta}_\mathcal{A}) \lambda(\mathbf{u}; \boldsymbol{\theta}_\mathcal{A}) [g(\mathbf{s}, \mathbf{u}) - 1] d\mathbf{u} + \mathbf{h}^*(\mathbf{s}; \boldsymbol{\theta}_\mathcal{A}) = \nabla \lambda(\mathbf{s}; \boldsymbol{\theta}_\mathcal{A}) / \lambda(\mathbf{s}; \boldsymbol{\theta}_\mathcal{A}).$$

In the spirit of Guan & Shen (2010), we assume that $\mathbf{h}^*(\mathbf{u}; \boldsymbol{\theta}_\mathcal{A}) \lambda(\mathbf{u}; \boldsymbol{\theta}_\mathcal{A}) \simeq \mathbf{h}^*(\mathbf{s}; \boldsymbol{\theta}_\mathcal{A}) \lambda(\mathbf{s}; \boldsymbol{\theta}_\mathcal{A})$ for proximate pairs of \mathbf{s}, \mathbf{u} and $g(\mathbf{s}, \mathbf{u}) \simeq 1$ for distant pairs, which yields a closed-form solution $\mathbf{h}^*(\mathbf{s}; \boldsymbol{\theta}_\mathcal{A}) = w^*(\mathbf{s}; \boldsymbol{\theta}_\mathcal{A}) \nabla \lambda(\mathbf{s}; \boldsymbol{\theta}_\mathcal{A}) / \lambda(\mathbf{s}; \boldsymbol{\theta}_\mathcal{A})$ with a spatially dependent weight function

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

It has the advantage that when the point process is second-order intensity-reweighted stationary and the spatial correlations in it vanish beyond a distance of m , $\int_{\mathcal{S}} [g(\mathbf{s}, \mathbf{u}) - 1] d\mathbf{u} = \int_{\mathcal{S}} [g(\mathbf{s} - \mathbf{u}) - 1] d\mathbf{u} = \int_0^m 2\pi r [g(r) - 1] dr$ can be approximated using Ripley’s K-function (see, e.g., Lieshout, 2019) by $K(m) - \pi m^2$.

B Derivation of the Additive Training Algorithm in Sec. 3.3

Recalling (4), we minimize the following penalized weighted Poisson likelihood loss function

$$L_w(\mathbf{f}) = \sum_{k=1}^K \Omega(f_k) - \sum_{\mathbf{x} \in \mathbf{x}} w(\mathbf{x}; \mathbf{f}) \phi(\mathbf{x}; \mathbf{f}) + \int_{\mathcal{S}} w(\mathbf{s}; \mathbf{f}) \exp[\phi(\mathbf{s}; \mathbf{f})] d\mathbf{s}$$

with $\Omega(f_k) = \gamma \sum_v |\theta_{kv}|$. Denote the estimated log-intensity over $k-1$ trees $\hat{\mathbf{f}}_{k-1} = \{\hat{f}_1, \dots, \hat{f}_{k-1}\}$ by $\hat{\phi}(\mathbf{s}; \hat{\mathbf{f}}_{k-1})$ with $\mathbf{s} \in \mathcal{S}$. At the k -th iteration, we add a tree predictor \hat{f}_k to minimize

$$L_w^{(k)}(f_k) = \Omega(f_k) - \sum_{\mathbf{x} \in \mathbf{x}} w(\mathbf{x}; \mathbf{f}) [\hat{\phi}(\mathbf{x}; \hat{\mathbf{f}}_{k-1}) + f_k(\mathbf{x})] + \int_{\mathcal{S}} w(\mathbf{s}; \mathbf{f}) \exp[\hat{\phi}(\mathbf{s}; \hat{\mathbf{f}}_{k-1}) + f_k(\mathbf{s})] d\mathbf{s}.$$

Without the knowledge of the tree structure q_k of f_k , we use the following iteratively updated spatial weight to better approximate $w(\mathbf{x}; \mathbf{f})$,

$$\hat{w}_k(\mathbf{s}) = \frac{\omega_k}{1 + \exp[\hat{\phi}(\mathbf{s}; \hat{\mathbf{f}}_{k-1})] [\hat{K}(m) - \pi m^2]}.$$

One can then quickly apply a quadratic approximation to $L_w^{(k)}(f_k)$ to obtain

$$\begin{aligned} L_w^{(k)}(f_k) &\simeq \Omega(f_k) - \sum_{\mathbf{x} \in \mathbf{x}} \hat{w}_k(\mathbf{x}) \hat{\phi}(\mathbf{x}; \hat{\mathbf{f}}_{k-1}) + \int_{\mathcal{S}} \hat{w}_k(\mathbf{s}) \exp[\hat{\phi}(\mathbf{s}; \hat{\mathbf{f}}_{k-1})] d\mathbf{s} - \sum_{\mathbf{x} \in \mathbf{x}} \hat{w}_k(\mathbf{x}) f_k(\mathbf{x}) \\ &\quad + \int_{\mathcal{S}} \hat{w}_k(\mathbf{s}) \exp[\hat{\phi}(\mathbf{s}; \hat{\mathbf{f}}_{k-1})] f_k(\mathbf{s}) d\mathbf{s} + \frac{1}{2} \int_{\mathcal{S}} \hat{w}_k(\mathbf{s}) \exp[\hat{\phi}(\mathbf{s}; \hat{\mathbf{f}}_{k-1})] f_k^2(\mathbf{s}) d\mathbf{s}. \end{aligned}$$

Since $\hat{w}_k(\mathbf{s})$ and $\hat{\phi}(\mathbf{s}; \hat{\mathbf{f}}_{k-1})$ are known, one can remove the constant terms to simplify it to

$$\begin{aligned} \tilde{L}_w^{(k)}(f_k) &\simeq \Omega(f_k) - \sum_{\mathbf{x} \in \mathbf{x}} \hat{w}_k(\mathbf{x}) f_k(\mathbf{x}) + \int_{\mathcal{S}} \hat{w}_k(\mathbf{s}) \exp[\hat{\phi}(\mathbf{s}; \hat{\mathbf{f}}_{k-1})] f_k(\mathbf{s}) d\mathbf{s} \\ &\quad + \frac{1}{2} \int_{\mathcal{S}} \hat{w}_k(\mathbf{s}) \exp[\hat{\phi}(\mathbf{s}; \hat{\mathbf{f}}_{k-1})] f_k^2(\mathbf{s}) d\mathbf{s}. \end{aligned}$$

Denote the set of locations where $\mathbf{z}(\mathbf{s})$ belonging to a leaf v by $I_{kv} = \{\mathbf{s} : q(\mathbf{s}) = v\}$. We can extract the contribution of this leaf to $\tilde{L}_w^{(k)}(f_k)$, expand $\Omega(f_k)$ and replace f_k by θ_{kv} to obtain

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

For any given tree structure q_k , it is straightforward to check that the minimizer of $\tilde{L}_{wv}^{(k)}$, denoted by $\hat{\theta}_{kv}$, is

$$\hat{\theta}_{kv} = \frac{\text{sgn}(R_{kv} - T_{kv}) \max(|R_{kv} - T_{kv}| - \gamma, 0)}{T_{kv}},$$

where

$$R_{kv} = \sum_{\mathbf{x} \in \mathbf{x}} \hat{w}_k(\mathbf{x}) 1\{\mathbf{x} \in I_{kv}\}, \quad T_{kv} = \int_{\mathcal{S}} \hat{w}_k(\mathbf{s}) \exp[\hat{\phi}(\mathbf{s}; \hat{\mathbf{f}}_{k-1})] 1\{\mathbf{s} \in I_{kv}\} d\mathbf{s}.$$

To approximate the integral throughout the derivation above, one may consider a numerical quadrature approximation.

Suppose that the observation window \mathcal{S} can be divided into m grid cells $\mathcal{T}_1, \dots, \mathcal{T}_m$, and each cell is centered at \mathbf{t}_i and has a volume $|\mathcal{T}_i|$. Then, the following integral can be approximated as

$$\int_{\mathcal{S}} \hat{w}_k(\mathbf{s}) h(\mathbf{s}) d\mathbf{s} \simeq \sum_{i=1}^m \hat{w}_k(\mathbf{t}_i) h(\mathbf{t}_i) |\mathcal{T}_i|,$$

where $h(\mathbf{s})$ is any measurable function defined on \mathcal{S} such that $\hat{w}_k(\mathbf{s}) h(\mathbf{s})$ is absolutely integrable.

C Supplementary Evaluation Results for the Simulation Study in Sec. 4

In this section, we supply the results of the integrated absolute error for the main simulation study in Sec. 4. The integrated absolute error reads $\int_{\mathcal{S}} |\lambda(s) - \hat{\lambda}(s)| ds$, where $\lambda(s)$ and $\hat{\lambda}(s)$ denote the theoretical and estimated intensities, respectively. We report the results on Poisson process data in Tab. 5 and those on log-Gaussian Cox and Neyman-Scott process data in Tab. 6 and 7.

In general, the findings convey similar messages to those reflected by Tab. 1, 2 and 3. Specifically, in the scenario of Sec. 4.1 where there are a small number of covariates, XGBoostPP obtains comparable performance against existing approaches on all data types under various parameter settings. It outperforms KIEs while APPs behave the best. In the scenario of Sec. 4.2 where there are a larger number of covariates, XGBoost significantly outperforms the APP method whose estimation error is about twice that of XGBoostPP. Comparing the estimation errors of XGBoostPP_p to XGBoostPP_{wp}, the latter are substantially reduced on clustered processes while remain almost unchanged on Poisson processes. Moreover, such reductions are greater when point patterns are more spatially varying and clustered.

In addition, comparing the scenario of Sec. 4.3 to Sec. 4.1, we find that, due to the presence of many nuisance covariates, the integrated absolute errors of both XGBoostPP and APP increase. However, the increase of XGBoostPP is much smaller than that of APP. This indicates that, while the true intensity function is simple, APP is much less robust than XGBoostPP when there is considerable uncertainty in choosing the most relevant covariates.

Table 5: Averaged integrated absolute errors (standard deviations) of different approaches on Poisson process data.

COVS	POISSON PROCESS	$\beta = 0.5$	$\beta = 1.0$
2 (§4.1)	KIE _{ra}	91.1(10.2)	170.0(8.9)
	KIE _{re}	129.9(10.3)	215.2(10.4)
	APP _{naive}	59.8 (11.0)	75.8 (11.1)
	APP _{rfm}	60.7(10.6)	77.7(10.6)
	XGBoostPP _p	86.0(10.3)	103.9(12.0)
	XGBoostPP _{wp}	86.1(10.3)	103.7(12.2)
10 (§4.3)	APP _{rfm}	191.7(7.2)	218.4(7.7)
	XGBoostPP _p	116.9 (12.0)	137.0 (11.3)
	XGBoostPP _{wp}	116.9 (12.0)	137.0 (11.3)
COVS	POISSON PROCESS	$\beta = 0.2$	$\beta = 0.4$
10 (§4.2)	APP _{rfm}	210.0(15.0)	226.5(7.9)
	XGBoostPP _p	109.3 (10.2)	149.8(11.9)
	XGBoostPP _{wp}	109.3 (10.2)	149.8 (11.8)

D Flexibility of XGBoostPP Demonstrated by Three Poisson Toy Examples

To intuitively visualize the flexibility of XGBoostPP in modeling various nonlinear relations between covariates and the point process intensity function, we demonstrate three Poisson toy examples.

The examples are created on the unit window as \mathcal{S} and take the x, y coordinates as covariates. The first example has the intensity function $\lambda(x, y) = \exp(3 + 2x + 7y)$, which is basically 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 modeling 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 function 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 Fig. 3. The pictures clearly show that our XGBoostPP model can detect different covariate relationships and perform an automatic variable selection.

Table 6: Averaged integrated absolute errors (standard deviations) of different approaches on log-Gaussian Cox process data.

COVS	LOG-GAUSSIAN COX PROCESS	$\tau^2 = 1$		$\tau^2 = 2$	
		$\sigma = 0.02$	$\sigma = 0.04$	$\sigma = 0.02$	$\sigma = 0.04$
$\beta = 0.5$					
2 (§4.1)	KIE _{ra}	93.7(14.7)	99.5(19.6)	96.8(18.0)	110.8(29.8)
	KIE _{re}	131.7(13.5)	135.7(19.8)	134.0(16.7)	144.3(30.3)
	APP _{naive}	65.9 (15.4)	78.0(23.0)	74.9(20.0)	101.7(38.2)
	APP _{rfm}	66.3(15.0)	77.9 (22.9)	74.8 (19.7)	100.8 (37.7)
	XGBoostPP _p	94.1(14.9)	102.6(20.3)	100.5(18.4)	120.0(31.2)
	XGBoostPP _{wp}	92.6(17.0)	97.7(21.1)	92.1(19.2)	108.4(29.2)
10 (§4.3)	APP _{rfm}	200.2(9.9)	213.2(17.5)	213.5(18.0)	243.9(36.0)
	XGBoostPP _p	118.8(16.1)	126.9(19.8)	125.8(20.3)	155.0(36.6)
	XGBoostPP _{wp}	115.1 (16.8)	121.2 (19.8)	116.0 (17.0)	141.5 (26.2)
$\beta = 1.0$					
2 (§4.1)	KIE _{ra}	171.5(12.3)	175.7(16.6)	174.0(15.9)	182.0(24.4)
	KIE _{re}	216.6(13.9)	220.7(21.6)	219.2(18.6)	226.9(34.8)
	APP _{naive}	84.0 (15.4)	98.7 (26.8)	97.2 (27.0)	125.7(52.2)
	APP _{rfm}	85.3(15.1)	99.2(26.5)	97.6(26.3)	124.7 (50.5)
	XGBoostPP _p	113.3(17.3)	127.1(25.4)	123.8(23.1)	146.2(40.7)
	XGBoostPP _{wp}	108.6(18.2)	121.8(26.8)	110.0(18.9)	131.8(33.6)
10 (§4.3)	APP _{rfm}	232.0(12.6)	247.4(23.0)	250.8(25.5)	280.1(50.7)
	XGBoostPP _p	146.1(14.8)	162.2(22.2)	164.4(22.3)	193.8(39.8)
	XGBoostPP _{wp}	142.2 (15.9)	160.2 (22.7)	147.5 (19.1)	179.1 (31.6)
$\beta = 0.2$					
10 (§4.2)	APP _{rfm}	208.0(12.2)	214.2(14.2)	214.2(14.9)	240.9(32.6)
	XGBoostPP _p	109.2(12.0)	116.9(17.7)	117.6(20.6)	142.0(34.3)
	XGBoostPP _{wp}	104.9 (11.8)	112.9 (16.8)	110.5 (16.4)	135.6 (26.4)
$\beta = 0.4$					
10 (§4.2)	240.9(32.6)	234.4(10.2)	246.1(17.5)	246.5(17.0)	272.7(40.1)
	XGBoostPP _p	149.5(14.3)	160.1(20.0)	156.0(20.5)	180.4(35.8)
	XGBoostPP _{wp}	140.9 (16.7)	154.3 (20.3)	139.7 (17.3)	166.9 (29.8)

E Additional Details for the Real Data Analyses in Sec. 5

In this section, we supply additional experimental details for the real data analyses in Sec. 5.

E.1 Covariates Information in the Two Data Sets

The covariates considered in the tropical forest data set (‘Bei’, ‘Capp’) and the kitchen fire data set (‘Fire’), along with their descriptions, are listed in Tab. 6 and 7, respectively.

E.2 The Test Poisson Log-likelihood Evaluation Metric

The evaluation metric used in the real data analyses is the cross-validated Poisson log-likelihoods of point patterns. Since we apply a four-fold cross-validation, the test Poisson log-likelihood over the data of one of the four subsets \mathbf{x}_i , with $\mathbf{x} = \{\mathbf{x}_1, \dots, \mathbf{x}_4\}$, reads

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

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

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

Table 7: Averaged integrated absolute errors (standard deviations) of different approaches on Neyman–Scott process data.

COVS	NEYMAN–SCOTT PROCESS	$\kappa = 100$		$\kappa = 200$	
		$\sigma = 0.02$	$\sigma = 0.04$	$\sigma = 0.02$	$\sigma = 0.04$
$\beta = 0.5$					
2 (§4.1)	KIE _{ra}	98.5(19.4)	97.8(18.1)	95.4(15.7)	94.0(14.4)
	KIE _{re}	136.8(19.0)	135.6(18.8)	132.7(15.0)	132.0(14.4)
	APP _{naive}	79.6(22.7)	72.8 (20.2)	69.2 (17.2)	66.3 (14.8)
	APP _{rfm}	79.3 (22.5)	72.9(20.0)	69.3(17.1)	66.8(14.5)
	XGBoostPP _p	103.9(19.3)	99.9(19.0)	96.8(16.1)	94.1(14.6)
	XGBoostPP _{wp}	95.3(20.1)	95.4(20.5)	92.4(18.4)	93.6(15.5)
10 (§4.3)	APP _{rfm}	223.3(16.0)	209.6(13.5)	206.8(11.3)	199.5(9.8)
	XGBoostPP _p	129.6(20.6)	124.6(19.2)	121.2(16.7)	119.9(15.3)
	XGBoostPP _{wp}	119.5 (17.5)	119.9 (18.6)	114.7 (17.2)	119.4 (15.5)
$\beta = 1.0$					
2 (§4.1)	KIE _{ra}	175.3(17.4)	174.5(14.6)	173.6(13.3)	172.3(11.7)
	KIE _{re}	221.6(21.2)	219.5(19.2)	218.2(15.6)	217.5(14.9)
	APP _{naive}	104.9(27.0)	92.6 (23.2)	90.1 (19.3)	84.8 (16.4)
	APP _{rfm}	104.9 (26.8)	93.6(22.7)	91.2(18.8)	85.9(16.1)
	XGBoostPP _p	129.7(25.4)	122.5(21.9)	120.6(19.7)	114.8(17.3)
	XGBoostPP _{wp}	112.0(22.3)	117.5(22.7)	105.0(17.6)	113.7(18.1)
10 (§4.3)	APP _{rfm}	263.4(20.8)	242.6(16.5)	240.7(14.3)	230.0(11.0)
	XGBoostPP _p	172.2(22.3)	155.9(20.1)	153.1(17.1)	146.1(15.8)
	XGBoostPP _{wp}	148.7 (22.2)	153.7 (20.1)	138.8 (17.9)	146.1 (15.8)
$\beta = 0.2$					
10 (§4.2)	APP _{rfm}	221.1(13.8)	211.0(11.6)	209.0(10.6)	207.3(11.4)
	XGBoostPP _p	120.0(19.4)	114.8(16.2)	110.0(13.3)	110.9(13.4)
	XGBoostPP _{wp}	113.9 (17.0)	111.2 (15.3)	104.9 (13.1)	109.9 (13.1)
$\beta = 0.4$					
10 (§4.2)	APP _{rfm}	253.9(14.7)	241.1(12.0)	238.7(11.1)	232.7(9.2)
	XGBoostPP _p	161.3(17.7)	158.1(16.8)	143.5(13.4)	153.3(14.2)
	XGBoostPP _{wp}	143.6 (17.5)	153.0 (17.8)	138.4 (12.9)	153.2 (14.2)

Table 8: The covariates involved in the forestry data set (‘Bei’ and ‘Capp’).

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

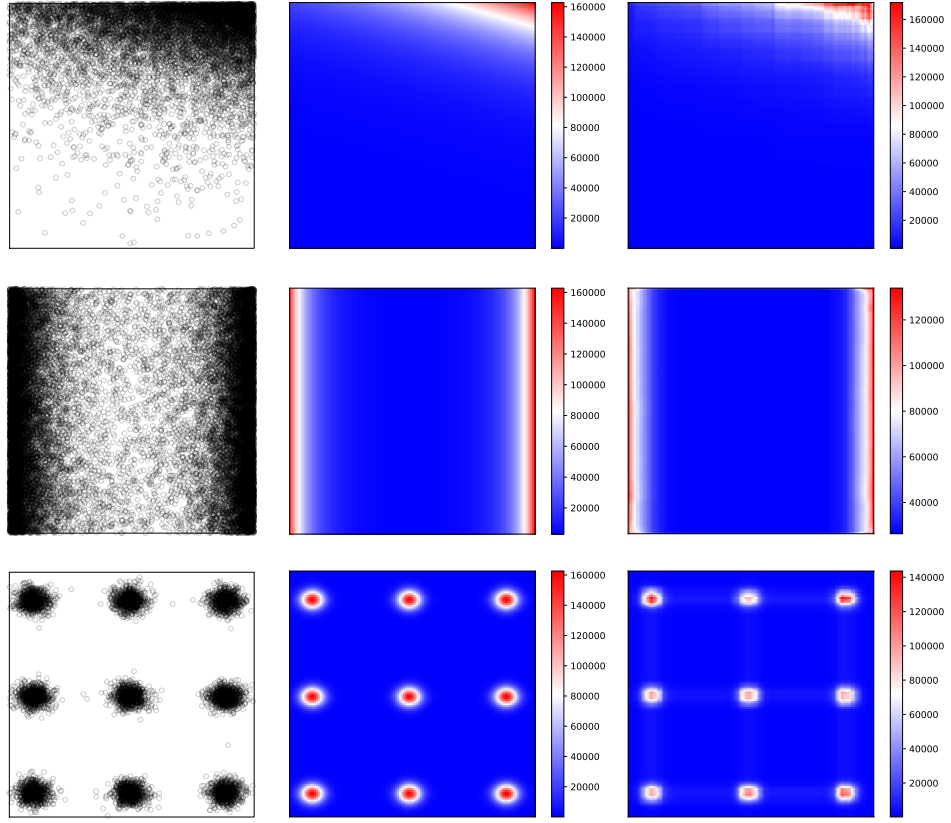


Figure 3: Point patterns, true intensities and the intensities estimated by XGBoostPP_p for the three toy examples.

Table 9: The covariates involved in the fire data set ('Fire').

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_frsl	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_frsl	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