Fused Spatial Point Process Intensity Estimation with Varying Coefficients on Complex Constrained Domains

Lihao Yin[a,b], Huiyan Sang[a]

[a] *Department of Statistics, Texas A&M University, College Station, TX, USA*
[b] *Institute of Statistics and Big Data, Renmin University, Beijing, , China*

Abstract

The availability of large spatial data geocoded at accurate locations has fueled a growing interest in spatial modeling and analysis of point processes. The proposed research is motivated by the intensity estimation problem for large spatial point patterns on complex domains in $\mathbb{R}^2$ (e.g., domains with irregular boundaries, sharp concavities, and/or interior holes due to geographic constraints) and linear networks, where many existing spatial point process models suffer from the problems of "leakage" and computation. We propose an efficient intensity estimation algorithm to estimate the spatially varying intensity function and to study the varying relationship between intensity and explanatory variables on complex domains. The method is built upon a graph regularization technique and hence can be flexibly applied to point patterns on complex domains such as regions with irregular boundaries and holes, or linear networks. An efficient proximal gradient optimization algorithm is proposed to solve the penalized composite likelihood problem for parameter estimations. Finally, we apply the method to study and visualize the intensity patterns of the accidents on the Western Australia road network, and the spatial variations in the effects of income, lights condition, and population density on the Toronto homicides occurrences.

Keywords: Graph regularization, Intensity estimation, Line network, Spatial point pattern, Varying coefficient models.

1. Introduction

Numerous problems in geosciences, social sciences, ecology, and urban planning nowadays involve extensive amounts of spatial point pattern data recording event occurrence. Examples include locations of invasive species, pick-up locations of Taxi trips, addresses of 911 calls, and traffic accidents on roads, to name a few. In many such applications,

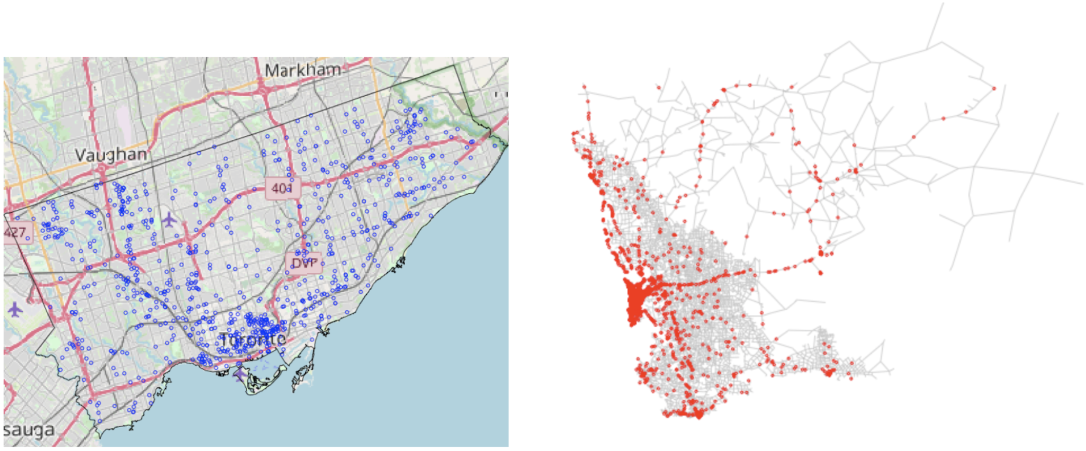

Figure 1: Left: Map of homicide locations in Toronto during $2000 - 2014$; Right: Traffic networks and road accidents on Western Australia in 2011;

the primary problem of interest is to characterize the probability of event occurrence. In the presence of additional covariates information, another problem of interest is to study the effect of these covarites on event occurrence probability, considering the spatial dependence of observations. Spatial point process models have been widely used for the analysis of point patterns, in which the intensity function, denoted as $\rho(u)$, is used to describe the likelihood for an event to occur at location u.

In practice, many spatial point patterns data are collected over complex domains with irregular boundaries, peninsulas, interior holes, or network geographical structures. In this paper, we consider two motivating data examples on such complex domains. The first one is the traffic accident locations on the Western Australia road network shown in the right panel of Figure 1, where the interest lies in studying the spatial variation of accident occurrences. The left panel in Figure 1 shows the homicides that occurred in Toronto, where the city boundary has a very irregular shape especially near Toronto islands. The Toronto data set also includes several additional covariates such as the records of average income, night lights, and population density. Therefore, the questions of interest include not only the intensity of crime events but also the relationships between crime intensity and regional characteristics. In particular, for a large city like Toronto, we may expect that such relationships can vary, and in some places rather abruptly, across the study domain.

Thus far, many methods have been introduced to model the first-order spatially varying intensity function $\rho(u)$. Popular point process models include the spatial Poisson point processes, the log-Gaussian Cox Processes, and the Gibbs point processes. See a review by Møller and Waagepetersen (2007). The intensity estimations of these models are often done using maximum composite likelihoods (Guan, 2006), estimating equations (Guan et al., 2015) or Bayesian inference methods (Leininger et al., 2017; Gonçalves and Gamerman, 2018; Shirota and Banerjee, 2019). Nonparametric methods have also been widely used for estimating the spatially varying intensity functions, including the edge-corrected kernel smoothing estimators by Diggle (1985); Jones et al. (1996), the Voronoi estimator by Barr and Schoenberg (2010) using the inverse of the area of the Voronoi cell for each observed location, and a local likelihood estimation procedure in analogy to geographically weighted regression by Fotheringham et al. (2003).

Yet, statistical analysis of point patterns on complex domains presents severe challenges to many of the classical point process models reviewed above. Mainly, the commonly adopted Euclidean assumption underpinning some of these methods no longer holds for point patterns on complex domains. For example, two locations on a road network that are close by Euclidean distance may actually lie on two separate roads. Moreover, the large data size will aggravate the challenges in modeling point patterns on complex domains. There is a great need to develop spatial point pattern analysis tools that are computationally efficient to solve the so called “leakage” problem encountered on complex domains.

For point patterns that occur along a line network, a particular type of complex domain, a number of methods have been developed recently. Kernel estimators of the intensity function on a line network were investigated in McSwiggan et al. (2017); Moradi et al. (2018); Rakshit et al. (2019), adapting the idea of edge-correction using path lengths. Other variations of kernel density estimation methods are reviewed in Baddeley et al. (2020). It is known that kernel estimators are, by nature, more suitable for estimating relatively smooth intensity functions because of the use of smoothing kernel functions. When intensity function exhibits discontinuities and abrupt changes in space, as discussed in Baddeley et al. (2020), piece-wise constant estimators become an appealing alternative as they have a strong adaptivity to changes. One research in this direction is the aforementioned Voronoi estimator by Barr and Schoenberg (2010). However, the method suffers from the high variance in the estimator. To reduce the variance, Moradi et al. (2019) extended it by a bootstrap resample smoothing procedure. Recently, Bassett and Sharpnack (2019) proposed to estimate the density of points on a network as opposed to the intensity function based on a total variation regularization method. While each represents advancements in estimating intensity or density of points, none has incorporated spatial covariates in estimation.

When spatial covariates are available, various methods (Baddeley et al., 2012; McSwiggan, 2019) have been developed to incorporate covariate information with the goal to investigate the effect of spatial covariates on point patterns. However, to the best of our knowledge, there has been very limited work for dealing with varying regression coefficients for spatial point patterns, even in the simpler case where point patterns are observed in the Euclidean space. One notable exception is the work by Pinto Junior et al. (2015), which modeled the regression coefficients as a multivariate Gaussian process in a similar fashion as the spatially varying coefficients (SVC) linear regression model proposed by Gelfand et al. (2003). Despite the model richness and flexibility, the SVC model is known to involve heavy computation in the presence of large spatial data due to the requirement of Metropolis MCMC and the need to invert a large covariance matrix. The intractability of the likelihood function of the spatial Poisson process further aggravates the issue. To address the computation issue, Pinto Junior et al. (2015) partitioned the study region into a small number of subregions according to administrative areas and assumed that latent spatial random effects take constant values within each subregion. However, in some applications, such a pre-determined partition may be unavailable or fail to accurately reflect the complex underlying environmental and geological conditions.

In light of these limitations in the current literature, we develop a simple yet effective approach based on a fused lasso regularization method on a graph for the estimation of piece-wise constant spatial intensity functions. We propose penalties on regression coefficients to encourage sparsity on the differences among regression coefficients that are close in space. The fused lasso methods have gained increasing popularity owning to its flexibility of learning clustered structures. However, to our knowledge, there is limited work that has investigated its performance for point pattern data analysis. In addition, we extend the approach to a piece-wise constant coefficient spatial point process model when explanatory variables are available, to model the varying relationships between point patterns and covariates. We formulate the estimation problem into penalized Poisson-based and Logistic based composite likelihoods optimizations, for which we solve by an efficient proximal gradient algorithm. We tailor the algorithm to utilize spatial graph structures to speed up computations. The choices of graphs play important roles in the modeling and computation of fused lasso problems. We consider various spatial graphs to represent spatial geometry of complex domains and compare their performance. Finally, we introduce this method to the analysis of the Western Australia accident data and the Toronto homicides data. The results of our analysis reveal several interesting clustering patterns of traffic accidents and the spatial crime distribution in relation to a number of key environmental, social, and economic variables.

The paper is organized as follows. In Section 2, we review the basic mathematical formulations and definitions of spatial point processes. We then introduce our method in Section 3.1, followed by the computation algorithm in 3.2. Sections 4 and 5 include the simulations to illustrate the model performance and the applications to the two real data sets. We offer conclusions and discussions in Section 6. The proof of Theorem 1 and additional implementation details and numerical results are in Appendix.

2. Preliminaries

2.1. Observation Domain

In this study, we consider spatial points on two important types of observation domains. The first type is a bounded domain $D \subset \mathbb{R}^2$ that can be fully covered by finitely many rectangles. The commonly assumed planar window $[a_1, a_2] \times [b_1, b_2]$ is a special case of this type. For any locations u_1, u_2 in a planar window, the Euclidean distance is used to measure the distance between two locations, denoted as $d(u_1, u_2)$. One example of this type is given in Figure 1, where the observation domain is the city of Toronto, which has irregular city limit boundaries. For any Borel subset $B \subseteq D$, the Lebesgue measure $|B|$ is the area of B.

In the second type, we assume D is a linear network. Let $[u, v] = \{tu + (1-t)v : 0 \leq t \leq 1\}$ denotes a line segment in the plane with endpoints $u, v \in \mathbb{R}^2$. A linear network is defined as the finite union $D = \cup_{i=1}^{\eta}[u_i, v_i]$ of line segments $[u_1, v_1], \cdots, [u_\eta, v_\eta]$ embedded in the same plane. One commonly used distance $d(u_1, u_2)$ is the shortest-path distance between u_1 and u_2 on the network. For any subset $B \subseteq D$, the measure $|B|$ represents the total length of all segments in B. An example of the line network is shown in the right panel of Figure 1, where the road network in the state of Western Australia is drawn in grey lines, and red points mark the traffic accident locations in 2011.

2.2. Spatial Point Processes

Let $\mathbf{X}$ be a spatial point process on D with the locally finite property, i.e., the random cardinality $N_{\mathbf{X}}(B) = \#\{u : u \in \mathbf{X} \cap B\}$ is almost surely finite for any $B \subset D$. Assume that, for any bounded $B \subset D$, if there exits a non-negative and locally integrable function $\rho(\cdot) : B \mapsto \mathbb{R}$ such that,

$$E\{N_{\mathbf{X}}(B)\} = \int_B \rho(u)du$$

then $\rho(\cdot)$ is called the intensity function of $\mathbf{X}$. The intensity function is of key interest in point pattern analysis as $\rho(u)|du|$ is interpreted as the approximate probability that an event occurs in the infinitesimal set du.

Poisson point processes are one of the most fundamental and tractable spatial point process models. In practice, $\rho(\cdot)$ is often varying over D, i.e. $\mathbf{X}$ is inhomogeneous and can also depend on some spatial covariates $\mathbf{z}(u)$. In our study, we model the intensity function with a general log-linear form,

$$\rho(u;\boldsymbol{\beta}) = \exp\{\mathbf{z}^T(u)\boldsymbol{\beta}\}, u \in D \tag{1}$$

where $\mathbf{z}(u) = \big(z_1(u), \cdots, z_p(u)\big)^T$ is a p-dimensional vector of spatial covariates associated with the spatial location u, and $\boldsymbol{\beta} \in \mathbb{R}^p$ is the vector of regression parameter.

There are several other popular parametric point process models whose marginal intensity functions take the same log-linear form as in (1). The class of Cox process models is one such example. Let $\boldsymbol{\Lambda} = \{\Lambda(u) : u \in D\}$ denote a real, nonnegative valued random field. If the conditional distribution of $\mathbf{X}$ given $\boldsymbol{\Lambda}$ is a Poisson process on D with intensity function $\boldsymbol{\Lambda}$, then X is said to be a Cox process driven by $\boldsymbol{\Lambda}$. Popular examples of Cox processes models include the Neyman-Scott process and the log Gaussian Cox process. See a review in Chapter 17 of Gelfand et al. (2010).

2.3. Composite Likelihoods

To estimate $\boldsymbol{\beta}$ in (1), one commonly used method is to construct unbiased estimating equations and obtain estimators by maximizing the corresponding composite likelihoods. The Poisson based composite log-likelihood function (Waagepetersen, 2007) and the logistic based composite log-likelihood function (Baddeley et al., 2014) have been used widely in the literature, which are respectively given by

$$\ell_{\mathrm{PL}}(\boldsymbol{\beta}) = \sum_{i=1}^{m} \log \rho(u_i;\boldsymbol{\beta}) - \int_D \rho(u;\boldsymbol{\beta})du \tag{2}$$

$$\ell_{\mathrm{LRL}}(\boldsymbol{\beta}) = \sum_{i=1}^{m} \log(\frac{\rho(u_i;\boldsymbol{\beta})}{\delta(u_i) + \rho(u_i;\boldsymbol{\beta})}) - \int_D \delta(u)\log(\frac{\rho(u;\boldsymbol{\beta}) + \delta(u)}{\delta(u)})du, \tag{3}$$

where $\{u_1, \cdots, u_m\}$ denotes a set of points from a point process, and $\delta(u)$ is a non-negative real-valued function. When point process is a Poisson process, the Poisson based composite log-likelihood function in (2) is identical to the full log-likelihood function. For other point processes models, the use of composite likelihood can be justified by the theory of estimating functions (Guan, 2006). It can be shown (see, e.g., Guan, 2006; Choiruddin et al., 2018) that the estimators obtained by maximizing both Poisson based and logistic based composite log-likelihood are the solution to the two corresponding unbiased estimating equations for $\boldsymbol{\beta}$.

Nevertheless, the composite likelihood based inference produces a less efficient estimator compared with the full likelihood based estimator, due to the loss of information incurred when only using the first-order moment property of the point process. To improve its efficiency, several methods have been developed to carefully select the weights when combining composite likelihood terms (Guan and Shen, 2010). For simplicity, we only consider the unweighted composite likelihoods in the paper, but remark that the methods can be potentially generalized to the use of weighted composite likelihoods.

In practice, numerical approximations are needed for the composite likelihood inference because both the evaluations of (2) and (3) involve integral terms. For equation (2), Berman and Turner (1992) developed a numerical quadrature method that employs Riemann sum approximation to the integral part. To implement this, the domain D is partitioned into $M - m$ quadrats. More details on how we divide a 2-D bounded domain and a line network can be found in Appendix A1. The $M - m$ dummy points, denoted by $\{u_i, i = m + 1, \cdots, M\}$, are then placed at the centroid of each quadrat. The Poisson based composite log likelihood is then approximated by

$$\ell_{PL}(\boldsymbol{\beta}) \approx \sum_{i=1}^{M} v_i\{y_i \log \rho(u_i; \boldsymbol{\beta}) - \rho(u_i; \boldsymbol{\beta})\}, \tag{4}$$

where $\{u_i \in D, i = 1, \cdots, M\}$ consists of the m observed points and $M - m$ dummy points. v_i is the quadrature weight corresponding to each u_i. We set $v_i = a_i/n_i$, where n_i denotes the total number of observed events and dummy points in the quadrat that u_i resides, and a_i denotes the Lebesgue measure of the quadrat of u_i such that $\sum_{i=m+1}^{M} a_i = |D|$. The working response data becomes $y_i = v_i^{-1}\Delta_i$, where Δ_i is an inndicator of whether point i is an observation ($\Delta_i = 1$) or a dummy point ($\Delta_i = 0$).

The Berman-Turner approximation in (4) often requires a great amount of dummy points, consequently incurring extra computational cost. Baddeley et al. (2014) showed that the estimates based on the logistic likelihood in (3) perform competitively with the Berman-Turner approximation using a smaller number of dummy points. The method approximates (3) by

$$\ell_{LRL}(\boldsymbol{\beta}) \approx \sum_{i=1}^{m} \log \frac{\rho(u_i; \boldsymbol{\beta})}{\delta(u_i) + \rho(u_i; \boldsymbol{\beta})} + \sum_{i=m+1}^{M} \log \frac{\delta(u_i)}{\delta(u_i) + \rho(u_i; \boldsymbol{\beta})} \tag{5}$$

where the integration term is calculated by Monte Carlo integration, and the dummy points are drawn from a Poisson point process over D, which has an intensity function $\delta(u)$ and is independent from $\mathbf{X}$. Applying the Campbell's formula (Moller and Waagepetersen, 2003), it is straightforward to show that the expectation of the second term in (5) equals to the integral part in (3). We follow the suggestion of Baddeley et al. (2014) and choose $\delta(u) = (M - m)/|D|$ in our numerical studies.

3. Methodology

3.1. Spatially Varying Coefficient Models

A traditional way to model the log-linear term of the intensity function is to treat regression coefficients as constants in space as in (1). In the proposed model, we are interested in estimating a piece-wise constant intensity function in an intercept-only log-linear model or detecting clustering patterns in $\boldsymbol{\beta}$ when covariates are available. Below, we introduce a varying coefficient log-linear intensity model (SVCI) for spatial point processes via a graph regularization method.

To elaborate, suppose a set of spatial points is observed at locations $u_1, \dots, u_m \in D$. We assume these spatial points are a realization from a point process $\mathbf{X}$ with an intensity function $\rho(u)$, which depends on the p-dimensional spatial explanatory variables $\mathbf{z}(u) = \{z_1(u), \dots, z_p(u)\}$. As an extension of the constant coefficients regression model, we assume that the regression coefficients are spatially varying across D, denoted as $\boldsymbol{\beta}(u) = \{\beta_1(u), \dots, \beta_p(u)\}^T$. The spatially varying coefficient models inherit the simplicity and easy interpretation of the traditional log-linear model in (1), yet they still enjoy great flexibility that allows practitioners to investigate locally varying relationships among variables.

Let $\boldsymbol{\beta}_k = \big(\beta_k(u_1), \dots, \beta_k(u_m)\big)^T$ denote the vector of regression coefficients associated with the k-th covariate, for $k = 1, \dots, p$. We assume that each $\boldsymbol{\beta}_k$ has its own spatially clustered pattern and is piece-wise constant on D, that is, the coefficients are homogeneous in the same spatial cluster and varying across different clusters. In many spatial applications involving point patterns including traffic accidents, crime locations, pick-up/drop-off locations of Taxi trips, to name a few, it is desired to consider spatially contiguous clustering configurations such that only adjacent locations are clustered together. This way, the practitioners can detect discontinuities across boundaries and easily interpret the clusters as local regions to facilitate subsequent regional analysis.

Before introducing our regularization method, we formally define spatially contiguous cluster of points using the notion of connected components in graph theory. Consider an undirected graph denoted as $\mathbb{G} = (\mathbb{V}, \mathbb{E})$, where $\mathbb{V} = \{u_i \in D, i = 1, \dots, M\}$ is the set of vertices which in our case consists of both observed points and dummy points, and $\mathbb{E}$ is the edge set consisting of a subset of $\{(u_i, u_j) : u_i, u_j \in \mathbb{V}\}$. In graph theory, a graph $\mathbb{G}$ is said to be connected if for any two vertices there exists a path between them. A subgraph $\mathbb{G}_s$ is called a connected component of $\mathbb{G}$ if it is connected and there is no path between any vertex in $\mathbb{G}_s$ and any vertex in $\mathbb{G} \setminus \mathbb{G}_s$, i.e., the difference between sets $\mathbb{V}$ and $\mathbb{V}_s$. Now we can define spatially contiguous clusters as the connected components of a graph $\mathbb{G}$. As a result, a spatially contiguous partition of $\mathbb{V}$ is defined as a collection of disjoint connect components such that the union of vertices is $\mathbb{V}$.

This motivates us to construct a graph based regularization model, which permits contiguous cluster identifications of regression coefficients for each covariate in the log-linear point process model. Let $\boldsymbol{\beta}_k^* = \big(\beta_k(u_{m+1}), \ldots, \beta_k(u_M)\big)^T$, for $k = 1, \ldots, p$, denote the vector of regression coefficients at the dummy points associated with the k-th covariate. Denote the vector of the stacked regression coefficients at both the observed and dummy points by $\boldsymbol{\beta} = (\boldsymbol{\beta}_1^T, \boldsymbol{\beta}_1^{*,T}, \ldots, \boldsymbol{\beta}_p^T, \boldsymbol{\beta}_p^{*,T})^T$. We estimate $\boldsymbol{\beta}$ by minimizing the penalized negative composite log likelihood objective function:

$$Q(\boldsymbol{\beta}) = -\frac{1}{|D|}\tilde{\ell}_c(\boldsymbol{\beta}) + \sum_{k=1}^{p} \sum_{(i,j)\in\mathbb{E}} P_\lambda(\beta_k(u_i) - \beta_k(u_j)) \tag{6}$$

where $\tilde{\ell}_c(\beta)$ is either the approximation of the Poisson based composite log-likelihood or the logistic regression-based composite log-likelihood function with the following expressions:

$$\tilde{\ell}_{PL}(\boldsymbol{\beta}) = \sum_{i=1}^{M} v_i\{y_i \log \rho(u_i; \boldsymbol{\beta}(u_i)) - \rho(u_i; \boldsymbol{\beta}(u_i))\}, \tag{7}$$

$$\tilde{\ell}_{LRL}(\boldsymbol{\beta}) = \sum_{i=1}^{m} \log \frac{\rho\big(u_i; \boldsymbol{\beta}(u_i)\big)}{\delta(u_i) + \rho\big(u_i; \boldsymbol{\beta}(u_i)\big)} + \sum_{i=m+1}^{M} \log \frac{\delta(u_i)}{\delta(u_i) + \rho\big(u_i; \boldsymbol{\beta}(u_i)\big)} \tag{8}$$

The second term in the objective function (6) adds a graph pairwise fused regularization to the negative composite log-likelihood function. $\mathbb{E}$ is the edge set of a graph $\mathbb{G}$, and $(i, j) \in \mathbb{E}$ implies that there is an edge in $\mathbb{E}$ connecting the points at u_i and u_j. $P_\lambda(\cdot)$, a non-negative function tuned by parameter λ, penalizes the pairwise difference of regression coefficients whose corresponding locations are connected by an edge in $\mathbb{E}$. One popular choice is the L_1-penalty,

$$P_\lambda(t) = \lambda\|t\|_1$$

which is often referred to as the graph fused lasso penalty in the literature (Tibshirani et al., 2011; Arnold and Tibshirani, 2016; Li and Sang, 2019). The L_1 penalty encourages sparsity in the pairwise differences between the coefficients of edge-connected locations. As a result, the edges in the graph can be classified into a set that corresponds to the non-zero elements of $|\beta_k(u_i) - \beta_k(u_j)|$, and another set that corresponds to the zero elements of $|\beta_k(u_i) - \beta_k(u_j)|$. This naturally leads to a piece-wise constant estimate of $\boldsymbol{\beta}_k$ for each covariate and hence a well defined spatially contiguous partition of the vertices for each covariate. λ is a non-negative tuning parameter that determines the strength of penalization and ultimately influences the estimates of clustered. To make a proper choice of the values of them, we use the Bayes information criterion (BIC) to select an optimal value of λ (Choiruddin et al., 2021). Specifically, $BIC = -2\tilde{\ell}_c + df \log m$, where $\tilde{\ell}_c$ is the approximated composite log likelihood as in (7) and (8), m is the number of observations, and df is the degree of freedom of $\hat{\boldsymbol{\beta}}$. Following Tibshirani et al. (2011), df is estimated by the summation of the number of clusters for each regression coefficient $\boldsymbol{\beta}_k$.

We remark that there are other choices of sparsity inducing penalty functions, including adaptive lasso (Zou, 2006), smoothly clipped absolute deviation (SCAD, Fan and Li, 2001), and minimax concave penalty (MCP, Zhang, 2010). And there are other criteria for tuning parameter selection, including Akaike information criterion (AIC), generalized cross-validation (GCV, Golub et al., 1979), and extended Bayesian information criterion (EBIC, Chen and Chen, 2012). In this paper, we choose to use Lasso together with BIC to demonstrate the utility of our method for its computational simplicity. The method can adopt other forms of penalty functions and model-selection criteria which may further improve its performance.

The selection of edge set $\mathbb{E}$ is a key ingredient in our SVCI model by playing two important roles. First, the corresponding graph $\mathbb{G}$ reflects the prior assumptions about the spatial structure and the contiguous constraint of the regression coefficients. In particular, we rely on $\mathbb{G}$ to incorporate the relational information among observations on complex constrained domains so that we can relax the Euclidean assumption. Second, as we will explain in Section 3.2, the computation speed and storage complexity of the optimization algorithm are largely determined by the structure of $\mathbb{G}$. We seek to construct a graph fused lasso regularization to achieve a good balance between model accuracy and computational efficiency.

For point patterns on a bounded observation domain, one natural choice is to construct a nearest neighbor graph that connects each vertex with its k nearest neighbors (K-NN) or neighbors within a certain radius (r-NN). In practice, the number of neighbors in K-NN or the radius in r-NN needs to be chosen with care to guarantee that $\mathbb{G}$ is a connected graph. It is known in machine learning literature (see, e.g., Shaw and Jebara, 2009) that K-NN graphs can effectively preserve the intrinsic manifold structure of the data. Another approach is the Delaunay triangulation (Lee, 1980), which constructs triangles with a vertex set such that

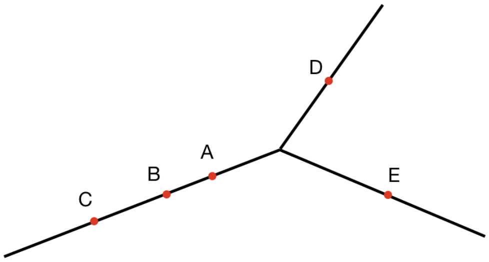

Figure 2: A simple illustration of connections in linear networks

no vertex is inside the circumcircle of any triangle. In practice, edges longer than a certain threshold are removed to ensure the spatial proximity of neighboring vertices. Triangular graphs have also shown their capabilities in preserving complex topological structures of the data. See Lindgren et al. (2011); Mu et al. (2018) for examples. Moreover, when a graph has certain simple structures such as a chain or a tree graph, several recent work (Padilla et al., 2018; Li and Sang, 2019) showed that these simple graph structures can be utilized to design efficient algorithms to solve the graph fused lasso problem. This motivates us to adopt a similar strategy to replace the original graph by a minimum spanning tree graph, defined as a subgraph that connects all vertices with no cycles and with minimum total edge weights. We will investigate and compare the performance of the proposed SVCI model with different types of graphs in the numerical studies in Section 4 and Appendix Section A3.

For point patterns on a linear network, we use an edge set that only connects pairs which are natural neighbors. To illustrate how we define natural neighbors, we provide a simple example of a linear network (black segments) and 5 spatial points (red nodes) near an intersection in Figure 2. For any interior point such as point B, defined as a point where there exists one other point on each side of the same line, we connect it with its two adjacent points $\{\mathtt{A},\mathtt{C}\}$. For any boundary point such as point A, defined as a point where there is no other point on the path between it and the intersection point, we connect it with $\{\mathtt{B},\mathtt{D},\mathtt{E}\}$, i.e., its adjacent interior point on the same line and its adjacent boundary points on the other lines that cross the same intersection.

3.2. Computation

Once we construct the edge set $\mathbb{E}$, we rewrite the objective function in (6) in matrix form and obtain the estimate of $\boldsymbol{\beta}$ by solving the following fused lasso optimization problem:

$$\hat{\boldsymbol{\beta}} = \underset{\boldsymbol{\beta} \in \mathbb{R}^{pM}}{\arg\min} \{ -\frac{1}{|D|} \tilde{\ell}_c(\boldsymbol{\beta}) + \lambda \sum_{k=1}^{p} \|\mathbf{H}\boldsymbol{\beta}_k\|_1 \}, \tag{9}$$

where $\mathbf{H}$ is an $m'' \times M$ incidence matrix corresponding to the edge set $\mathbb{E}$ with m'' edges. Specifically, for the l-th edge of $\mathbb{E}$ connecting vertices u_i and u_j, the penalty term $|\beta_k(u_i) - \beta_k(u_j)|$ is represented as $|\mathbf{H}_l \boldsymbol{\beta}_k|$, where $\mathbf{H}_l$ is the l-th row of $\mathbf{H}$ and contains only two nonzero elements; 1 at the i-th index and -1 at the j-th.

The path following type of algorithms (Arnold and Tibshirani, 2016) and alternating direction methods of multipliers (ADMM, Boyd et al., 2011) have been developed to solve graph fused lasso problems. However, the computation of these algorithms can be expensive for a general graph with a large number of nodes. Recall the number of nodes in our graph is the summation of the numbers of observations and dummy points, typically a large number in practice. It is, therefore, challenging to directly apply these conventional algorithms for the implementation of our model.

It is noted that the two approximated log composite likelihood functions in (7) and (8) coincide with the forms of the log likelihood function of a weighted Poisson linear regression and a logistic linear regression, respectively, both of which are concave functions of $\boldsymbol{\beta}$. Below, we propose to combine the proximal gradient method and the alternating direction method of multipliers to solve the convex optimization problem in (9). In particular, we take advantage of the structures of our selected spatial graphs to speed up computation.

Specifically, with the current estimate of the parameters being $\boldsymbol{\beta}^{(t)}$, we follow the proximal gradient method (Beck and Teboulle, 2009) to update the value of $\boldsymbol{\beta}$ iteratively by solving:

$$\boldsymbol{\beta}^{(t+1)} = \underset{\boldsymbol{\beta}}{\arg\min} \frac{1}{2} \left\| \boldsymbol{\beta} - \mathbf{R}^{(t)} \right\|_2^2 + \frac{\lambda}{L} \sum_{k=1}^{p} \|\mathbf{H}\boldsymbol{\beta}_k\|_1, \tag{10}$$

where L is the local Lipschitz constant of $-\frac{1}{|D|}\tilde{\ell}_c(\boldsymbol{\beta}^{(t)})$, $\nabla\tilde{\ell}_c(\boldsymbol{\beta}^{(t)})$ is the first derivative of $\tilde{\ell}_{PL}(\boldsymbol{\beta})$ or $\tilde{\ell}_{LRL}(\boldsymbol{\beta})$ evaluated at $\boldsymbol{\beta}^{(t)}$ for the Poisson based or logistics regression log likelihood respectively, and $\mathbf{R}^{(t)} = \boldsymbol{\beta}^{(t)} + (1/L)\frac{1}{|D|}\nabla\tilde{\ell}_c(\boldsymbol{\beta}^{(t)})$. We can choose L to be the maximum eigenvalue of the Hessian matrix of $-\frac{1}{|D|}\tilde{\ell}_c(\boldsymbol{\beta})$ evaluated at $\boldsymbol{\beta}^{(t)}$.

Now the optimization at each iteration boils down to solving (10), which we propose to use the ADMM algorithm (Wahlberg et al., 2012). By introducing auxiliary variables

$\boldsymbol{\theta} = \{\boldsymbol{\theta}_1, \cdots, \boldsymbol{\theta}_p\}$, Equation (10) is equivalent to:

$$\underset{\boldsymbol{\beta},\boldsymbol{\theta}}{\arg\min} \frac{1}{2} \left\|\boldsymbol{\beta} - \mathbf{R}^{(t)}\right\|_2^2 + \frac{\lambda}{L} \sum_{k=1}^{p} \|\boldsymbol{\theta}_k\|_1 \qquad s.t. \quad \boldsymbol{\theta}_k = \mathbf{H}\boldsymbol{\beta}_k, \quad \forall k = 1, \cdots, p$$

Its augmented Lagrangian function is:

$$\frac{1}{2} \left\|\boldsymbol{\beta} - \mathbf{R}^{(t)}\right\|_2^2 + \frac{\lambda}{L} \sum_{k=1}^{p} \|\boldsymbol{\theta}_k\|_1 + \frac{\gamma}{2} \sum_{k=1}^{p} \|\mathbf{H}\boldsymbol{\beta}_k - \boldsymbol{\theta}_k\|_2^2 + \gamma \sum_{k=1}^{p} \mathbf{u}_k^T (\mathbf{H}\boldsymbol{\beta}_k - \boldsymbol{\theta}_k)$$

where $\mathbf{u} = \{\mathbf{u}_1, \cdots, \mathbf{u}_p\}$ are Lagrangian multipliers, and γ is a penalty parameter. ADMM alternately optimizes $\{\boldsymbol{\beta}, \boldsymbol{\theta}, \mathbf{u}\}$ by solving the following three subproblems:

$$\begin{cases} \boldsymbol{\beta}^{(t+1)} = \underset{\boldsymbol{\beta}}{\arg\min} \{ \left\|\boldsymbol{\beta} - \mathbf{R}^{(t)}\right\|_2^2 + \gamma \sum_{k=1}^{p} \|\mathbf{H}\boldsymbol{\beta}_k - \boldsymbol{\theta}_k^{(t)} + \mathbf{u}_k^{(t)}\|_2^2 \}, \\ \boldsymbol{\theta}_k^{(t+1)} = \underset{\boldsymbol{\theta}_k}{\arg\min} \{ \frac{\lambda}{L} \|\boldsymbol{\theta}_k\|_1 + \frac{\gamma}{2} \|\mathbf{H}\boldsymbol{\beta}_k^{(t+1)} - \boldsymbol{\theta}_k + \mathbf{u}_k^{(t)}\|_2^2 \}, \quad k = 1, \cdots, p \\ \mathbf{u}_k^{(t+1)} = \mathbf{u}_k^{(t)} + \mathbf{H}\boldsymbol{\beta}_k^{(t+1)} - \boldsymbol{\theta}_k^{(t+1)}, \quad k = 1, \cdots, p \end{cases}$$

where t denotes the t-th iteration.

The above sub optimization problems have the following analytical results:

$$\begin{cases} \boldsymbol{\beta}_k^{(t+1)} : (\mathbf{I}_n + \gamma \mathbf{H}^T \mathbf{H})^{-1} [\mathbf{R}_k^{(t)} + \gamma \mathbf{H}^T (\boldsymbol{\theta}_k^{(t)} - \mathbf{u}_k^{(t)})] \\ \theta_k^{(t+1)} : \mathcal{S}(\mathbf{H}\boldsymbol{\beta}_k^{(t+1)} + \mathbf{u}_k^{(t)}; \frac{\lambda}{L\gamma}), \\ \mathbf{u}_k^{(t+1)} = \mathbf{u}_k^{(t)} + \mathbf{H}\boldsymbol{\beta}_k^{t+1} - \boldsymbol{\theta}_k^{t+1}, \end{cases}$$

for each $k = 1, \cdots, p$, where $\mathcal{S}(z, \lambda)$ is the soft-thresholding operator, and $\mathbf{R}_k^{(t)} = \boldsymbol{\beta}_k^{(t)} + (1/L)\frac{1}{|D|}\nabla\tilde{\ell}_c(\boldsymbol{\beta}_k^{(t)})$. It is noted that the above optimization steps are separable for the parameters associated with each k, and hence can be conveniently solved in a parallel fashion. In addition, under our choice of the spatial graphs, the graph Laplacian matrix $\mathbf{H}^T\mathbf{H}$ is a sparse matrix. As a result, the update of $\boldsymbol{\beta}^{(t)}$ only involves the linear solver of the sparse matrix $(\mathbf{I}_n + \gamma \mathbf{H}^T \mathbf{H})^{-1}$, whose sparse Cholesky factorization can be pre-computed efficiently using `R` package `Matrix`. We iterate between (8) and the above ADMM steps until convergence.

4. Simulation Studies

In this section, we conduct simulation studies to investigate the performance of the SVCI model. We design two different data generation scenarios:

- Scenario 1: Point patterns are generated from a Poisson point process on a planar window, where the log intensity is a linear function of an intercept and two covariates with clustered regression coefficients, i.e., $\rho(u; \boldsymbol{\beta}(u)) = \exp\{\beta_0(u) + z_1(u)\beta_1(u) + z_2(u)\beta_2(u)\}$.
- Scenario 2: Point patterns are generated from a Poisson point process on a linear network. We consider two sub-scenarios: (a) The log intensity function is piece-wise constant, i.e., $\rho(u; \boldsymbol{\beta}(u)) = \exp\{\beta_0(u)\}$; (b) The log intensity is a linear function of an intercept and two covariates with clustered regression coefficients as in Scenario 1.

In Scenario 1, we focus on examining the performance of our method under different model choices, including the choice of graphs used in the graph fused lasso penalty and the choice between the Poisson likelihood based SVCI (SVCI-PL) and the logistic likelihood based SVCI (SVCI-LRL). For comparison studies, to the best of our knowledge, there are very limited existing methods available for spatially clustered coefficient log linear point process models on complex domains as reviewed in the Introduction, except for the simple case of an intercept-only log-linear model. As such, Scenario 2(a) is included so that we can compare SVCI with the nonparametric kernel density estimation method on a linear network (KDE.lpp) proposed in McSwiggan et al. (2017), the fast KDE method (KDEQuick.lpp) in Rakshit et al. (2019), and the resample-smoothed Voronoi intensity estimation method (Voronoi.lpp) in Moradi et al. (2019). For the case that has spatial covariates as in Scenario 2.(b), the comparison is made with the LGCP model (Møller et al., 1998), in which the inhomogeneity of the intensity function is modeled by a latent spatial Gaussian process random effects model.

Given the estimator $\hat{\boldsymbol{\beta}}$ defined in (9), we can predict the coefficients at any given new location $u \in D \backslash \{u_1, \cdots, u_M\}$ according to $\hat{\boldsymbol{\beta}}(u) = \sum_{i=1}^{M} \mathbf{1}_{\{u_i \in \mathcal{N}_K(u)\}} \hat{\boldsymbol{\beta}}(u_i)/\mathrm{K}$, where $\mathcal{N}_K(u)$ denote the K nearest neighbors of u. To quantify the performance of parameter estimation, we evaluate the estimation accuracy of $\beta_k(u)$ by the mean integrated squared error (MISE_β, Davis (1977)), defined as

$$\mathrm{MISE}_\beta = \frac{1}{p|D|} \sum_{k=1}^{p} \int_D (\beta_k(u) - \hat{\beta}_k(u))^2 du$$

We implement our methods in R (see https://github.com/LihaoYin/SVCI). The data generations are done using the R package `spatstat` (Baddeley and Turner, 2005). The competing KDE.lpp and KDEQuick.lpp methods are implemented using `density.lpp` and `densityQuick.lpp` in R package `spatstat`, respectively. Voronoi.lpp is implemented using `densityVoronoi.lpp` also in R package `spatstat`. The competing LGCP method is implemented in R using the `lgcp` function provided in `geostatsp` (Brown, 2015). All computations were performed on a Mac Pro with 2.4 GHz Intel Core i7 laptop with 8GB of memory.

4.1. Simulation Scenario 1

In Simulation Scenario 1, we consider a spatial 2D window $D = [0, R]^2 \subset \mathbb{R}^2$, where the true regression coefficients in the log-intensity function are assumed to have clustering patterns as shown in the top panel of Figure 3. We simulate the two covariates $\{z_1(u)\}$ and $\{z_2(u)\}$ from two independent realizations of a spatial GP with mean zero and an isotropic exponential covariance function taking the form of $\mathrm{Cov}\{z_k(u), z_k(v)\} = \sigma^2 \exp(-\|u-v\|/\phi)$, $k = 1, 2$, $u, v \in [0, R]^2$, where the range parameter $\phi = 0.3R$ corresponding to a moderate spatial correlation setting, and $\sigma^2 = 1$.

Under one chosen fixed coefficient pattern, we experiment with a range of R values to simulate one realization from the Poisson point process model described in Scenario 1 such that the number of simulated points ranges from 800 to 6000 on average, in order to examine the performance of SVCI as the sample size increases with the expanding domain. Furthermore, we report the model performance for three different numbers of dummy points $\ltimes$: (a) $\mathtt{nd}^2 < m$; (b) $\mathtt{nd}^2 = m$; (c) $\mathtt{nd}^2 > m$, where m is the number of the observed points. We also compare with an LGCP model with intensity function $\log \rho(u) = z(u)^T \boldsymbol{\beta} + \phi(u)$, where $\boldsymbol{\beta}$ are the constant-coefficients across the domain, and $\phi(u)$ is a spatial Gaussian process with a zero mean and a Mátern correlation function.

As discussed in Section 3, the selection of connection graphs for the fused lasso penalty plays critical roles on the estimation accuracy and computation speed. In this study, we compare the performance of SVCI using three types of connection graphs, including the minimum spanning tree graph (MST), the Delaunay triangulation (DTs) and the K-nearest neighbor graph (K-NNs, K is set to be $3, 4, 5$). Also see a comparison study between K-NNs and r-NNs in Appendix Section A3. We run 100 repeated experiments of the SVCI model using each connection graph for both SVCI-PL and SVCI-LRL with a fix number of dummy points $\mathtt{nd}^2 = m$.

In Table 1, we report the averaged MISE of the estimated $\boldsymbol{\beta}$ (MISE_β). There are several noticeable observations. First, a denser graph such as the 5-NN graph produces a more

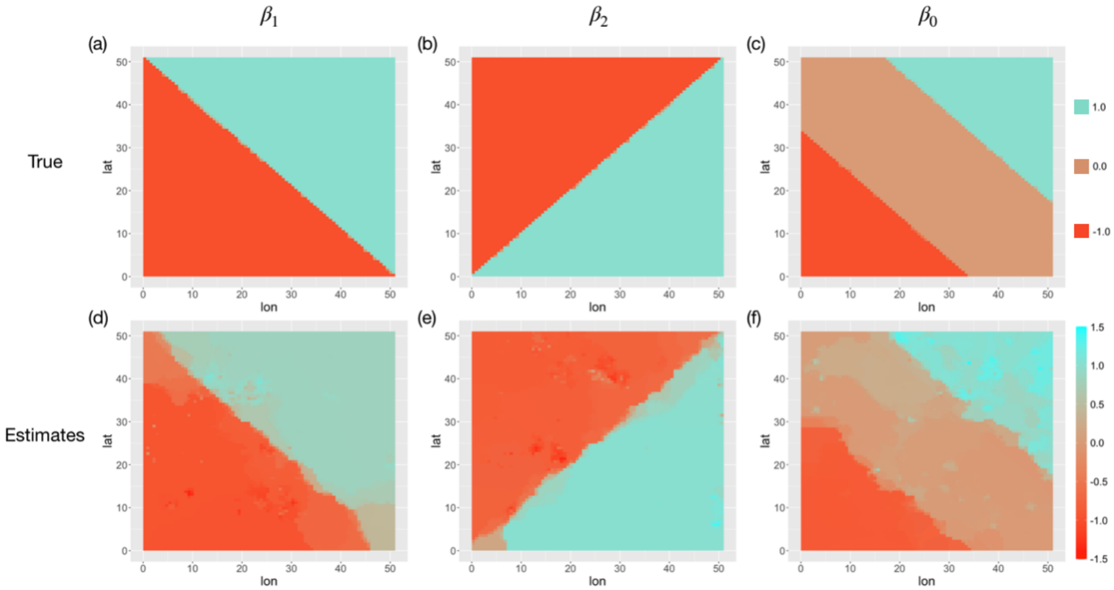

Figure 3: The top panel (a-c): the true coefficients $\boldsymbol{\beta}_0$, $\boldsymbol{\beta}_1$ and $\boldsymbol{\beta}_2$ in Scenario 1; The bottom panel (d-f): the estimated coefficients from one simulation using the 5NN graph when $m = 2400$, $\texttt{nd}^2 = m$

accurate estimation result compared with that of a sparser graph such as the MST or 3-NN graph. The bottom panel of Figure 3 illustrates an example of the estimated coefficients using the 5-NN graph when $m = 2000$, $\texttt{nd}^2 = m$, which demonstrates the capability of SVCI in capturing the cluster structure in the regression coefficients and detecting the abrupt changes across the boundaries of adjacent clusters. However, there is clearly a trade off between the estimation accuracy and computation efficiency when using different graphs; as reported in the left panel of Figure 4, the computation time (in seconds) using the 5-NN or Delaunay triangulation graph is roughly 1.5 times of the computation time using the MST. Second, the parameter estimation is more accurate when m grows larger, as evidenced by the decreasing value of MISE_β. Finally, SVCI-LRL produces comparable results with those from SVCI-PL when $m = 800$ or using the MST graph, but it notably outperforms SVCI-PL when $m \geq 1600$. This is consistent with the findings in Baddeley et al. (2014), which showed that for datasets with a large number of points or a highly structured point pattern, the logistic likelihood method produces a less biased estimator than its Poisson counterpart.

We further examine the performance of the SVCI model in terms of recovering the intensity function. Given $\hat{\boldsymbol{\beta}}(u)$, we obtain the estimate of the log intensity function by $\log\hat{\rho}(u) = \mathbf{z}^T(u)\hat{\boldsymbol{\beta}}(u)$ for $u \in D$. The right panel of Figure 4 compares the MISE of $\log\hat{\rho}(u)$ from SVCI-PL, SVCI-LRL and LGCP, respectively. It is noted that SVCI-LRL maintains its superior performance when predicting the intensity function in comparison with SVCI-PL. Besides, both versions of SVCI produce more accurate estimates than LGCP when estimating the intensity function with clustered regression coefficients.

Next we examine the performance of SVCI in recovering the clusters of coefficients. Table 2 reports the Rand index for each of the SVCI estimates $\hat{\boldsymbol{\beta}}_1$, $\hat{\boldsymbol{\beta}}_2$ and $\hat{\boldsymbol{\beta}}_0$ averaged over 100 simulations, using the 5-NN graph and setting $\texttt{nd}^2 = m$. Rand index measures the proportion of pairs consisting of a true parameter and the corresponding estimated parameter that agree by virtue of belonging either to the same cluster or to different clusters. Overall, SVCI achieves an accurate cluster recovery result, evidenced by the relatively high Rand index value ranging from 0.73 to 0.93 in all settings. We also find that SVCI-LPL surpasses SVCI-PL in detecting spatial clusters. Finally, an interesting observation is that $\hat{\boldsymbol{\beta}}_0$ has a lower Rand index value than both $\hat{\boldsymbol{\beta}}_1$ and $\hat{\boldsymbol{\beta}}_2$, which might be the consequence of having more clusters in the true function of $\boldsymbol{\beta}_0$.

Finally we check the sensitivity of the model performance to the number of dummy points. We fix $m = 1600$ and consider three different numbers of dummy points denoted by $\texttt{nd}^2$. Table 3 presents the averaged MISE_β and the associated computation time over 100 simulations. For the Poisson likelihood, the default choice suggested in the R package `spatstat` is $\texttt{nd}^2 \approx 4m$. In our experiments, however, as presented in Table 3, both SVCI-PL and SVCI-LRL achieve the minimal MISE_β when $\texttt{nd}^2 = 60^2$, i.e. when the number of dummy points roughly equals the number of points. Moreover, based on the results in Table 3, we observe that when $\texttt{nd}^2$ is not too large, both SVCI-PL and SVCI-LRL seem to achieve a smaller MISE_β but at a higher computation cost as $\texttt{nd}^2$ increases. Weighing the trade-off between computation efficiency and estimation accuracy, we recommend to use $\texttt{nd}^2 \approx m$ in practice.

Table 1: Scenario 1: Mean integrated squared error of $\boldsymbol{\beta}$ (MISE_β) averaged over 100 simulations for different values of m with $\texttt{nd}^2 = m$, different connection graphs, and the Poisson-based SVCI-PL method and the logistic regression based SVCI-LRL method.

Method	$m = 800$		$m = 1600$		$m = 2400$		$m = 3600$		$m = 6000$	
	PL	LRL	PL	LRL	PL	LRL	PL	LRL	PL	LRL
MISE_β										
MST	0.234	0.243	0.222	0.224	0.189	0.191	0.152	0.146	0.130	0.125
3-NN	0.223	0.230	0.204	0.182	0.189	0.184	0.155	0.142	0.127	0.115
4-NN	0.214	0.209	0.200	0.181	0.157	0.132	0.133	0.116	0.115	0.098
5-NN	0.209	0.195	0.184	0.153	0.141	0.119	0.122	0.105	0.095	0.078
DT	0.215	0.211	0.174	0.141	0.128	0.111	0.113	0.097	0.080	0.072

Table 2: Scenario 1: Rand index of the estimates $\hat{\boldsymbol{\beta}}_1$, $\hat{\boldsymbol{\beta}}_2$ and $\hat{\boldsymbol{\beta}}_0$ (averaged over 100 Monte Carlo simulations) for SVCI-PL and SVCI-LRL for different values of m, using $\texttt{nd}^2 = n$ and 5-NN connection graphs.

	$m = 800$		$m = 1600$		$m = 2400$		$m = 3600$		$m = 6000$	
	PL	LRL	PL	LRL	PL	LRL	PL	LRL	PL	LRL
Rand Index										
$\hat{\boldsymbol{\beta}}_1$	0.817	0.839	0.857	0.874	0.883	0.903	0.897	0.915	0.917	0.930
$\hat{\boldsymbol{\beta}}_2$	0.817	0.836	0.851	0.867	0.879	0.906	0.893	0.917	0.914	0.925
$\hat{\boldsymbol{\beta}}_0$	0.729	0.760	0.742	0.773	0.767	0.786	0.787	0.806	0.803	0.825

4.2. Simulation Scenario 2

In Scenario 2, we generate spatial points on the `chicago` linear network from the `R` package `spatstat`. The network is shown in the left panel of Figure 5, which consists of the road network in an area of Chicago, USA near the University of Chicago (Baddeley

Table 3: Scenario 1: Comparing the MISE_β (averaged over 100 simulations) and computation time (in seconds) between SVCI-PL and SVCI-LRL for different numbers of dummy points, using $m = 1600$ and 5-NN connection graphs.

dummy points	SVCI-PL		SVCI-LRL	
	MISE_β	time(s)	MISE_β	time(s)
$\texttt{nd}^2 = 30^2$	0.195	1.54	0.164	1.61
$\texttt{nd}^2 = 40^2$	0.184	1.86	0.153	1.81
$\texttt{nd}^2 = 60^2$	0.182	2.38	0.151	2.30
$\texttt{nd}^2 = 80^2$	0.190	3.35	0.167	3.24

and Turner, 2005). We bound the linear network in a window $D = R \times [0,1]^2$ and increase R to expand D, so that the linear network that resides in D grows with D at the same rate to obtain an increasing number of realizations on the network.

We first consider a simplified case where there is no covariate available. We focus on the estimation of intensity function whose true value is a piece-wise constant function, that is, the intensity function $\rho(u) = \exp\{\beta_0(u)\}$, and the true value of $\beta_0(u)$ has a clustered pattern as shown in the left panel of Figure 5. The original graph is constructed following the method described in the last paragraph of Section 3.1. The upper part of Table 4 presents the MISE of $\log \rho$, i.e., the log intensity function for each value of m. In general, we obtain similar findings on the linear network as on a planar window presented in Scenario 1; MISE from both the Poisson based and logistic based SVCI models show a convergence tendency as the domain expands and m goes up, and the logistic likelihood based method achieves a slightly more accurate estimation than the Poisson based method with a large number of points. It is clear from Table 4 that both the SVCI-PL and SVCI-LRL models outperform the KDE based and resample-smoothed Voronoi based intensity estimation methods in almost all settings. Previous studies (Barr and Schoenberg, 2010) show that KDE estimators may suffer from the problem of having substantial bias and high variance when there are abrupt changes in the intensity. Both SVCI and Voronoi.lpp are designed to alleviate this problem, as evidenced by their improved performance over the two KDE methods in Table 4. Nevertheless, SVCI seems to be more effective than Voronoi.lpp to capture abrupt changes or clustering patterns. We illustrate an example in the right panel of Figure 5, which plots the true and the estimated log intensity along a selected road segment from one simulation. It clearly shows that SVCI captures the intensity with discontinuities more efficiently than KDE.lpp.

We also compare the computation time of each method and report the detailed results

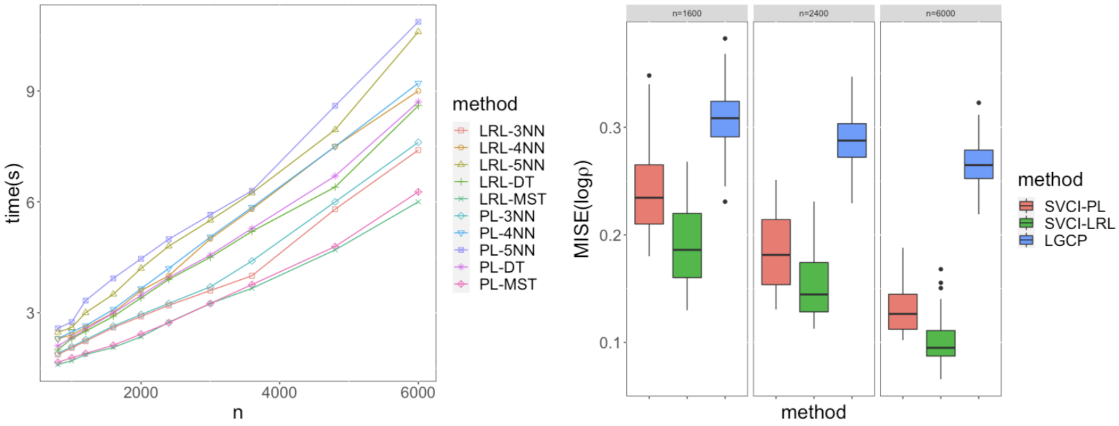

Figure 4: Left: Computation time to solve the optimization for one tuning parameter using different connection graphs; Right: the boxplots of MISE$_\beta$ for SVCI-PL, SVCI-LRL with 5-NN graphs and LGCP. Reported results are averaged over 100 Monte Carlo simulations under Scenario 1.

in Appendix Table A2 for various values of m. Taking $m = 2400$ as an example, to get one estimate, KDE.lpp requires 4.95 seconds, KDEQuick.lpp requires 0.084 seconds, and Voronoi.lpp requires 4.95 seconds. In contrast, SVCI-LRL need 0.93 seconds to construct the connection graph and 1.11 seconds to get an estimate. Although SVCI is not the fastest among the compared methods, overall, its computation is still reasonable and competitive, especially considering its superior performance in intensity estimations.

We then consider the case with an intercept and two covariates as described in Scenario 2(b). The true regression coefficients are plotted in the subfigures (a-c) of Figure 6. The subfigures (d-f) of Figure 6 give the estimated coefficients from SVCI on the `chicago` network. The results demonstrate that our method is also capable of capturing the clustered coefficient patterns on a linear network. In addition, the log intensity estimation results presented in the lower part of Table 4 are in general consistent with the findings presented in Scenario 1 and Scenario 2(a), that is, the performance of the SVCI model with the logistic regression likelihood or with a larger m is more preferable. Table 4 displays the results from LGCP as a comparison, which indicate a clear improvement of using SVCI over LGCP in terms of estimation accuracy.

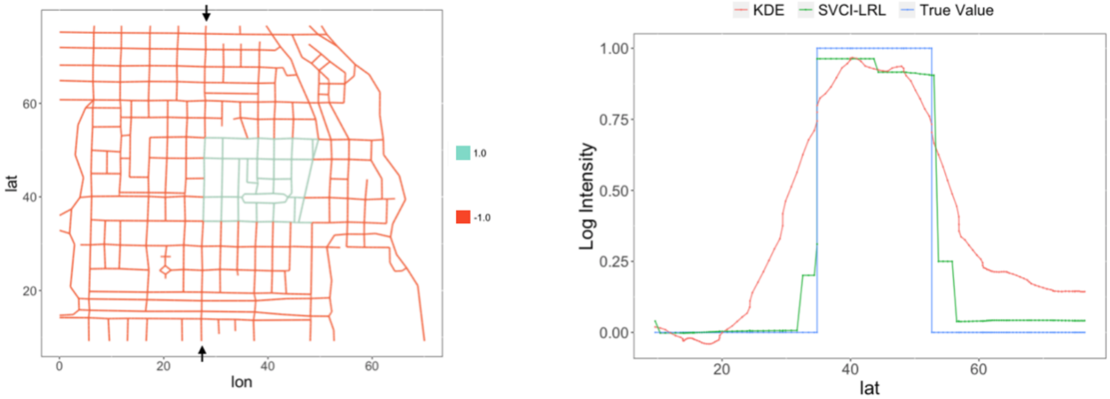

Figure 5: Left: The true log intensity β_0 on the `chicago` network in Scenario 2(a); Right: The true and the estimated log intensity along one road segment corresponding to the line between the two black arrows in the left panel of Figure 5.

Table 4: Scenario 2: mean integrated squared error of log intensity ($\text{MISE}_{\log\rho}$) averaged over 100 simulations for different values of m. We compare the two estimating equations, the Poisson likelihood (PL) and the logistic regression likelihood (LRL) with their competitors.

Method	$\text{MISE}_{\log\rho}$				
	$m = 800$	$m = 1600$	$m = 2400$	$m = 3600$	$m = 6000$
(a): $\rho(u) = \exp\{\beta_0(u)\}$					
SVCI-PL	0.128	0.101	0.084	0.057	0.041
SVCI-LRL	0.117	0.095	0.074	0.042	0.030
KDE.lpp	0.157	0.140	0.112	0.084	0.061
KDEQuick.lpp	0.133	0.127	0.109	0.075	0.054
Voronoi.lpp	0.128	0.120	0.094	0.067	0.048
(b): $\rho(u) = \exp\{z_1(u)\beta_1(u) + z_2(u)\beta_2(u) + \beta_0(u)\}$					
SVCI-PL	0.177	0.152	0.135	0.114	0.085
SVCI-LRL	0.165	0.141	0.122	0.097	0.072
LGCP	0.227	0.202	0.188	0.154	0.137

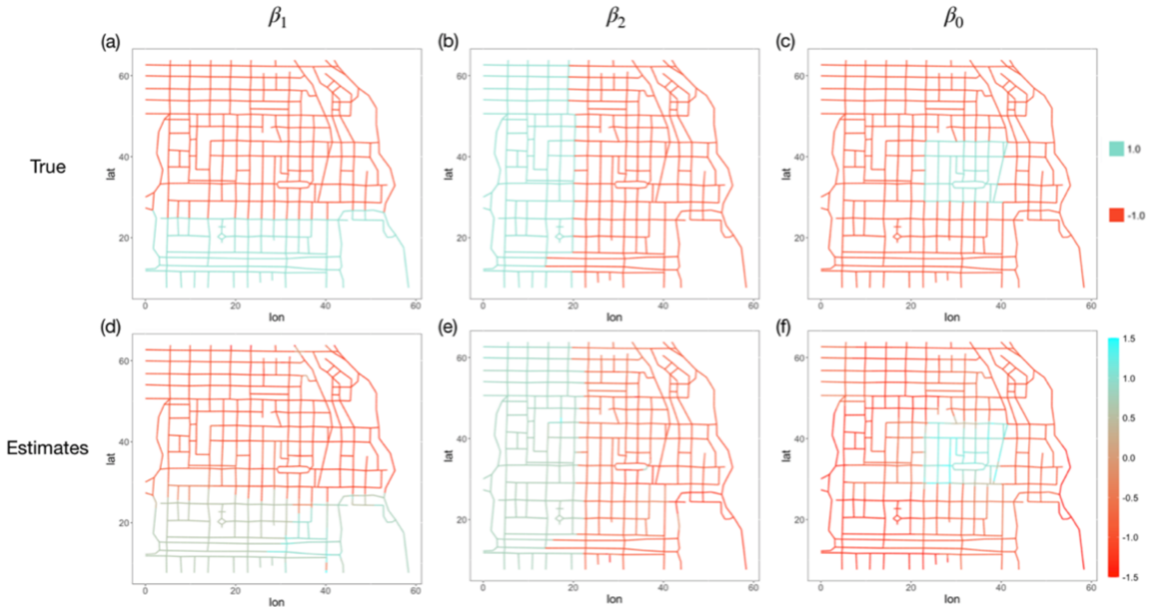

Figure 6: The top three panels (a-c): the spatial structures of true coefficients $\boldsymbol{\beta}_1$, $\boldsymbol{\beta}_2$ and $\boldsymbol{\beta}_0$ in Scenario 2(b); The upper three panels (d-f): the estimated coefficient surfaces from in one simulation using SVCI-LRL with $m = 2400$.

5. Real Data Analysis

We consider two real data examples to illustrate the performance of the proposed method. The first Toronto Homicide data example has a moderate data size with 1398 points and three explanatory variables on a domain with irregular boundaries. And the second Western Australia Traffic Accidents data has a larger data size with $14,562$ points on a linear road network. In both studies, we use SVCI-LRL and $\texttt{nd}^2 \approx m$, due to their favorable performance in our simulation studies.

5.1. Toronto Homicide Dataset

We apply the proposed SVCI model to the analysis of the Toronto Homicide dataset. The raw dataset contains the information of 1398 homicides occurred in Toronto, Canada during 1990 to 2014, recording the locations of murder scenes, homocide types and information of victims obtained from the Toronto Star Newspaper (`http://www.thestar.com/news/crime/torontohomicidemap.html`). We select the more recent years since 2000 and delete the data which have duplicated locations. There remains 764 homicide cases for the final analysis. Figure 1 shows the entire Toronto city and the locations of the selected cases within a 42 × 31 km rectangle window. Notably, the old Toronto region in the middle of the coast has more frequent occurrences of homicides.

The data also contains the records of average income, night lights and population density of Toronto city in 2006, and we use them as the explanatory variables. Figures 7 (a-c) show the observations of the three variables. As can be seen, there is a large spatial variation of these variables across the city. We scaled and centered each spatial covariates before running our point process models.

We focus on the investigation of the relationship between the distribution of homicides and the three explanatory variables. We first fit a standard log-Gaussian cox process model with constant regression coefficients as a benchmark for comparisons. The intensity function of LGCP takes the form

$$\log\{\rho(u)\} = \beta_0 + \mathrm{Income}(u)\beta_1 + \mathrm{Night}(u)\beta_2 + \mathrm{Pdens}(u)\beta_3 + \phi(u)$$

where β_1, β_2 and β_3 are the constant-coefficients across the domain and $\phi(u)$ is a spatial Gaussian process with a zero mean and a Matern correlation function. After centering and scaling the covariates, we obtain the parameter estimates from LGCP as $\beta_1 = -0.918$, $\beta_2 = 0.378$, $\beta_3 = 0.207$ and $\beta_0 = 1.096$. These estimates imply that the homicides are more likely to occur in the area with a lower average income, a better lights condition and a denser residential population.

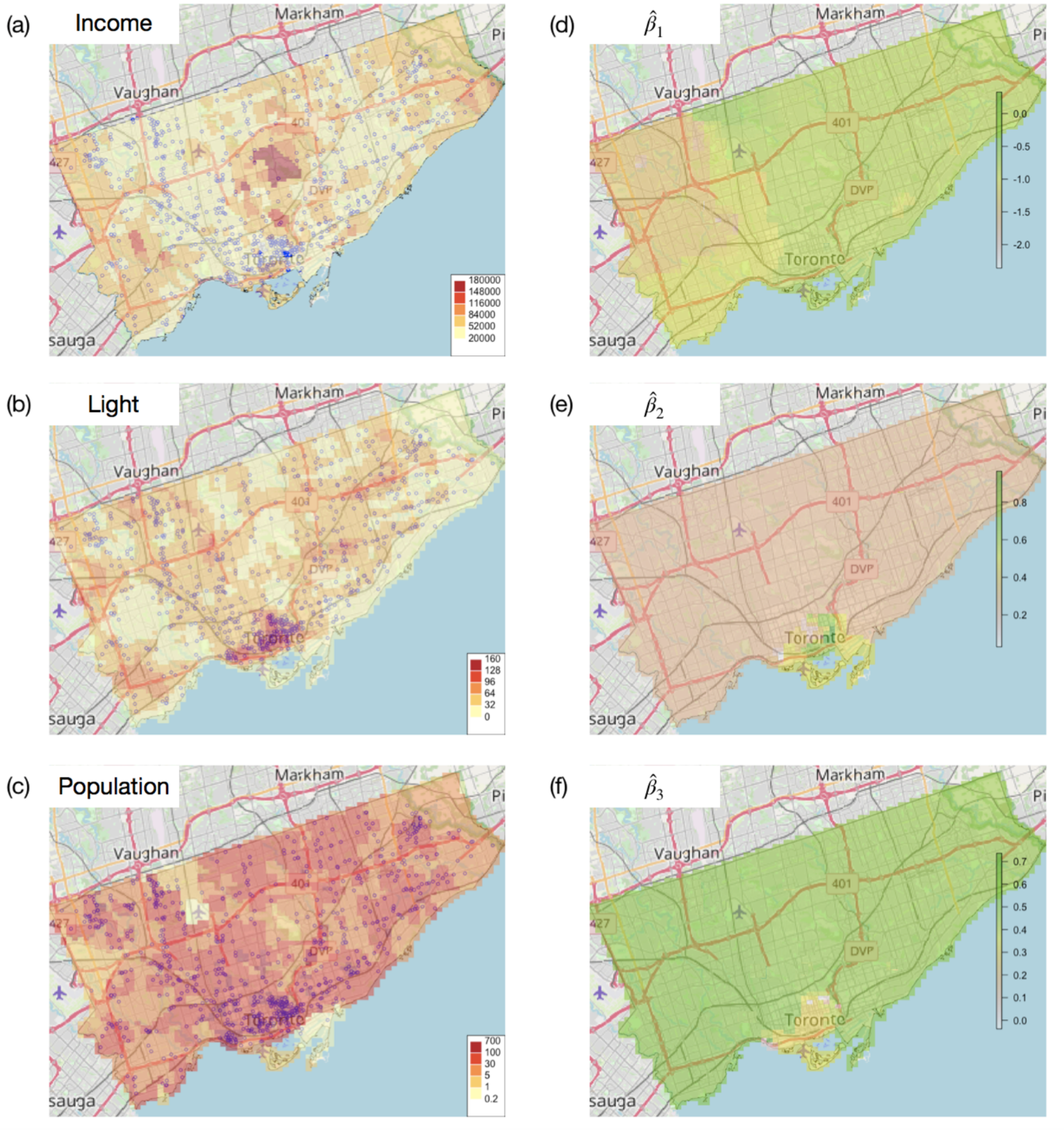

Figure 7: Left Panel: patterns of covariates; Right Panel: patterns of estimates; (a,d) Average income of the residents in Toronto; (b,e) Light intensity in Toronto night; (c,f) Population density in Toronto;

We then fit the SVCI model with a 5-NN graph assuming that the homicide locations follow a spatial point process with the following intensity function,

$$\log\{\rho(u)\} = \text{Income}(u)\beta_1(u) + \text{Night}(u)\beta_2(u) + \text{Pdens}(u)\beta_3(u) + \beta_0(u)$$

Here $\beta_k(u)$, $k = 0, 1, 2, 3$ are spatially varying coefficients, whose estimates are shown in Figure 7(d-f). It takes about 0.072 seconds to construct the 5-NN graph and 0.69 seconds to get an estimate of $\boldsymbol{\beta}$ for each tuning parameter. Clearly, the results of SVCI reveal more details about the effects of covariates than those from LGCP. 3, 4 and 3 major clusters were detected for β_1, β_2 and β_3 respectively. Overall, the signs of $\beta_k(u)$ are the same as the results of LGCP. For example, the estimates of $\boldsymbol{\beta}_1(u)$ range from -2 to 0, indicating a negative relationship between income and homicide occurrence as is expected. Such a negative relationship is most prominent in the western region of Toronto whereas a weaker relationship is observed near the upper east corner of Toronto City. For both β_2 and β_3, we observe a small cluster at the Old Toronto region, which has the most concentrated homicide cases. It is notable that the relationships between light intensity/population density and homicides occurrence in the Old Toronto region differ significantly from the rest of Toronto city; a stronger positive relationship is observed for both variables.

5.2. Western Australia Traffic Accidents

In this section, we study the traffic accidents data in the state of Western Australia for the year 2011, as shown in Figure 1. The data were originally provided by the Western Australian State Government Department of Main Roads and are made publicly available as part of the Western Australian Whole of Government Open Data Policy. The data can also be accessed from the `R` package `spatstat.Knet`. It consists of $14,562$ locations of accidents on a road network with $115,169$ road segments constrained in a $[217.4, 1679.1] \times [6114.9, 7320.6]$ km rectangle window.

The grey lines in Figure 1 represent the traffic network of Western Australia and each red point denotes an accident spot. It is clear from this Figure that accidents are highly concentrated around the Perth metropolitan area located in the western coastal region. This region contains nearly 75% of the population in Western Australia. By contrast, the remote eastern region has a much sparser road network and a smaller number of traffic accidents. Our goal is to estimate the intensity function over this network to investigate the spatial variation of accident occurrences.

We build the SVCI model of the intensity function with a spatially varying intercept, $\rho(u; \beta) = \exp\{\beta_0(u)\}$. In this study, we don't have any spatial covariates available and hence we focus on detecting the clustered patterns of the intensity function $\rho(u)$. SVCI

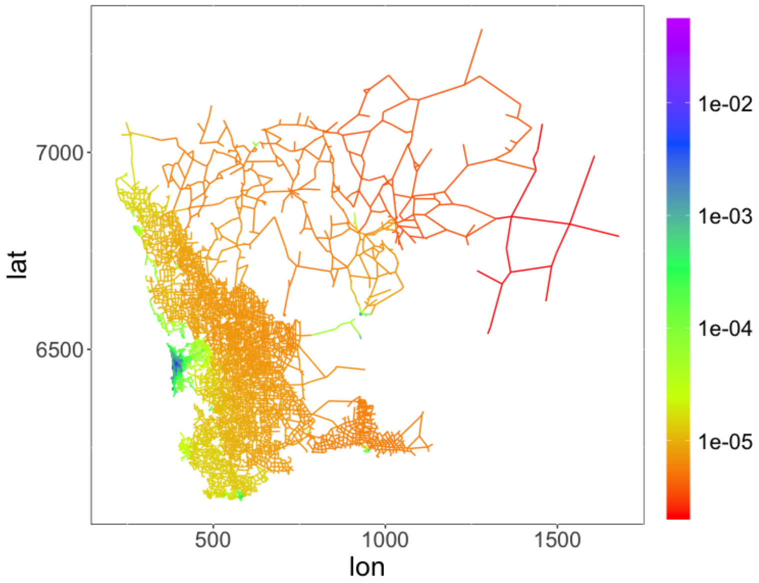

Figure 8: Intensity estimates for the accidents on the Western Australian road network.

takes about 1.75 minutes to construct a connection graph using the graph construction method in Section 3.1 and takes on average 5.01 seconds to get an estimate of $\boldsymbol{\beta}$ for each tuning parameter. 23 clusters were detected, and Figure 8 plots the estimated log intensity $\log \hat{\rho}(u)$ on the road network. In the western part of the city near downtown, there are many small clusters, indicating more local variations in traffic accidents intensities in these regions. In contrast, the eastern and northern part of the city has a fewer number of clusters. We also notice that $\hat{\rho}(u)$ has a large spatial variation, ranging from 0 per kilometer in some remote eastern areas to nearly 50 accidents per kilometer in some busy roads in the Perth metropolitan area.

We zoom into the sub-region of $[372, 431] \times [6434, 6501]$ km, which is displayed in Figure 9 (a), to have a detailed investigation of the traffic accident rates in the densely populated Perth metropolitan area. It is clear that several roads are having substantially higher intensities than the rest of the roads, many of which are along the major freeways of the city. In particular, we observe very high intensity values at or near the center of the city marked by the purple color. Indeed, these roads and intersection are located at the Perth Central Business District. In contrast, although having dense local road networks, many residential areas away from highways have relatively lower intensity values. One advantage

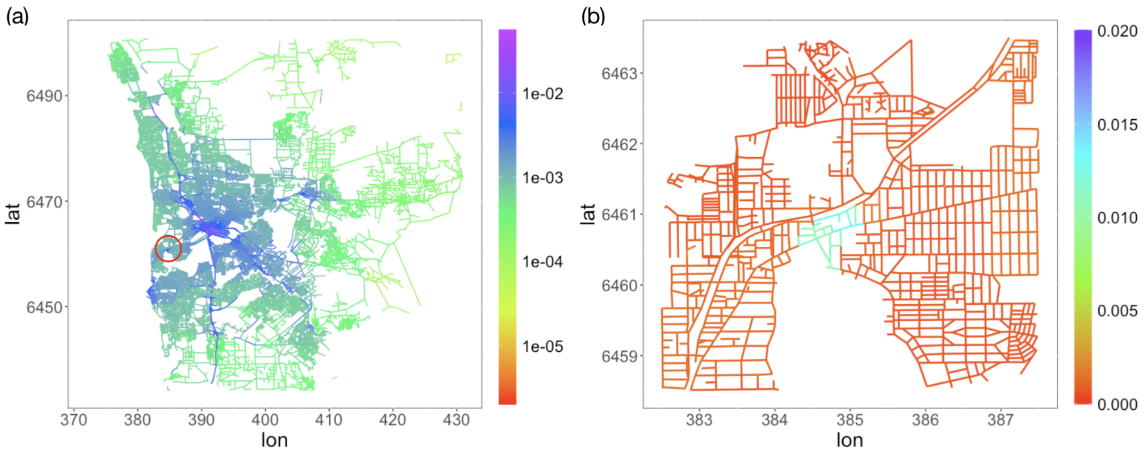

Figure 9: (a): Intensity estimates map for the accidents in the metropolitan Perth Area; (b): Intensity estimates map by zooming into the red circle area in the left panel.

of SVCI lies in its capability of capturing intensity functions with abrupt changes. To give an example, we highlight a road segment on Highway 5 in Perth by a red circle in Figure 9 (a), and show the zoomed map in Figure 9 (b). It is noticeable that a sudden jump in the estimated intensity function appears near the southwest end of the road. After verifying with the Google satellite image, we confirm that the northeast part of the road passes through a large residential area, whereas the southwest part is a commercial and public service area (restaurants/shops/school/church/hospital) that is expected to have a higher rate of accidents.

6. Conclusion

In this study, we propose a varying coefficient log-linear intensity model, referred to as the SVCI model, for the visualization and analysis of spatial point processes. We utilize a graph fused lasso regularization to estimate the clustering patterns of the regression coefficients. The method guarantees spatially contiguous clustering configurations with highly flexible cluster shapes and data-driven cluster sizes. It supplements the current research on intensity estimation, which primarily focuses on relatively smooth intensity functions without covariates or spatially constant regression coefficients. The method also has the advantage of being applicable to a broad range of complex domains such as line networks and spatial domains with irregular boundaries. The computation of the model is made highly efficient by using a proximal gradient optimization algorithm. Numerical studies show that our method produces more accurate intensity estimations than several competing methods such as the KDE-based methods (McSwiggan et al., 2017; Rakshit et al., 2019) and the resample-smoothing Voronoi intensity estimation method (Moradi et al., 2019), when intensity functions exhibit discontinuous changes on linear networks. The computation of SVCI is in general reasonable compared to its competitors considered in this paper. The method is applied to identify spatially heterogeneous patterns in the determinants of Toronto crime events and the intensity of traffic accidents in Western Australia.

Moving forward, this work could be further refined in several ways. First, SVCI only considers a small fixed number of covariates. However, in practice, practitioners may face a large number of available covariates but lack a strong theory to inform variable selection. There is a need of a more general model that allows researchers to undergo variable selection and spatial cluster identification simultaneously for point processes. Second, the SVCI estimator does not come with an uncertainty measure that makes it hard for statistical inference, a common issue shared by many regularization based approaches. We may consider a Bayesian version of the method or a bootstrapping based approach to address the inference problem. Third, an interesting research direction is to extend the finite dimensional graph regularization based method to an infinite dimensional process-based clustered coefficient model such that spatial predictions can be done in a more rigorous way. Finally, the method can be extended by considering a weighted composite log-likelihood to further improve statistical efficiency for non-Poisson point processes (Guan and Shen, 2010).

Appendix

A1. Quadrat Scheme

For the quadrature approximation in (4), we need to divide the domain into small subdivisions or quadrats. For the 2-D square domain, it is easy to add dummy points and draw regular equal-sized rectangular quadrats on the domain using `dummy.ppm` and `quad.ppm` in `R` library `spatstat`. For planar domains with irregular boundary, we follow the routine quadrature approximation methods for irregular domains in $\mathbb{R}^2$ (Shen et al., 2009), where the domain is masked by regular pixel grids, and we expand the domain slightly to include the whole pixel if it has intersections with domain boundary. See the left panel of Figure A1 for an example.

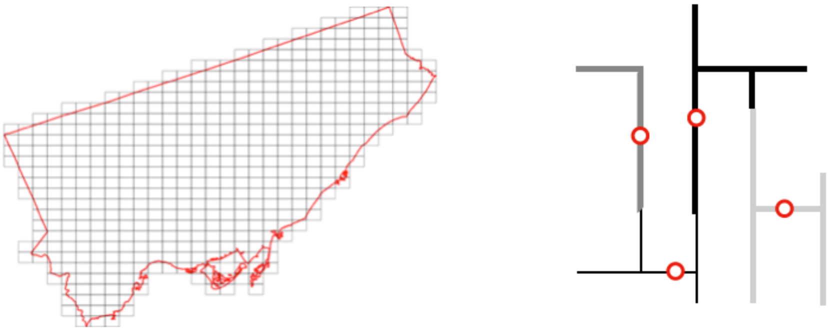

Figure A1: Illustration of quadrat schemes for Toronto city (left) and a toy linear network example (right); Left panel: boundary lines (red) of Toronto city and the grids (grey) covering the irregular domain; Right panel: dummy points (red nodes) and their corresponding Voronoi quadrats (marked by different shades of grey).

As for the partition on a linear work, Chapter 9 of Okabe and Sugihara (2012) discussed the implementation of both equal-length and unequal-length network cells. Furuta et al. (2008); Shiode (2008) proposed computational methods for dividing a network into equal-length network cells. However, their methods are not a guarantee of success if we want to insert enough dummy points. We propose to randomly draw dummy points from a homogeneous Poisson point process on the linear network with intensity function $\delta(u) = (M - m)/|D|$. Then we obtain the quadrats on linear networks using the network Voronoi tessellation method (Chapter 4 of Okabe and Sugihara, 2012), each of which contains one dummy point as its centroid. See the right panel of Figure A1 for an example.

A2. Addition Numerical Results

We report the integrated squared bias (ISB) and variance (IV) for each $\hat{\boldsymbol{\beta}}_1$, $\hat{\boldsymbol{\beta}}_2$ and $\hat{\boldsymbol{\beta}}_0$ under both Scenario 1 and Scenario 2(b) in Table A1. Specifically, ISE and IV are defined as

$$\mathrm{ISB}(\boldsymbol{\beta}_i) = \frac{1}{|D|}\int_D (\hat{\beta}(u) - \mathbb{E}\hat{\beta}(u))^2 du, \qquad \mathrm{IV}(\boldsymbol{\beta}_i) = \mathrm{MISE}(\boldsymbol{\beta}_i) - \mathrm{ISB}(\boldsymbol{\beta}_i).$$

The results generally agree well with the findings based on Rand index and the MISE_β, in the sense that SVCI-LRL achieves a slightly smaller bias and variance compared with SVCI-PL.

Table A1: Scenario 1: The integrated squared bias and variance (in parentheses) of $\hat{\boldsymbol{\beta}}_1$, $\hat{\boldsymbol{\beta}}_2$ and $\hat{\boldsymbol{\beta}}_0$ respectively, in the case that $m = 2400$, and $\texttt{nd}^2 = m$ based on 5-NN connection graphs.

	$\boldsymbol{\beta}_1$	$\boldsymbol{\beta}_2$	$\boldsymbol{\beta}_0$
Scenario 1			
SVCI-PL	0.115(0.035)	0.116(0.032)	0.078(0.038)
SVCI-LRL	0.098(0.034)	0.105(0.033)	0.057(0.030)
Scenario 2(b)			
SVCI-PL	0.099(0.028)	0.104(0.030)	0.108(0.035)
SVCI-LRL	0.105(0.028)	0.101(0.027)	0.093(0.030)

Under the setting of Scenario 1, we examine the performance of SVCI-PL based on K-NN graphs with $K = 3, 4, 5, 6, 9$ and r-NN graphs with $r = 0.02R, 0.03R$. Figure A2 shows the MISE_β versus the number of edges for each graph. We notice that MISE does not always decrease as the graph includes more neighbors by increasing K of K-NN or r of r-NN. Both K-NN and r-NN lose some estimation accuracy if there are too many neighbors. It also seems that K-NN slightly outperform r-NN in terms of MISE when K and r are chosen such that the two graphs have comparable number of edges.

Under the setting of Scenario 2(a), we compare the performance of SVCI-LRL using the 3-NN graphs constructed based on the shortest-path distance and Euclidean distance metrics, respectively. Overall, we observe very similar results between these two choices of distance metrics, especially when m and $\texttt{nd}^2$ go up. For example, when $m = 2400$, the MISEs of SVCI-LRL using shortest-path distance and Euclidean distance metrics are 0.121 and 0.124, respectively, which are very close to the result reported in Table 4 obtained using the graph constructed by connecting natural neighbors on the Chicago network.

Table A2 reports the computation time of different methods with one tuning parameter, under the setting of Scenario 2(a). We notice that the computation time of KDE.lpp and Voronoi.lpp vary notably with their tuning parameter, so we selected a range of tuning parameters in the proximity of the optimal tuning parameter and computed the average computation times.

For the Toronto Homicide data considered in Section 5.1, we compare the estimated log intensity surfaces obtained by SVCI and LGCP in Figure A3. Both methods seem to be capable of capturing the inhomogeneity pattern in intensities and agree well with each

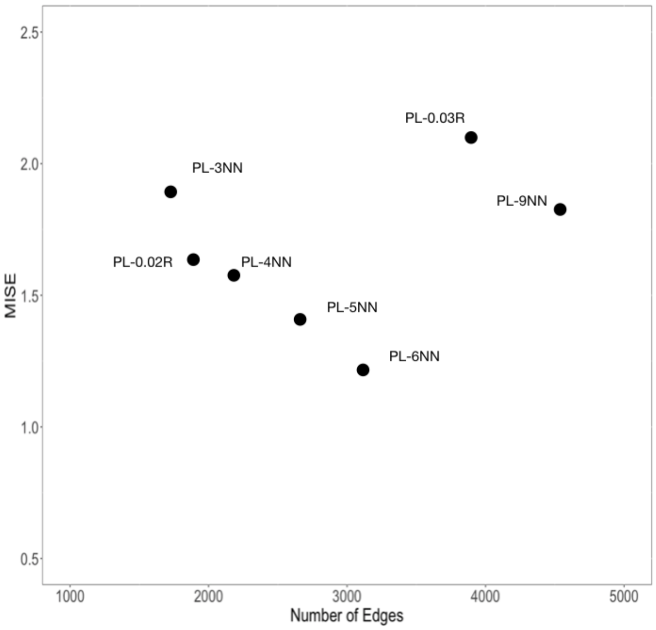

Figure A2: Illustration of MISE_β versus the number of edges, under the setting of Scenario 1, with $m = 2400$ and $\texttt{nd}^2 = m$. ϵR indicates the SVCI-PL method based on an r-NN graph with a radius $\epsilon \times R$.

Table A2: Scenario 2: Comparison of computation times (in seconds).

Method	$m = 800$	$m = 1600$	$m = 2400$	$m = 3600$	$m = 6000$
		Scenario 2(a): $\rho(u) = \exp\{\beta_0(u)\}$			
SVCI-LRL	0.64	0.73	1.11	1.40	1.92
KDE.lpp	4.76	3.74	3.15	2.65	2.47
KDEQuick.lpp	0.055	0.062	0.081	0.094	0.102
Voronoi.lpp	3.24	3.73	4.35	4.52	4.83

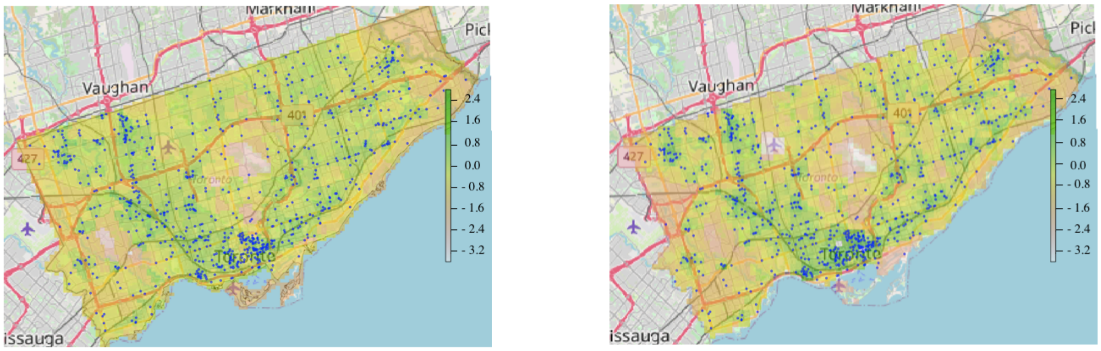

Figure A3: Illustrate of the estimated log intensity function of the Toronto data with the observed locations overlaid in blue dots; Left: the estimated log intensity surface by SVCI; Right: the posterior mean estimate of the log intensity surface by LGCP;

other in most areas. The most notable difference between the two methods occurs near Toronto islands, where we observe more variations in intensity estimations by LGCP than those by SVCI possibly due to the piece-wise homogeneity assumptions made in the latter.

References

Arnold, T.B., Tibshirani, R.J., 2016. Efficient implementations of the generalized Lasso dual path algorithm. Journal of Computational and Graphical Statistics 25, 1–27.

Baddeley, A., Chang, Y.M., Song, Y., Turner, R., 2012. Nonparametric estimation of the dependence of a spatial point process on spatial covariates. Statistics and its interface 5, 221–236.

Baddeley, A., Coeurjolly, J.F., Rubak, E., Waagepetersen, R., 2014. Logistic regression for spatial Gibbs point processes. Biometrika 101, 377–392.

Baddeley, A., Nair, G., Rakshit, S., McSwiggan, G., Davies, T.M., 2020. Analysing point patterns on networks—a review. Spatial Statistics , 100435.

Baddeley, A., Turner, R., 2005. Spatstat: an R package for analyzing spatial point patterns. Journal of Statistical Software 12, 1–42.

Barr, C.D., Schoenberg, F.P., 2010. On the Voronoi estimator for the intensity of an inhomogeneous planar Poisson process. Biometrika 97, 977–984.

Bassett, R., Sharpnack, J., 2019. Fused density estimation: theory and methods. Journal of the Royal Statistical Society: Series B (Statistical Methodology) 81, 839–860.

Beck, A., Teboulle, M., 2009. Fast gradient-based algorithms for constrained total variation image denoising and deblurring problems. IEEE Transactions on Image Processing 18, 2419–2434.

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

Boyd, S., Parikh, N., Chu, E., Peleato, B., Eckstein, J., et al., 2011. Distributed optimization and statistical learning via the alternating direction method of multipliers. Foundations and Trends in Machine learning 3, 1–122.

Brown, P.E., 2015. Model-based geostatistics the easy way. Journal of Statistical Software 63, 1–24.

Chen, J., Chen, Z., 2012. Extended BIC for small-n-large-P sparse GLM. Statistica Sinica 22, 555–574.

Choiruddin, A., Coeurjolly, J.F., Letué, F., et al., 2018. Convex and non-convex regularization methods for spatial point processes intensity estimation. Electronic Journal of Statistics 12, 1210–1255.

Choiruddin, A., Coeurjolly, J.F., Waagepetersen, R., 2021. Information criteria for inhomogeneous spatial point processes. To appear in Australian & New Zealand Journal of Statistics .

Davis, K.B., 1977. Mean integrated square error properties of density estimates. The Annals of Statistics 5, 530–535.

Diggle, P., 1985. A kernel method for smoothing point process data. Journal of the Royal Statistical Society: Series C (Applied Statistics) 34, 138–147.

Fan, J., Li, R., 2001. Variable selection via nonconcave penalized likelihood and its oracle properties. Journal of the American Statistical Association 96, 1348–1360.

Fotheringham, A.S., Brunsdon, C., Charlton, M., 2003. Geographically weighted regression: the analysis of spatially varying relationships. John Wiley & Sons.

Furuta, T., Suzuki, A., Okabe, A., 2008. A Voronoi heuristic approach to dividing networks into equal-sized sub-networks. Forma 23, 73–79.

Gelfand, A.E., Diggle, P., Guttorp, P., Fuentes, M., 2010. Handbook of spatial statistics. CRC press.

Gelfand, A.E., Kim, H.J., Sirmans, C., Banerjee, S., 2003. Spatial modeling with spatially varying coefficient processes. Journal of the American Statistical Association 98, 387–396.

Golub, G.H., Heath, M., Wahba, G., 1979. Generalized cross-validation as a method for choosing a good ridge parameter. Technometrics 21, 215–223.

Gonçalves, F.B., Gamerman, D., 2018. Exact Bayesian inference in spatiotemporal Cox processes driven by multivariate Gaussian processes. Journal of the Royal Statistical Society: Series B (Statistical Methodology) 80, 157–175.

Guan, Y., 2006. A composite likelihood approach in fitting spatial point process models. Journal of the American Statistical Association 101, 1502–1512.

Guan, Y., Jalilian, A., Waagepetersen, R., 2015. Quasi-likelihood for spatial point processes. Journal of the Royal Statistical Society. Series B (Statistical methodology) 77, 677.

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

Jones, M.C., Marron, J.S., Sheather, S.J., 1996. A brief survey of bandwidth selection for density estimation. Journal of the American statistical association 91, 401–407.

Lee, D.T., 1980. Two-dimensional Voronoi diagrams in the L_p-metric. Journal of the ACM 27, 604–618.

Leininger, T.J., Gelfand, A.E., et al., 2017. Bayesian inference and model assessment for spatial point patterns using posterior predictive samples. Bayesian Analysis 12, 1–30.

Li, F., Sang, H., 2019. Spatial homogeneity pursuit of regression coefficients for large datasets. Journal of the American Statistical Association 114, 1050–1062.

Lindgren, F., Rue, H., Lindström, J., 2011. An explicit link between Gaussian fields and Gaussian Markov random fields: the stochastic partial differential equation approach. Journal of the Royal Statistical Society: Series B (Statistical Methodology) 73, 423–498.

McSwiggan, G., 2019. Spatial point process methods for linear networks with applications to road accident analysis. (PhD thesis) University of Western Australia .

McSwiggan, G., Baddeley, A., Nair, G., 2017. Kernel density estimation on a linear network. Scandinavian Journal of Statistics 44, 324–345.

Møller, J., Syversveen, A.R., Waagepetersen, R.P., 1998. Log Gaussian Cox processes. Scandinavian Journal of Statistics 25, 451–482.

Moller, J., Waagepetersen, R.P., 2003. Statistical inference and simulation for spatial point processes. CRC Press.

Møller, J., Waagepetersen, R.P., 2007. Modern statistics for spatial point processes. Scandinavian Journal of Statistics 34, 643–684.

Moradi, M.M., Cronie, O., Rubak, E., Lachieze-Rey, R., Mateu, J., Baddeley, A., 2019. Resample-smoothing of Voronoi intensity estimators. Statistics and Computing 29, 995–1010.

Moradi, M.M., Rodríguez-Cortés, F.J., Mateu, J., 2018. On kernel-based intensity estimation of spatial point patterns on linear networks. Journal of Computational and Graphical Statistics 27, 302–311.

Mu, J., Wang, G., Wang, L., 2018. Estimation and inference in spatially varying coefficient models. Environmetrics 29, e2485.

Okabe, A., Sugihara, K., 2012. Spatial analysis along networks: statistical and computational methods. John Wiley & Sons.

Padilla, O.H.M., Sharpnack, J., Scott, J.G., Tibshirani, R.J., 2018. The DFS fused Lasso: linear-time denoising over general graphs. Journal of Machine Learning Research 18, 176–1.

Pinto Junior, J.A., Gamerman, D., Paez, M.S., Fonseca Alves, R.H., 2015. Point pattern analysis with spatially varying covariate effects, applied to the study of cerebrovascular deaths. Statistics in Medicine 34, 1214–1226.

Rakshit, S., Davies, T., Moradi, M.M., McSwiggan, G., Nair, G., Mateu, J., Baddeley, A., 2019. Fast kernel smoothing of point patterns on a large network using two-dimensional convolution. International Statistical Review 87, 531–556.

Shaw, B., Jebara, T., 2009. Structure preserving embedding. International Conference on Machine Learning (ICML) .

Shen, L., Young, D., Lo, D., Sun, C., 2009. Local differential quadrature method for 2-D flow and forced-convection problems in irregular domains. Numerical Heat Transfer, Part B: Fundamentals 55, 116–134.

Shiode, S., 2008. Analysis of a distribution of point events using the network-based quadrat method. Geographical Analysis 40, 380–400.

Shirota, S., Banerjee, S., 2019. Scalable inference for space-time Gaussian Cox processes. Journal of Time Series Analysis 40, 269–287.

Tibshirani, R.J., Taylor, J., et al., 2011. The solution path of the generalized Lasso. The Annals of Statistics 39, 1335–1371.

Waagepetersen, R.P., 2007. An estimating function approach to inference for inhomogeneous Neyman–Scott processes. Biometrics 63, 252–258.

Wahlberg, B., Boyd, S., Annergren, M., Wang, Y., 2012. An ADMM algorithm for a class of total variation regularized estimation problems. IFAC Proceedings Volumes 45, 83–88.

Zhang, C.H., 2010. Nearly unbiased variable selection under minimax concave penalty. Annals of Statistics 38, 894–942.

Zou, H., 2006. The adaptive Lasso and its oracle properties. Journal of the American statistical association 101, 1418–1429.