

# Spatial and spatio-temporal point patterns on linear networks

Dissertation submitted by Mohammad Mehdi Moradi to apply for the joint doctorate degree from the Universitat Jaume I, Universidade Nova de Lisboa and Universität Münster.



## European Joint Doctorate Marie Skłodowska-Curie in Geoinformatics

Doctoral School from Universitat Jaume I



**Mohammad Mehdi Moradi**

**Supervisors:**

Jorge Mateu                      Ana Cristina Costa                      Edzer Pebesma  
(Universitat Jaume I)    (Universidade Nova de Lisboa)    (Universität Münster)

**Castellón de la Plana (Spain), November 2018**





This dissertation is funded by the European Commission within the Marie Skłodowska-Curie Actions (ITN-EJD). Grant Agreement num. 642332 - GEO-C - H2020-MSCA-ITN-2014.



---

*To my parents, my brother and my sister*



## Acknowledgments

---

First of all I would like to express my sincere gratitude to my supervisor Jorge Mateu for his continuous support, patience, motivation, and immense knowledge. Besides, I would like to thank my co-supervisor Edzer Pebesma who provided me with the opportunity of visiting his research group during my external semester and also for his insightful comments, encouragement and being always in my disposal. I also thank my co-supervisor Ana Cristina Costa for her thoughtful comments and encouragement.

My sincere thanks also goes to Adrian Baddely, Ottmar Cronie and Ege Rubak who provided me the opportunity to visit their research group and collaborate with them in several projects. I am also grateful to Gopalan Nair, Tilman M. Davies, Suman Rakshit and Raphael Lachieze-Rey that I had the pleasure to work with them during my phd.

I take this opportunity to thank the members of GEOTEC for their help and encouragement, especially Carlos Granel and Sergi Trilles. Special thanks go to my friends Fernando Benitez, Carolina Jimenez, Diego Pajarito, Manuel Portela and Khoi Manh for their support, encouragement and helps.

I would also thank Janine Illian and Rasmus Waagepetersen for their helpful comments and suggestions.

Last but not least, I would like to give further thanks to my family for their continued support and encouragement throughout my studies and life.





## Resumen

---

Los datos con componente espacial suelen formarse con diferentes tipos de geometrías, como pueden ser lineales, poligonales o puntuales, entre otras. Estas últimas, las puntuales, están formadas por pares de coordenadas, las cuales se consideran patrones y los puntos son llamados eventos. Estos pueden estar distribuidos de forma regular o irregular dentro de un mismo espacio. Esta distribución puede darse en contextos muy variados, como puede ser en las ciencias geoespaciales, ecología, astronomía, econometría, criminología, entre otros. Por ejemplo, los puntos pueden representar desde árboles dentro de un bosque, accidentes de tráfico o crímenes en una ciudad. En Møller and Waagepetersen (2003); Baddeley and Turner (2005); Illian et al. (2008); Gelfand et al. (2010); Okabe and Sugihara (2012); Diggle (2003); Baddeley et al. (2015) se pueden encontrar diferentes ejemplos de este tipo de datos. Muchos de los patrones observados también pueden incluir una componente temporal, y son llamados patrones espacio-temporales. En ocasiones, la ubicación de los objetos se registra de forma regular o irregular a través del tiempo. En este caso, cada objeto tiene un conjunto de puntos consecutivos que definen una trayectoria. Con lo que se obtiene, no solo unos patrones de puntos, sino un patrón de las trayectorias, como puede ser el movimiento de taxis, animales, humanos, etc. La actual tesis, se centra principalmente en los patrones puntuales espaciales y espacio-temporales que suceden en una red lineal, como pueden ser los accidentes en las calles de una ciudad. Finalmente, también se incluyen el análisis de las trayectorias. La tesis se estructura de la siguiente manera:

En primer lugar, se presenta un capítulo introductorio (Capítulo 1) en el que se muestra un breve resumen de la teoría de los procesos puntuales espaciales en  $\mathbb{R}^2$ , redes lineales y trayectorias. De ellos, se presentan conceptos básicos a nivel

abstracto e introduciendo de ellos algunas características importantes. También, se definen teóricamente algunos modelos clásicos de procesos puntuales. Se presentan los estimadores de intensidad, summary-statistics y relative risk.

El Capítulo 2 tiene como objetivo revisar y proponer algunos estimadores de intensidad basados en kernel para patrones de puntos espaciales en redes lineales, junto con el estudio de sus propiedades. La estimación de la intensidad de los patrones de puntos en redes lineales, como es el caso de una red de carreteras, parece ser una tarea complicada (Rakshit et al., 2018). Varias técnicas publicadas en la literatura en los contextos de geografía e informática, han resultado ser erróneas carreteras, parece ser una tarea complicada (Rakshit et al., 2018). Las técnicas existentes también son computacionalmente costosas, especialmente cuando se analizan gran cantidad de datos, es decir, conjuntos de datos con una alta concentración de puntos y/o una gran red (Rakshit et al., 2018). Existen trabajos en la literatura que proponen estimadores de intensidad Borruso (2003, 2005, 2008); Xie and Yan (2008); Okabe et al. (2009); McSwiggan et al. (2017). Además, en este mismo capítulo se revisan algunos de los estimadores de intensidad actuales y se proponen dos nuevos estimadores, con el objetivo de mejorar los actuales, tanto estadísticamente como computacionalmente. En primer lugar, se presenta un estimador de intensidad basado en kernel, el cual aplica una corrección de borde y revisa sus propiedades estadísticas (Moradi et al., 2017) tales como unbiasedness, mass conservation y variance. Mediante una simulación, se prueba su rendimiento estadístico y se compara con el de Okabe et al. (2009); Okabe and Sugihara (2012). En segundo lugar, se propone un método computacionalmente eficiente y con principios estadísticos de 2D convolution (Rakshit et al., 2018). Para ello se utiliza la transformada rápida de Fourier, la cual trabaja de forma eficiente incluso en redes grandes y para grandes bandwidth, y es robusta frente a posibles errores en la geometría de la red. También se discute el sesgo, variance, asymptotics, selección del bandwidth, estimación de variance, estimación de relative risk y adaptive smoothing. Además, se realiza un análisis de su rendimiento y se compara con el de McSwiggan et al. (2017), tanto estadísticamente como computacionalmente. A lo largo del mismo capítulo, y utilizando los métodos propuestos, también se analizan varios conjuntos de datos reales como datos del crimen de Castellón, (España) y Chicago (EEUU) y los accidentes de tráfico de Medellín (Colombia) y Western Australia (Australia).

El Capítulo 3 propone una nueva técnica para proporcionar un estimador de intensidad adaptativo para procesos de puntos espaciales independientemente del state space (Moradi et al., 2018a). La técnica es introducida y aplicada a los estimadores de intensidad de Voronoi. Además, se revisan sus propiedades estadísticas y se estudia su comportamiento a través de diferentes simulaciones. Los estimadores de intensidad de Voronoi son adaptativos y no necesitan definir previamente parámetros; en ellos la estimación de intensidad está condicionada por el tamaño recíproco de la celda de Voronoi que contiene esa ubicación. Su principal inconveniente, se debe a que tiende a suavizar los datos en regiones donde la densidad puntual del patrón de punto observado es alta, siendo demasiado suave en regiones donde la densidad de puntos es baja. Para remediar este problema, es decir, para encontrar un término medio entre el suavizado excesivo y el subestimado, se propone una técnica de suavizado adicional para estimadores de intensidad Voronoi para procesos puntuales en espacios métricos arbitrarios, que se basa en aplicar repetidamente raleos independientes del proceso puntual (Moradi et al., 2018a). Se muestra que la técnica de remuestreo suavizado mejora la estimación de manera significativa. Además, se estudia las propiedades estadísticas, como la imparcialidad y la varianza, y se propone una regla del pulgar y un enfoque de validación cruzada basada en datos para elegir la cantidad de dilución/suavizado que se aplicará. También se aplica a dos conjuntos de datos reales, en los accidentes de tráfico en un área de Houston (EEUU) (Patrón de punto de red lineal) y Finish pines que consiste en la ubicación de árboles en un bosque finlandés (patrón de puntos de dos dimensiones).

El Capítulo 4 se centra en los procesos puntuales espacio-temporales en redes lineales. En algunos trabajos, solo se considera el dominio espacial y se analizan los patrones de puntos independientemente del tiempo, mientras que ellos están ocurriendo inherentemente de forma conjunta en el espacio y el tiempo. Sin embargo, puede haber preguntas que el análisis espacial no puede responder (Diggle, 2003). Para ello, se presentan varias características de patrones de puntos espacio-temporales cuando las ubicaciones espaciales están restringidas a una red lineal (Moradi et al., 2018b). Se propone un estimador de intensidad basado en kernel para resaltar la concentración alta/baja de eventos dentro de la red y el tiempo, ya sea de forma conjunta o por separado. También se utilizan características de segundo orden de patrones de puntos espacio-temporales en redes lineales, como la función  $K$ -function y la función de correlación de pares

para analizar el tipo de interacción entre los puntos (Moradi et al., 2018b). Los cuales son independientes de la geometría de la red y tienen valores conocidos para los procesos de puntos de Poisson. Por lo que, se pueden usar para medir la desviación de Poisson y también para la selección del modelo. Finalmente, también se han utilizado diferentes conjuntos de datos reales, como por ejemplo los accidentes de tráfico en una área de Houston (EEUU), en una área de Medellín (Colombia) y una área de Eastbourne (Reino Unido).

El Capítulo 5 presenta se centra en las trayectorias. En el cual se proponen varias clases, métodos y metodologías estadísticas, provenientes del paquete **trajectories** de R para analizar conjuntos de datos en movimiento (Moradi et al., 2018c). Hay que tener en consideración que el paquete **trajectories** amplía aún más las capacidades del paquete **spacetime** (Pebesma, 2012). El R paquete **trajectories** proporciona códigos/funciones para manejar, simular y analizar estadísticamente datos de movimiento independientemente del dominio y convertir un patrón de trayectoria en una lista de patrones de puntos basados en timestamps regulares. Para comprender el comportamiento de los objetos en movimiento, se propone un método de estimación de la intensidad (para resaltar las calles y caminos más visitadas), se realiza un análisis de distancia, también se presenta un suavizado de movimientos y se calcula el área de variabilidad de las características de segundo orden (para ver los cambios del tipo de interacción entre objetos a través del tiempo). Con el fin de comparar la intensidad estimada (observada) con la intensidad esperada, se introducen los chimaps, que resaltan las áreas con intensidad estimada más alta/más baja que la esperada. También se utilizan datos reales, en este caso de taxis en movimiento de Beijing (China).

**Palabras clave:** *Características de segundo orden, Espacio-temporal, Estimador adaptativo, Intensidad, Kernel, Procesos puntuales, Red lineal, Remuestreo-suavizado, Trayectoria, Voronoi.*

## Abstract

---

The last decade witnessed an extraordinary increase in interest in the analysis of network related data and trajectories. This pervasive interest is partly caused by a strongly expanded availability of such datasets. In the spatial statistics field, there are numerous real examples such as the locations of traffic accidents and geo-coded locations of crimes in the streets of cities that need to restrict the support of the underlying process over such linear networks to set and define a more realistic scenario. Examples of trajectories are the path taken by moving objects such as taxis, human beings, animals, etc.

Intensity estimation on a network of lines, such as a road network, seems to be a surprisingly complicated task. Several techniques published in the literature, in geography and computer science, have turned out to be erroneous. We propose several adaptive and non-adaptive intensity estimators, based on kernel smoothing and Voronoi tessellation. Theoretical properties such as bias, variance, asymptotics, bandwidth selection, variance estimation, relative risk estimation, and adaptive smoothing are discussed. Moreover, their statistical performance is studied through simulation studies and is compared with existing methods.

Adding the temporal component, we also consider spatio-temporal point patterns with spatial locations restricted to a linear network. We present a non-parametric kernel-based intensity estimator and develop second-order characteristics of spatio-temporal point processes on linear networks such as  $K$ -function and pair correlation function to analyse the type of interaction between points.

In terms of trajectories, we introduce the R package **trajectories** that contains different classes and methods to handle, summarise and analyse trajectory data. Simulation and model fitting, intensity estimation, distance analysis, movement smoothing, Chi maps and second-order summary statistics are discussed.

Moreover, we analyse different real datasets such as a crime data from Chicago (US), anti-social behaviour in Castellón (Spain), traffic accidents in Medellín (Colombia), traffic accidents in Western Australia, motor vehicle traffic accidents in an area of Houston (US), locations of pine saplings in a Finnish forest, traffic accidents in Eastbourne (UK) and one week taxi movements in Beijing (China).

**Keywords:** *Adaptive estimator, Intensity estimator, Kernel, Linear network, Point process, Resample-smoothing, Second-order summary statistics, Space-time, Trajectory, Voronoi.*

# Contents

---

<b>Resumen</b>	<b>v</b>
<b>Abstract</b>	<b>ix</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Data examples . . . . .	1
1.2 Spatial point processes on $\mathbb{R}^2$ . . . . .	3
1.2.1 Point process models . . . . .	5
1.2.2 Intensity estimators . . . . .	6
1.2.3 Relative risk . . . . .	10
1.2.4 Second-order summary statistics . . . . .	11
1.3 Spatial point processes on linear networks . . . . .	13
1.3.1 Linear networks . . . . .	14
1.3.2 Second-order summary statistics . . . . .	16
1.4 Trajectories . . . . .	19
1.5 Organization of the thesis . . . . .	21
<b>2 Kernel smoothing for network events</b>	<b>25</b>
2.1 Introduction . . . . .	25
2.2 Datasets . . . . .	28
2.2.1 Chicago crime data . . . . .	28
2.2.2 Castellón anti-social behaviour . . . . .	29
2.2.3 Traffic accident in Medellín . . . . .	30

---

2.2.4	Traffic accident in Western Australia . . . . .	30
2.3	Equal-split intensity estimators . . . . .	31
2.4	Adapted Jones-Diggle estimator . . . . .	33
2.4.1	Statistical properties . . . . .	35
2.4.2	Simulation study . . . . .	36
2.4.3	Chicago crime data . . . . .	40
2.4.4	Castellón anti-social behaviour . . . . .	42
2.5	Heat kernel intensity estimator . . . . .	42
2.6	Fast kernel smoothing using 2D convolution . . . . .	44
2.6.1	Fast computation . . . . .	46
2.6.2	Chicago example . . . . .	48
2.6.3	Theoretical properties . . . . .	50
2.6.4	Toy example . . . . .	52
2.6.5	Simulation experiments . . . . .	56
2.6.6	Relative risk and smoothing on a network . . . . .	60
2.6.7	Traffic accidents on urban roads of Medellín . . . . .	63
2.6.8	Adaptive smoothing . . . . .	65
2.6.9	Traffic accidents in Western Australia . . . . .	68
2.7	Summary . . . . .	80
<b>3</b>	<b>Resample-smoothing of Voronoi estimators</b>	<b>83</b>
3.1	Introduction . . . . .	83
3.2	Setup . . . . .	85
3.2.1	Independent thinning . . . . .	85
3.2.2	Voronoi tessellations . . . . .	86
3.2.3	Voronoi intensity estimation . . . . .	86
3.3	Resample-smoothing of intensity estimators . . . . .	87
3.3.1	Definition of Resample-Smoothing . . . . .	88
3.3.2	Theoretical properties . . . . .	89
3.3.3	Choosing the smoothing parameters . . . . .	95
3.3.4	Large scale data and sparsity . . . . .	96
3.4	Numerical experiments . . . . .	96



---

3.4.1	Homogeneous Poisson process . . . . .	97
3.4.2	Inhomogeneous Poisson process . . . . .	99
3.4.3	Log-Gaussian Cox process . . . . .	105
3.4.4	Thinned simple sequential inhibition point process . . . . .	106
3.5	Houston traffic accident . . . . .	113
3.6	Finnish pines . . . . .	114
3.7	Summary . . . . .	115
<b>4</b>	<b>Spatio-temporal point patterns on networks</b>	<b>117</b>
4.1	Introduction . . . . .	117
4.2	Setup . . . . .	118
4.3	Methodologies . . . . .	121
4.3.1	First-order characteristics . . . . .	121
4.3.2	Homogeneous second-order characteristics . . . . .	122
4.3.3	Inhomogeneous second-order characteristics . . . . .	126
4.4	Data analysis . . . . .	127
4.4.1	Traffic accidents in Houston . . . . .	128
4.4.2	Traffic accidents in Medellín . . . . .	129
4.4.3	Traffic accidents in Eastbourne . . . . .	131
4.5	Summary . . . . .	133
<b>5</b>	<b>Trajectory analysis</b>	<b>137</b>
5.1	Introduction . . . . .	137
5.2	Classes and methods . . . . .	138
5.2.1	Track . . . . .	138
5.2.2	Tracks . . . . .	141
5.2.3	TracksCollection . . . . .	141
5.2.4	segments . . . . .	142
5.2.5	Methods . . . . .	143
5.3	Simulation and model fitting . . . . .	145
5.3.1	Trajectory simulation . . . . .	145
5.3.2	Model fitting . . . . .	147

5.4	Exploratory data analysis . . . . .	149
5.4.1	Data . . . . .	149
5.4.2	Distance analysis . . . . .	150
5.4.3	Movement smoothing . . . . .	153
5.4.4	Intensity function . . . . .	156
5.4.5	Chi maps . . . . .	158
5.4.6	Second-order summary statistics . . . . .	160
5.5	Summary and discussion . . . . .	162
<b>6</b>	<b>Conclusions and Future work</b>	<b>165</b>
<b>A</b>	<b>Publications and research visits</b>	<b>169</b>
	<b>Bibliography</b>	<b>173</b>

## List of tables

---

2.1	Computation time (in milliseconds) for the diffusion algorithm and the convolution algorithm applied to the Chicago data with different bandwidths $\sigma$ or $\tau$ (in feet). . . . .	49
2.2	Computation time (in <i>minutes</i> ) of the diffusion algorithm for different step-sizes (in km), the adaptive algorithm for different number of bins $m$ , and the convolution algorithm (U = uniform correction; J = Jones-Diggle correction), applied to the Western Australian accident data with different bandwidths $\tau$ or $\sigma$ (in km). . . . .	75
3.1	Estimates of IAB, ISB and IV for $\hat{\lambda}_{p,m}^V(u)$ , $u \in W = [0, 1]^2$ , $m = 200, 300, 400$ , $p = 0.1, \dots, 1$ , based on 500 realisations of a homogeneous Poisson process in $W = [0, 1]^2$ with intensity $\lambda = 60$ . . . . .	98
3.2	Cross-validation selections of $p$ for $m = 200$ in a geometric sequence, based on 100 realisations of a homogeneous Poisson process in $W = [0, 1]^2$ with intensity $\lambda = 60$ . . . . .	99
3.3	Estimates of IAB, ISB and IV for $\hat{\lambda}_{p,m}^V(u)$ , $u \in W = [0, 1]^2$ , $m = 200, 300, 400$ , $p = 0.1, \dots, 1$ , based on 500 realisations of an inhomogeneous Poisson process on $W = [0, 1]^2$ with intensity $\lambda(x, y) =  10 + 90 \sin(16x) $ . . . . .	99
3.4	Cross-validation selections of $p$ in a geometric sequence for $m = 200$ , based on 100 realisations of an inhomogeneous Poisson process in $W = [0, 1]^2$ with intensity $\lambda(x, y) =  10 + 90 \sin(16x) $ . . . . .	102

3.5	Estimates of IAB, ISB and IV for $\widehat{\rho}_{p,m}^V(u)$ , $u \in W = [0, 1]^2$ , $m = 200, 300, 400$ , $p = 0.1, \dots, 1$ , based on 500 realisations of a log-Gaussian Cox process in $W = [0, 1]^2$ with mean function $(x, y) \mapsto \log(40 \sin(20x) )$ and $((x_1, y_1), (x_2, y_2)) \mapsto 2 \exp\{-\ (x_1, y_1) - (x_2, y_2)\ /0.1\}$ as covariance function for the driving random field. . . . .	106
3.6	Cross-validation selections of $p$ in a geometric sequence for $m = 200$ , based on 100 realisations of a log-Gaussian Cox process in $W = [0, 1]^2$ with mean function $(x, y) \mapsto \log(40 \sin(20x) )$ and covariance function $((x_1, y_1), (x_2, y_2)) \mapsto 2 \exp\{-\ (x_1, y_1) - (x_2, y_2)\ /0.1\}$ for the driving random field. . . . .	106
3.7	Estimates of IAB, ISB and IV for $\widehat{\rho}_{p,m}^V(u)$ , $u \in W = [0, 1]^2$ , $m = 200, 300, 400$ , $p = 0.1, \dots, 1$ , based on 500 realisations of an independently thinned simple sequential inhibition process in $W = [0, 1]^2$ with intensity $\rho(x, y) = 450p(x, y)$ , $p(x, y) = \mathbf{1}\{x < 1/3\} x - 0.02  + \mathbf{1}\{1/3 \leq x < 2/3\} x - 0.5  + \mathbf{1}\{x \geq 2/3\} x - 0.95 $ , $x, y \in W$ . . . . .	110
3.8	Cross-validation selections of $p$ in a geometric sequence for $m = 200$ , based on 100 realisations of an independently thinned simple sequential inhibition process in $W = [0, 1]^2$ with intensity $\rho(x, y) = 450p(x, y)$ , $p(x, y) = \mathbf{1}\{x < 1/3\} x - 0.02  + \mathbf{1}\{1/3 \leq x < 2/3\} x - 0.5  + \mathbf{1}\{x \geq 2/3\} x - 0.95 $ , $x, y \in W$ . . . . .	113
3.9	Cross-validation selected values for $p$ , based on the sequence $m = 100, 110, \dots, 200$ . . . . .	114
3.10	Cross-validation selected values for $p$ , based on the sequence $m = 100, 110, \dots, 200$ . . . . .	115
5.1	Methods implemented in the package <b>trajectories</b> for objects from class ‘Track’, ‘Tracks’ and ‘TrackCollection’. . . . .	144

## List of figures

---

1.1	Locations of trees in a forest in New Zealand. . . . .	1
1.2	Locations of street crimes close to the University of Chicago, US. .	2
1.3	Atlantic tropical storm trajectories in the period 2009-2012. . . . .	3
2.1	Point patterns. <i>Left</i> : Non-randomly distributed points. <i>Right</i> : Randomly distributed points on a linear network. . . . .	26
2.2	Locations of street crimes close to the University of Chicago, US. .	28
2.3	Castellón anti-social behaviour during January 2013. . . . .	29
2.4	Traffic accidents in Medellín, Colombia during the year 2016 which caused <i>Left</i> : personal injury, <i>Middle</i> : fatal, <i>Right</i> : property damage.	30
2.5	Traffic accidents in Western Australia during the year 2011. . . . .	31
2.6	Realisations of inhomogeneous Poisson processes. <i>Left</i> : with intensity function (2.10) on network $L_1$ , <i>Right</i> : with intensity function (2.12) on network $L_2$ . . . . .	37
2.7	Bandwidth selection, ISE versus a sequence of bandwidth smoothing parameter $\sigma$ . . . . .	38
2.8	Intensity visualisation for the pattern on the left hand side of Figure 2.6. <i>Left</i> : true intensity, <i>Middle</i> : adapted Jones-Diggle corrected estimator (2.7) with ISE = 9.57, <i>Right</i> : equal-split discontinuous (2.2) with ISE = 9.66. . . . .	38
2.9	Bandwidth selection, ISE versus a sequence of bandwidth smoothing parameter $\sigma$ . . . . .	40

2.10 Intensity visualisation for the pattern on the right hand side of Figure 2.6. <i>Left</i> : true intensity, <i>Middle</i> : adapted Jones-Diggle corrected estimator (2.7) with ISE = 8.710, <i>Right</i> : equal-split discontinuous (2.2) with ISE = 15.384. . . . .	41
2.11 Bandwidth selection for Chicago crime data. $ \hat{L}  = \sum_{i=1}^n 1/\hat{\lambda}_L^{JD}(x_i)$ against a sequence of bandwidth smoothing parameter $\sigma$ (based on feet). Horizontal dashed line shows the total length of the network.	41
2.12 Estimated intensity using adapted Jones-Diggle corrected estimator for Chicago street crime data with a bandwidth parameter of $\sigma = 650$ feet. . . . .	42
2.13 Bandwidth selection for Castellón anti-social behaviour data. Horizontal dashed line shows the total length of the network. . . . .	43
2.14 Estimated intensity using adapted Jones-Diggle corrected estimator for Castellón anti-social behaviour data with a bandwidth parameter of $\sigma = 0.9$ km. . . . .	43
2.15 Kernel estimates of intensity for Chicago data. Perspective views with height representing the function value. <i>Left</i> : diffusion estimate with bandwidth 125 feet. <i>Right</i> : convolution method with bandwidth 100 feet and uniform correction. . . . .	49
2.16 Toy example. Simulated point pattern of 4 points on a network of total length 3 units. . . . .	53
2.17 Kernel estimates of the intensity for the toy example. <i>Left</i> : diffusion estimate with bandwidth $\tau = 0.225$ units. <i>Middle</i> : convolution estimate with uniform correction, bandwidth $\sigma = 0.15$ units. <i>Right</i> : convolution estimate with Jones-Diggle correction, bandwidth $\sigma = 0.15$ units. Perspective views with height representing the function value. Vertical scales are equal. . . . .	53
2.18 Edge correction denominator $c_L(u)$ for the toy network of Figure 2.16 with bandwidth $\sigma$ equal to 0.015, 0.15 and 1.5 units (left to right). Perspective views with height representing the function value, using different vertical scales. . . . .	54

- 2.19 Predicted performance on the toy example. Assuming a uniform Poisson process with intensity 2, and kernel smoothing with bandwidth 0.15. *Top Left*: variance (= MSE) of the uniform correction estimator. *Top Right*: variance of the Jones-Diggle correction estimator. *Bottom Left*: bias of the Jones-Diggle correction estimator, with positive values shown by solid grey colour and negative values by diagonal shading. *Bottom Right*: MSE of the Jones-Diggle correction estimator. Variance and MSE panels use the same vertical scale. . . . . 55
- 2.20 Typical simulated realisations for each of the eight scenarios. *Top row*: Chicago street network. *Bottom row*: southern part of the city of Perth, extracted from the Western Australian road network. The Gaussian mixture and LGCP realisation scenarios are based on an initial 2D surface defined on  $W$ . The diffusion estimate scenario is based on the original data observed on the relevant network. Simulated realisations all have size  $n = 500$ . . . . . 56
- 2.21 Integrated squared error (ISE) of the convolution and diffusion estimators applied to simulated data. *Left column*: Chicago network; *Right column*: southern Perth network. *Rows* represent the four scenarios in Figure 2.20. The bottom horizontal axis in each panel shows the bandwidth  $\sigma$  of the convolution estimator; the top horizontal axis shows the bandwidth  $\tau$  of the diffusion estimator. *Box-plots* show the numerically computed ISEs for the diffusion estimator. *Lines* show the theoretically calculated ISEs for  $\hat{\lambda}_{L,con}^U$  (red, solid) and  $\hat{\lambda}_{L,con}^{JD}$  (green, dashed). Bandwidths are in feet for Chicago, and in km for southern Perth. . . . . 59
- 2.22 Mean relative execution time per estimate for  $\hat{\lambda}_L^H$  relative to both  $\hat{\lambda}_{L,con}^U$  and  $\hat{\lambda}_{L,con}^{JD}$  across all four scenarios for the Chicago and southern Perth networks (*left* and *right* respectively). . . . . 61
- 2.23 Estimated intensity functions for each type of accident in Medellín data, using convolution estimator and uniform correction (2.14). *Left*: property damage, bandwidth 0.36 km; *Middle*: personal injury, bandwidth 0.36 km; *Right*: fatal, bandwidth 0.67 km. Intensity values are reported as accidents per km. . . . . 64

2.24	Estimated relative risk of different types of accidents, relative to accidents which caused only property damage. <i>Left</i> : personal injury; <i>Right</i> : fatality. . . . .	65
2.25	Likelihood cross-validation criterion $cv(\sigma)$ plotted against bandwidth $\sigma$ for the kernel estimators of intensity of traffic accidents on the Western Australian road network. <i>Left</i> : uniform-corrected estimator; the vertical line shows the optimum at $\sigma = 9.1$ km. <i>Right</i> : Jones-Diggle estimator; the vertical line shows the optimum at $\sigma = 10.9$ km. . . . .	69
2.26	Fixed-bandwidth estimate of intensity for the accidents on the Western Australian road network using the uniform correction with $\sigma = 9.1$ km. Intensity values are accidents per km. . . . .	70
2.27	Fixed-bandwidth estimate of intensity for the accidents on the Western Australian road network using the Jones-Diggle correction with $\sigma = 10.9$ km. Logarithmic colour map. Intensity values are accidents per km. . . . .	71
2.28	Adaptive-bandwidth intensity estimate for the accidents on the Western Australian road network using Jones-Diggle correction. Logarithmic colour map. Intensity values are accidents per km. . . . .	72
2.29	Accidents recorded in the Perth metropolitan area. . . . .	73
2.30	Adaptive-bandwidth intensity estimate for the accidents in metropolitan Perth using Jones-Diggle correction. Linear colour map with gamma-corrected colour sequence. Intensity values are accidents per km. Map is 60 km wide. . . . .	74
2.31	Accidents recorded in the Central Business District of the city of Perth. . . . .	76
2.32	Intensity estimate in the Perth CBD using fixed-bandwidth uniform correction with automatically selected bandwidth $\sigma = 0.091$ km. Intensity values mapped to colours. . . . .	77
2.33	Intensity estimate in the Perth CBD using fixed-bandwidth uniform correction with automatically selected bandwidth $\sigma = 0.091$ km. Intensity values are proportional to line width. . . . .	78
2.34	Perspective view of fixed-bandwidth uniform correction estimate. . . . .	79



- 2.35 A screenshot of the fixed and adaptive intensity estimates of the Perth CBD data shown as interactive HEN plots. Accessible at the URL noted in the text. . . . . 79
- 3.1 Estimation error plots for a realisation of a homogeneous Poisson process  $X$  in  $W = [0, 1]^2$  with intensity  $\lambda = 60$ . *Left:*  $p = 0.2$  and  $m = 200$ . *Right:*  $p = 1$ . . . . . 98
- 3.2 Estimated bias for  $\widehat{\lambda}_{p,m}^V(u)$  and  $\widehat{\lambda}^U(u)$ ,  $u \in W = [0, 1]^2$ ,  $m = 200$ , based on 500 realisations of a homogeneous Poisson process  $X \subset W = [0, 1]^2$  with intensity  $\lambda = 60$ . From top-left to bottom-right:  $\widehat{\lambda}_{p,m}^V(u)$  with  $p = 0.1, 0.3, 0.5, 0.7, 0.9, 1$ , and  $\widehat{\lambda}^U(u)$  using bandwidth selection (1.12) and (1.13). . . . . 100
- 3.3 Estimated variance for  $\widehat{\lambda}_{p,m}^V(u)$  and  $\widehat{\lambda}^U(u)$ ,  $u \in W = [0, 1]^2$ ,  $m = 200$ , based on 500 realisations of a homogeneous Poisson process  $X \subset W = [0, 1]^2$  with intensity  $\lambda = 60$ . From top-left to bottom-right:  $\widehat{\lambda}_{p,m}^V(u)$  with  $p = 0.1, 0.3, 0.5, 0.7, 0.9, 1$ , and  $\widehat{\lambda}^U(u)$  using bandwidth selection (1.12) and (1.13). . . . . 101
- 3.4 True intensity and estimation error plots for a realisation of an inhomogeneous Poisson process on  $W = [0, 1]^2$  with intensity  $\lambda(x, y) = |10 + 90 \sin(16x)|$ . *Left:*  $p = 0.2$  and  $m = 200$ . *Middle:*  $p = 1$ , *Right:* True intensity. . . . . 102
- 3.5 Estimated bias for  $\widehat{\lambda}_{p,m}^V(u)$  and  $\widehat{\lambda}^U(u)$ ,  $u \in W = [0, 1]^2$ ,  $m = 200$ , based on 500 realisations of an inhomogeneous Poisson process  $X \subseteq W = [0, 1]^2$  with intensity  $\rho(x, y) = |10 + 90 \sin(16x)|$ . From top-left to bottom-right:  $\widehat{\lambda}_{p,m}^V(u)$  with  $p = 0.1, 0.3, 0.5, 0.7, 0.9, 1$ , and  $\widehat{\lambda}^U(u)$  using bandwidth selection (1.12) and (1.13). . . . . 103
- 3.6 Estimated variance for  $\widehat{\lambda}_{p,m}^V(u)$  and  $\widehat{\lambda}^U(u)$ ,  $u \in W = [0, 1]^2$ ,  $m = 200$ , based on 500 realisations of an inhomogeneous Poisson process  $X \subseteq W = [0, 1]^2$  with intensity  $\rho(x, y) = |10 + 90 \sin(16x)|$ . From top-left to bottom-right:  $\widehat{\lambda}_{p,m}^V(u)$  with  $p = 0.1, 0.3, 0.5, 0.7, 0.9, 1$ , and  $\widehat{\lambda}^U(u)$  using bandwidth selection (1.12) and (1.13). . . . . 104

- 3.7 True intensity and estimation error plots for a realisation of a log-Gaussian Cox process in  $W = [0, 1]^2$  with mean function  $(x, y) \mapsto \log(40|\sin(20x)|)$  and  $((x_1, y_1), (x_2, y_2)) \mapsto 2 \exp\{-\|(x_1, y_1) - (x_2, y_2)\|/0.1\}$  as covariance function for the driving random field. *Left:*  $p = 0.2$  and  $m = 200$ . *Middle:*  $p = 1$ . *Right:* True intensity. . . . . 107
- 3.8 Estimated bias for  $\hat{\rho}_{p,m}^V(u)$ ,  $u \in W = [0, 1]^2$ ,  $m = 200$ , based on 500 realisations of a log-Gaussian Cox process  $X \subseteq W = [0, 1]^2$  where the driving Gaussian random field has mean function  $(x, y) \mapsto \log(40|\sin(20x)|)$  and  $((x_1, y_1), (x_2, y_2)) \mapsto 2 \exp\{-\|(x_1, y_1) - (x_2, y_2)\|/0.1\}$  as covariance function. From top-left to bottom-right:  $\hat{\lambda}_{p,m}^V(u)$  with  $p = 0.1, 0.3, 0.5, 0.7, 0.9, 1$ , and  $\hat{\lambda}^U(u)$  using bandwidth selection (1.12) and (1.13). . . . . 108
- 3.9 Estimated variance for  $\hat{\rho}_{p,m}^V(u)$ ,  $u \in W = [0, 1]^2$ ,  $m = 200$ , based on 500 realisations of a log-Gaussian Cox process  $X \subseteq W = [0, 1]^2$  where the driving Gaussian random field has mean function  $(x, y) \mapsto \log(40|\sin(20x)|)$  and  $((x_1, y_1), (x_2, y_2)) \mapsto 2 \exp\{-\|(x_1, y_1) - (x_2, y_2)\|/0.1\}$  as covariance function. From top-left to bottom-right:  $\hat{\lambda}_{p,m}^V(u)$  with  $p = 0.1, 0.3, 0.5, 0.7, 0.9, 1$ , and  $\hat{\lambda}^U(u)$  using bandwidth selection (1.12) and (1.13). . . . . 109
- 3.10 True intensity and estimation error plots for a realisation of an independently thinned simple sequential inhibition process in  $W = [0, 1]^2$  with intensity  $\rho(x, y) = 450p(x, y)$ ,  $p(x, y) = \mathbf{1}\{x < 1/3\}|x - 0.02| + \mathbf{1}\{1/3 \leq x < 2/3\}|x - 0.5| + \mathbf{1}\{x \geq 2/3\}|x - 0.95|$ ,  $x, y \in W$ . *Left:*  $p = 0.2$  and  $m = 200$ . *Middle:*  $p = 1$ . *Right:* True intensity. . . . . 110
- 3.11 Estimated bias for  $\hat{\rho}_{p,m}^V(u)$ ,  $u \in W = [0, 1]^2$ ,  $m = 200$ , based on 500 realisations of an independently thinned simple sequential inhibition process in  $W = [0, 1]^2$  with intensity  $\rho(x, y) = 450p(x, y)$ ,  $p(x, y) = \mathbf{1}\{x < 1/3\}|x - 0.02| + \mathbf{1}\{1/3 \leq x < 2/3\}|x - 0.5| + \mathbf{1}\{x \geq 2/3\}|x - 0.95|$ ,  $x, y \in W$ . From top-left to bottom-right:  $\hat{\lambda}_{p,m}^V(u)$  with  $p = 0.1, 0.3, 0.5, 0.7, 0.9, 1$ , and  $\hat{\lambda}^U(u)$  using bandwidth selection (1.12) and (1.13). . . . . 111

- 3.12 Estimated variance for  $\hat{\rho}_{p,m}^V(u)$ ,  $u \in W = [0, 1]^2$ ,  $m = 200$ , based on 500 realisations of an independently thinned simple sequential inhibition process in  $W = [0, 1]^2$  with intensity  $\rho(x, y) = 450p(x, y)$ ,  $p(x, y) = \mathbf{1}\{x < 1/3\}|x - 0.02| + \mathbf{1}\{1/3 \leq x < 2/3\}|x - 0.5| + \mathbf{1}\{x \geq 2/3\}|x - 0.95|$ ,  $x, y \in W$ . From top-left to bottom-right:  $\hat{\lambda}_{p,m}^V(u)$  with  $p = 0.1, 0.3, 0.5, 0.7, 0.9, 1$ , and  $\hat{\lambda}^U(u)$  using bandwidth selection (1.12) and (1.13). . . . . 112
- 3.13 *Left*: Motor vehicle traffic accidents in an area of Houston, US, during April, 1999. *Right*: Resample-smoothed Voronoi intensity estimate for  $m = 200$  and  $p = 0.15$ . . . . . 114
- 3.14 The estimate  $\hat{\rho}_{p,m}^V(u)$ ,  $u \in W$ ,  $m = 200$ , for  $p = 0.2$  (*left*) and  $p = 0.5$  (*right*), together with the locations of 126 pine saplings in a Finnish forest, within a rectangular window  $W = [-5, 5] \times [-8, 2]$  (metres). 115
- 4.1 The motor vehicle traffic accidents in Houston near the university of Houston in 2001 which caused non-incapacitating injury such as bump on the head, abrasions or minor lacerations. *Left*: The projection of the data onto the network. *Right*: Cumulative number of data points versus time. . . . . 129
- 4.2 Intensity estimates for motor vehicle traffic accidents in Houston. *Left*: Intensity estimate of daily hours together with the frequency of accidents per hour (bar plot). *Right*: Intensity estimate of the projection onto the network. . . . . 130
- 4.3 Estimated second-order characteristics for motor vehicle traffic accidents in Houston. *Left*: Inhomogeneous  $K$ -function. *Right*: Inhomogeneous pair correlation function. Gray surfaces are envelopes based on 99 simulations and significance level 5% from the complete spatio-temporal randomness. . . . . 130
- 4.4 Traffic accidents in Medellín during the year 2016. *Left*: The projection of data onto the network. *Right*: Cumulative number of data points versus occurrence time. . . . . 131
- 4.5 Intensity estimates for traffic accidents in Medellín. *Left*: Intensity estimate of daily hours together with the frequency of accidents per hour (bar plot). *Right*: Intensity estimate of the projection onto the network. . . . . 132

4.6	Estimated second-order characteristics for traffic accidents in Medellín. <i>Left:</i> Inhomogeneous $K$ -function. <i>Right:</i> Inhomogeneous pair correlation function. Gray surfaces are envelopes based on 99 simulations and significance level 5% from complete spatio-temporal randomness. . . . .	133
4.7	Traffic accidents in the down-town of Eastbourne (UK) in . <i>Left:</i> The projection of data onto the network. <i>Right:</i> Cumulative number of data points versus occurrence time. . . . .	134
4.8	Intensity estimates for traffic accidents in the down-town of Eastbourne. <i>Left:</i> Intensity estimate of daily hours together with the frequency of accidents per hour (bar plot). <i>Right:</i> Intensity estimate of the projection onto the network. . . . .	135
4.9	Estimated second-order characteristics for traffic accidents in the down-town of Eastbourne. <i>Left:</i> Inhomogeneous $K$ -function. <i>Right:</i> Inhomogeneous pair correlation function. Gray surfaces are point-wise envelopes based on 99 simulations and significance level 5% from complete spatio-temporal randomness. . . . .	135
5.1	Three consecutive movements of 50 taxis in Beijing, China on Feb 2008 per 20 minutes. . . . .	137
5.2	Single track A1 passed by person A. . . . .	140
5.3	Classes for trajectory data in the package <b>trajectories</b> . Solid arrows denote inheritance. Arrows show the corresponding slot's class and slot's names are displayed using lines accordingly. . . . .	143
5.4	Co2 consumption over time. . . . .	145
5.5	Simulated random tracks using <code>rTrack</code> . <code>x</code> is random track with all defaults. <code>y</code> is a random track transformed to a unit box. <code>w</code> is a random track transformed to the box $[0, 10] \times [0, 10]$ and <code>z</code> is in a same box as <code>w</code> but with random number of points. . . . .	148
5.6	Map of the studied area in Beijing, China. . . . .	150
5.7	Average pairwise distance between taxis in Beijing, China. <i>Left:</i> Within the period 2-8, Feb 2008. <i>Right:</i> During 3-rd of Feb 2008. . .	152

---

5.8	Movement smoothing for taxi data in Beijing, China based on timestamp = "20 mins" and movements with length longer than 1000 meters. . . . .	155
5.9	Average length of movements by taxis in Beijing, China versus time based on timestamp = "20 mins", and movements with length longer than 1000 meters. <i>Left</i> : Within the period 2-8 Feb 2008. <i>Right</i> : During the 3- <i>rd</i> of Feb 2008. . . . .	156
5.10	Estimated intensity function. <i>Left</i> : Beijing. <i>Right</i> : Beijing metropolitan area. . . . .	158
5.11	Chi maps. <i>Left</i> : in the morning, <i>Middle</i> : in the afternoon, <i>Right</i> : at night. Exact time is reported on top of each plot. . . . .	160
5.12	Variability area of second-order summary statistics for taxi data in Beijing, China. <i>Left</i> : <i>K</i> -function, <i>Right</i> : pair correlation function. .	162



## CHAPTER 1

---

# Introduction

---

### 1.1 Data examples

One type of spatial data is when it comes as a set of points which are regularly/irregularly distributed within a region of space. Such data might be seen in different contexts such as geo-science, ecology, astronomy, econometrics, criminology, etc. Examples may include trees in a forest, traffic accidents, street crimes, mobile phone calls, animal sightings, or cases of a rare disease. Figure 1.1 shows the location of 86 trees in a forest in New Zealand in a region of approximately 140 by 85 feet. This data was firstly collected by Mark and Esler (1970) and then extracted and analysed by Ripley (2005).

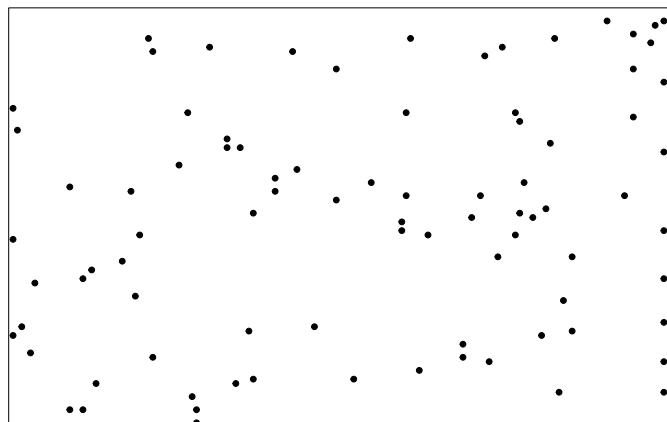


Figure 1.1: Locations of trees in a forest in New Zealand.

Figure 1.2 displays the locations of street crimes reported in the period of 25 April to 8 May 2002, in an area of Chicago, US close to the University of Chicago. The original crime map was published in the Chicago Weekly News in 2002 and it has later been analysed by Ang et al. (2012); Baddeley et al. (2015); Moradi et al. (2017); Rakshit et al. (2018). In real world, there are numerous and various events that are strongly constrained by networks, such as traffic accidents and fast-food shops located alongside streets (Okabe and Sugihara, 2012).



Figure 1.2: Locations of street crimes close to the University of Chicago, US.

Such patterns, as those shown in Figure 1.1 and 1.2, are called “*spatial point patterns*” and the locations are referred as “*events*”. Both displayed datasets are available in the R package **spatstat**. More of such examples can be found in Møller and Waagepetersen (2003); Baddeley and Turner (2005); Illian et al. (2008); Gelfand et al. (2010); Okabe and Sugihara (2012); Diggle (2003); Baddeley et al. (2015).

Sometimes the location of objects are recorded over time and according to regular/irregular timestamps. Thus, in this case, we have a set of consecutive points per object and it can be displayed as a *track* instead of points, and the final pattern is a pattern of tracks which is called “*trajectory pattern*”. Figure 1.3 shows the Atlantic tropical storm trajectories, in the period of 2009-2012. The dataset is stored in the R package **trajectories** (Moradi et al., 2018c).



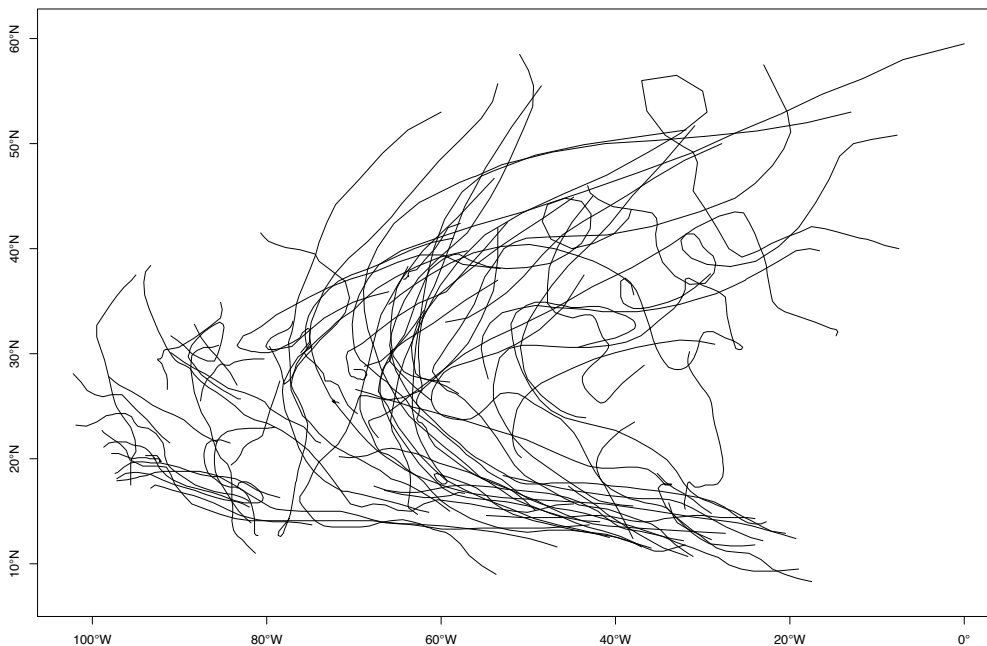


Figure 1.3: Atlantic tropical storm trajectories in the period 2009-2012.

Datasets in such form are often considered as outcomes of some random mechanism (Baddeley et al., 2015) that a researcher might be interested to discover. For instance, in Figure 1.1 and 1.2, one may be interested to know if events are uniformly distributed over the corresponding region or if there is any particular interaction between them. In Figure 1.3 we may wish to know if the behaviour of objects changes over time.

This chapter is devoted to review the preliminaries that are needed for analysing such datasets. Section 1.2 provides some definitions and reviews some characteristics of spatial point processes on  $\mathbb{R}^2$ . A brief introduction to spatial point processes on linear networks is provided in Section 1.3, and Section 1.4 reviews some preliminaries about trajectory patterns.

## 1.2 Spatial point processes on $\mathbb{R}^2$

A *spatial point process* is a random countable subset of state space  $\mathbb{R}^2$ . Throughout this thesis, we only consider finite and simple point processes such that the total number of points has finite second moment, see Daley and Vere-Jones (2003, Chapter 5). The outcome of such process is called a *spatial point pattern* which is

a dataset giving the observed spatial locations of things or events. However, in practice, we only observe some points placed in a bounded observation window  $W \subset \mathbb{R}^2$ . Hereafter, we denote point processes with capital letters such as  $X, Y, \dots$  and the corresponding point patterns as  $\mathbf{x}, \mathbf{y}, \dots$ . For any arbitrary bounded subset  $A \subset \mathbb{R}^2$ , its cardinality,  $N(X \cap A)$ , is the number of points falling in  $A \subset \mathbb{R}^2$ . For any point process  $X$ ,

$$\mathbb{E}[N(X \cap A)] = \int_A \lambda(u) du, \quad u \in \mathbb{R}^2, \quad (1.1)$$

where  $\lambda(\cdot)$  is called *intensity function* which governs the behaviour of the underlying point process and shows how it uses the corresponding space. When studying point patterns, one of the very first steps is to investigate the behaviour of the intensity function. Intuitively, the intensity function  $\lambda(u)$  is the expected number of points per unit area in the vicinity of location  $u \in \mathbb{R}^2$ . If the intensity function  $\lambda(\cdot)$  is constant, then the point process  $X$  is called homogeneous, otherwise it is an inhomogeneous point process on  $\mathbb{R}^2$ . Equation (1.1) might be extended to higher orders, e.g. for every subsets  $A, B \subset \mathbb{R}^2$ ,

$$\mathbb{E}[N(X \cap A)N(X \cap B)] = \int_A \int_B \lambda_2(u, v) du dv, \quad u, v \in \mathbb{R}^2, \quad (1.2)$$

where  $\lambda_2(\cdot, \cdot)$  is the second-order product density function of  $X$ .

A spatial point process  $X$  on state space  $\mathbb{R}^2$  is stationary if its distribution is invariant under translations, i.e. for any arbitrary point  $a \in \mathbb{R}^2$  the distributions of  $X$  and  $X + a$  are the same (Møller and Waagepetersen, 2003). Although such aspect might add much to the convenience, in practice we usually face datasets with lack of stationarity. The point process  $X$  is second-order intensity-reweighted stationary if the second moment measure

$$M(A, B) = \mathbb{E} \sum_{x_i \in X \cap A} \sum_{x_j \in X \cap B} \frac{1}{\lambda(x_i)\lambda(x_j)}, \quad (1.3)$$

is second-order stationary, i.e.  $M(A, B) = M(A + a, B + a)$  for all  $a \in \mathbb{R}^2$  (Baddeley et al., 2000). The point process  $X$  is isotropic if its distribution is invariant under rotations around the origin (Møller and Waagepetersen, 2003).

We denote a realisation of a point process  $X$  with  $n$  points as  $\mathbf{x} = \{x_1, x_2, \dots, x_n\}$  where the number of points  $n$  is not fixed previously. In other words, it varies from

one realisation to another one. For any non-negative and measurable function  $f$  on  $\mathbb{R}^2$ , the intensity function  $\lambda(\cdot)$  satisfies

$$\mathbb{E} \sum_{x \in X} f(x) = \int_{\mathbb{R}^2} f(u) \lambda(u) du. \quad (1.4)$$

Equation (1.4) has been widely used in the literature of point processes and it is called *Campbell's formula* and it can be extended to higher orders so that e.g. for second-order is of the form

$$\mathbb{E} \sum_{x, y \in X}^{\neq} f(x, y) = \int_{\mathbb{R}^2} \int_{\mathbb{R}^2} f(u, v) \lambda_2(u, v) du dv, \quad (1.5)$$

where  $\sum^{\neq}$  means that the sum is over distinct pair of points.

### 1.2.1 Point process models

A very popular and important model for point processes is the *Poisson* point process which satisfies the following conditions:

1.  $N(X \cap A)$  follows a Poisson distribution with parameter  $\int_A \lambda(u) du$ .
2. Given  $N(X \cap W) = n$ , events in the window  $W$  constitute an independent random sample from the uniform distribution over  $W$ .
3. For any finite number of disjoint subsets  $A_i \subset W (i = 2, \dots, m)$ , the random variables  $N(A_i \cap W)$  are independent.

Such conditions lead to have no interaction amongst the events so that they do not tend to either happen closely or distantly. Such model might bring ease, however as listed above, a crucial assumption of Poisson point process models is that points are independent of each other, which does not seem to be an appropriate assumption when analysing real data. Due to simplicity and having known values for summary statistics (see Section 1.2.4), Poisson processes are usually used as a benchmark for model selection. They are also used to build more complex and flexible models such as *Cox* point processes which are clustered due to an environmental random heterogeneity (Møller and Waagepetersen, 2003). The source of such environmental heterogeneity might itself be stochastic in nature (Diggle, 2003).

A Cox point process is a Poisson process when its intensity function is considered as a realisation of a random field, i.e. *doubly stochastic* behaviour is here considered. This being said, the conditional distribution of a Cox process given the underlying random field is a Poisson point process. For details of random fields see Adler (1981). A Cox process is assumed stationary if the underlying random field is stationary. For a broad introduction to Cox processes, see Møller and Waagepetersen (2003, Chapter 5).

In some real applications events might tend to not get close to each other more than a particular distance. A nice example in biology is provided in van Lieshout (2000, Chapter 2) where in a pattern of cells, no nucleus can be closer to other nucleus more than a particular distance. A family of point processes which might cover such behaviour is *Markov* point processes that considers a density for point processes with respect to a Poisson process. For key references to Markov point processes see van Lieshout (2000) and Møller and Waagepetersen (2003).

## 1.2.2 Intensity estimators

One of the very first steps to get into the behaviour of the point pattern  $\mathbf{x}$  is to study the intensity function  $\lambda(\cdot)$  since it might reveal the distribution of the underlying point process  $X$ . If the point process  $X$  has constant intensity function, then

$$\hat{\lambda} = \frac{n(\mathbf{x})}{|W|}, \quad (1.6)$$

is an unbiased estimator for the intensity function  $\lambda(\cdot)$  where  $n(\mathbf{x})$  is the number of points of pattern  $\mathbf{x}$  and  $|W|$  is the volume of window  $W$ . However, in practice, this assumption seems hard to be realistic.

### 1.2.2.1 Kernel smoothing

Kernel smoothing is usually the most popular method to estimate the intensity function. The uncorrected kernel estimator is of the form

$$\hat{\lambda}^0(u) = \sum_{i=1}^n \kappa(u - x_i), \quad u \in W, \quad (1.7)$$

where the kernel  $\kappa$  is a probability density function on  $\mathbb{R}^2$ . This estimator generates bias close to the edges which is caused by not having access to data points

outside the window. In order to improve the efficiency of the estimator (1.7), Diggle (1985) introduced an uniformly corrected version as

$$\widehat{\lambda}^U(u) = \frac{1}{e_W(u)} \sum_{i=1}^n \kappa(u - x_i), \quad u \in W, \quad (1.8)$$

where

$$e_W(u) = \int_W \kappa(u - v) dv, \quad (1.9)$$

is the mass of the kernel centred at  $u$  that falls inside the window. The correction (1.9) is often called the “uniform” edge correction, because it ensures that  $\widehat{\lambda}^U(\cdot)$  is a pointwise unbiased estimator when the true intensity is uniform. That is, if  $\lambda(u) \equiv \lambda > 0$ , then  $\mathbb{E}[\widehat{\lambda}^U(u)] \equiv \lambda$ . Jones (1993) proposed an alternative estimator as

$$\widehat{\lambda}^{JD}(u) = \sum_{i=1}^n \frac{\kappa(u - x_i)}{e_W(x_i)}, \quad u \in W, \quad (1.10)$$

often confusingly called the “Diggle correction” in the spatial statistics literature. We shall call it the “Jones-Diggle” correction. This estimator conserves total mass, that is,

$$\int_W \widehat{\lambda}^{JD}(u) du = n. \quad (1.11)$$

The initial difference between (1.8) and (1.10) is the way they apply the correction that makes other differences in terms of efficiency. The degree of smoothing in the estimators above strongly depends on the smoothing parameter bandwidth. A small bandwidth might result in under-smoothing while a large bandwidth may over smooth the intensity. Moreover, small bandwidth results in small bias and large variance, while large bandwidth makes larger bias and less variation. For more details see Baddeley et al. (2015, Chapter 6).

However, to compute the aforementioned intensity estimators, a smoothing bandwidth parameter needs to be picked previously. Bandwidth selection process has always been a challenge when estimating intensity functions using kernel smoothing methods. Prior researches such as Diggle (1985); Silverman (1986); Berman and Diggle (1989); Scott (1992); Wand and Jones (1994); Jones et al. (1996); Loader (1999) on bandwidth selection method show that we are unlikely to find a data-based method which performs uniformly well. Diggle (1985) aimed at

finding a proper choice of bandwidth by minimising the “*mean square error*” when the underlying point process is assumed to be a stationary Cox process. However, the very common method to find a proper bandwidth might be “*leave-one-out*” cross validation based on the Poisson point process likelihood in which it selects a bandwidth which maximises

$$CV(\sigma) = \sum_{i=1}^n \log(\hat{\lambda}_{\sigma}^{-i}(x_i)) - \int_S \hat{\lambda}_{\sigma}(u) du, \quad (1.12)$$

where  $\sigma$  is bandwidth and  $\hat{\lambda}_{\sigma}^{-i}(x_i)$  is the intensity estimator at data point  $x_i$  when we have excluded  $x_i$  from the point pattern in question. In other words, the contribution of  $x_i$  in the sum in equations (1.8) and (1.10) has been omitted. For estimators which provide mass conservation such as estimator (1.10), we might exclude the integral part in the cross validation since it is equal to the number of data points.

A simple way to at least reach a range of trial values of  $\sigma$  is the Scott’s rule of thumb (Scott, 1992) according to which,  $j$ -th Cartesian coordinate ( $j = 1, 2$ ) should be smoothed with bandwidth  $\sigma_j = c_{n,2}s_j$  where  $s_j$  is the sample standard deviation of the  $j$ -th Cartesian coordinate values for the data points, and  $c_{n,2} = (4n)^{-1/6}$  where  $n$  is the number of data points.

Cronie and van Lieshout (2018) defined a non-model-based approach to choose the bandwidth based on the fact that

$$\mathbb{E} \sum_{x \in X \cap W} \frac{1}{\lambda(x)} = \int_W \frac{1}{\lambda(u)} \lambda(u) du = |W|, \quad u \in W. \quad (1.13)$$

Equation above results from using the Campbell formula (1.4). Cronie and van Lieshout (2018) considered a proper choice of the bandwidth as the one which minimises the discrepancy between the total volume of the corresponding window  $W$  and the left side of equation (1.13) when  $\lambda(x)$  is replaced by its estimator. An extensive comparison with other methods based on “*leave-one-out*” cross validation and “*mean square error*” is also provided by Cronie and van Lieshout (2018).

### 1.2.2.2 Adaptive kernel intensity estimators

Intensity estimators such as (1.7),(1.8) and (1.10) are based on “*fixed-bandwidth smoothers*”, i.e. same kernel and the same bandwidth are considered to calculate

the estimator at different spatial locations in which local effects on the estimation might be neglected. Especially, a fixed smoothing bandwidth is unsatisfactory when the true intensity varies greatly across the spatial domain, because it is likely to cause over-smoothing in the high-intensity areas and under-smoothing in the low intensity areas. Kernel estimators such as (1.7),(1.8) and (1.10) might suffer from sharp boundaries between areas with high and low intensity, because this boundary will be smoothed out (Baddeley et al., 2015). In such situations, adaptive (variable-bandwidth) kernel estimation can perform substantially better (Abramson, 1982; Hall and Marron, 1988) where an alternative is to allow the degree of smoothing to be adapted to local smoothing requirements. Adaptive versions of (1.8) and (1.10) are in the form of

$$\widehat{\lambda}_a^U(u) = \frac{1}{e_W(u, \sigma(u))} \sum_{i=1}^n \kappa_{\sigma(x_i)}(u - x_i), \quad u \in W, \quad (1.14)$$

and

$$\widehat{\lambda}_a^{JD}(u) = \sum_{i=1}^n \frac{\kappa_{\sigma(x_i)}(u - x_i)}{e_W(x_i, \sigma(x_i))}, \quad u \in W, \quad (1.15)$$

where  $\sigma(\cdot)$  is the spatially-varying bandwidth function (Marshall and Hazelton, 2010; Rakshit et al., 2018). Although estimator (1.14) is analogous to the estimator (1.8), it does not retain the same unbiasedness property. However, estimator (1.15) still provides mass conservation. Estimators (1.14) and (1.15) take longer time to be computed.

### 1.2.2.3 Adaptive Voronoi estimators

The Voronoi cell associated with a particular data point  $x_i$  is the area of  $\mathbb{R}^2$  which is closer to  $x_i$  than any other points when the observed point pattern  $\mathbf{x} = \{x_1, x_2, \dots, x_n\}$  is distributed over  $W \subset \mathbb{R}^2$ . We then denote a Dirichlet cell as

$$\mathcal{V}_{x_i} = \mathcal{V}_{x_i}(\mathbf{x}, W) = \{u \in W : \|u - x_i\| \leq \|u - x_j\| \text{ for all } x_j \in \mathbf{x} \setminus \{x_i\}\}, \quad (1.16)$$

where  $\|\cdot\|$  stands for the Euclidean distance. Finding the Dirichlet cell of all data points results in a set of non-overlapping, disjoint and convex polygons which build a Voronoi tessellation out of the observed point pattern. The adaptive Voronoi intensity estimator  $\widehat{\lambda}(u)$  at location  $u \in W$  is then the reciprocal size of  $\mathcal{V}_u(\mathbf{x}, W)$ , the cell where  $u$  belongs to. This estimator is unbiased when the underlying point process is homogeneous, conserves mass and performs well when there

are abrupt changes in the intensity of observed data (Baddeley et al., 2015, Chapter 6). However, it generates a huge variance (Barr and Schoenberg, 2010; Moradi et al., 2018a). Apparently, it was first introduced by Brown (1965) and Ord (1978). Ebeling and Wiedenmann (1993) used Voronoi estimators to study spatial concentration of photons locally. Other applications such as estimating neuronal density can be found in Duyckaerts et al. (1994); Duyckaerts and Godefroy (2000). Barr and Schoenberg (2010) investigate statistical properties of such intensity estimator when the underlying point process is supposed to be an inhomogeneous process. Under particular conditions, they also discussed being approximately ratio-unbiased when the underlying point process is an inhomogeneous Poisson process. Moreover, this Voronoi technique has also been used in statistical seismology (Ogata, 2011; Baddeley et al., 2015).

### 1.2.3 Relative risk

Sometimes the point pattern in question contains points with different marks. A very known example might be a spatial case-control study in which some points play the role of control and others the spatial locations of a set of disease cases. The spatially varying risk of the case study might be measured by the ratio of the intensity function of the two point processes (Bithell, 1991; Kelsall and Diggle, 1995a,b; Diggle, 2003; Baddeley et al., 2015).

Consider there are two different observed point patterns  $\mathbf{x}$  and  $\mathbf{y}$ , generated by point processes  $X$  and  $Y$ , respectively. The relative risk of the two point process is of the form

$$\rho(u) = \log \frac{\lambda_X(u)}{\lambda_Y(u)}, \quad u \in S, \quad (1.17)$$

where  $\lambda_X(u)$ ,  $\lambda_Y(u)$  are the intensity functions of the underlying point processes  $X$  and  $Y$ , respectively. However, in practice,  $\rho$  needs to be estimated and it demands to estimate  $\lambda_X(u)$  and  $\lambda_Y(u)$  individually. An enough good estimate of  $\rho$  provides us with the spatially-varying relative frequency of each type of points.

The very first idea to estimate  $\rho$  is to use the plug-in estimator based on kernel estimators of  $\lambda_X(u)$  and  $\lambda_Y(u)$ . Here, the challenge is to choose bandwidth smoothing parameter. As the two patterns  $\mathbf{x}$  and  $\mathbf{y}$  are generated by two different point process, one might estimate  $\lambda_X(u)$  and  $\lambda_Y(u)$  using different bandwidth parameters. Nevertheless, theoretical arguments provided by Kelsall and Diggle



(1995a) suggest to use equal bandwidths. Davies et al. (2016) show that using different bandwidth parameters might lead in potentially misleading methodological artefacts in the resulting estimates. Different bandwidth selection methods for the optimal estimation of relative risk  $\rho$  are discussed in Lawson and Williams (1993); Kelsall and Diggle (1995a,b); Diggle et al. (2005); Hazelton (2008); Davies and Hazelton (2010); Davies (2013); Davies and Baddeley (2018). An overview of various estimators and bandwidth selection methods is also provided by Davies et al. (2018).

Some jointly optimal bandwidth selection methods including minimising the approximate mean integrated squared error (MISE) of the log-transformed risk surface (Kelsall and Diggle, 1995a), a weighted-by-control MISE (Hazelton, 2008) and a crude plug-in approximation to the asymptotic MISE (Davies, 2013) are implemented in R package **sparr** (Davies et al., 2018).

## 1.2.4 Second-order summary statistics

In order to get into the type of the interaction between events, we might aim at studying the correlation between events. In the literature of point processes, the type of such correlation can be determined by using summary statistics. Thinking of correlation, the first thing that comes to mind might be using “*second-moment*” quantities which are built based on pairs of points. Note that to be able to calculate the second-moment quantities we need to estimate the first moment previously. We now review some summary statistics in point process literature.

A summary function that is mostly used to analyse the spatial correlation of point events is  $K$ -function which is firstly introduced by Ripley (1977). Considering  $X$  a stationary point process on  $\mathbb{R}^2$  with constant intensity  $\lambda$ , Ripley (1977) defined the  $K$ -function so that  $\lambda K(r)$  is the expected number of further points of  $X$  lying within a distance  $r$  of a typical point of  $X$ , i.e. for any location  $u \in \mathbb{R}^2$ ,

$$K(r) = \frac{1}{\lambda} \mathbb{E} [N(X \cap b(u, r) \setminus \{u\}) \mid u \in X], \quad (1.18)$$

where  $b(u, r)$  is a disc of radius  $r$  centred at  $u$ . The expectation in (1.18) is formally defined as an expectation with respect to the Palm distribution of  $X$  at  $u$ , see Daley and Vere-Jones (2003) and Møller and Waagepetersen (2003). Since  $X$  is stationary, this expectation does not depend on the choice of location  $u$ . For any bounded region  $B \subset \mathbb{R}^2$  with area  $|B| > 0$ ,  $K(r)$  can be expressed as a

second-moment quantity,

$$K(r) = \frac{1}{\lambda^2|B|} \mathbb{E} \left[ \sum_{x_i \in X \cap B} \sum_{x_j \in X} \mathbf{1}\{0 < \|x_j - x_i\| \leq r\} \right], \quad (1.19)$$

so that  $\lambda^2|B|K(r)$  is the expected number of ordered pairs of distinct points  $(x_i, x_j)$  lying at most  $r$  units apart, with the first point  $x_i$  falling in  $B$ . Again, the stationarity of  $X$  implies that the right hand side of (1.19) does not depend on the choice of  $B$ .

When a realisation of  $X$  is observed inside the window  $W \subset \mathbb{R}^2$ , giving a point pattern  $\mathbf{x} = \{x_1, \dots, x_n\}$  of points  $x_i \in W$ , the estimator of  $K(r)$  is of the form

$$\hat{K}(r) = \frac{|W|}{n(n-1)} \sum_i \sum_{j \neq i} \mathbf{1}\{\|x_i - x_j\| \leq r\} e_W(x_i, x_j, r), \quad (1.20)$$

where  $e_W(x_i, x_j, r)$  is a correction for edge effect bias (Baddeley et al., 2015).

Note that (1.20) is a *biased* estimator of  $K(r)$ . The double sum on the right of (1.20) is an unbiased estimate of  $\lambda^2|W|K(r)$  for any stationary point process. The denominator  $n(n-1)$  effectively serves as an estimator of  $\lambda^2|W|^2$ , but is only unbiased if  $X$  is a Poisson process. Writing  $\hat{K}(r) = A/B$  where  $A$  is the double sum in (1.20) and  $B = n(n-1)/|W|$ , we have  $\mathbb{E}[A]/\mathbb{E}[B] = K(r)$  if  $X$  is a Poisson process. Under reasonable assumptions,  $\hat{K}(r)$  is a consistent estimator of  $K(r)$ . In practice the bias is usually small when  $r$  is not too large (Moradi et al., 2018d).

However, when analysing real data, it rarely happens to have the point process in question stationary. Thus, the empirical  $K$ -function might be misleading if applied to point pattern data which are spatially inhomogeneous. By developing an analogue of  $K$ -function (1.19), Baddeley et al. (2000) extended the  $K$ -function to the cases in which the pattern is spatially inhomogeneous. For second-order intensity-reweighted stationary point processes, they defined the  $K$ -function as

$$K_{inhom}(r) = \frac{1}{|B|} \mathbb{E} \left[ \sum_{x_i \in X \cap B} \sum_{x_j \in X} \frac{\mathbf{1}\{0 < \|x_j - x_i\| \leq r\}}{\lambda(x_i)\lambda(x_j)} \right], \quad (1.21)$$

which does not depend on the choice of  $B$ . Baddeley et al. (2000) further proposed the following pointwise unbiased estimator

$$\hat{K}_{inhom}(r) = \frac{1}{|W|} \sum_i \sum_{j \neq i} \frac{\mathbf{1}\{\|x_i - x_j\| \leq r\} e_W(x_i, x_j, r)}{\lambda(x_i)\lambda(x_j)}, \quad (1.22)$$

where similar to (1.20),  $e_W(x_i, x_j, r)$  is an edge correction. In the literature, different edge corrections have been developed and an overview and comparison can be found in Gabriel (2014).

In practice, however,  $\lambda(\cdot)$  needs to be estimated as it is not known previously. Thus, an estimate of  $\lambda(\cdot)$  is needed in advance to compute (1.22). Notwithstanding, replacing  $\lambda(\cdot)$  by  $\hat{\lambda}(\cdot)$  in (1.22) makes it severely under-biased which might be caused from the bias of the intensity estimator. Baddeley et al. (2000) suggested to use the “leave-one-out” kernel smoother to estimate the intensity function which seems to provide a better estimator for the  $K$ -function.

Regardless of homogeneity, for Poisson point processes  $K(r) = \pi r^2$ . Values larger than  $\pi r^2$  indicate clustering behaviour while values smaller than  $\pi r^2$  point out repulsion. Another second-order summary statistic which has been frequently used is the pair correlation function that is defined as

$$g(u, v) = \frac{\lambda_2(u, v)}{\lambda(u)\lambda(v)}, \quad u, v \in \mathbb{R}^2. \quad (1.23)$$

For Poisson processes,  $g(\cdot, \cdot) = 1$  as  $\lambda_2(\cdot, \cdot) = \lambda(\cdot)\lambda(\cdot)$ . Values larger than one indicate clustering behaviour whereas values less than one show repulsion. This comes from the fact that values larger than one mean  $\lambda_2(\cdot, \cdot) > \lambda(\cdot)\lambda(\cdot)$  which shows the tendency of points to be closer to each other. With the assumption that  $g$  is translation invariant, i.e.  $g(u, u+h) = g_0(h)$  for some function  $g_0 : \mathbb{R}^2 \rightarrow [0, \infty)$ , then  $X$  is second-order intensity-reweighed stationary and

$$K_{inhom}(r) = \int_{b(o,r)} g_0(h)dh, \quad (1.24)$$

where  $b(o, r)$  denotes a disc centred at origin with radius  $r > 0$ .

### 1.3 Spatial point processes on linear networks

The last decade witnessed an extraordinary increase in interest in the analysis of network related data within numerous disciplines. This pervasive interest is partly caused by a strongly expanded availability of network data. In the spatial statistics field, there are numerous real examples such as the location of traffic accidents, geo-coded locations of crimes or anti-social behaviour events in the streets of cities that need to restrict the support of the underlying process over

such linear networks to set and define a more realistic scenario. Network events can be categorised to *on-network* events such as traffic accidents which inherently happen on street networks and *alongside-network* events such as street crimes on side-walk which are usually committed alongside to street networks. The inherent geometry of a linear network makes the analysis of spatial point patterns as a different story with respect to the analysis of such patterns occurring on planar regions (reviewed in Section 1.2). For example, measuring distance between events play an important role when events are living on a linear network.

A nice introduction on spatial analysis along networks can be found in Okabe and Sugihara (2012). Following, we provide the extension of spatial point processes to network events.

### 1.3.1 Linear networks

A line segment in the plane with endpoints  $u$  and  $v$  can be written in a parametric form as  $[u, v] = \{tu + (1 - t)v : 0 \leq t \leq 1\}$  with  $u, v \in \mathbb{R}^2$ . A linear network  $L$  can be defined as the union of a finite collection of line segments embedded in the plane (Ang et al., 2012). The endpoints of the segments are called *nodes* and the degree of a node  $\mathbf{n}$  (denoted by  $\delta$ ) is the number of line segments that share the same node (Okabe and Sugihara, 2012). The total length of all line segments in  $L$  is denoted by  $|L|$ . The distance between two points  $u$  and  $v$  in the network  $L$  is usually computed by the *shortest-path* distance  $d_L(u, v)$  which is the minimum of the length of all possible paths between  $u$  and  $v$ . However, different possible distances have been discussed in Rakshit et al. (2017); Anderes et al. (2017).

The disc of radius  $r > 0$  centred at the point  $u \in L$  is given by  $b_L(u, r) = \{v \in L : d_L(u, v) \leq r\}$  which includes all points within a shortest-path distance less than  $r$ , and its relative boundary  $\partial b_L(u, r)$  is the set of points lying exactly  $r$  units away from  $u$ . Moreover, the circumference  $m(u, r)$  is the number of points in the relative boundary with centre at  $u$  and radius  $r$  which are on the linear network  $L$ , i.e.  $m(u, r) = \#\{\partial b_L(u, r) \cap L\}$ . The circumference is finite for  $r < \infty$  and  $m(u, \infty) = \infty$  by convention. In addition,  $R(L)$  is the largest value so that  $m(u, r) \neq 0$  for all events  $u \in X$  and all  $r \leq R(L)$  (for more details see (Baddeley et al., 2015, Chapter 17)).

Similar to planar point processes, a point process  $X$  on a linear network  $L$  with no overlapping points is a random countable subset of  $\mathbb{R}^2$ . Applying some

modifications, most of the provided details in Section 1.2 are also valid for network events. For instance, the intensity measure (1.1) is in the form

$$\mathbb{E}[N(X \cap A)] = \int_A \lambda(u) d_1 u, \quad u \in L, \quad (1.25)$$

where  $d_1 u$  denotes one-dimensional integration over the line segment (Federer, 1996; Ang et al., 2012),  $\lambda(\cdot)$  is considered as the intensity function of  $X$  and  $A \subset L$ . Here,  $\lambda(u)$  gives the expected number of points per unit length of the network in the vicinity of location  $u$ . For any non-negative and measurable function  $f$  on  $L$ , *Campbell's formula* is of the form

$$\mathbb{E} \sum_{x \in X} f(x) = \int_L f(u) \lambda(u) d_1 u. \quad (1.26)$$

Surprisingly, intensity estimation on a network of lines, such as a road network, seems to be a complicated task. Several techniques published in the literature, in geography and computer science, have turned out to be erroneous in which some of the existing techniques are also computationally expensive (Xie and Yan, 2008; Okabe et al., 2009; McSwiggan et al., 2017; Moradi et al., 2017). In Chapter 2, we will review the current intensity estimations together with their advantages and/or disadvantages. Some new intensity estimators are also proposed in Chapters 2 and 3.

Due to the nature of networks, there is still no clear picture of how stationarity can be defined on linear networks. For example, using the classical shift transformation, there is no guarantee if the point will still live on the network after transformation. Baddeley et al. (2017) considered some popular procedures to construct point patterns on networks and pointed out that models are no longer stationary when distance is measured using the shortest-path distance. Having no clear picture of an applicable transformation method on networks might make some restriction of developing more complex models and summary statistics such as higher order ones (van Lieshout, 2011). However, Anderes et al. (2017) developed some parametric classes of covariance functions for linear networks, defined on the nodes and edge points. Their covariance functions are also *isotropic* i.e. they only depend on the geodesic distance or resistance metric which was considered in Anderes et al. (2017). The resistance metric here is an extension of the resistance

metric developed in electrical network theory (Anderes et al., 2017). van Lieshout (2017) also defined nearest-neighbour point processes on linear networks.

### 1.3.2 Second-order summary statistics

Another consequence of considering a linear network as a state space for the point process  $X$  is coming up when dealing with summary statistics such as  $K$ -function and pair correlation function. Okabe and Yamada (2001) adapted the  $K$ -function to the linear network by only replacing the Euclidean distance by shortest-path distance as

$$\widehat{K}_{net}(r) = \frac{|L|}{n(n-1)} \sum_{i=1}^n \sum_{j \neq i} \mathbf{1}\{d_L(x_i, x_j) \leq r\}. \quad (1.27)$$

However, this modification was affected by the geometry of the linear network and its interpretation was not easy. The  $\widehat{K}_{net}$  can be reformed as a Palm expectation and then it discloses that  $\widehat{K}_{net}$  depends on locations, even for Poisson processes, see Ang et al. (2012). Thus, there is no simple benchmark to measure the deviation from Poisson processes. Moreover, no connection between  $\widehat{K}_{net}$  and pair correlation function could be provided.

Knowing this, Ang et al. (2012) proposed a modification as

$$\widehat{K}_L(r) = \frac{|L|}{n(n-1)} \sum_{i=1}^n \sum_{j \neq i} \frac{\mathbf{1}\{d_L(x_i, x_j) \leq r\}}{m(x_i, d_L(x_i, x_j))}, \quad (1.28)$$

for  $0 \leq r \leq R(L)$ . The  $m(x_i, d_L(x_i, x_j))$  in the denominator of (1.28) is somehow playing the role of an edge correction by adding a contribution from each pair of points. This is actually analogous to the isotropic edge correction, since  $m(x_i, d_L(x_i, x_j))$  is a measure of the size of the boundary of the ball of radius  $d_L(x_i, x_j)$  centred at  $x_i$  (Ripley, 1977; Ang et al., 2012). For the point process  $X$  on the linear network  $L$  with constant intensity  $\lambda > 0$ , Ang et al. (2012) defined the geometrically corrected  $K$ -function as

$$K_L(u, r) = \frac{1}{\lambda} \mathbb{E} \left[ \sum_i \frac{\mathbf{1}\{d_L(u, x_i) \leq r\}}{m(u, d_L(u, x_i))} \mid u \in X \right], \quad (1.29)$$

and called  $X$  second-order pseudostationary if  $K_L(u, r)$  does not depend on  $u$ . They also showed that any homogeneous Poisson process is second-order

pseudostationary according to which  $K_L(u, r) = K_L(r) = r$ . Therefore, this can be used as a benchmark when measuring deviation from Poisson processes. They next defined the pair correlation function with the form

$$g_L(u, v) = \frac{\lambda_2(u, v)}{\lambda^2}, \quad u, v \in L, \quad (1.30)$$

where  $\lambda_2(u, v)$  is the second-order product density function of  $X$ . When  $g_L(u, v) = g_0(d_L(u, v))$ , i.e. the pair correlation only depends on the shortest-path distance between two points in question, then  $X$  is second-order pseudostationary and in this case

$$K_L(r) = \int_0^r g_0(t) dt. \quad (1.31)$$

The estimator (1.28) is *ratio-unbiased* and its asymptotic bias and variance is studied in Baddeley et al. (2000); Ang et al. (2012).

Equations (1.29) and (1.30) are built for homogeneous point processes which is not realistic when dealing with real datasets. In order to make the network  $K$ -function (1.29) and pair correlation function (1.30) applicable to a class of non-stationary processes, Ang et al. (2012) considered a point process  $X$  on a linear network  $L$  with intensity function  $\lambda(u)$  and product density function  $\lambda_2(u, v)$ , and introduced the pair correlation function as

$$g_{LI}(u, v) = \frac{\lambda_2(u, v)}{\lambda(u)\lambda(v)}, \quad u, v \in L, \quad (1.32)$$

and it can be interpreted informally as the standardized probability density that a pair of events occur in two infinitesimal discs  $d_1u$  and  $d_1v$ . The point process  $X$  is called second-order reweighted pseudostationary (SORS) if the pair correlation function depends only on the shortest-path distance  $g_{LI}(u, v) = g_0(d_L(u, v))$  (see Ang et al. (2012)). We point out that for Poisson process on linear networks, the product density function satisfies that  $\lambda_2(u, v) = \lambda(u)\lambda(v)$  and therefore  $g_{LI}(u, v) = 1$ .

In general, the inhomogeneous geometrically corrected pair correlation function (1.32) has a direct relationship with the inhomogeneous geometrically corrected  $K$ -function (see Illian et al. (2008), Baddeley et al. (2015)) according to which

$$K_{LI}(r) = \int_0^r g_0(t) dt, \quad r \geq 0, \quad (1.33)$$

where this direct relationship can be rewritten as

$$g_0(r) = \frac{d}{dr} K_{LI}(r), \quad r \geq 0. \quad (1.34)$$

The inhomogeneous geometrically corrected  $K$ -function has also a simple form for Poisson processes on linear networks, and it is given by

$$K_{LI}(r) = r, \quad \text{for } r \geq 0. \quad (1.35)$$

Similar to the case of planar point processes,  $K_{LI} > r$  ( $g_{LI} > 1$ ) indicates clustering behaviour and  $K_{LI} < r$  ( $g_{LI} < 1$ ) shows inhibition among points.

An approximately unbiased estimator of the inhomogeneous geometrically corrected  $K$ -function (1.33) is given by

$$\widehat{K}_{LI}(r) = \frac{1}{|L|} \sum_{i=1}^n \sum_{j \neq i} \frac{\mathbf{1}\{d_L(x_i, x_j) \leq r\}}{\widehat{\lambda}(x_i) \widehat{\lambda}(x_j) m(x_j, d_L(x_i, x_j))}, \quad 0 \leq r \leq R(L), \quad (1.36)$$

where  $\widehat{\lambda}(\cdot)$  is an estimate of intensity function. However, the estimator (1.36) has a high variance. Ang et al. (2012) proposed to replace  $|L|$  by  $\sum_{i=1}^n 1/\widehat{\lambda}(x_i)$  to decrease the variance due to the positive correlation between numerator and denominator. The new estimator then takes the form

$$\widehat{K}_{LI}(r) = \frac{1}{\sum_{i=1}^n 1/\widehat{\lambda}(x_i)} \sum_{i=1}^n \sum_{j \neq i} \frac{\mathbf{1}\{d_L(x_i, x_j) \leq r\}}{\widehat{\lambda}(x_i) \widehat{\lambda}(x_j) m(x_j, d_L(x_i, x_j))}, \quad 0 \leq r \leq R(L). \quad (1.37)$$

A non-parametric estimator of the inhomogeneous geometrically corrected pair correlation function on a linear network  $L$  is

$$\widehat{g}_{LI}(r) = \frac{1}{\sum_{i=1}^n 1/\widehat{\lambda}(x_i)} \sum_{i=1}^n \sum_{j \neq i} \frac{\kappa(d_L(x_i, x_j) - r)}{\widehat{\lambda}(x_i) \widehat{\lambda}(x_j) m(x_j, d_L(x_i, x_j))}, \quad 0 \leq r \leq R(L), \quad (1.38)$$

where  $\kappa_\epsilon$  is a one-dimensional kernel function.

It is noted that in the calculation of  $\widehat{K}_{LI}(r)$  and  $\widehat{g}_{LI}(r)$  we need an intensity function estimator. Thus, the estimation of the first-order intensity plays an important role (and it is crucial) in the estimation of the inhomogeneous geometrically corrected  $K$ - and pair correlation functions (Baddeley et al., 2015; Moradi et al., 2018d). Point processes on linear networks and in particular their intensity estimators will be discussed in Chapters 2 and 3.



## 1.4 Trajectories

Modern data collection techniques allow tracking objects continuously. This means that we do not only know the current location of a moving object, we also track the objects over time. A set of some tracks from different moving objects may be considered a trajectory pattern (see Figure 1.3). Examples can be found when studying traffic flow, taxis movements, animal movements, etc.

Güting and Schneider (2005) focused on moving objects databases and extended database technology to deal with moving objects. Challa et al. (2011) presented an introduction to the field of object tracking and provided solid foundation to the collection of diverse algorithms developed by academics, scientific researchers and engineers. Hanks et al. (2015) proposed a continuous-time discrete-space (CTDS) model for animal movement. Russell et al. (2016) introduced an approach that models dependent movement by augmenting a dynamic marginal movement model with a spatial point process interaction function within a weighted distribution framework. Niu et al. (2016) considered a multivariate Ornstein Uhlenbeck diffusion process to model the movement of animals in continuous time. Hooten and Johnson (2017) presented a natural basis function approach to constructing appropriate covariance models for movement processes.

There are already some R packages available in CRAN Task View: Handling and Analyzing Spatio-Temporal Data to handle moving objects. Most of them are focused on handling and analysing animal movements such as **adehabitatLT** (Calenge, 2006), **tripEstimation** (Sumner et al., 2009), **argosfilter** (Freitas, 2012), **V-Track** (Campbell et al., 2012), **animalTrack** (Farrell and Fuiman, 2013), **BBMM** (Nielson et al., 2013), **bcpa** (Eliezer, 2014), **BayesianAnimalTracker** (Yang, 2014), **TrackReconstruction** (Battaile, 2014), **mkde** (Jeff et al., 2014), **SimilarityMeasures** (Toohey, 2015), **smam** (Jun and Vladimir, 2016), **trip** (Sumner, 2016), **moveHMM** (Michelot et al., 2016), **FLightR** (Rakhimberdiev et al., 2017). In particular, the package **adehabitatLT** (Calenge, 2006) provides tools to simulate trajectories using a Brownian motion, correlated Random walks and Levy walks. Toohey (2015) presented four different similarity measures in **SimilarityMeasures**. Michelot et al. (2016) in **moveHMM** provided animal movement modelling using hidden Markov models. Using multiple regression, **fishmove** (Radinger and Wolter, 2014) provides functions to predict fish movement parameters. R package **trackeR** (Frick and Kosmidis, 2017) provides infrastructure for

running and cycling Data.

However, to the best of our knowledge, R is still missing a complete set of generic data structures and methods to effectively analyse trajectories without being limited to a particular domain. Looking at trajectory patterns as they live in space and move over time, we aim at building statistical methodologies for trajectory patterns using the literature of spatial point pattern. Following, we provide a definition of a trajectory pattern from the point process perspective (Moradi et al., 2018c).

**Definition 1.1.** *A trajectory pattern is a dataset which provides observed tracks ( $s_i$ ) of a set of moving objects such as cars, humans, etc over a finite time period  $T$ . We denote a trajectory pattern consisting of  $n > 0$  single tracks as  $S = \{s_i : s_i \subset \mathbb{R}^2, i = 1, \dots, n\}$ , that is a countable set of tracks. Each  $s_i$  is itself a countable set of points, e.g.  $s_i = \{x_1^{s_i}, x_2^{s_i}, \dots, x_{m_i}^{s_i}\}$  with  $i = 1, \dots, n$  where each  $s_i$  consists of  $m_i < \infty$  points that are associated with an increasing set of time stamps  $t_1, \dots, t_{m_i}$ ,  $t_j < t_{j+1} \forall j$ .*

We point out that the length of each of the tracks  $s_i$  is not necessarily the same for all tracks. In other words, they might have different start/end times. Each single track  $s_i$  represents the movement of a moving object within a finite time/area. It is usually supposed that locations of a moving object are recorded in regular timestamps. However, if timestamps are not regular, one can still interpolate the locations in regular timestamps. Therefore, each single track in  $S$  might be seen as a set of points corresponding to the considered timestamps. Discretising all tracks of  $S$  according to regular timestamps results in a list of point patterns (one per each time) which enables us to consider a trajectory pattern as a point pattern which is changing over time. Therefore, using point pattern methodology, one can study the behaviour of moving objects over time. For instance, the spatially varying distribution of objects and the type of interaction between them over time can be of interest. Assume that the trajectory pattern  $S$  is observed within the time period  $T$ , thus discretising  $T$  into a time sequence  $\{t_i : t_i \subset T, i = 1, \dots, k \text{ where } t_i < t_j \text{ if } i < j\}$  generates a collection of spatial point patterns, say,  $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_k$  ( $k > 1$ ). Nevertheless, one may still consider  $S$  as a point process on  $\mathbb{R}^{2 \otimes k}$ , where  $\mathbb{R}^{2 \otimes k}$  means  $\mathbb{R}^2 \times \dots \times \mathbb{R}^2$  for  $k$  times when the length of timestamps is  $k$ . We do not discuss this other approach here.

## 1.5 Organization of the thesis

The rest of the thesis is organized as follows. Chapter 2 is devoted to review and propose some kernel-based intensity estimators for spatial point patterns on linear networks together with studying their properties. Intensity estimation of point patterns on linear networks, such as a road network, seems to be a complicated task. Several techniques published in the literature, in geography and computer science, have turned out to be erroneous. Existing techniques are also computationally expensive, especially when analysing big data i.e. too many points and/or a huge network. Prior works proposing intensity estimators include Borruso (2003, 2005, 2008); Xie and Yan (2008); Okabe et al. (2009); McSwiggan et al. (2017). In Chapter 2, we review some of the current intensity estimators and propose two new estimators with the aim of improving current estimators, both statistically and computationally. First, we present a kernel-based edge-corrected intensity estimator and review its statistical properties (Moradi et al., 2017) such as unbiasedness, mass preservedness and variance. Through a simulation study, its statistical performance is studied and compared with that of Okabe et al. (2009); Okabe and Sugihara (2012). Second, we propose a computationally efficient and statistically principled method using 2D convolution (Rakshit et al., 2018). This can be computed rapidly using the Fast Fourier Transform, even on large networks and for large bandwidths, and is robust against errors in network geometry. We also discuss its bias, variance, asymptotics, bandwidth selection, variance estimation, relative risk estimation and adaptive smoothing. Moreover, its performance is analysed and compared with that of McSwiggan et al. (2017), both statistically and computationally. Throughout the Chapter 2 and using proposed methods, several real datasets are also analysed such as crime data in an area near the university of Chicago, Us, anti-social behaviour data in Castellón, Spain, traffic accidents in Medellín, Colombia and Western Australia.

Chapter 3 proposes a new technique to provide an adaptive intensity estimator for spatial point processes independently of the state space (Moradi et al., 2018a). The technique is introduced and applied to Voronoi intensity estimators, and its statistical properties are reviewed and its behaviour is studied through a simulation study. Voronoi intensity estimators are both parameter-free and adaptive; the intensity estimate at a given location is given by the reciprocal size of the Voronoi/Dirichlet cell containing that location. Their major drawback is that

they tend to under-smooth the data in regions where the point density of the observed point pattern is high and over-smooth in regions where the point density is low. To remedy this problem, i.e. to find some middle-ground between over- and under-smoothing, we propose an additional smoothing technique for Voronoi intensity estimators for point processes in arbitrary metric spaces, which is based on repeated independent thinnings of the point process/pattern (Moradi et al., 2018a). We show that our resample-smoothing technique improves the estimation significantly. In addition, we study statistical properties such as unbiasedness and variance, and propose a rule-of-thumb and a data-driven cross-validation approach to choose the amount of thinning/smoothing to apply. We also apply it to two real datasets, traffic accidents in an area of Houston, US (linear network point pattern) and Finish pines which consists of location of trees in a Finnish forest (planar point pattern).

Chapter 4 focuses on spatio-temporal point processes on linear networks. Sometimes we only consider the spatial domain and analyse point patterns regardless of time while they are inherently happening jointly in space and time. Nevertheless, there might be questions that spatial analysis can not answer (Diggle, 2003). We present several characteristics of spatio-temporal point patterns when spatial locations are restricted to a linear network (Moradi et al., 2018b). A non-parametric kernel-based intensity estimator is proposed to highlight the high/low concentration of events within the network and time, either jointly or separately. We also develop second-order characteristics of spatio-temporal point patterns on linear networks such as  $K$ -function and pair correlation function to analyse the type of interaction between points (Moradi et al., 2018b). They are independent of network's geometry and have known values for Poisson point processes. Thus they can be used to measure deviation from being Poisson and also for model selection. Three traffic accidents datasets from Houston (US), Medellín (Colombia) and Eastbourne (UK) are also analysed using the developed methodologies.

In Chapter 5, we pay special attention to trajectories. We propose several classes, methods and statistical methodologies, accommodated in R package **trajectories** to analyse movement datasets (Moradi et al., 2018c). Note that package **trajectories** further extends the capabilities of the R package **spacetime** (Pebesma, 2012). The R package **trajectories** provides functions to handle, simulate and statistically analyse movement data regardless of the domain and by converting a trajectory pattern to a list of point patterns based on regular

---

timestamps. To understand the behaviour of moving objects, we propose an intensity estimation method (to highlight the more visited streets, dense paths, etc), distance analysis, movements smoothing and variability area of second-order summary statistics (to see the changes of the type of interaction between objects over time). In order to compare the estimated (observed) intensity with the expected intensity, we further introduce chimaps that highlights the areas with higher/lower estimated intensity than expected intensity. Using the developed methodologies in package **trajectories**, we finally analyse a taxi movement data from Beijing, China.

Lastly, the thesis ends with Chapter 6 providing a summary of our findings and some future lines of research.



## CHAPTER 2

---

# Kernel intensity estimators for network events

---

## 2.1 Introduction

When the data is of the form of point patterns on linear networks, it is challenging to estimate the intensity function. The classical statistical methodologies were originally designed for analysing events on a plane, however as a matter of convenience, they were used to get into the behaviour of some other events which are inherently not planar point patterns. Although this might facilitate the analysing process, it may also lead to false conclusions.

Figure 2.1 shows two point patterns, the original pattern is the plot on the right side of Figure 2.1. The same point pattern after retaining the same data points and removing the network is displayed on the left plot of Figure 2.1. Looking only on the left plot, one may feel that such data is generated by a cluster point process while considering the network as the support of data points makes it hard to judge. This simple example might be a warning to consider the network as state space when data points are inherently living on networks. We point out that the pattern on the right plot in Figure 2.1 is a realisation of 120 points randomly distributed over the network. Yamada and Thill (2004) compared the network and planar  $K$ -function by applying them to traffic accident data and showed that using the methods for planar point patterns to study network events might be problematic since they may indicate a significant chance of over-detecting clustered patterns.

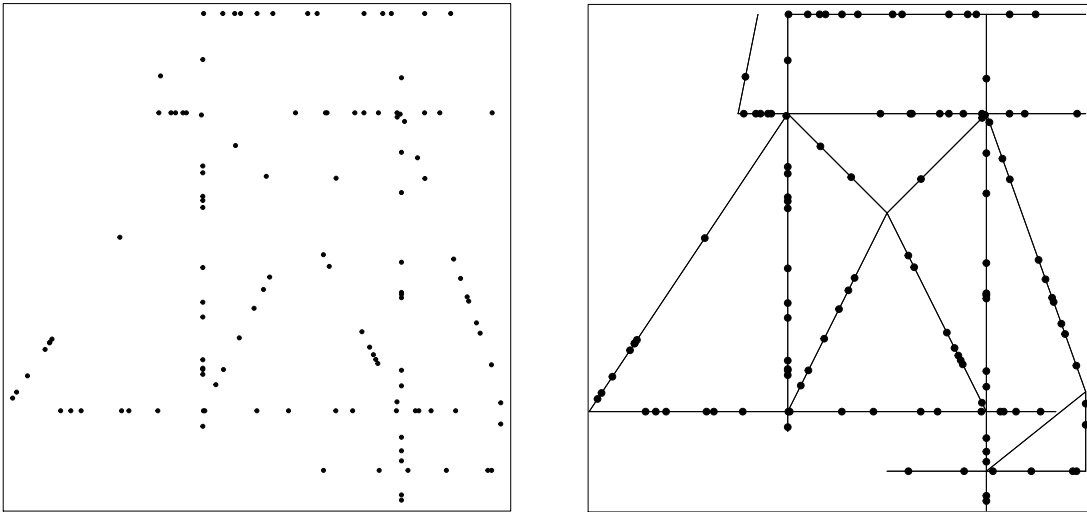


Figure 2.1: Point patterns. *Left*: Non-randomly distributed points. *Right*: Randomly distributed points on a linear network.

However, considering the network as the state space of data points demands further modification in the corresponding methodologies. The very first necessary change is the way to measure the distance between data points. Initially, the shortest-path distance was considered instead of Euclidean distance (Okabe and Yamada, 2001; Borruo, 2005; Xie and Yan, 2008; Okabe and Sugihara, 2012), although, later Rakshit et al. (2017); Anderes et al. (2017) discussed other kind of distances.

In order to consider a more realistic scenario when dealing with network events, there has been some attempts. Okabe and Yamada (2001) replaced the Euclidean distance by the shortest-path distance, and modified the empirical Ripley  $K$ -function by adapting it to the case of a linear network. Ver Hoef et al. (2006) developed spatial autocovariance models for stream networks that incorporate flow and use stream distance. Okabe and Satoh (2006) provided a simple procedure to convert a non-uniform network to an uniform one so that the graph of the new network is not a straight-line graph, i.e. its edges may be a polygonal line segment or it may be curved.

Borruo (2003, 2005, 2008) offered several *ad hoc* proposals for kernel smoothing of network data. One is the “Euclidean, divide-by-length” smoother (Borruo,



2008)

$$\widehat{\lambda}_L^B(u) = \frac{N(\mathbf{x} \cap b(u, r))}{|L \cap b(u, r)|}, \quad u \in \mathbb{R}^2, \quad (2.1)$$

where  $b(u, r) = \{v \in L : \|v - u\| \leq r\}$  is the two-dimensional disc of fixed radius  $r > 0$  centred at the query location  $u$ . Note that Borruo (2008) allows  $u$  to be any location in a two-dimensional space. The numerator of (2.1) is the number of data points lying at most  $r$  units away from the query location, while the denominator is the total length of network lying at most  $r$  units from  $u$ , measured by Euclidean distance. This was described as a “pure” density estimate (Borruo, 2008, p. 382) but no justification was given and no statistical properties were discussed. Xie and Yan (2008) proposed a non-parametric kernel-based estimator to analyse traffic accidents, a network version of uncorrected intensity estimator (1.7). However, this estimator did not consider the structure of the linear network around the points, and as a result it provided a highly biased estimation.

Okabe et al. (2009) introduced equal-split network kernel density estimators, both under continuous and discontinuous schemes, but as these estimators are based on nodes between pairs, if the path (buffer) between two points in question does not contain any node we then suspect whether they can detect the intensity. Details of these estimators are provided in Section 2.3. Using a constructive process based on stream distance and moving average functions, Ver Hoef and Peterson (2010) developed spatial autocovariance models for spatially continuous data on stream networks. The  $K$ -function adapted to the linear network case by Okabe and Yamada (2001) was affected by the geometry of the linear network and its interpretation was not easy. Therefore, Ang et al. (2012) corrected it by defining a geometrically corrected  $K$ -function which does not depend on the geometry of the network. Baddeley et al. (2014) assumed multi-type point patterns and discussed the first- and second-order characterizations of such a pattern on a linear network. O’Donnell et al. (2014) mentioned data which are collected over river networks and developed methods of flexible regression for such a data, discussed local fitting, penalized methods and a spatio-temporal model. In order to fix effects estimation, prediction and the estimation of only covariance parameters, Som et al. (2014) considered optimal sampling designs for stream networks.

In Section 2.2, we present some real datasets of traffic accidents and street crimes. Section 2.3 reviews the kernel smoothing method defined by Okabe et al. (2009); Okabe and Sugihara (2012). The adapted Jones-Diggle intensity estimator

defined by Moradi et al. (2017) is presented in Section 2.4. Section 2.5 is devoted to review the heat kernel intensity estimator by McSwiggan et al. (2017) and Section 2.6 proposes a fast kernel smoothing using 2D convolution (Rakshit et al., 2018).

## 2.2 Datasets

This Section is devoted to present some real examples of network events such as street crimes and traffic accidents.

### 2.2.1 Chicago crime data

The spatial locations of 116 reported street crimes in the period 25 April to 8 May 2002, recorded over a fortnight in an area of Chicago, US are shown in Figure 2.2. The original crime map was published in the Chicago weekly news in 2002. The data points are labelled by the type of crime such as assault, burglary, car-theft, damage, robbery, theft and trespass. The street network contains 338 intersections, 503 line segments and a total length of 31150 feet (5.9 miles, 9.5 km).

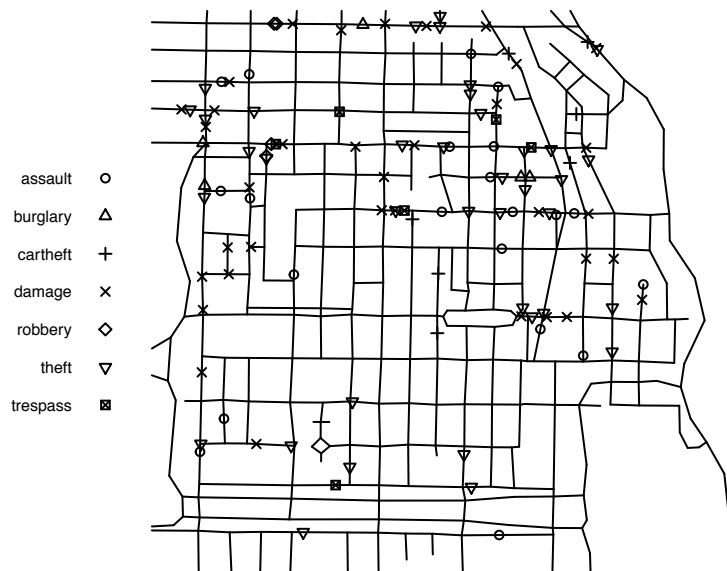


Figure 2.2: Locations of street crimes close to the University of Chicago, US.

After ignoring the type of crimes, this dataset was firstly analysed by Ang et al. (2012) and (Baddeley et al., 2015, p. 721). Ang et al. (2012) considered a

log-quadratic intensity and fitted a parametric model by approximate maximum likelihood using a one-dimensional variant of the Berman-Turner algorithm (Berman and Turner, 1992).

### 2.2.2 Castellón anti-social behaviour

The Castellón anti-social behaviour data reports geo-referenced coordinates of phone calls received by the Police station in the city of Castellón, Spain during January 2013. The listed calls were received by the local Police call centre or transferred by 112 emergency service to the local Police call centre. Geo-codification was performed indirectly by local officials based on precise address information provided by the callers. The calls comprise up to nine different types of crimes or anti-social behaviour categories, but we here only focus on anti-social actions comprising a total number of 184 events in the streets of Castellón. The city of Castellón is divided into 108 census tracks with an overall surface of 108.6 km<sup>2</sup>. The linear network contains 450 nodes and 2242 line segments, with a total length of 161.17 km. This linear network shows a fairly high degree of complexity given by the old-fashion design of the historical centre of the city (see Figure 2.3).

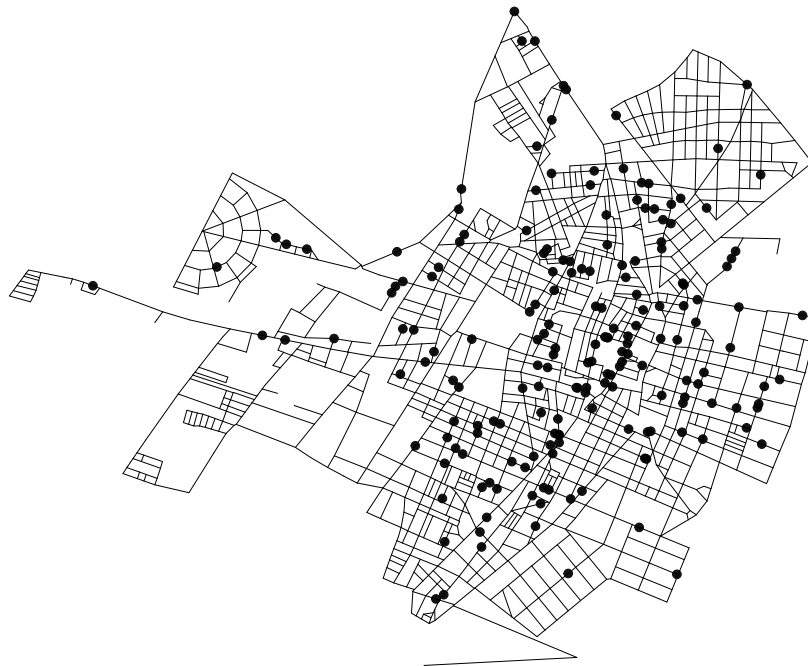


Figure 2.3: Castellón anti-social behaviour during January 2013.

### 2.2.3 Traffic accident in Medellín

Figure 2.4 shows the locations of traffic accidents (based on accident severity) in the urban area of Medellín, Colombia, in 2016 which were published in the OpenData portal of Medellín Town Hall<sup>1</sup>. After cleaning, the data consists of 10,764 points, classified by accident severity as either “property damage”, “personal injury” or “fatal” accidents (comprising 4627, 6004 and 133 points respectively). The three types of accident are displayed in separate panels in Figure 2.4. The road network has 54,164 road segments and total length 1244 km with maximum degree of node 6.

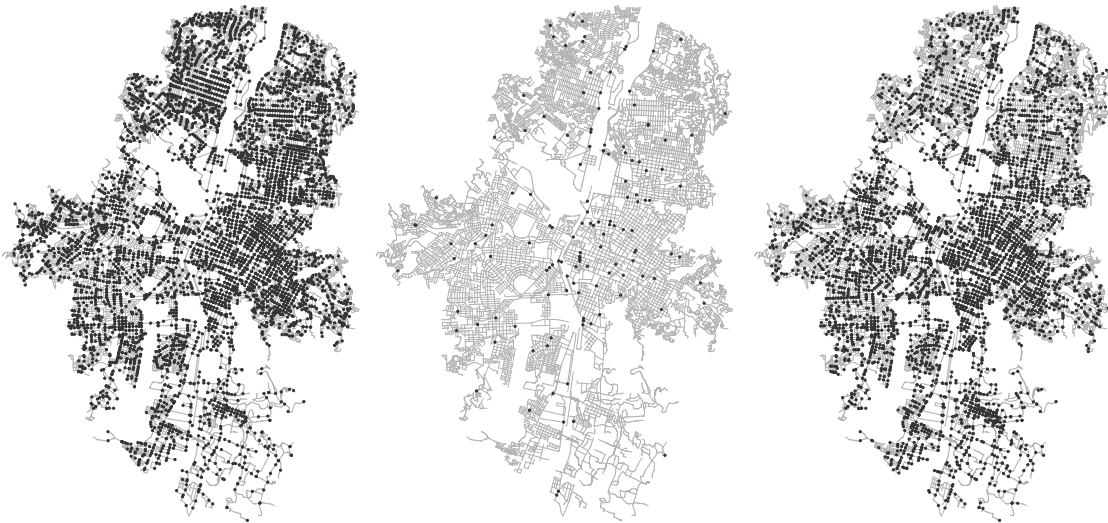


Figure 2.4: Traffic accidents in Medellín, Colombia during the year 2016 which caused *Left*: personal injury, *Middle*: fatal, *Right*: property damage.

### 2.2.4 Traffic accident in Western Australia

The most challenging dataset in this Chapter is shown in Figure 2.5; it gives the locations of 14,562 traffic accidents, recorded for the year 2011, on the road network in the southern half of the state of Western Australia. The road network data originally contained 626,031 individual road segments but this has been reduced to 115,169 segments for analysis. The accident locations and the road

<sup>1</sup><https://www.medellin.gov.co/geomedellin/>

network itself are highly concentrated in the urban area along the west coast of the state. Major highways are also clearly visible from the accident pattern.



Figure 2.5: Traffic accidents in Western Australia during the year 2011.

## 2.3 Equal-split intensity estimators

Okabe et al. (2009) considered a sub-network  $L_u$  of  $L$  where the shortest-path distance between  $u$  and any point on  $L_u$  is less or equal to  $\epsilon$ , and they called it “buffer network” of  $u$  with width  $\epsilon$ . This is equivalent to the concept of a disc centre at  $u$  and radius  $\epsilon/2$ , i.e.  $b_L(u, \epsilon/2)$ . Then, Okabe et al. (2009) defined the network kernel density estimator (network KDE), an intensity estimator for a linear network, as

$$\hat{\lambda}_L^O(u) = \frac{1}{n} \sum_{i=1}^n K_{v_i}(u), \quad u \in L, \quad (2.2)$$

where  $K_v(u)$  is the equal-split discontinuous/continuous kernel function according to which they are producing discontinuous/continuous function on a network  $L$ . It

is important to point out that here continuity at nodes is of interest.

Consider  $\mathbf{N}_{(1)}$  the set of nodes with degree one, and  $\mathbf{N}_{(\geq 3)}$  the set of nodes with degree larger or equal than three. Let  $L_{\mathbf{n}_{(1)q}}$  be the discs (buffer networks) of the  $q$ -th node in  $\mathbf{N}_{(1)}$  with width  $2\epsilon$ , and  $L_{\mathbf{n}_{(\geq 3)q}}$  be the discs (buffer networks) of the  $q$ -th node in  $\mathbf{N}_{(\geq 3)}$  with width  $2\epsilon$ . Then

$$L_{(1)} = \{L_{\mathbf{n}_{(1)q}} | \mathbf{n}_{(1)q} \in \mathbf{N}_{(1)}\} \quad \text{and} \quad L_{(\geq 3)} = \{L_{\mathbf{n}_{(\geq 3)q}} | \mathbf{n}_{(\geq 3)q} \in \mathbf{N}_{(\geq 3)}\}. \quad (2.3)$$

The complement set of above discs (buffer networks) with respect to  $L$  is given by

$$L_{(2)} = \{L_{\mathbf{n}_{(2)q}} | \mathbf{n}_{(2)q} \in (\mathbf{N}_{(1)} \cup \mathbf{N}_{(\geq 3)})^c\}, \quad (2.4)$$

where  $L_{(2)}$  is the set of discs (buffer networks) with exactly two endpoints. Network KDE methods vary from  $L_{(1)}$  to  $L_{(2)}$  and  $L_{(\geq 3)}$ . Okabe and Sugihara (2012) claim that the network KDE on  $L_{(1)}$  and  $L_{(2)}$  is equivalent to the case of one-dimensional probability density estimation (see Silverman (1986)). For  $L_{(\geq 3)}$  the equal-split discontinuous kernel density function must be calculated for two different cases as follows:

**Case 1:** If the point  $u \in L$  does not coincide with a node, the general idea to count is to traverse all the paths over the disc  $b_L(u, \epsilon)$  from  $u$  to the points in  $\partial b_L(u, \epsilon)$  visiting the nodes located on the shortest-path between  $u$  and  $v$ , and equally divide the value of  $\kappa(d_L(u, v))$  by the degree of the respective node minus one for each node, and repeating the same procedure for each path on the disc. The final expression for an equal-split discontinuous kernel density function is given by

$$\mathbf{K}_v(u) = \begin{cases} \frac{\kappa(d_L(u, v))}{(\delta_1 - 1)(\delta_2 - 1) \cdots (\delta_q - 1)}, & 0 \leq d_L(u, v) \leq \epsilon, \\ 0, & d_L(u, v) > \epsilon, \end{cases} \quad (2.5)$$

where  $q$  is the index for the order of the nodes located on the shortest-path between  $u$  and  $v$ , and  $\kappa$  is a one-dimensional kernel function. Note that, if the path between  $u$  and  $v$  does not contain any node, then  $\mathbf{K}_v(u) = \kappa(d_L(u, v))$ , i.e.  $\mathbf{K}_v(u)$  turns into the uncorrected kernel estimator of the intensity function (see equation (1.7) and (Baddeley et al., 2015, p. 168)). A simple example that produces buffers with no node is the real line.

**Case 2:** If the point  $u \in L$  coincides with a node, the expression for an equal-split discontinuous kernel density function is given by

$$K_v(u) = \begin{cases} \frac{2\kappa(d_L(u, v))}{\delta_0(\delta_1 - 1)(\delta_2 - 1) \cdots (\delta_q - 1)}, & 0 \leq d_L(u, v) \leq \epsilon, \\ 0, & d_L(u, v) > \epsilon, \end{cases} \quad (2.6)$$

where  $\delta_0$  is the degree of  $u$ . Okabe and Sugihara (2012) showed that the equal-split functions satisfy the conditions of being a kernel function, and provided a proof for their unbiasedness property when density is given as a uniform probability density function since being complete spatial randomness (CSR) is one of the most fundamental hypotheses in spatial analysis. In case that observed data does not support being uniform, they suggested to use the probability integral transformation in Okabe and Satoh (2006), convert the non-uniform network to a uniform one and then obtain unbiased estimators. However, in practice modifying the structure of a network to reach unbiasedness can not be a good choice (see more details in Okabe and Satoh (2006) and Okabe and Sugihara (2012)).

In brief, equal-split discontinuous kernel method divides the mass at nodes into outgoing segments equally and it results a discontinuous function on the network. Okabe et al. (2009); Okabe and Sugihara (2012) thought that there appears to be no systematic method available for finding a continuous KD function on a network (Okabe and Sugihara, 2012, p.183), hence, by only changing the manner of splitting and applying the same algorithm like equal-split discontinuous one, they defined the equal-split continuous which produces unbiased (assuming similar assumptions on density) and continuous function on the linear network (Okabe and Sugihara, 2012, p. 192). Note that in order to produce a continuous function, Okabe and Sugihara (2012) increased the weight of outgoing segments and decreased the weight of incoming segments in the discontinuous algorithm. Hence, although it results in an unbiased, mass preserved and continuous function on the network, it seems to be somewhat artificial. More formal details about equal-split continuous KD function can be found in Okabe et al. (2009) and Okabe and Sugihara (2012, Section 9.2.3, p. 183).

## 2.4 Adapted Jones-Diggle estimator

We extended the non-parametric kernel-based edge-corrected intensity estimator 1.10 to the linear network case (Moradi et al., 2017). The natural extension is

of the form

$$\widehat{\lambda}_L^{JD}(u) = \sum_{i=1}^n \frac{\kappa(d_L(u, x_i))}{e_L(x_i)}, \quad u \in L, \quad (2.7)$$

where  $\kappa$  is a one-dimensional kernel function, and

$$e_L(v) = \int_L \kappa(d_L(v, u)) d_1 u, \quad v \in L, \quad (2.8)$$

is an edge-correction factor. Following Federer (1996) and Ang et al. (2012), the computation of the edge-correction factor in (2.8) on a linear network is given by

$$\begin{aligned} e_L(v) &= \int_L \kappa(d_L(u, v)) d_1 u = \int_0^\infty \sum_{u \in L: d_L(u, v) = r} \kappa(d_L(u, v)) dr \\ &= \int_0^\infty \sum_{u \in L: d_L(u, v) = r} \kappa(r) dr = \int_0^\infty \kappa(r) m(v, r) dr. \end{aligned} \quad (2.9)$$

Note that the edge-correction in (2.9) involves the circumference function  $m(v, r)$  which provides local information about the structure of the linear network on each point. Therefore, Moradi et al. (2017) takes into account the essence in Okabe and Sugihara (2012) which is counting over discs (buffer networks), but in this case the weight assigned by the kernel is equally distributed over the segments that composed the respective disc. It is also important to point out that the estimator (2.7) is independent of the choice of the kernel, i.e. kernel functions used to estimate the density function non-parametricly can be accommodated in equation (2.7).

Equation (1.8) can also be extended to the network case as

$$\widehat{\lambda}_L^U(u) = \frac{1}{e_L(u)} \sum_{i=1}^n \kappa(d_L(u, x_i)), \quad u \in L, \quad (2.10)$$

where  $e_L(u)$  is the network version of uniform edge correction and can be calculated using (2.9).



### 2.4.1 Statistical properties

The analogous estimator of (2.7) for the planar case (i.e. (1.10)) has been widely studied in the literature of spatial point processes. Properties such as unbiasedness, the variance and the appropriate bandwidth selection are analysed in (Møller and Waagepetersen, 2003; Illian et al., 2008; Diggle, 2003; Baddeley et al., 2015). Following, we review some statistical properties of the intensity estimator (2.7).

**Property 2.4.1.**  $\int_L \widehat{\lambda}_L^{JD}(u) d_1 u$  is an unbiased estimator of  $\mu(L) = \int_L \lambda(u) d_1 u$ .

*Proof.* In order to show the unbiasedness, by using Campbell's formula (1.26) and Fubini's theorem,

$$\begin{aligned} \mathbb{E} \left[ \int_L \widehat{\lambda}_L^{JD}(u) d_1 u \right] &= \mathbb{E} \left[ \int_L \sum_{x \in X} \frac{\kappa(d_L(u, x))}{e_L(x)} d_1 u \right] = \int_L \mathbb{E} \left[ \sum_{x \in X} \frac{\kappa(d_L(u, x))}{e_L(x)} \right] d_1 u \\ &= \int_L \int_L \frac{\kappa(d_L(u, v))}{e_L(v)} \lambda(v) d_1 v d_1 u \\ &= \int_L \lambda(v) d_1 v = \mu(L). \end{aligned}$$

□

**Property 2.4.2.** If the intensity function  $\lambda$  is constant, then the estimator (2.10) is unbiased while the estimator (2.7) is not.

*Proof.*

$$\begin{aligned} \mathbb{E}[\widehat{\lambda}_L^U(u)] &= \mathbb{E} \left[ \sum_{x \in X} \frac{\kappa(d_L(u, x))}{e_L(u)} \right] = \lambda \int_L \frac{\kappa(d_L(u, v))}{e_L(u)} d_1 v = \lambda. \\ \mathbb{E}[\widehat{\lambda}_L^{JD}(u)] &= \mathbb{E} \left[ \sum_{x \in X} \frac{\kappa(d_L(u, x))}{e_L(x)} \right] = \lambda \int_L \frac{\kappa(d_L(u, v))}{e_L(v)} d_1 v. \end{aligned}$$

□

**Property 2.4.3.** The intensity estimator (2.7) produces a continuous function on a network  $L$ , that is, arbitrary points  $u_i \in L$  with a same distance  $d = d_L(u_i, v)$  from  $v \in L \cap X$  receive the same mass  $\kappa(d)/e_L(v)$  regardless of the path between  $u_i$  and  $v$ . In other words, when the path from  $v$  to  $u_i$  touches a node, each outgoing segment receives a copy of the kernel tail.

**Property 2.4.4.** Assume  $X$  is a Poisson point process on a linear network  $L$ . Then, the variance of  $\widehat{\lambda}_L^{JD}(u)$  is

$$V(\widehat{\lambda}_L^{JD}(u)) = \int_L \left[ \frac{\kappa(d_L(u, v))}{e_L(v)} \right]^2 \lambda(v) d_1 v$$

and its unbiased estimator is  $\widehat{V}(u) = \sum_{i=1}^n [\kappa(d_L(u, v_i))/e_L(v_i)]^2$  (Baddeley et al., 2015, p. 173).

We note that similar properties for the estimator (2.10) can be shown.

## 2.4.2 Simulation study

We carried out a simulation experiment to compare the accuracy and performance of the proposed intensity estimator (2.7) with the equal-split discontinuous kernel density estimator (2.2) under inhomogeneous Poisson processes (Moradi et al., 2017). Okabe and Sugihara (2012) mentioned that both discontinuous and continuous network KDE are unbiased, but the equal-split continuous density estimator takes longer time to be computed (see Okabe and Sugihara (2012, Section 9.3) for more details) due to a different manner of splitting. Nevertheless, it also depends on code implementation and it may have a fast implementation later.

Moradi et al. (2017) built two linear networks  $L_1$  and  $L_2$  so that  $L_1$  (on the left side of Figure 2.6) has 19 nodes, 26 lines, and a total length of 39.38 units, within a window  $[0, 5] \times [0, 5]$  units<sup>2</sup>. It also contains 12 intersections with degrees varying between 2 and 6. Network  $L_2$  (on the right side of Figure 2.6) has 40 nodes, 48 lines, and a total length of 45.62 units, within a window  $[0, 6] \times [0, 7.1]$  units<sup>2</sup>. The function `density.lpp` with argument `continuous=FALSE` from the R package **spatstat (1.48-0)** was used to compute the equal-split discontinuous density estimator. It is worth pointing out that throughout this simulation study, both the estimator (2.7) and the equal-split discontinuous density estimator (2.2) are computed by using Gaussian kernel function (Silverman, 1986) and the tail of the kernel for the distances larger than  $4\sigma$  has not been considered. In other words, a tail of the kernel with total mass less than  $M = 1 - p(x < 4\sigma)$  has been deleted, where  $x$  follows the same distribution as the kernel function  $\kappa$  and  $\sigma$  is the smoothing bandwidth parameter. In this simulation study,  $\kappa$  is the Gaussian distribution with mean 0 and standard deviation  $\sigma$ .

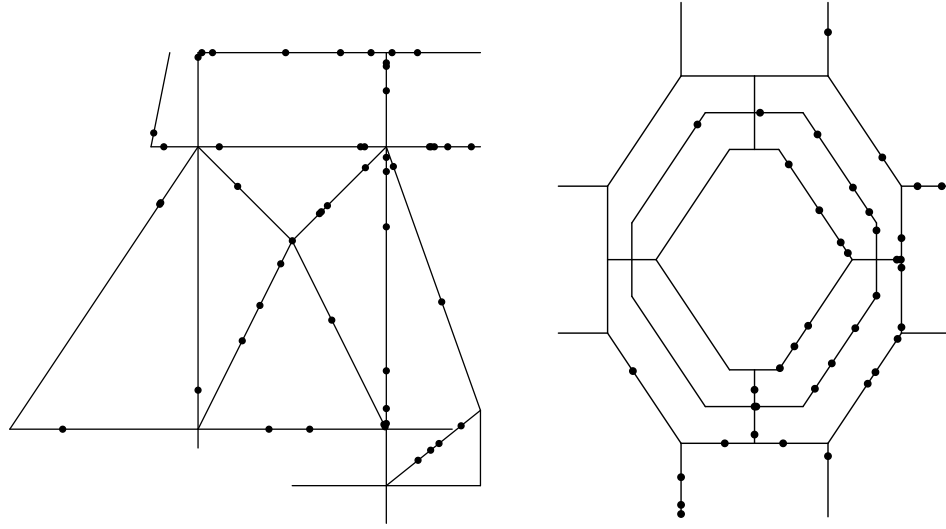


Figure 2.6: Realisations of inhomogeneous Poisson processes. *Left*: with intensity function (2.10) on network  $L_1$ , *Right*: with intensity function (2.12) on network  $L_2$ .

In order to draw the comparison, we considered two inhomogeneous Poisson process so that Figure 2.6 shows single realisations of both (Moradi et al., 2017). Note that the distribution of events is exponentially changing over both networks and forming the inhomogeneity.

Throughout the simulation study, as the true intensity function is known, the optimal bandwidth is selected using

$$\text{ISE} = \int_L (\hat{\lambda}_L^{JD}(u) - \lambda(u))^2 d_1 u, \quad (2.9)$$

meaning that an optimum bandwidth is the one that minimises the integrated squared error (ISE).

**Example 2.4.1.** Consider an inhomogeneous Poisson process on the linear network  $L_1$  with expected number of points  $\mathbb{E}[N(L_1)] \simeq 49$  and intensity function

$$\lambda_1(x, y) = 0.2 e^{0.3(x+y)} \quad \text{with} \quad (x, y) \in L_1. \quad (2.10)$$

A single realization is shown in the left hand side of Figure 2.6. In order to estimate the intensity function, we first need to find an optimum bandwidth so that we use equation (2.9). Figure 2.7 shows that  $\sigma = 1.2$  is the one which minimises the integrated squared error (ISE).

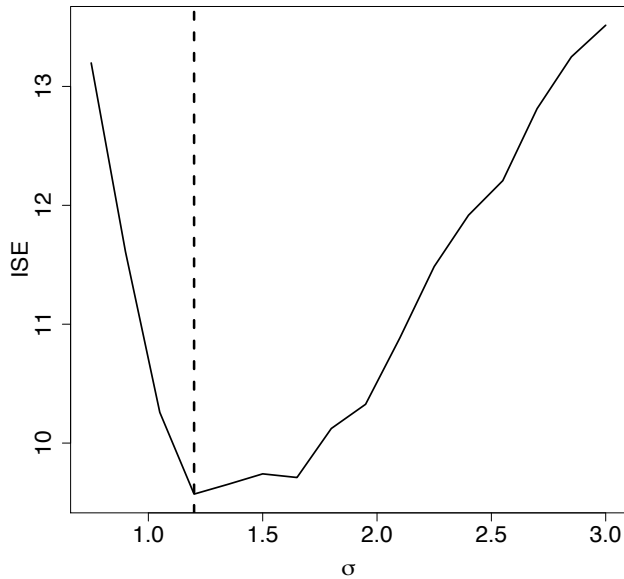


Figure 2.7: Bandwidth selection, ISE versus a sequence of bandwidth smoothing parameter  $\sigma$ .

Using  $\sigma = 1.2$  unit, we estimate the intensity of the pattern in the left hand side of Figure 2.6 through both intensity estimators (2.2) and (2.7), and together with the true intensity are exhibited in Figure 2.8.

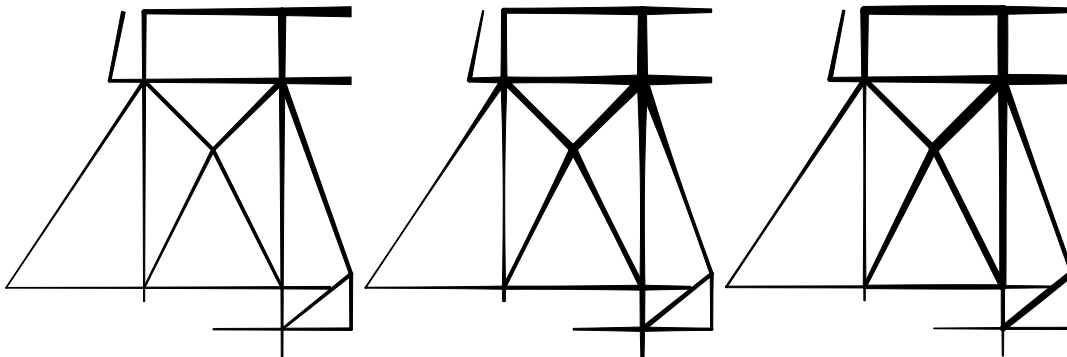


Figure 2.8: Intensity visualisation for the pattern on the left hand side of Figure 2.6. *Left*: true intensity, *Middle*: adapted Jones-Diggle corrected estimator (2.7) with ISE = 9.57, *Right*: equal-split discontinuous (2.2) with ISE = 9.66.

By comparing both estimated intensities in Figure 2.8, it is apparently possible to see that the equal-split discontinuous density estimator assigns a specific thickness along some line segments which can be somewhat equivalent to consider a

constant estimated intensity value along each line segment of the linear network  $L_1$ . However, this is clearly not the case, as the point pattern in Figure 2.6 is not homogeneous and its intensity is changing exponentially over the network. On the other side, adapted Jones-Diggle corrected estimator assigns more precise values of the intensity to the different parts of the line segment, as shown in the middle plot in Figure 2.8. It is thus able to identify high/low intensity values along each line segment in the linear network. Moreover, in terms of ISE, adapted Jones-Diggle corrected estimator (ISE=9.57) outperforms equal-split discontinuous density estimator (ISE=9.66).

In order to have a better comparison, we then simulated 500 realisations over the linear network  $L_1$  of an inhomogeneous Poisson process with intensity function (2.10). By using  $\sigma = 1.2$  units (see Figure 2.7), both adapted Jones-Diggle corrected estimator (2.7) and equal-split discontinuous (2.2) were computed for each one, and we evaluated

$$\begin{aligned} \text{MISE} &= \mathbb{E}[\text{ISE}] = \mathbb{E} \left[ \int_L \left( \widehat{\lambda}_L(u) - \lambda(u) \right)^2 d_1u \right] \\ &= \int_L \text{var} \left[ \widehat{\lambda}_L(u) \right] d_1u + \int_L \text{bias} \left[ \widehat{\lambda}_L(u) \right]^2 d_1u \end{aligned} \quad (2.11)$$

according to which the adapted Jones-Diggle corrected estimator (2.7) has MISE = 10.95 and the equal-split discontinuous (2.2) has MISE = 13.56. Therefore, in this example the adapted Jones-Diggle corrected estimator evidently has a better performance than the equal-split discontinuous.

**Example 2.4.2.** Consider an inhomogeneous Poisson process on the linear network  $L_2$  with expected number of points  $\mathbb{E}[N(L_2)] \simeq 42$  and intensity function

$$\lambda_2(x, y) = e^{(x-y)/x} \quad \text{with} \quad (x, y) \in L_2. \quad (2.12)$$

Similar to Example 2.4.1, we consider ISE in equation (2.9) to find the optimum bandwidth according to which Figure 2.9 shows ISE versus a sequence of bandwidths  $\sigma$ . Figure 2.9 exhibits that  $\sigma = 1.95$  units is the optimal bandwidth and it minimises the ISE.

Figure 2.10 shows the estimated intensity through the equal-split discontinuous (right plot) and adapted Jones-Diggle corrected estimator (middle plot) together with the true intensity (left plot). From Figure 2.10 we can see that the adapted Jones-Diggle corrected estimator (2.7) outperforms the equal-split discontinuous

estimator (2.2) along the segments and it can detect the low/high intensity parts of segments well. Also, ISE for adapted Jones-Diggle corrected estimator is 8.54 while it has the amount of 11.27 for equal-split discontinuous estimator. According to Figure 2.8 and 2.10, we now suspect whether the equal-split discontinuous estimate can disclose the low/high intensity parts of segments, when events are distributed as an inhomogeneous Poisson process.

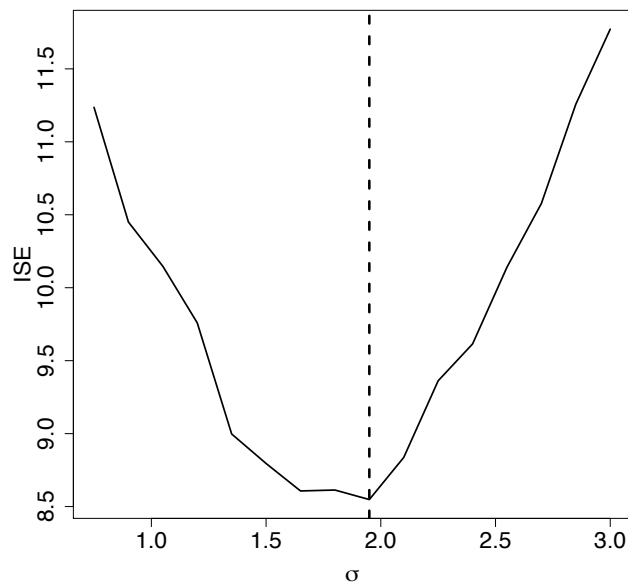


Figure 2.9: Bandwidth selection, ISE versus a sequence of bandwidth smoothing parameter  $\sigma$ .

Similar to Example 2.4.1, we continue with simulating 500 realisations of an inhomogeneous Poisson process with intensity function (2.12) on the network  $L_2$  and we again lie on MISE (2.11) to compare the performance of adapted Jones-Diggle corrected estimator and equal-split discontinuous one. Considering  $\sigma = 1.95$  as an optimum bandwidth parameter (see Figure 2.9), both adapted Jones-Diggle corrected estimator and equal-split discontinuous are computed for 500 realisations, and they result in a MISE = 7.81 and MISE = 10.00, respectively.

### 2.4.3 Chicago crime data

In order to find an optimum bandwidth parameter we use equation (1.13) defined by Cronie and van Lieshout (2018). Figure 2.11 shows that  $\sigma = 650$  feet can minimise the discrepancy in (1.13) while  $\sigma = 60$  feet, used by Baddeley et al.

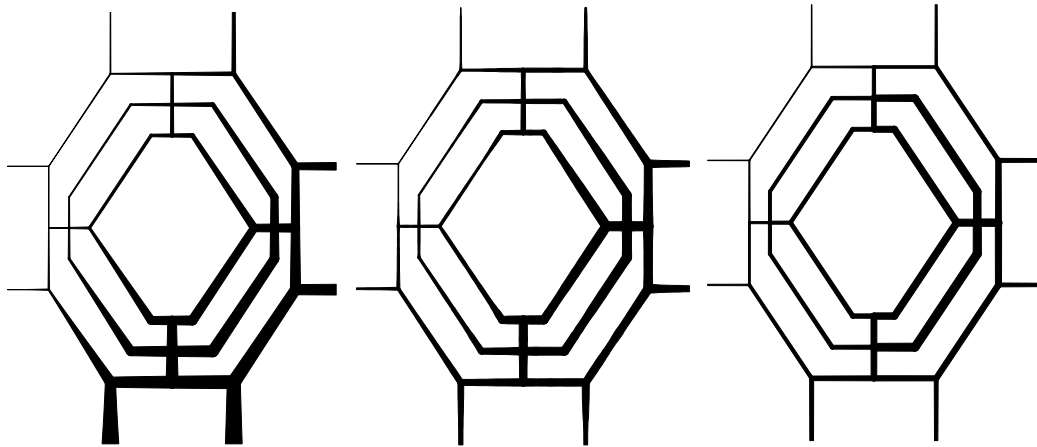


Figure 2.10: Intensity visualisation for the pattern on the right hand side of Figure 2.6. *Left*: true intensity, *Middle*: adapted Jones-Diggle corrected estimator (2.7) with ISE = 8.710, *Right*: equal-split discontinuous (2.2) with ISE = 15.384.

(2015), produce a large bias. We note that this bandwidth, although allowing to show a nice visualisation, does not provide a good intensity estimation.

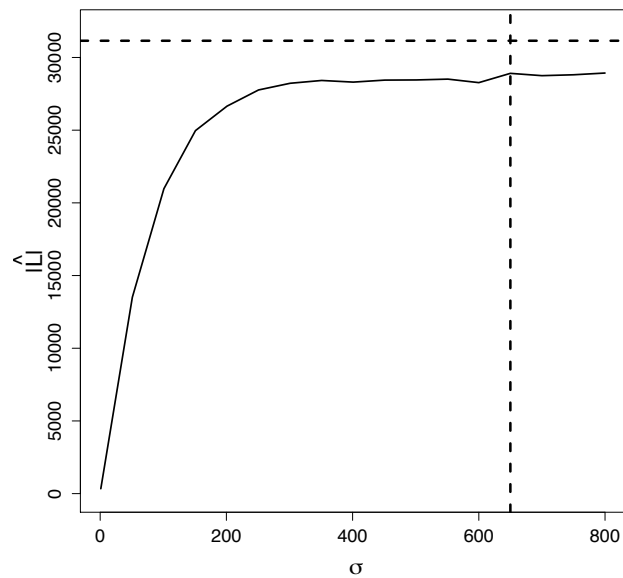


Figure 2.11: Bandwidth selection for Chicago crime data.  $|\hat{L}| = \sum_{i=1}^n 1/\hat{\lambda}_L^{JD}(x_i)$  against a sequence of bandwidth smoothing parameter  $\sigma$  (based on feet). Horizontal dashed line shows the total length of the network.

Bandwidth  $\sigma = 650$  is a much more larger value than the one used in Baddeley

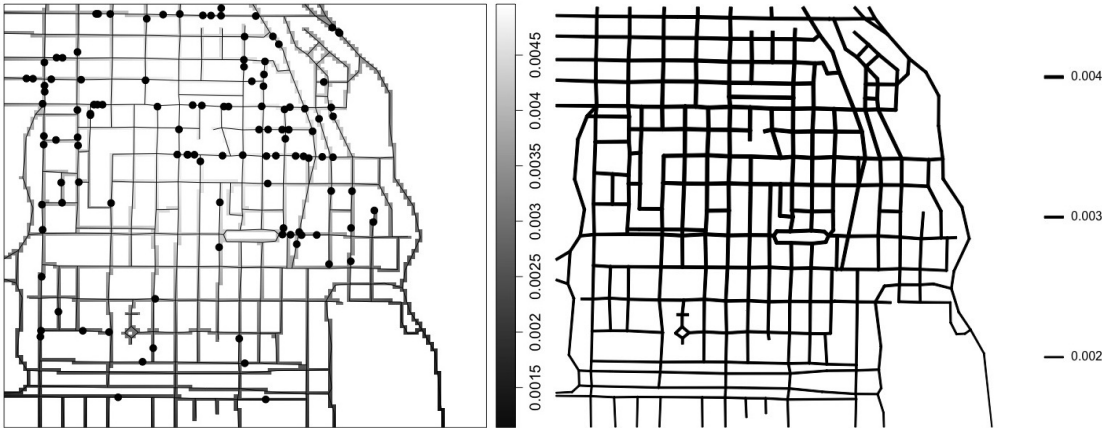


Figure 2.12: Estimated intensity using adapted Jones-Diggle corrected estimator for Chicago street crime data with a bandwidth parameter of  $\sigma = 650$  feet.

et al. (2015, p.721), but we guarantee less bias with this amount of bandwidth. The data is markedly inhomogeneous, and though we are using a non-parametric estimation only based on the locations of crimes, we note that Figure 2.12 shows the same result as in Ang et al. (2012) who used a parametric model, confirming higher intensity in the upper side of the network.

#### 2.4.4 Castellón anti-social behaviour

Figure 2.13 shows  $|\hat{L}| = \sum_{i=1}^n 1/\hat{\lambda}_L^{JD}(x_i)$  versus a sequence of bandwidth smoothing parameter  $\sigma$  (based on km) which suggests a bandwidth of  $\sigma = 0.9$  km based on benchmark (1.13). The adapted Jones-Diggle estimator (2.7) is represented in Figure 2.14, providing indication of inhomogeneity in the pattern. It also indicates higher intensity in the city centre, possibly due to existing areas of night life with pubs and discos that make noise and general disturbances among the population living in these areas. Adding covariate information would enrich the intensity estimation, but we have preferred to focus only on the non-parametric estimation of the intensity.

## 2.5 Heat kernel intensity estimator

McSwiggan et al. (2017) developed a statistically principled kernel estimator on a linear network by exploiting the connection between kernel smoothing and



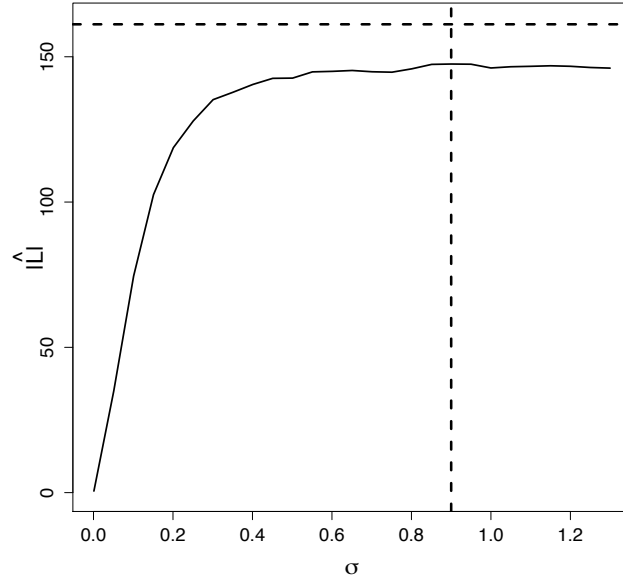


Figure 2.13: Bandwidth selection for Castellón anti-social behaviour data. Horizontal dashed line shows the total length of the network.

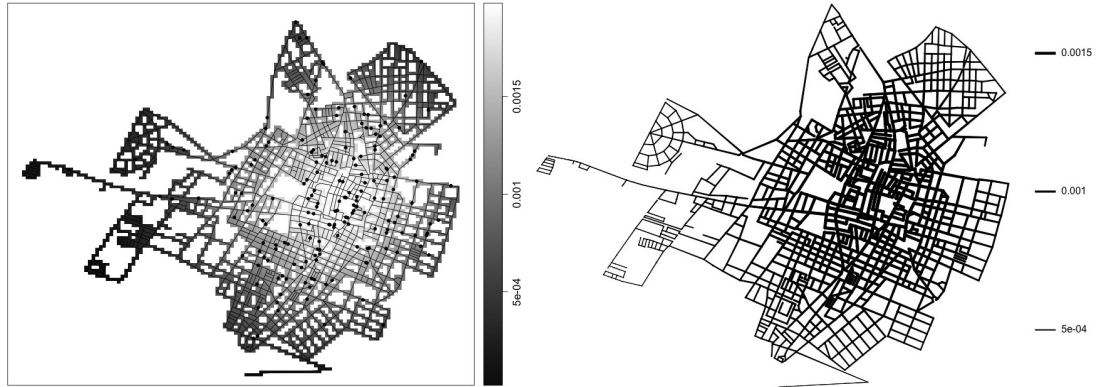


Figure 2.14: Estimated intensity using adapted Jones-Diggle corrected estimator for Castellón anti-social behaviour data with a bandwidth parameter of  $\sigma = 0.9$  km.

diffusion (Chaudhuri and Marron, 2000; Botev et al., 2010).

Their *diffusion* estimator  $\hat{\lambda}_L^H(u)$  can be expressed as a kernel sum

$$\hat{\lambda}_L^H(u) = \sum_{i=1}^n \kappa_\tau(x_i, u), \quad u \in L, \quad (2.13)$$

where  $\kappa_\tau(x_i, v)$  is the *heat kernel*, the probability density at time  $t = \tau^2$  of a particle executing a Brownian motion on the network, starting from location  $x_i$  at time 0.

There is no simple closed-form expression for  $\kappa_\tau$ . McSwiggan et al. (2017, Lemma 2, p. 12) showed that the diffusion estimator is formally equivalent to the “equal-split continuous” rule (Sugihara et al., 2010; Okabe and Sugihara, 2012, Chap. 9) applied to the Gaussian kernel. The diffusion estimator conserves mass and is unbiased when the true density is uniform (McSwiggan et al., 2017, Section 7). The diffusion estimator also satisfies the classical time-dependent heat equation, so that (2.13) can be computed by solving the heat equation up to time  $t = \tau^2$ . An implementation by McSwiggan et al. (2017) is available in the R package **spatstat** (Baddeley and Turner, 2005; Baddeley et al., 2015). Table 2.1 in Section 2.6.2 shows timings for the Chicago data in Figure 2.12 using default settings for the heat equation algorithm for the diffusion estimate. Unfortunately, computation on a large network, such as that illustrated in Figure 2.5, can be costly, depending on the bandwidth and the spatial resolution.

## 2.6 Fast kernel smoothing using 2D convolution

In this Section we propose a kernel estimator on a linear network based on a *two-dimensional* smoothing kernel by Rakshit et al. (2018). The original motivation was speed: this estimator can be expressed in terms of two-dimensional convolutions of the kernel, so it can be computed very rapidly using the FFT. However it also has theoretical and methodological advantages.

**Definition 2.1.** Let  $\mathbf{x} = \{x_1, \dots, x_n\}$  be a point pattern on a linear network  $L$ . Let  $\kappa$  denote a bivariate kernel function, that is, a probability density on  $\mathbb{R}^2$ . The convolution kernel estimator of intensity is, with the uniform correction,

$$\widehat{\lambda}_{L,con}^U(u) = \frac{1}{c_L(u)} \sum_{i=1}^n \kappa(u - x_i), \quad u \in L, \quad (2.14)$$

and with the Jones-Diggle correction

$$\widehat{\lambda}_{L,con}^{JD}(u) = \sum_{i=1}^n \frac{\kappa(u - x_i)}{c_L(x_i)}, \quad u \in L, \quad (2.15)$$

where

$$c_L(u) = \int_L \kappa(v - u) d_1 v, \quad u \in L. \quad (2.16)$$

Notice that (2.14) and (2.15) are similar to (1.8) and (1.10) respectively, and could have been formally “derived” by replacing the two-dimensional integral over

the window  $W$  by the one-dimensional integral over the network  $L$ . They can also be “derived” from (2.10) and (2.7), if we replace the one-dimensional kernel over the network  $L$  by a two-dimensional kernel over the window  $W$ . However, a rigorous justification needs to be found, and statistical properties analysed in this new setting.

The sum in (2.14) is the usual kernel estimator of intensity in two dimensions without edge correction. The denominator  $c_L(u)$  defined in (2.16) is the convolution of the kernel  $\kappa$  with the arc-length measure on the network. Both functions are now evaluated only at locations on the network.

Intuitively it may seem wrong to use a two-dimensional kernel to estimate a one-dimensional density. However, the normalisation in (2.14)–(2.15) ensures that the result has the correct dimension. Values of  $\kappa$  have dimension  $length^{-2}$  (points per unit area) while values of  $c_L(\cdot)$  are integrals of  $\kappa$  against length, so they have dimension  $length^{-1}$ . Consequently (2.14) and (2.15) have the correct dimension  $length^{-1}$  (points per unit length).

In the special case where  $\kappa$  is the uniform density on a disc of fixed radius  $r > 0$ , i.e.  $\kappa(u) = a \mathbf{1}\{\|u\| \leq r\}$  where  $a = 1/(\pi r^2)$ , the sum in (2.14) is equal to  $a$  times the number  $N(\mathbf{x} \cap b_L(u, r))$  of data points falling in the disc of radius  $r$  centred at  $u$ . The denominator  $c_L(u)$  is equal to  $a$  times the total length  $|L \cap b_L(u, r)|$  of network segments falling in the same disc. The uniform-correction estimator (2.14) reduces to

$$\widehat{\lambda}_{L,con}^U(u) = \frac{N(\mathbf{x} \cap b(u, r))}{|L \cap b(u, r)|}, \quad u \in L, \quad (2.17)$$

the number of points divided by the total length of network in a disc of radius  $r$  centred on  $u$ . This is Borruso’s estimator (2.1), but now restricted to query locations  $u$  lying on the network. The Jones-Diggle correction estimator (2.15) reduces to

$$\widehat{\lambda}_{L,con}^{JD}(u) = \sum_{x_i \in b(u, r)} \frac{1}{|L \cap b(x_i, r)|}, \quad u \in L. \quad (2.18)$$

This corresponds to associating, with each data point  $x_i$ , a unit mass which is then uniformly spread over the part of the network lying within an Euclidean distance  $r$  of  $x_i$ .

Returning to the general case, the normalising factor  $c_L(u)$  in (2.16) is also very

closely related to the “network density” proposed by Borruso (2003),

$$B(u) = \frac{1}{m} \sum_{j=1}^m \kappa(u - v_j), \quad u \in \mathbb{R}^d, \quad (2.19)$$

where  $v_1, \dots, v_m$  are random sample points on the network. The function  $B(u)$  for  $u \in \mathbb{R}^2$  was used by Borruso (2003) in order to investigate the spatial density of the lines themselves. By Campbell’s formula (1.26), if the sample points are uniformly distributed,  $\mathbb{E}[B(u)] = c_L(u)/|L|$  for  $u \in L$ . Thus,  $c_L(u)$  or  $B(u)$  has the interpretation of a reference or baseline value that would be obtained if the data points were uniformly randomly distributed on the network.

The uniform correction estimator (2.14) could also be motivated by techniques for estimating spatially-varying relative risk in a case-control study (see Section 1.2.3 and 2.6.6). The data points  $x_i$  are viewed as the locations of cases, and a random sample of points uniformly distributed on the network will serve as the controls. The standard estimator of relative risk is the ratio of kernel estimates of probability density for the cases and for the controls. This is effectively the same as (2.14).

It is theoretically permissible to choose a kernel  $\kappa$  which is not isotropic, i.e. not invariant under rotation. This seems undesirable in practice, except in situations where the coordinate system is not isometric, such as the latitude–longitude coordinates on a globe. We shall assume that  $\kappa$  is isotropic.

Unlike estimators of intensity based on path distances in the network, the convolution estimators are robust against errors in the geometry of the network. If  $\kappa$  is uniformly continuous, the quantities (2.14), (2.15), (2.16) are continuous functions of the point pattern  $\mathbf{x}$  and the linear network  $L$  under the Wasserstein metric (Rüschendorf, 1994).

### 2.6.1 Fast computation

The sums in (2.14)–(2.15) and the integral (2.16) can be recognised as convolutions of the kernel  $\kappa$  with different measures  $M$  on  $\mathbb{R}^2$ . These may be computed rapidly using the FFT, using the rule (Pinsky, 2002, p. 13, 92)

$$\kappa * M = \mathcal{F}^{-1}(\mathcal{F}(\kappa) \cdot \mathcal{F}(M)), \quad (2.20)$$

where  $*$  denotes convolution,  $\mathcal{F}$  is the Fourier transform, and “ $\cdot$ ” is pointwise multiplication of functions.

The sum in (2.14) is the convolution of the kernel  $\kappa$  with the counting measure on data points (Silverman, 1982). It would usually be computed by discretising the point pattern  $\mathbf{x}$  onto a fine grid of pixels (where each pixel value equals the number of data points falling in the pixel), then applying (2.20). Similarly, the estimator (2.15) is the convolution of  $\kappa$  with the measure that puts mass  $1/c_L(x_i)$  on data point  $x_i$  for  $i = 1, \dots, n$ . A similar computational strategy can be used.

The denominator  $c_L(u)$  in (2.16) is the convolution of  $\kappa$  with arc length measure on the network. Here one can discretise the *network* onto a pixel grid, where each pixel value equals the *length* of network segments intersecting the pixel, and apply (2.20). To suppress artefacts such as aliasing and numerical underflow, Rakshit et al. (2018) recommends that pixel size should be smaller than one-tenth of the smoothing bandwidth.

Alternatively, for appropriate choices of the kernel,  $c_L(u)$  can be computed analytically. Decompose  $c_L(u)$  into contributions from each network segment

$$c_L(u) = \int_L \kappa(u - v) d_1 v = \sum_{j=1}^N \int_{l_j} \kappa(u - v) d_1 v = \sum_{j=1}^N c_{l_j}(u). \quad (2.21)$$

When  $\kappa$  is the two-dimensional isotropic Gaussian kernel with mean 0 and standard deviation  $\sigma$ , it can easily be calculated that

$$c_{l_j}(u) = \phi_\sigma(u_2) |\Phi_\sigma(s_j - u_1) - \Phi_\sigma(u_1)|, \quad (2.22)$$

where  $\phi_\sigma, \Phi_\sigma$  are, respectively, the probability density function and cumulative distribution function of the Gaussian  $N(0, \sigma^2)$  distribution,  $s_j = |l_j|$  is the length of the segment  $l_j$ , and  $(u_1, u_2)$  are the coordinates of  $u$  in a coordinate system in which the origin is an endpoint of  $l_j$  and the first coordinate axis is parallel to  $l_j$ . The function (2.22) can be evaluated rapidly at a fine grid of pixels  $u$ .

We implemented these procedures in the R language using the package **spatstat** (Baddeley and Turner, 2005; Baddeley et al., 2015). Computation of (2.14)–(2.16) was implemented in R (relying on existing implementations of the FFT) while the analytic computation (2.22) was implemented in C. Timings given in this Section (i.e. entire Section 2.6) refer to the implementation of (2.14)–(2.16) using byte-compiled R code (Tierney, 2001) and the **fftw** library for FFT computation (Frigo and Johnson, 2005; Karim, 2017).

## 2.6.2 Chicago example

Table 2.1 shows computation times for our proposed convolution algorithm (Definition 2.1) and the diffusion algorithm (Section 2.5) applied to the Chicago data for various values of the bandwidth. Computation time increases steeply with bandwidth for the diffusion algorithm but is roughly constant for the convolution algorithm. Note however that, for a given bandwidth  $\tau$ , the diffusion algorithm also provides kernel estimates for a finely-spaced sequence of bandwidths  $\eta$  with  $0 < \eta \leq \tau$ .

Comparison of the two methods is complicated by the fact that the diffusion bandwidth  $\tau$  and convolution bandwidth  $\sigma$  have different meanings. The bandwidth  $\tau$  for the diffusion method can be interpreted as a path distance on the network, while the bandwidth  $\sigma$  for the convolution method is a Euclidean distance in two-dimensional space. A rough conversion rule is needed for practical purposes. To investigate this, one may inspect a scatter plot of shortest-path distance against Euclidean distance for a random sample of points on the network. In our experience, an approximately proportional relationship between shortest-path distance  $d_L(u, v)$  and Euclidean distance  $\|u - v\|_2$  (index 2 to emphasize the dimension) holds for some kinds of networks, in particular for street networks. That is,

$$d_L(u, v) \approx \beta \|u - v\|_2 \quad (2.23)$$

for independent uniform random locations  $u$  and  $v$  on  $L$ , where  $\beta$  is the conversion factor. For the Chicago network we estimate  $\beta \approx 1.25$ . In any network we have  $\|u - v\|_2 \leq d_L(u, v)$  for all  $u, v$ . In a perfect grid composed of  $a \times b$  rectangles, it is easy to show that  $d_L(u, v) = \|u - v\|_1 + q(u, v)$ , where  $\|\cdot\|_1$  denotes the  $L^1$  norm and  $q(u, v) \leq 2(a + b)$ . Since  $\|u - v\|_1 \leq \sqrt{2}\|u - v\|_2$ , the approximation (2.23) with  $\beta = 1.25$  should be reasonable in any rectangular grid.

For general networks, we shall estimate the conversion factor  $\beta$  by generating a large sample of independent uniform random points on the network, evaluating the pairwise Euclidean distances and pairwise shortest-path distances, and fitting a proportional regression (2.23) with standard deviation proportional to the mean.

Figure 2.15 shows intensity estimates for the Chicago crimes using the diffusion and convolution methods, with comparable bandwidths 125 feet and 100 feet respectively. These are three-dimensional perspective views in the style of Okabe and Sugihara (2012) in which the function values are represented by the heights

Table 2.1: Computation time (in milliseconds) for the diffusion algorithm and the convolution algorithm applied to the Chicago data with different bandwidths  $\sigma$  or  $\tau$  (in feet).

$\sigma$ or $\tau$	50	100	150	200	250	300	350	400	450	500
$\widehat{\lambda}_L^H$	53	77	116	176	246	327	429	546	676	824
$\widehat{\lambda}_{L,con}^U$	44	41	41	41	41	42	44	42	42	41
$\widehat{\lambda}_{L,con}^{JD}$	41	41	41	42	43	41	41	41	41	38

of vertical walls erected above each segment of the network. While their overall appearance is very similar, the two estimates show some clear differences near the boundary, for example, on the eastern-most road and the north-west terminal nodes. The eastern-most road is geographically close to crime locations but the corresponding network path distance is much longer.

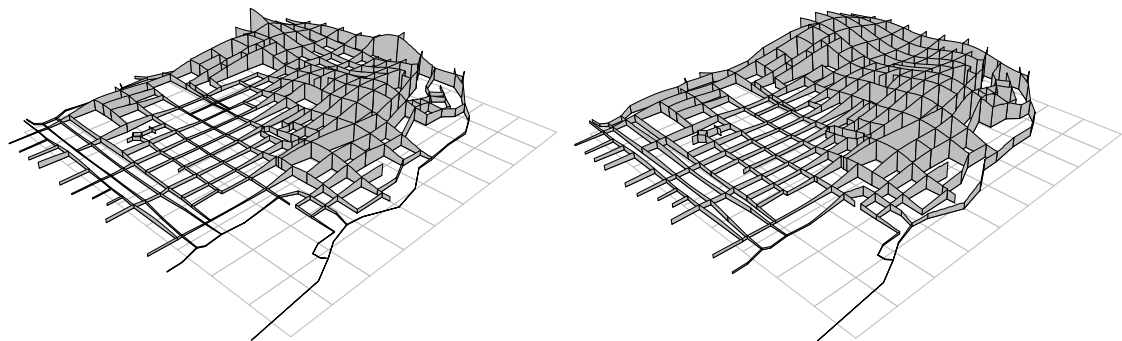


Figure 2.15: Kernel estimates of intensity for Chicago data. Perspective views with height representing the function value. *Left*: diffusion estimate with bandwidth 125 feet. *Right*: convolution method with bandwidth 100 feet and uniform correction.

## 2.6.3 Theoretical properties

### 2.6.3.1 Bias

If  $X$  is a point process on  $L$  with intensity function  $\lambda(u)$ ,  $u \in L$ , then by Campbell's formula (1.26) the estimators (2.14) and (2.15) have mean values

$$\mathbb{E}[\widehat{\lambda}_{L,con}^U(u)] = \frac{1}{c_L(u)} \mathbb{E} \sum_i \kappa(u - x_i) = \frac{\int_L \kappa(u - v) \lambda(v) d_1 v}{\int_L \kappa(u - v) d_1 v}, \quad (2.24)$$

$$\mathbb{E}[\widehat{\lambda}_{L,con}^{JD}(u)] = \int_L \frac{\kappa(u - v)}{c_L(v)} \lambda(v) d_1 v, \quad (2.25)$$

for  $u \in L$ . The right-hand sides can be regarded as smoothed versions of the intensity function. In particular if  $X$  has uniform intensity  $\lambda(u) \equiv \lambda > 0$ , then

$$\mathbb{E}[\widehat{\lambda}_{L,con}^U(u)] \equiv \lambda, \quad (2.26)$$

$$\mathbb{E}[\widehat{\lambda}_{L,con}^{JD}(u)] = \lambda \int_L \frac{\kappa(u - v)}{c_L(v)} d_1 v, \quad (2.27)$$

for  $u \in L$ , so that the uniform correction estimator (2.14) is pointwise unbiased, while the Jones-Diggle correction estimator (2.15) is not.

### 2.6.3.2 Variance

If  $X$  is a *Poisson* point process on  $L$  with intensity function  $\lambda(u)$ , then the general variance formula

$$\text{var} \left[ \sum_{i=1}^n h(x_i) \right] = \int_L h(u)^2 \lambda(u) d_1 u \quad (2.28)$$

holds for any function  $h : L \rightarrow \mathbb{R}$  for which the right-hand side is integrable. Hence the pointwise variances of the uniform correction estimator (2.14) and Jones-Diggle correction estimator (2.15) are, respectively,

$$\text{var} \left[ \widehat{\lambda}_{L,con}^U(u) \right] = \frac{1}{c_L(u)^2} \int_L \kappa(u - v)^2 \lambda(v) d_1 v, \quad (2.29)$$

$$\text{var} \left[ \widehat{\lambda}_{L,con}^{JD}(u) \right] = \int_L \frac{\kappa(u - v)^2}{c_L(v)^2} \lambda(v) d_1 v, \quad (2.30)$$

for  $u \in L$ . In particular, if  $X$  is a uniform Poisson process with intensity  $\lambda$ , then

$$\text{var} \left[ \widehat{\lambda}_{L,con}^U(u) \right] = \frac{\lambda}{c_L(u)^2} \int_L \kappa(u - v)^2 d_1 v, \quad u \in L, \quad (2.31)$$

$$\text{var} \left[ \widehat{\lambda}_{L,con}^{JD}(u) \right] = \lambda \int_L \frac{\kappa(u - v)^2}{c_L(v)^2} d_1 v, \quad u \in L. \quad (2.32)$$



The variance is not constant over the network, even if the true intensity is uniform.

The variances (2.29) and (2.30) can also be estimated unbiasedly from data  $\{x_1, \dots, x_n\}$  by (respectively)

$$\widehat{V}_U(u) = \frac{1}{c_L(u)^2} \sum_{i=1}^n \kappa(u - x_i)^2, \quad (2.33)$$

$$\widehat{V}_{JD}(u) = \sum_{i=1}^n \frac{\kappa(u - x_i)^2}{c_L(x_i)^2}, \quad (2.34)$$

for  $u \in L$ . These expressions can again be evaluated using FFT methods. This is another advantage of the convolution method: although analogous formulae exist for the variances of other intensity estimators, their computational cost is usually prohibitive. For the convolution method, computation of variance estimates (2.33) and (2.34) takes roughly the same time as computing the intensity estimates (2.14) and (2.15) themselves.

### 2.6.3.3 Asymptotics

Here we briefly sketch some asymptotic properties of our proposed intensity estimator, and its asymptotic equivalence with other estimators.

Consider the kernel with bandwidth  $\sigma > 0$  defined by  $\kappa_\sigma(u) = \sigma^{-2} \kappa_1(u/\sigma)$  for  $u \in \mathbb{R}^2$ , where  $\kappa_1$  is a given, fixed, probability density on  $\mathbb{R}^2$ . We assume  $\kappa_1(x)$  is continuous at  $x = 0$ . The estimators (2.14) and (2.15) computed using  $\kappa = \kappa_\sigma$  will be denoted by  $\widehat{\lambda}_\sigma^U$  and  $\widehat{\lambda}_\sigma^{JD}$  respectively.

#### Large bandwidth, fixed data

Suppose the point pattern dataset  $\mathbf{x}$  is fixed and the bandwidth  $\sigma$  increases. Then the convolution estimate (Definition 2.1) converges to a constant intensity on the entire network  $L$ .

To prove this, since  $\kappa_1$  is continuous at 0, as  $\sigma \rightarrow \infty$  we have  $\kappa_\sigma(x-y) \sim \sigma^{-2} \kappa_1(0)$  for any fixed  $x, y \in L$ . Hence  $\widehat{\lambda}_\sigma^U(u) \rightarrow \bar{\lambda}$  and  $\widehat{\lambda}_\sigma^{JD}(u) \rightarrow \bar{\lambda}$ , uniformly in  $u$ , where  $\bar{\lambda} = n/|L|$  is the average intensity over the network.

By comparison, the diffusion estimate (2.13) converges as  $\sigma \rightarrow \infty$  to a function which is constant on each connected component of the network (McSwiggan et al., 2017, Section 7.2).

### Increasing intensity, decreasing bandwidth

Standard asymptotic results for kernel density estimation on the infinite real line can be generalised to a linear network, adapting the approach of Botev et al. (2010). Let  $N \rightarrow \infty$  and suppose the true intensity is  $\lambda_N(u) = N\lambda_1(u)$  on  $L$ , where  $\lambda_1(u)$  is twice continuously differentiable. Assume the bandwidth  $\sigma_N$  satisfies  $\sigma_N \rightarrow 0$  and  $N\sigma_N \rightarrow \infty$ . Then adapting Botev et al. (2010, Theorem 1), for any location  $u$  that is not a node, the behaviour of  $\widehat{\lambda}_\sigma^U(u)/N$  and  $\widehat{\lambda}_\sigma^{JD}(u)/N$  is asymptotically equivalent to that which would occur if the network were an infinite straight line. Both estimators are asymptotically equivalent to the kernel smoother on the real line with (one-dimensional) kernel  $\zeta_\sigma(x) = \kappa_\sigma((0, x))/Z_1(\sigma)$ , where  $Z_1(\sigma)$  is the normalising constant  $Z_1(\sigma) = \int_{-\infty}^{\infty} \kappa_\sigma((0, x))dx$ . By change of variables  $Z_1(\sigma) = (1/\sigma)Z_1(1)$  so that  $\zeta_\sigma(x) = \sigma^{-1}\zeta_1(x/\sigma)$  in accordance with the usual scaling behaviour of a kernel in one dimension.

In the special case where the two-dimensional kernel  $\kappa_\sigma$  is the isotropic Gaussian density with standard deviation  $\sigma$  in each coordinate, the corresponding one-dimensional density  $\zeta_\sigma$  is the Gaussian density with standard deviation  $\sigma$ , and the normalised bias and variance of the convolution estimator  $\widehat{\lambda}^U(u)/N$  satisfy, for any fixed  $u \in L$ ,

$$\mathbb{E} \left[ N^{-1}\widehat{\lambda}_{L,con}^U(u) - \lambda_1(u) \right] = \frac{\sigma^2}{2} \frac{\partial^2 \lambda_1}{\partial u^2} + O(\sigma^4), \quad (2.35)$$

$$\text{var} \left[ N^{-1}\widehat{\lambda}_{L,con}^U(u) \right] = \frac{\lambda_1(u)}{2\sqrt{\pi}N\sigma} + o(1), \quad (2.36)$$

and similarly for  $N^{-1}\widehat{\lambda}_{L,con}^{JD}(u)$ . At a node  $v$  of degree  $m$ , the estimate  $\widehat{\lambda}_{L,con}^U(v)$  is equal to  $1/m$  times the sum of  $m$  asymptotically independent contributions with asymptotic bias and variance of the form (2.35)–(2.36), hence these asymptotics also hold at a node.

### 2.6.4 Toy example

It is instructive to consider a very small example. Figure 2.16 shows a simulated pattern of 4 points on a network which has total length 3 units and bounding box about 0.8 by 0.9 units.

Kernel estimates of the intensity for this pattern are shown in Figure 2.17. The left panel shows the diffusion kernel estimate (2.13) with bandwidth  $\tau = 0.225$  units.

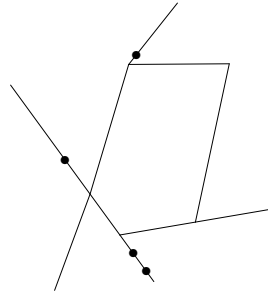


Figure 2.16: Toy example. Simulated point pattern of 4 points on a network of total length 3 units.

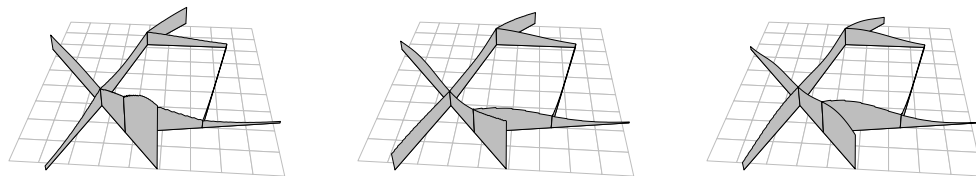


Figure 2.17: Kernel estimates of the intensity for the toy example. *Left*: diffusion estimate with bandwidth  $\tau = 0.225$  units. *Middle*: convolution estimate with uniform correction, bandwidth  $\sigma = 0.15$  units. *Right*: convolution estimate with Jones-Diggle correction, bandwidth  $\sigma = 0.15$  units. Perspective views with height representing the function value. Vertical scales are equal.

The middle and right panels show the convolution estimates using an isotropic Gaussian kernel with standard deviation 0.15 units (The bandwidths  $\tau = 0.225$  and  $\sigma = 0.15$  are comparable for reasons explained in Section 2.6.2). The middle panel is the uniform correction estimate (2.14) and the right panel is the Jones-Diggle correction estimate (2.15), using the same vertical scale. Maximum estimated intensity is 4.2 for the uniform correction and 3.4 for the Jones-Diggle correction.

These estimates behave differently near the “boundary” of the network. In particular, at a terminal endpoint of the network, the diffusion estimate always has zero slope (McSwiggan et al., 2017). For the convolution method, the uniform correction tends to inflate the intensity estimate near a terminal endpoint, while

the Jones-Diggle correction estimate tends to taper off more sharply.

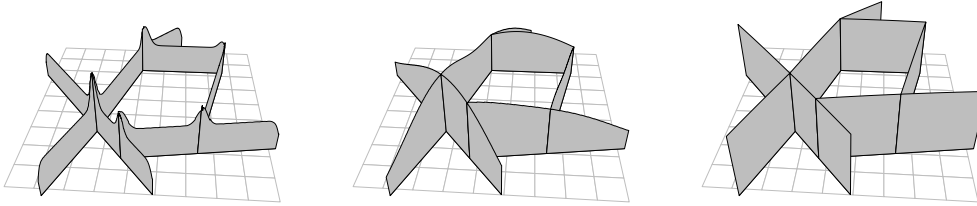


Figure 2.18: Edge correction denominator  $c_L(u)$  for the toy network of Figure 2.16 with bandwidth  $\sigma$  equal to 0.015, 0.15 and 1.5 units (left to right). Perspective views with height representing the function value, using different vertical scales.

Figure 2.18 shows the edge correction denominator  $c_L(u)$  defined in (2.16) evaluated for the toy example using the isotropic Gaussian kernel with three different bandwidths. For small values of  $\sigma$ , the function  $c_L(u)$  is roughly constant along each segment, with larger values near the nodes. For moderate to large values of  $\sigma$ , the function achieves a maximum near the centre of the network, and declines toward the periphery. For very large values of  $\sigma$  the function is roughly constant.

If we define the “boundary” of the network as consisting of locations  $u$  where  $c_L(u)$  takes its smallest values, then Figure 2.18 shows that the boundary may include segments and nodes of the network which are spatially extreme — say, which are close to the boundary of the convex hull of the network — as well as terminal nodes of the network.

The top left and top right panels of Figure 2.19 show the predicted variance of the uniform and Jones-Diggle corrections respectively, with smoothing bandwidth 0.15 units, and assuming a Poisson process with uniform intensity  $\lambda = 2$  (mean total number of points  $\lambda|L| = 6$ ). The variance was computed using equations (2.29) and (2.30). The uniform correction tends to inflate the variance near the boundary, and has larger maximum variance than the Jones-Diggle correction (5.4 against 3.2) but interestingly the two estimates have roughly equal integrated variance (6.6) so that overall the uniform correction has slightly smaller ISE (6.6 against 6.9). The pointwise mean square error of the Jones-Diggle estimate is shown in the bottom right panel of Figure 2.19.

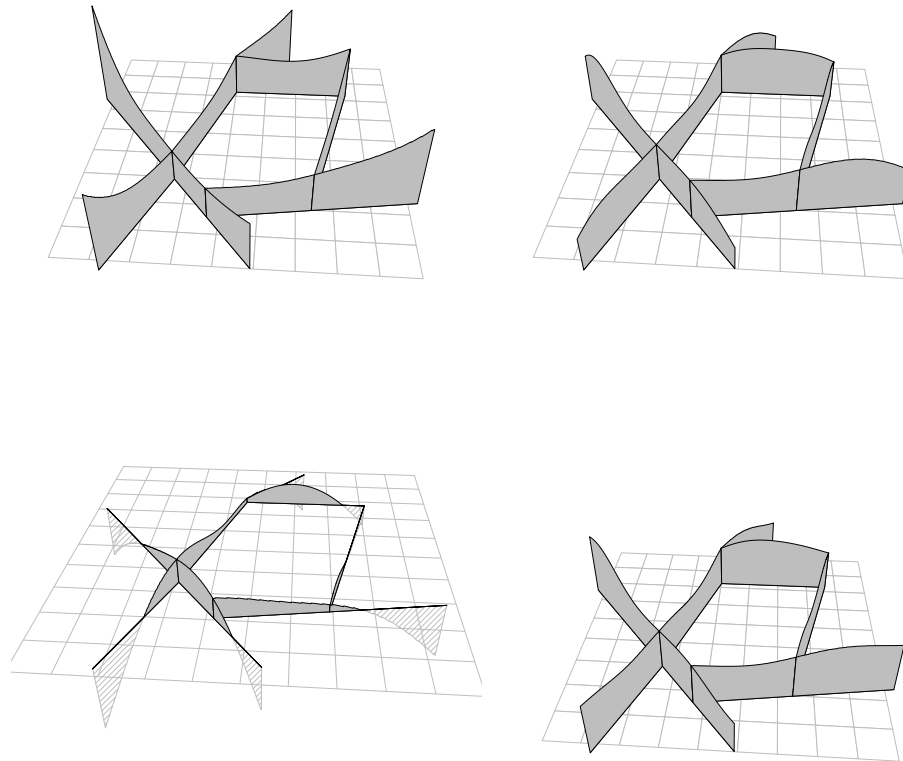


Figure 2.19: Predicted performance on the toy example. Assuming a uniform Poisson process with intensity 2, and kernel smoothing with bandwidth 0.15. *Top Left*: variance (= MSE) of the uniform correction estimator. *Top Right*: variance of the Jones-Diggle correction estimator. *Bottom Left*: bias of the Jones-Diggle correction estimator, with positive values shown by solid grey colour and negative values by diagonal shading. *Bottom Right*: MSE of the Jones-Diggle correction estimator. Variance and MSE panels use the same vertical scale.

The bottom left panel of Figure 2.19 shows the predicted bias of the Jones-Diggle correction estimator (2.15), assuming a uniform intensity  $\lambda = 2$ . The bias was calculated by expressing (2.27) as a convolution and using the FFT. The bias ranges from  $-1$  to  $+0.4$ ; it is large and negative near an isolated endpoint of the network.

## 2.6.5 Simulation experiments

Simulation experiments are necessary in order to compare the performance of our convolution method (2.14)–(2.15) with that of the diffusion method (2.13).

### 2.6.5.1 Description of experiments

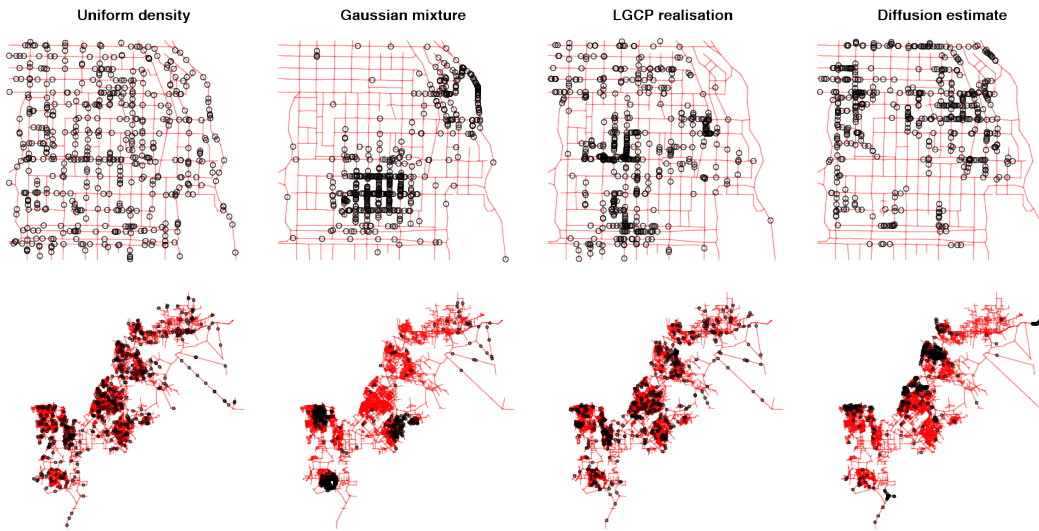


Figure 2.20: Typical simulated realisations for each of the eight scenarios. *Top row*: Chicago street network. *Bottom row*: southern part of the city of Perth, extracted from the Western Australian road network. The Gaussian mixture and LGCP realisation scenarios are based on an initial 2D surface defined on  $W$ . The diffusion estimate scenario is based on the original data observed on the relevant network. Simulated realisations all have size  $n = 500$ .

We simulated eight different scenarios, illustrated in Figure 2.20. In the top row, the network  $L$  is taken to be the street network from the Chicago data (Figure 2.2). In the bottom row,  $L$  is a subset of the Western Australian road

network (Figure 2.5) representing the southern part of the city of Perth, comprising 18,870 road segments.

In the first column of Figure 2.20, the intensity is constant on the network. In the second column, a mixture of  $K$  isotropic Gaussian probability densities ( $K = 2$  for Chicago;  $K = 4$  for southern Perth), with different variances, and with mean vectors lying inside the convex hull of the network, was chosen arbitrarily and restricted to the network. In the third column, a realisation of a stationary log-Gaussian random field was generated and restricted to the network. In the fourth column, an intensity function was computed by applying the diffusion method to the original point pattern data for the relevant network. Each intensity was scaled to integrate to an expected sample size of 500. These eight functions were then held fixed for the rest of the experiment.

For each given scenario with true intensity  $\lambda$  on  $L$ , we generated 100 simulated realisations of a Poisson process with intensity  $\lambda$ . For each realisation we computed the intensity estimates  $\hat{\lambda}_{L,con}^U$  and  $\hat{\lambda}_{L,con}^{JD}$  for several bandwidths  $\sigma$ , and computed the diffusion estimate  $\hat{\lambda}_L^H$  over the corresponding bandwidths  $\tau = \beta\sigma$ , where  $\beta$  is the conversion factor in (2.23). We used  $\beta \approx 1.25$  for the Chicago network and  $\beta \approx 1.42$  for the southern Perth network. Pixel images were computed with resolution  $128 \times 128$  for the Chicago network and  $2048 \times 2048$  for the southern Perth network to ensure sufficient accuracy. Performance was measured by computing, for each estimate  $\hat{\lambda}$ , the integrated squared error ISE, and by noting the computation time.

### 2.6.5.2 Details of simulation scenarios

For the scenarios based on the Chicago network, the data are contained in the rectangle  $W = [0.4, 1281.9] \times [153.1, 1276.6]$  feet. Scenario 2 (Gaussian mixture) is the restriction to the network of a mixture of 2 isotropic Gaussian densities plus a small additive constant. The two Gaussian densities have equal mixture probabilities of  $p = 0.5 - 0.05/|W|$  each and the additive constant is  $1 - 2p$ . Their parameters  $(\mu_1, \mu_2, \sigma)$  are equal to  $(500, 500, 100)$  and  $(1000, 1000, 100)$  feet. Scenario 3 is the restriction to the network of a log-Gaussian random field with zero log mean and an exponential correlation function with variance and scale parameters 2 and 100 respectively. Scenario 4 is the result of diffusion kernel smoothing of the original data with bandwidth 50 feet. All intensity functions were

normalised to integrate to 500 on  $L$ .

For the scenarios based on South Perth, the network  $L$  is the subset of Western Australia data in the rectangle  $W = [382.269, 438.7022] \times [6417.982, 6480.644]$  km. Segments of length zero were removed. The resulting network is connected, and has 18873 segments, 15043 vertices, total length 3259.7 km, and maximum node degree 8. Scenario 2 (Gaussian mixture) is the restriction to the network of a mixture of 4 isotropic Gaussian densities plus a small additive constant. The four Gaussian densities have equal mixture probabilities of  $p = 0.25 - 0.1/|W|$  each and the additive constant is  $1 - 4p$ . Their parameters  $(\mu_1, \mu_2, \sigma)$  are equal to  $(430, 6470, 5)$ ,  $(410, 6445, 2)$ ,  $(390, 6430, 1)$  and  $(388, 6448, 1)$ . Scenario 3 is the restriction to the network of a log-Gaussian random field with zero log mean and an exponential correlation function with variance and scale parameters 2.8 and 0.3 respectively. Scenario 4 is the result of diffusion kernel smoothing of the subset of the WA crashes data that falls on  $L$  with bandwidth 1.75 km. All intensity functions were normalised to integrate to 500 on  $L$ .

All simulation results are based on 100 independently generated realisations of the Poisson process for each unique scenario.

### 2.6.5.3 Results

Figure 2.21 shows statistical performance of the estimators. Boxplots show the spread of ISE values for the diffusion estimates computed from 100 simulated realisations, and are plotted against the diffusion kernel bandwidth  $\tau$  indicated at the top of each panel. Curves show the mean integrated squared error MISE for the convolution method, computed by evaluating the analytic formulae for bias and variance in Section 2.6.3, and plotted against the convolution kernel bandwidth  $\sigma$  indicated at the bottom of each panel. We verified that these analytic formulae accurately predicted the sample means of ISE obtained in the simulations.

For the scenarios with uniform intensity, shown in the top row of Figure 2.21, MISE decreases with increasing bandwidth. This is expected, since the optimal estimate is constant, which is obtained as the bandwidth approaches infinity. The uniform correction estimator  $\hat{\lambda}_{L,con}^U$  has the best empirical performance, as expected.

For the other three scenarios, performance is optimised at an intermediate value of the bandwidth. This is the familiar tradeoff between bias and variance. For the



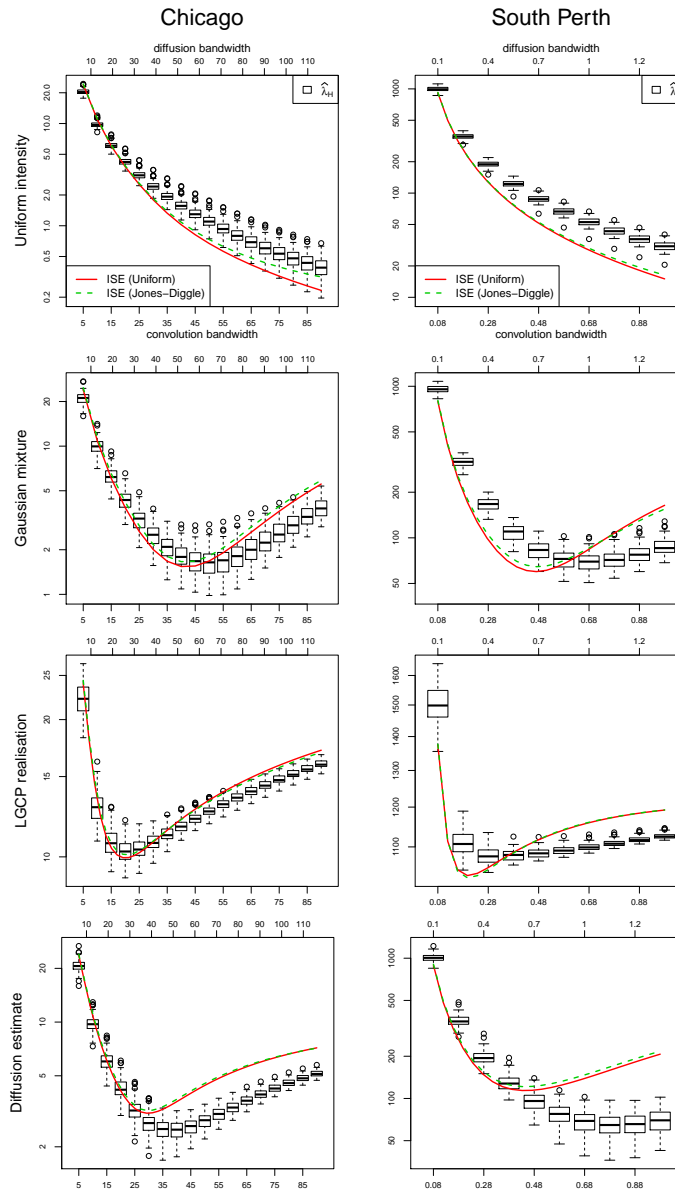


Figure 2.21: Integrated squared error (ISE) of the convolution and diffusion estimators applied to simulated data. *Left column*: Chicago network; *Right column*: southern Perth network. *Rows* represent the four scenarios in Figure 2.20. The bottom horizontal axis in each panel shows the bandwidth  $\sigma$  of the convolution estimator; the top horizontal axis shows the bandwidth  $\tau$  of the diffusion estimator. *Boxplots* show the numerically computed ISEs for the diffusion estimator. *Lines* show the theoretically calculated ISEs for  $\hat{\lambda}_{L,con}^U$  (red, solid) and  $\hat{\lambda}_{L,con}^{JD}$  (green, dashed). Bandwidths are in feet for Chicago, and in km for southern Perth.

convolution method, this tradeoff can be quantified using the analytic formulae for bias and variance in Section 2.6.3.

In the Gaussian mixture and LGCP scenarios, the competing methods have roughly equal performance overall. For large bandwidths, the diffusion method has better performance than the convolution method. Interestingly, this advantage disappears near the optimal bandwidth, where the two methods have roughly equal MISE. This suggests that, provided we do not greatly over- or under-smooth, the convolution estimators are capable of performing as well as the diffusion method. The uniform correction has a slight advantage over the Jones-Diggle correction.

The diffusion estimate scenario shows results that clearly favour the diffusion estimator  $\hat{\lambda}_L^H$  for large bandwidths. This is unsurprising since the behaviour of the diffusion estimate near a terminal vertex precisely matches the behaviour of the “true” intensity surface in this synthetic problem. For nearly optimal bandwidths, there is no great difference in performance.

#### 2.6.5.4 Computation times

For the convolution method, execution times were roughly constant as a function of the bandwidth, and were roughly independent of the spatial pattern of points. Execution times were around 0.06 sec to produce a  $128 \times 128$  pixel image for the Chicago network, and around 20 seconds for a  $2048 \times 2048$  image for the southern Perth network.

In both networks, execution time for  $\hat{\lambda}_L^H$  is comparable to that for  $\hat{\lambda}_{L,con}^U$  and  $\hat{\lambda}_{L,con}^{JD}$  for small to moderate bandwidths, but rises steadily with increasing bandwidth, reaching 0.2 sec and 80 seconds in Chicago and southern Perth respectively at the largest bandwidth considered. This is emphasised by Figure 2.22 which shows mean execution times of  $\hat{\lambda}_L^H$ , relative to the uniform and Jones-Diggle convolution estimators, across all eight scenarios.

#### 2.6.6 Relative risk and smoothing on a network

Here we sketch how the convolution method can be generalised for the purpose of estimating spatially-varying relative risk and for smoothing numerical responses observed at sample points on a network.

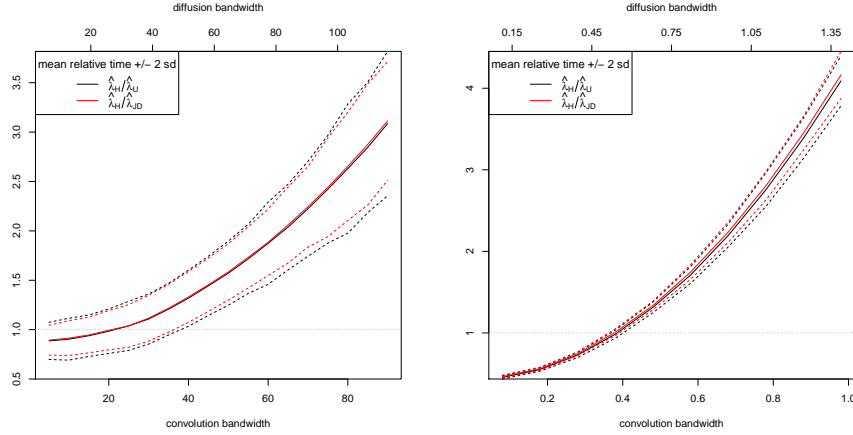


Figure 2.22: Mean relative execution time per estimate for  $\hat{\lambda}_L^H$  relative to both  $\hat{\lambda}_{L,con}^U$  and  $\hat{\lambda}_{L,con}^{JD}$  across all four scenarios for the Chicago and southern Perth networks (*left and right* respectively).

### 2.6.6.1 Weighted kernel estimators

Numerical weights  $w_i$  associated with the data points  $x_i$  can be incorporated in the kernel estimators (2.14)–(2.15) giving

$$\hat{\lambda}_{L,con}^U(u) = \frac{1}{c_L(u)} \sum_{i=1}^n w_i \kappa(u - x_i), \quad u \in L, \quad (2.37)$$

$$\hat{\lambda}_{L,con}^{JD}(u) = \sum_{i=1}^n w_i \frac{\kappa(u - x_i)}{c_L(x_i)}, \quad u \in L. \quad (2.38)$$

Weighted kernel estimators serve many purposes. The weight may represent multiplicity: for example if  $w_i$  is the number of vehicles involved in the traffic accident that occurred at location  $x_i$ , then the weighted intensity estimate is the spatially-varying number of vehicles involved in accidents per unit length. Weights may also be used to adjust for a denominator: if  $w_i = 1/v_i$ , where  $v_i$  is the traffic volume (average number of cars passing this location per hour), then the weighted intensity estimate is a spatially-varying accident risk (accidents per car–km).

### 2.6.6.2 Relative risk

The literature of relative risk is briefly reviewed in Section 1.2.3. In this Section and using the convolution method, we provide non-parametric estimators of the network version of (1.17)

A standard approach is to estimate  $\rho(\cdot)$  using a plug-in estimator based on kernel estimators of  $\lambda_X$  and  $\lambda_Y$ . It would be possible to use different smoothing bandwidths for the two point patterns, although there are theoretical arguments for using equal bandwidths (Kelsall and Diggle, 1995a). Recent research suggests that unequal bandwidths can cause undesirable “halo” artefacts in the estimate of  $\rho(\cdot)$  (Davies et al., 2016), at least in adaptive estimation. Hence we recommend using the same bandwidth  $\sigma$  to estimate both  $\lambda_X$  and  $\lambda_Y$ . If the uniform correction (2.14) is used, then the plug-in estimate of  $\rho(\cdot)$  is

$$\hat{\rho}^U(u) = \log \frac{\sum_i \kappa_\sigma(u - x_i)}{\sum_j \kappa_\sigma(u - y_j)}, \quad (2.39)$$

in which the edge correction factor  $c_L(u)$  has cancelled out (Hazelton, 2008). This is equivalent to the usual uniform-correction estimate of relative risk in two-dimensional space, restricted to locations on the network. For the Jones-Diggle correction (2.15) the plug-in estimate of  $\rho(\cdot)$  is

$$\hat{\rho}^{JD}(u) = \log \frac{\sum_i \kappa_\sigma(u - x_i)/c_L(x_i)}{\sum_j \kappa_\sigma(u - y_j)/c_L(y_j)}. \quad (2.40)$$

Bandwidth selection for the optimal estimation of relative risk is different from bandwidth selection for intensity estimation. For spatial relative risk in two dimensions, several techniques for bandwidth selection are canvassed by Diggle (2003); Kelsall and Diggle (1995a); Diggle et al. (2005); Lawson and Williams (1993); Hazelton and Davies (2009); Davies et al. (2018) and summarised in Davies et al. (2018, Section 4.2). These can be adapted immediately to a linear network: we do not discuss them here.

For statistical inference, Hazelton and Davies (2009) introduced asymptotic  $p$ -value surfaces for relative risk. These could also be adapted to the network setting.

### 2.6.6.3 Spatial smoothing

Suppose we observe numerical responses  $z_1, \dots, z_n$  at a set of sample locations  $x_1, \dots, x_n$ , and seek to estimate the spatially-varying mean response. In the Nadaraja-Watson approach to smoothing (Nadaraya, 1964; Watson, 1964; Nadaraya, 1989), the estimate of the mean response  $Z(u)$  at a location  $u$  is a weighted average of the observed responses  $z_i$  at nearby points  $x_i$ , weighted by

the smoothing kernel. For our convolution method, the uniform and Jones-Diggle corrections give respectively

$$\widehat{Z}^U(u) = \frac{\sum_i z_i \kappa(u - x_i)}{\sum_i \kappa(u - x_i)}, \quad (2.41)$$

$$\widehat{Z}^{JD}(u) = \frac{\sum_i z_i \kappa(u - x_i) / c_L(x_i)}{\sum_i \kappa(u - x_i) / c_L(x_i)} \quad (2.42)$$

for locations  $u \in L$ . These estimates can be calculated rapidly using the techniques described in Section 2.6.1. Bandwidth selection for spatial smoothing can be based on minimising the sum of squared errors of interpolation  $S = \sum_i [z_i - \widehat{Z}^{-i}(x_i)]^2$ , where  $\widehat{Z}^{-i}(x_i)$  is the leave-one-out estimate of  $Z(x_i)$  obtained by omitting terms associated with  $x_i$  from both numerator and denominator of (2.41)–(2.42).

### 2.6.7 Traffic accidents on urban roads of Medellín

The locations of traffic accidents in the urban area of Medellín, Colombia, in 2016 are shown in Figure 2.4 which were published in the OpenData portal of Medellín Town Hall<sup>2</sup>. Details of the dataset are provided in Section 2.2.3.

We first estimate the intensity of each type of accident using the convolution estimate with the uniform correction (2.14). For these individual intensity estimates, bandwidths were selected using the extension of Scott's rule of thumb (see Section 1.2.2.1 and Rakshit et al. (2018)), yielding bandwidths of 0.36, 0.36 and 0.67 km for property damage, personal injury and fatal accidents, respectively. For comparison, likelihood cross-validation yielded bandwidths of 0.24, 0.24 and 1.07 km respectively. Figure 2.23 below shows the estimated intensity of each type of accident using the convolution estimate with the uniform correction (2.14). Due to the complexity of the road network, the intensity values are now rendered as colours rather than line thicknesses or vertical heights. Each panel in Figure 2.23 is a  $500 \times 500$  pixel image and required computation time of about 9 seconds. Using the default resolution  $128 \times 128$  would have reduced the computation time to 1.3 sec.

Figure 2.24 shows plug-in estimates of relative risk for different types of accidents, relative to accidents which caused only property damage. Due to the complexity of the road network, the relative risk values are now rendered as colours rather than vertical heights. The estimates were computed by taking ratios of intensity estimates with the uniform correction (2.14), using the same bandwidth

<sup>2</sup><https://www.medellin.gov.co/geomedellin/>

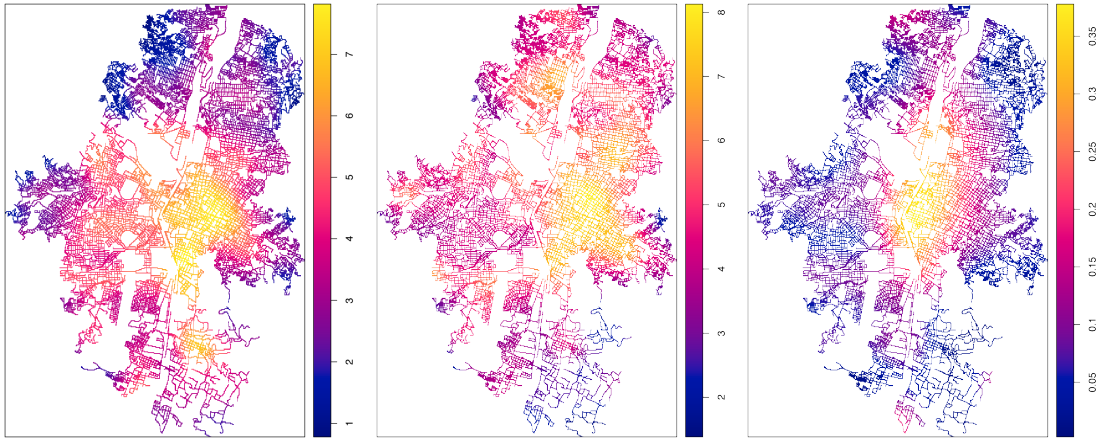


Figure 2.23: Estimated intensity functions for each type of accident in Medellín data, using convolution estimator and uniform correction (2.14). *Left*: property damage, bandwidth 0.36 km; *Middle*: personal injury, bandwidth 0.36 km; *Right*: fatal, bandwidth 0.67 km. Intensity values are reported as accidents per km.

$\sigma$  for numerator and denominator. For estimation of relative risk,  $\sigma$  was selected using the weighted mean integrated squared error criterion of Hazelton (2008), available in the R package **sparr** (Davies et al., 2018). This yielded bandwidths of 0.715 km for the relative risk of personal injury to property damage, and 0.88 km for the relative risk of fatality to property damage. Each panel in Figure 2.24 is a  $500 \times 500$  pixel image and required computation time of about 20 seconds. Using the default resolution  $128 \times 128$  would have reduced the computation time to 3 seconds.

The left panel of Figure 2.24 shows the relative risk of accidents involving personal injury relative to accidents involving property damage. The average relative risk is  $6004/4627 \approx 1.3$ . Estimated relative risks which are 3 times higher than average occur in the northern part of the urban area. The right panel shows the relative risk of fatal accidents to those involving property damage (average relative risk  $133/4627 \approx 0.029$ ). Estimated relative risks almost 3 times higher than average occur in central and northern areas.

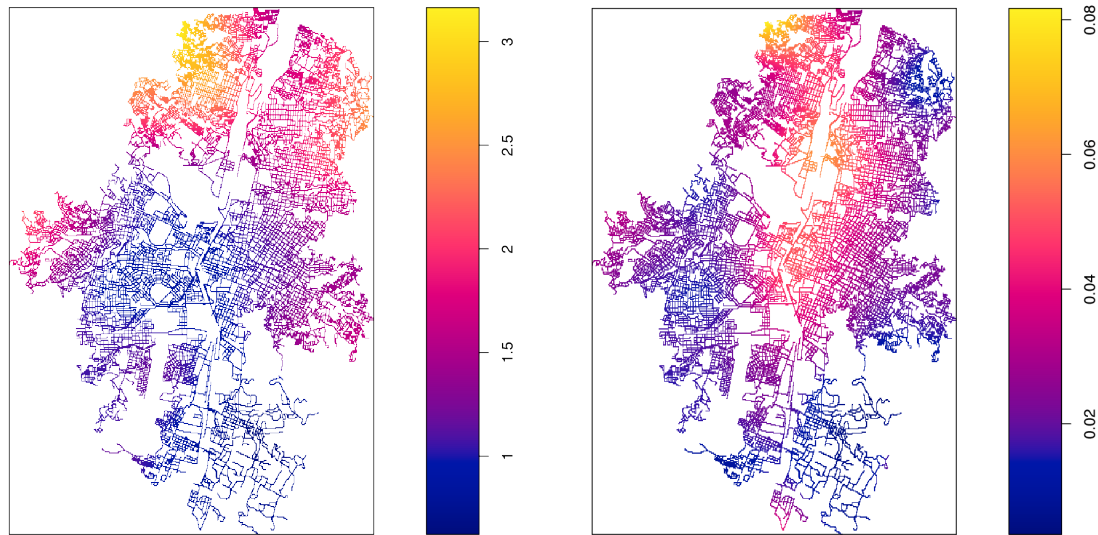


Figure 2.24: Estimated relative risk of different types of accidents, relative to accidents which caused only property damage. *Left*: personal injury; *Right*: fatality.

## 2.6.8 Adaptive smoothing

The Western Australian road accident data shown in Figure 2.5 exhibit huge spatial variation in the concentration of data points. In such situations it is well known that fixed-bandwidth kernel estimation can perform poorly. Dense concentrations of accidents in the urban areas will be over-smoothed, and sparse accidents in the remote desert will be under-smoothed. Adaptive (variable-bandwidth) kernel estimation can perform substantially better in this context (Abramson, 1982; Hall and Marron, 1988).

A novel complication is that the network itself has huge spatial variation. This makes it more difficult to judge by eye the density of accidents per unit length of road. It also causes computational difficulties, since a much finer spatial resolution is required in urban areas than in the remote desert.

A pragmatic approach might be to divide the state of Western Australia into different sub-regions which are then analysed separately. We show one example in Section 2.6.9.3. To avoid artefacts at the boundaries between sub-regions, one could instead construct a spatially-varying bandwidth function  $\sigma(u)$ ,  $u \in L$  and

estimate the intensity by

$$\widehat{\lambda}^u(u) = \frac{1}{c_L(u, \sigma(u))} \sum_{i=1}^n \kappa_{\sigma(x_i)}(u - x_i), \quad u \in L, \quad (2.43)$$

analogous to the uniform correction (2.14), although it does not retain the same unbiasedness property. Estimators of the form (2.43) in 2D were proposed by Marshall and Hazelton (2010).

An alternative is to allow each data point  $x_i$  to have its own smoothing bandwidth  $\sigma_i$ , and estimate the intensity by

$$\widehat{\lambda}^{\text{JD}}(u) = \sum_{i=1}^n \frac{\kappa_{\sigma_i}(u - x_i)}{c_L(x_i, \sigma_i)}, \quad u \in L, \quad (2.44)$$

analogous to the Jones-Diggle correction (2.15), where

$$c_L(u, \sigma) = \int_L \kappa_{\sigma}(u - v) d_1 v, \quad u \in L. \quad (2.45)$$

This estimator conserves mass, i.e.  $\int_L \widehat{\lambda}^{\text{JD}}(u) d_1 u = n$ . Notice that (2.43) and (2.44) are actually the network version of (1.14) and (1.15).

Direct computation of either (2.43) or (2.44) would be very costly. For computational efficiency we can follow the partitioning strategy of Davies and Baddeley (2018, Section 4) in which continuously-varying bandwidths  $\sigma$  are mapped to discretised bandwidths  $\sigma^* = h(\sigma)$  where the function  $h$  takes only the values  $h_1, \dots, h_m$  with  $m \ll n$ .

For the Jones-Diggle style estimate (2.44), let  $\sigma_i^* = h(\sigma_i)$  be the discretised bandwidth associated with data point  $x_i$ . Then  $\widehat{\lambda}^{\text{JD}}(u)$  is approximated by

$$\widehat{\lambda}^{\text{JD}^*}(u) = \sum_{i=1}^n \frac{\kappa_{\sigma_i^*}(u - x_i)}{c_L(x_i, \sigma_i^*)}, \quad u \in L. \quad (2.46)$$

This can be computed by dividing the point pattern  $\mathbf{x}$  into sub-patterns  $\mathbf{x}^{(1)}, \dots, \mathbf{x}^{(m)}$  according to the value of the discretised bandwidth, computing the fixed-bandwidth kernel estimate for each sub-pattern, and summing the results

$$\widehat{\lambda}^{\text{JD}^*}(u) = \sum_{j=1}^m \sum_{i: \sigma_i^* = h_j} \frac{\kappa_{h_j}(u - x_i)}{c_L(x_i, h_j)} = \sum_{j=1}^m \widehat{\lambda}_{h_j}^{\text{JD}}(u \mid \mathbf{x}^{(j)}), \quad u \in L, \quad (2.47)$$



where  $\widehat{\lambda}_\sigma^{JD}(u | \mathbf{y})$  is the Jones-Diggle corrected estimate (2.15) with bandwidth  $\sigma$  computed for point pattern  $\mathbf{y}$ . Total computation time for (2.47) is about  $m$  times the cost of the fixed-bandwidth Jones-Diggle corrected estimate.

For the uniform-style estimate (2.43), the discretised approximation to  $\widehat{\lambda}^U(u)$  is

$$\widehat{\lambda}^{U*}(u) = \frac{1}{c_L(u, h(\sigma(u)))} \sum_{i=1}^n \kappa_{\sigma^*(x_i)}(u - x_i), \quad u \in L. \quad (2.48)$$

To compute this, we again partition the point pattern  $\mathbf{x}$  into sub-patterns according to the value of  $\sigma_i^* = h(\sigma(x_i))$ , but evaluate the *uncorrected* fixed-bandwidth estimate for each sub-pattern

$$\widehat{\lambda}_{h_j}(u | \mathbf{x}^{(j)}) = \sum_{i:\sigma_i^*=h_j} \kappa_{h_j}(u - x_i), \quad u \in L, \quad (2.49)$$

then sum these estimates and normalise

$$\widehat{\lambda}^{U*}(u) = \frac{1}{c_L(u, \sigma^*(u))} \sum_{j=1}^m \widehat{\lambda}_{h_j}(u | \mathbf{x}^{(j)}), \quad u \in L. \quad (2.50)$$

Total computation time for (2.50) is again about  $m$  times the cost of the fixed-bandwidth estimate. The approximation (2.48)–(2.50) could produce artefacts associated with abrupt transitions in the value of  $\sigma^*(u)$ . These artefacts could be avoided by using the three-dimensional FFT technique for adaptive estimation proposed by Davies and Baddeley (2018), instead of partitioning the bandwidths.

It remains to choose the bandwidths  $\sigma_i$  associated with each data point, or the bandwidth function  $\sigma(u)$  for locations  $u \in L$ . Abramson (1982) argues that the optimal smoothing bandwidth should be inversely proportional to the square root of the true density at the given location. This can be approximated using a pilot estimate of the intensity. The pilot estimate could be a fixed-bandwidth intensity estimate, or another type of intensity estimate that can be calculated rapidly.

Given a pilot estimate of the intensity  $\widetilde{\lambda}(x_i)$  for each data point  $x_i$ , the prescription of Abramson (1982) is to compute initial bandwidths

$$a_i = \left( \frac{\widetilde{\lambda}(x_i)}{n} \right)^{-1/2}, \quad (2.51)$$

then to derive the smoothing bandwidths

$$\sigma_i = \sigma^\# \frac{a_i}{a}, \quad (2.52)$$

where  $a = (\prod_i a_i)^{1/n}$  is the geometric mean of the initial bandwidths, and  $\sigma^\#$  is the *global bandwidth*. Bandwidth selection for the adaptive case mainly involves selection of the value of  $\sigma^\#$ .

The adaptive estimation of *relative risk* was introduced by Davies and Hazelton (2010), who also developed asymptotic tolerance contours. Davies et al. (2016) argued that the adaptive risk estimator should be symmetrised to use common bandwidth factors, extending an argument of Kelsall and Diggle (1995a) to the adaptive case. These results can be extended to the case of a linear network.

## 2.6.9 Traffic accidents in Western Australia

Figure 2.5 shows the spatial locations of 14,562 road traffic accidents recorded for the year 2011 on the road network of the State of Western Australia. The data were provided by the Western Australian state government department of Main Roads, and are made available for publication as part of the Western Australian Whole of Government Open Data Policy. The road network was simplified from its original 626,031 segments to 115,169 segments, with a total length of approximately 97,165 km. Duplicate accident locations were removed because these are typically separate records of each of the vehicles involved in a multiple-vehicle collision.

### 2.6.9.1 Fixed bandwidth estimation

We first use the fixed-bandwidth estimators (2.14) and (2.15) to estimate the intensity of accidents on the network. Figure 2.25 shows the cross-validation criterion  $cv(\sigma)$  plotted against bandwidth  $\sigma$  for both estimators. Optimal bandwidths are  $\sigma = 9.1$  km for the uniform-corrected estimator and 10.9 km for the estimator with Jones-Diggle correction. The modified Scott rule gives  $\sigma = 8.9$  km and would have been acceptable in practice.

Figure 2.26 shows the fixed-bandwidth uniform correction estimate of accident intensity for the state of Western Australia using  $\sigma = 9.1$  km. Figure 2.27 plots the fixed-bandwidth intensity estimate using the Jones-Diggle correction (2.15) with the optimal bandwidth 10.9 km. A logarithmic colour scale is used here in order to retain visual detail, because the intensity estimates vary by 20 orders of magnitude across the image. To avoid artefacts, intensity values falling below a threshold of  $10^{-5}$  accidents per km (corresponding to an average of one accident across the entire network) have been replaced by this threshold.

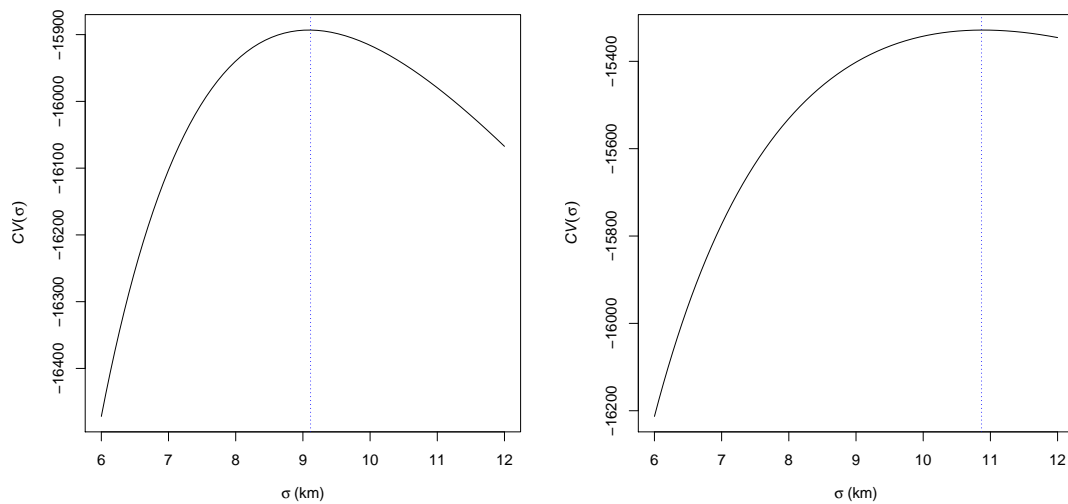


Figure 2.25: Likelihood cross-validation criterion  $cv(\sigma)$  plotted against bandwidth  $\sigma$  for the kernel estimators of intensity of traffic accidents on the Western Australian road network. *Left*: uniform-corrected estimator; the vertical line shows the optimum at  $\sigma = 9.1$  km. *Right*: Jones-Diggle estimator; the vertical line shows the optimum at  $\sigma = 10.9$  km.

The two corrections give very similar estimates. Both corrections show the effect of over-smoothing in the urban areas along the west coast, and under-smoothing in the remote east. These are common features of fixed bandwidth estimates applied to spatially heterogeneous patterns (Loftsgaarden and Quesenberry, 1965; Breiman et al., 1977; Davies and Baddeley, 2018).

In the remote eastern areas where no accidents have been recorded, some intensity estimates are close to zero and may be negative due to numerical error. This occurs more frequently with the uniform correction. Zero values are quite likely if the kernel  $\kappa$  has bounded support.

### 2.6.9.2 Adaptive bandwidth estimation

It is evident from Figure 2.5 that accidents are highly concentrated in one area, the location of the capital city of Perth. This contains more than 75% of the state's population, while the remote eastern part of the state has very low population density and traffic volume. Although the road network itself is also much denser in Perth than elsewhere, the estimated accident rate per kilometre shown in

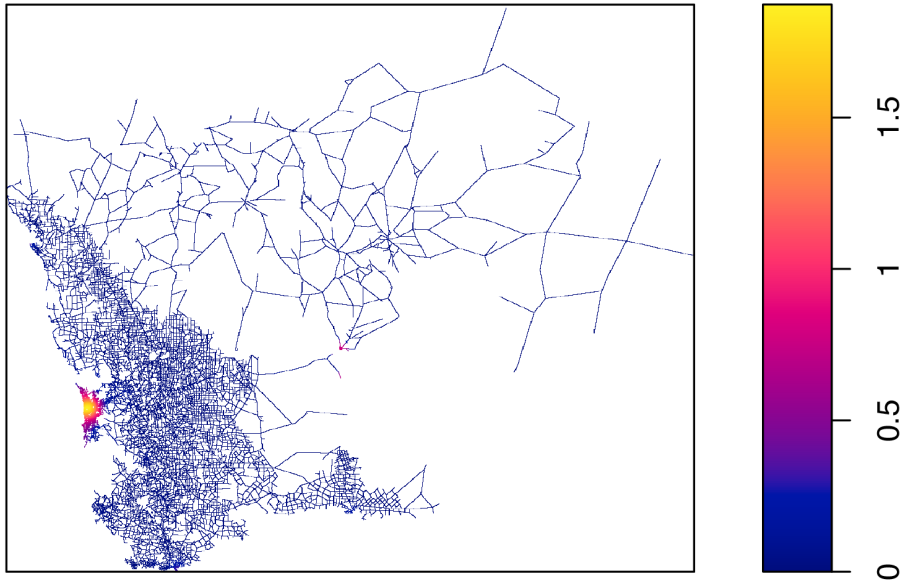


Figure 2.26: Fixed-bandwidth estimate of intensity for the accidents on the Western Australian road network using the uniform correction with  $\sigma = 9.1$  km. Intensity values are accidents per km.

Figures 2.26 and 2.27 is still much higher in Perth. Fixed-bandwidth estimators are not suitable for analysing such spatially heterogeneous point patterns.

Figure 2.28 shows the adaptive estimate of intensity for the Western Australian road accidents using the Jones-Diggle style estimator (2.44). The pilot estimate of the intensity was taken to be the fixed-bandwidth Jones-Diggle corrected convolution estimate (2.15) with bandwidth  $\sigma = 10.9$  km shown in Figure 2.27. Adaptive bandwidths  $\sigma_i$  for each data point  $x_i$  were then derived using Abramson's rule (2.51)–(2.52) with global bandwidth  $\sigma^\# = 10.9$  km. Values of  $\sigma_i$  ranged between 7.5 and 171 km. The adaptive bandwidths  $\sigma_i$  were divided into 30 groups of equal size, and the adaptive intensity estimate was computed using the discrete approximation (2.46) in the form (2.47).

The Figure 2.28 shows that the adaptive kernel-smoothed estimates have removed under-smoothing in the remote east, while retaining slightly more detail of the high intensity in the west.

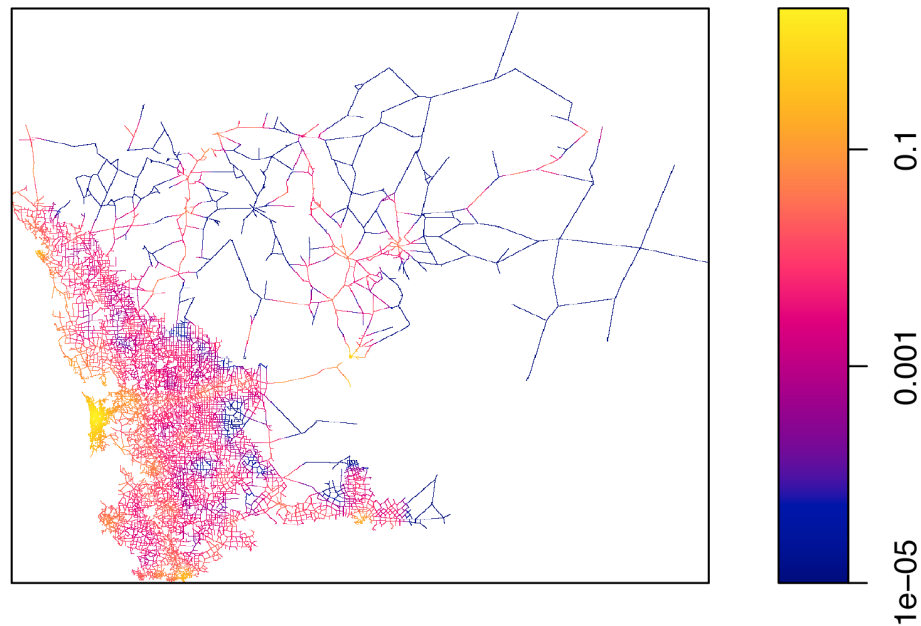


Figure 2.27: Fixed-bandwidth estimate of intensity for the accidents on the Western Australian road network using the Jones-Diggle correction with  $\sigma = 10.9$  km. Logarithmic colour map. Intensity values are accidents per km.

### 2.6.9.3 Perth metropolitan area

Figure 2.29 shows accidents recorded in the Perth metropolitan area, defined as the rectangle  $[372, 431] \times [6434, 6501]$  km. There are 12408 accident locations; the network has a total length of 10318.8 km; the average accident density is 1.20 points per km. Figure 2.30 shows an adaptive estimate of accident intensity for the metropolitan Perth area (Figure 2.29), which is actually a  $60 \times 67$  km subregion of Figure 2.5. The pilot estimate was a fixed-bandwidth Jones-Diggle estimate with a bandwidth of 0.35 km selected by cross-validation. Adaptive bandwidths ranged from 0.1 to 1.1 km. The Figure 2.30 reveals that the accident data contain exquisite detail about the urban road system, including the main freeways and their tributaries, the central business district and the coastal town of Fremantle.

### 2.6.9.4 Computation times

Table 2.2 reports the time taken to compute intensity estimates for the Western Australian accident data using the diffusion algorithm, the fixed bandwidth con-

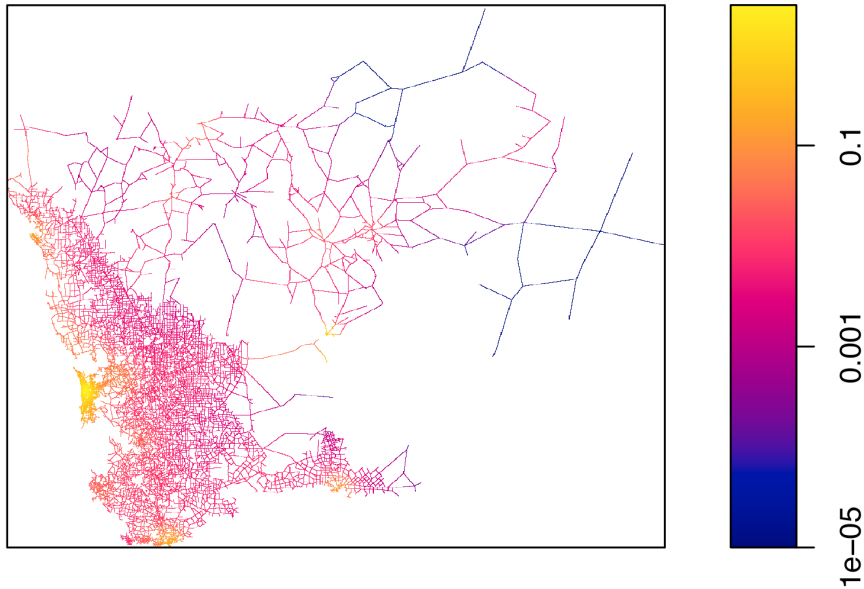


Figure 2.28: Adaptive-bandwidth intensity estimate for the accidents on the Western Australian road network using Jones-Diggle correction. Logarithmic colour map. Intensity values are accidents per km.

volution algorithms with uniform and Jones-Diggle corrections, and the adaptive bandwidth algorithm corresponding to the Jones-Diggle style correction.

All computations were performed on a 2.4 GHz Intel Core i7 laptop with 8 GB RAM using byte-compiled R code. The Fast Fourier Transform was computed using the `fftw` library.

As expected, the computation time for the diffusion estimator increased with bandwidth  $\tau$ , while computation time for the convolution estimators was not dependent on bandwidth  $\sigma$ .

Timings depend crucially on algorithm parameters, including the spatial resolution of the resulting pixel image (pixel width  $\epsilon$ ), the step-size  $\delta$  of the sample points along the linear network for the diffusion algorithm, and the number  $m$  of discretised bandwidths in the adaptive estimator. These experiments generated a  $512 \times 512$  pixel image so that  $\epsilon \approx 2.8$  km.

In the case of the Western Australian accidents, the effect of  $\delta$  is crucial. Heuristically, computation time for the diffusion algorithm should be roughly proportional to  $\delta^{-3}$  for small  $\delta$ , and this is confirmed by Table 2.2. Six hours of computation was

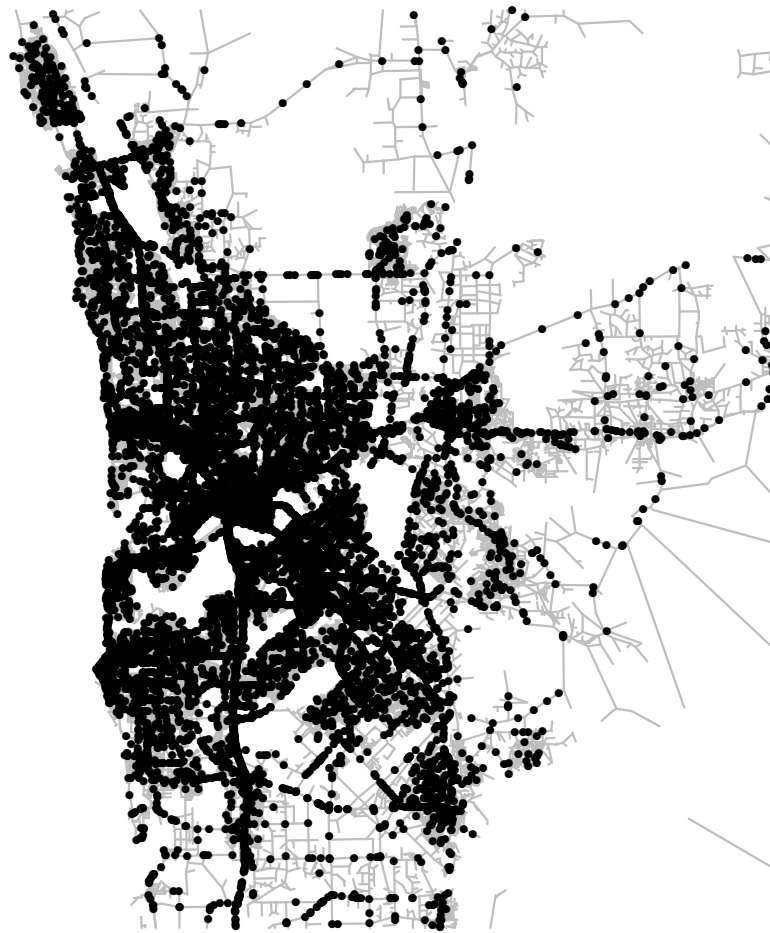


Figure 2.29: Accidents recorded in the Perth metropolitan area.

required in one case. Our implementation of the diffusion algorithm chooses  $\delta = \epsilon$  by default, and for this choice, the diffusion estimator is as fast as the convolution estimators. However, this also causes substantial bias in the diffusion estimates, because the Western Australian network contains many urban road segments shorter than  $\delta = 2.8$  km, and the diffusion algorithm approximates these segment lengths by rounding upward to a multiple of  $\delta$ . A reasonably accurate estimate is obtained only when  $\delta \leq 0.05$  km, but this inflates the computation time by several orders of magnitude.

Computation of the fixed bandwidth convolution estimates took approximately 12 seconds, irrespective of  $\sigma$ . For adaptive estimation, computation time ranged from 2 to 8 minutes, increasing with the number of discretised bandwidths  $m$ .

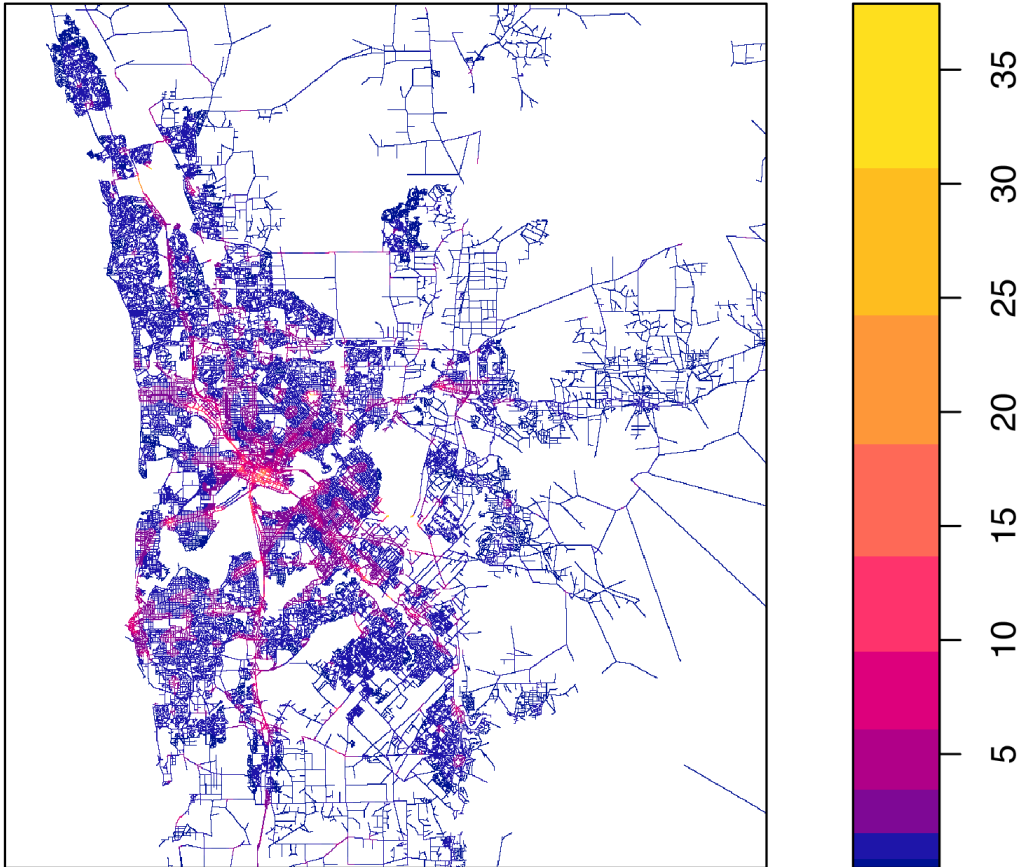


Figure 2.30: Adaptive-bandwidth intensity estimate for the accidents in metropolitan Perth using Jones-Diggle correction. Linear colour map with gamma-corrected colour sequence. Intensity values are accidents per km. Map is 60 km wide.



Table 2.2: Computation time (in *minutes*) of the diffusion algorithm for different step-sizes (in km), the adaptive algorithm for different number of bins  $m$ , and the convolution algorithm (U = uniform correction; J = Jones-Diggle correction), applied to the Western Australian accident data with different bandwidths  $\tau$  or  $\sigma$  (in km).

		$\tau$ or $\sigma$ (km)				
	step size $\delta$ (km)	6.0	8.0	10.0	12.0	14.0
diffusion	0.05	60	107	165	240	330
	0.10	8	14	22	30	42
	0.15	2.5	4.2	6.5	9.2	12.5
	0.20	1.2	2.0	3.0	4.1	5.5
	0.25	0.7	1.1	1.6	2.2	3.0
	0.30	0.5	0.8	1.1	1.4	1.8
	1.0	0.2	0.2	0.2	0.2	0.2
	2.0	0.2	0.2	0.2	0.2	0.2
	3.0	0.2	0.2	0.2	0.2	0.2
convolution (U)		0.2	0.2	0.2	0.2	0.2
convolution (J)		0.2	0.2	0.2	0.2	0.2
adaptive	$m = 10$	2.0	2.0	2.1	2.2	2.2
	$m = 50$	7.1	7.1	7.3	6.8	6.1
	$m = 100$	8.7	9.1	9.1	8.8	8.5

### 2.6.9.5 Accidents in the Perth Central Business District

Figure 2.31 shows accidents recorded in the Central Business District of the city of Perth, defined as the rectangle  $[389.2845, 394.8112] \times [6462.688, 6466.778]$  km. There are 1355 accident locations; the network has a total length of 251.9 km; the average accident density is 5.4 points per km.

Figure 2.32 shows a fixed-bandwidth estimate of the intensity using the uniform correction, with cross-validated bandwidth  $\sigma = 0.091$  km or about 90 metres. The main north-south freeway stands out clearly.

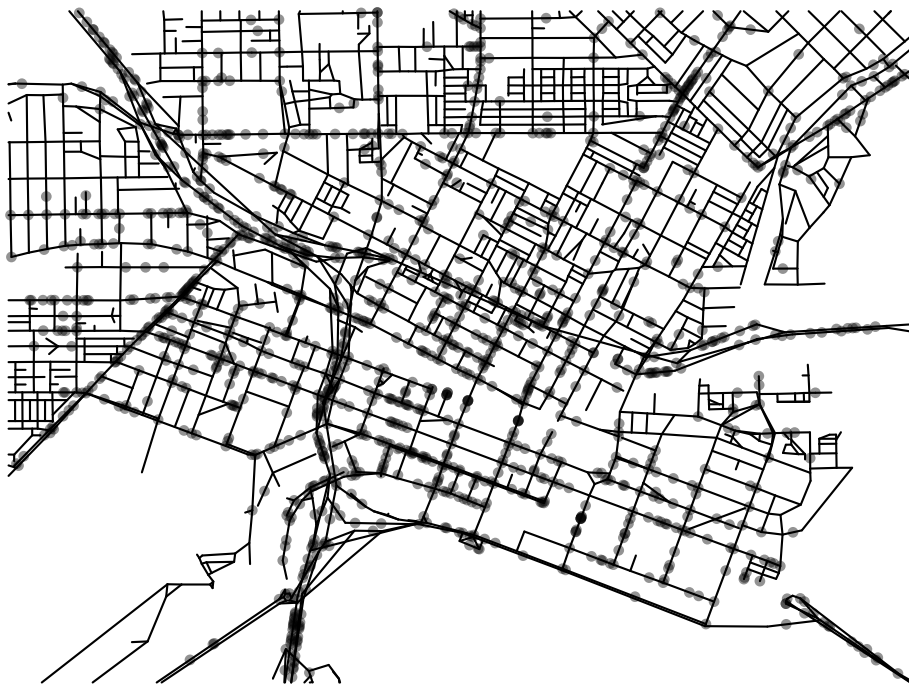


Figure 2.31: Accidents recorded in the Central Business District of the city of Perth.

Figures 2.33–2.35 below show several alternative styles of plotting the same intensity values that are depicted in Figure 2.32.

In Figure 2.33, line segments are drawn with variable width, proportional to the intensity function value. Figure 2.34 is a perspective view in the style of Okabe and Sugihara (2012) in which the function values are represented by the heights of vertical walls erected above each segment of the network.

Figure 2.35 shows two panels representing the fixed-bandwidth and adapt-



Figure 2.32: Intensity estimate in the Perth CBD using fixed-bandwidth uniform correction with automatically selected bandwidth  $\sigma = 0.091$  km. Intensity values mapped to colours.

ive estimates respectively as three-dimensional surfaces. We refer to these as “heightened network” (HEN) plots. They are similar in style to those of Borruso (2008), in that they portray a function defined on the two-dimensional plane as a surface viewed in perspective. In our case, the function is simply the extension of the estimator (2.14), (2.15), (2.43) or (2.44) to all locations  $u \in \mathbb{R}^2$ , which is an intermediate result in the convolution method calculations. The surface height is proportional to this function value; the surface colour also represents the function value; and the network itself is overlaid onto the surface.

Figure 2.35 is a screenshot of an interactive 3D graphics tool which can be viewed at [http://www.stats.otago.ac.nz/~tdavies/wacbd\\_hen.html](http://www.stats.otago.ac.nz/~tdavies/wacbd_hen.html). The plots can be controlled using the mouse; left-click and hold to rotate, scroll up/down to zoom in/out.

The left panel of Figure 2.35 represents the fixed-bandwidth estimate described above. The right panel is an adaptive estimate from the same data computed using the partitioning approach detailed in Section 2.6.8 (using the fixed bandwidth estimate as the pilot, and setting the global bandwidth at the same value used;  $\sigma^\# = 0.091$  km). Characteristically, we see a smoother adaptive estimate in areas



Figure 2.33: Intensity estimate in the Perth CBD using fixed-bandwidth uniform correction with automatically selected bandwidth  $\sigma = 0.091$  km. Intensity values are proportional to line width.

of relatively low point density when compared to the fixed bandwidth estimate, with taller peaks than the fixed bandwidth estimate in the most dense areas.

#### 2.6.9.6 Findings

Intensity estimates for the Western Australian road accident data range from 0 to 40 accidents per km. The highest intensity values occur in the main conurbations and along major highways.

The intensity of accidents is a measure of accident *frequency* (number of accidents per km) rather than *risk* (probability of an accident occurring to a given vehicle at this location). The intensity estimates may be useful for planning emergency response, but need further analysis if they are to be used for road safety research. Estimation of accident risk would require a denominator such as the traffic volume (number of vehicles passing the given location per hour) and explanatory variables such as speed limit.

In this analysis the choice of graphical display method for the intensity estimate was quite influential. The intensity values could have been represented as vertical

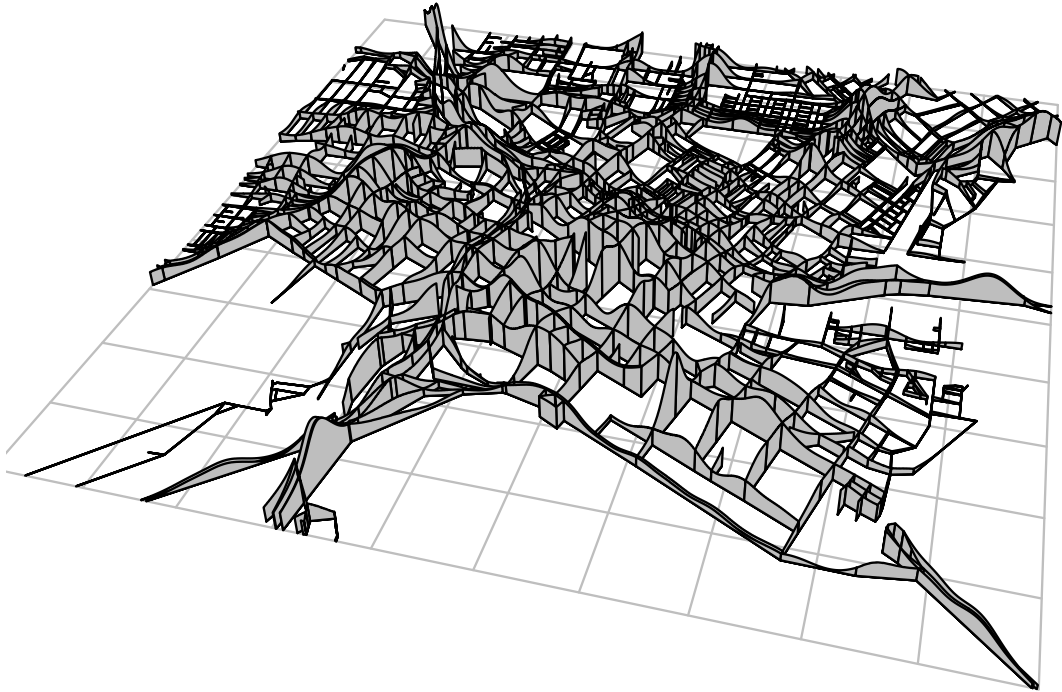


Figure 2.34: Perspective view of fixed-bandwidth uniform correction estimate.

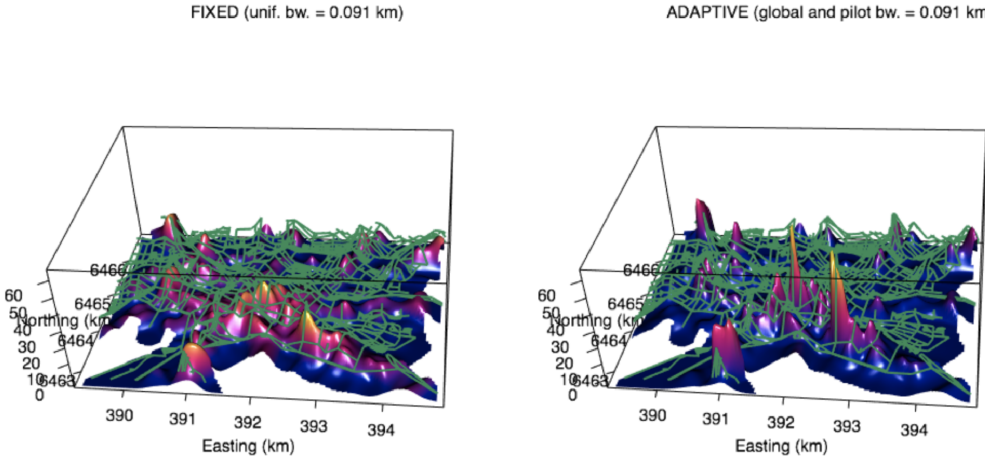


Figure 2.35: A screenshot of the fixed and adaptive intensity estimates of the Perth CBD data shown as interactive HEN plots. Accessible at the URL noted in the text.

heights (e.g. Figure 2.15), colours (with different kinds of colour map shown in e.g. Figure 2.24), line thicknesses (e.g. Figure 2.33) or an interactive 3D graphic (e.g. Figure 2.35). Naturally, the best choice regarding visualisation depends on the complexity of the network and the variability in the estimated intensity values.

Instead of removing duplicate points which refer to the same accident, an alternative would have been to count the number of vehicles involved in each accident, and retain this as an attribute of the accident location. This would have enabled separate study of single- and multiple-vehicle accidents. The number of vehicles could also serve as a weight associated with each location: the weighted kernel estimator would provide the spatially-varying number of *vehicles involved in accidents* per unit length, as discussed in Section 2.6.6.1.

## 2.7 Summary

In this Chapter, we reviewed the current kernel intensity estimators for spatial point patterns on linear networks and also proposed some new intensity estimators.

In Sections 2.3 and 2.4, we reviewed the equal-split intensity estimators (Okabe et al., 2009; Okabe and Sugihara, 2012) and the adapted Jones-Diggle estimator (Moradi et al., 2017) and their theoretical properties are also studied. A simulation study confirms a better performance for adapted Jones-Diggle estimator (2.7) by Moradi et al. (2017). We also applied the estimator (2.7) to a street crime data in Chicago, US and a dataset of anti-social behaviour in Castellón, Spain in which their spatially varying behaviour is represented in Figures 2.12 and 2.14.

The diffusion kernel estimator proposed by McSwiggan et al. (2017) was reviewed in Section 2.5 and later compared with the convolution method (definition 2.1) by (Rakshit et al., 2018). The convolution method is well defined in Section 2.6 and its theoretical properties such as bias, variance and asymptotics are studied. It is very fast to compute for any value of bandwidth. There is an exact, tractable variance formula and a variance estimator which can easily be computed. The calculation of leave-one-out estimates is also easy. Estimation can easily be confined to a subregion of space. Compared to other techniques which depend on path lengths in the network, the convolution method is much less sensitive to changes or errors in the connectivity of the network. This may be either an advantage or disadvantage in different contexts. It may be considered inappropriate to allow mass to “tunnel” between parts of the network which are

not interconnected; a similar issue arises in two-dimensional density estimation (Barry and McIntyre, 2011). However this property also makes it much easier to estimate spatio-temporal variation when the network itself is changing over time: for example, urban road networks are continually being modified. We have also analysed two big datasets of traffic accidents in Medellín, Colombia and Western Australia, represented in Figures 2.4 and 2.5 respectively.

In general, the kernel smoothing methods need to deal with bandwidth selection, which it is often selected based on some assumptions such as being Poisson (cross-validation), stationary Cox, etc. Therefore, in Chapter 3, we discuss Voronoi intensity estimators which are fully non-parametric and also propose an additional smoothing to improve their general performance.





## CHAPTER 3

---

# Resample-smoothing of Voronoi intensity estimators

---

### 3.1 Introduction

In point pattern analysis (van Lieshout, 2000; Diggle, 2003; Chiu et al., 2013; Baddeley et al., 2015), exploratory investigation often starts with non-parametric analysis of the spatial intensity of points. The intensity function, which is a first order moment characterisation of the point process assumed to have generated the data, reflects the abundance of points in different regions and may be seen as a “heat map” for the events (see Chapter 2). For most datasets, it is not realistic to assume that the underlying point process is homogeneous, i.e. that its intensity function is constant; rather it is natural to start by assuming inhomogeneity.

The most prominent approach to non-parametric intensity estimation is undoubtedly kernel estimation that is widely discussed in Section 1.2.2.1 and in Chapter 2. The degree of smoothing is controlled by a smoothing parameter, bandwidth, and the resulting estimates heavily depend on the choice of bandwidth. A small bandwidth may result in under-smoothing whereas a large bandwidth might result in over-smoothing of the intensity. Data-based procedures for bandwidth selection have been studied extensively (Diggle, 1985; Silverman, 1986; Berman and Diggle, 1989; Scott, 1992; Wand and Jones, 1994; Jones et al., 1996; Loader, 1999) including some recent advances by Cronie and van Lieshout (2018) (see Section 1.2.2.1). A further problem with kernel estimation is that, if there are wide

variations in intensity across the spatial domain, it may be impossible to find a single fixed bandwidth value which is satisfactory for smoothing every part of the spatial domain. Consequently the bandwidth must be spatially-varying, giving rise to a spatially “adaptive” kernel estimator (Diggle, 2003; Davies and Hazelton, 2010; Davies et al., 2016; Davies and Baddeley, 2018) at the cost of increased complexity (see Section 1.2.2.2). Recently there has been increasing interest in point patterns on linear networks (Okabe and Sugihara, 2012; Ang et al., 2012; Baddeley et al., 2015; Rakshit et al., 2018). Kernel smoothing of network events was already studied in Chapter 2.

As a consequence of underlying causes such as demography and human mobility, it is quite common to encounter sharp boundaries between high and low concentrations of events. For example, street crimes and traffic accidents tend to happen in particularly busy streets, which may be surrounded by quiet neighbourhoods. The classical kernel estimation approach is often unsuitable for such types of data.

In this Chapter, echoing Barr and Schoenberg (2010), we argue that kernel-based approaches may be unsatisfactory when there are sharp boundaries between parts with high and low intensity. Fixed bandwidth kernel smoothing results in over-smoothing in high-intensity areas, under-smoothing in low-intensity areas, and a blurring of sharp boundaries (Baddeley et al., 2015). By using a spatially adaptive kernel estimator we may reduce such problems when estimating the intensity function, but optimal bandwidth selection becomes even more challenging and important (Davies and Hazelton, 2010).

As an alternative, one could consider an approach without any choice of tuning parameters, e.g. a tessellation-based approach (van Lieshout, 2012; Schaap, 2007). One such approach is provided by Voronoi intensity estimation (Ord, 1978; Barr and Schoenberg, 2010; Okabe and Sugihara, 2012), defined such that within a given Voronoi cell of the point pattern the intensity estimate is set to the reciprocal of the size of that cell (Okabe et al., 2000). When employing the Voronoi intensity estimator, one thing that quickly becomes evident is that it often accentuates local features too much, in particular in regions with high event density. This reflects a previously observed phenomenon: adaptive estimators, such as the Voronoi intensity estimator, may smooth too little whereas kernel estimators may smooth too much in dense regions (Baddeley et al., 2015, Section 6.5.2). Hence, one should be able to find some middle ground and we here aim at providing a

contribution to that.

Section 4.2 provides the necessary preliminaries and we propose the resample-smoothing of intensity estimators in Section 3.3. A simulation study is drawn in Section 3.4 which assesses the performance of the proposed methodology numerically. Traffic accident in an area of Houston, US and location of trees in a Finnish forest are also analysed in Sections 3.5 and 3.6, respectively.

## 3.2 Setup

Throughout this Chapter we consider spatial point processes on general state space  $S$  assumed to be a complete separable metric space with distance metric  $d(\cdot, \cdot)$ , unless we specifically consider  $\mathbb{R}^2$  or a linear network  $L$ . We further assume there is a reference measure  $A \mapsto |A|$  for  $A \subseteq S$ , which is sigma-finite and locally finite.

### 3.2.1 Independent thinning

A key ingredient in our smoothing technique is independent thinning (Chiu et al., 2013, Section 5.1): given some measurable retention probability function  $p(u) \in (0, 1]$ ,  $u \in S$ , we run through the points of the point process  $X$  and delete a point  $x \in X$  with probability  $1 - p(x)$ , independently of the deletions carried out for the other points of  $X$ . The resulting thinned process has intensity

$$\lambda_{th}(u) = p(u)\lambda(u), \quad u \in S,$$

where  $\lambda(\cdot)$  is the intensity of the original process  $X$  (Chiu et al., 2013, Chapter 5.1). For further details on the thinning of point processes, see e.g. Møller and Schoenberg (2010) and Daley and Vere-Jones (2008, Section 11.3.).

It is worth mentioning that a Poisson process stays Poissonian after independent thinning (Daley and Vere-Jones, 2008, Exercise 11.3.1) and, in addition, the independent thinning of an arbitrary point process  $X$  with low retention probability results in a point process which, from a distributional point of view, is approximately a Poisson process (Baddeley et al., 2015, Section 9.2.2).

### 3.2.2 Voronoi tessellations

The next key ingredient in our estimation scheme is the Voronoi/Dirichlet tessellation of a point pattern  $\mathbf{x} = \{x_1, \dots, x_n\} \subseteq W \subseteq S$  (Chiu et al., 2013; Okabe et al., 2000) which we briefly reviewed in Section 1.2.2.3. Generally speaking, a tessellation of  $W$  is a tiling such that i) the union of all tiles constitutes all of  $W$ , and ii) the interiors of any two tiles have empty intersections.

The Voronoi/Dirichlet cell  $\mathcal{V}_x$  associated with  $x \in \mathbf{x}$  consists of all  $u \in S$  which are closer to  $x$  than any  $y \in \mathbf{x} \setminus \{x\}$ , i.e.

$$\mathcal{V}_x = \mathcal{V}_x(\mathbf{x}, W) = \{u \in W : d(x, u) \leq d(y, u) \text{ for all } y \in \mathbf{x} \setminus \{x\}\}, \quad (3.1)$$

and the tiling  $\{\mathcal{V}_x\}_{x \in \mathbf{x}}$  is termed the Voronoi/Dirichlet tessellation generated by  $\mathbf{x}$ . Clearly, the shape of each  $\mathcal{V}_x$  depends on the distance  $d(\cdot, \cdot)$  chosen for  $S$  and its size,  $|\mathcal{V}_x|$ , depends on the chosen reference measure  $|\cdot|$ . Note that (3.1) is the general version of (1.16).

### 3.2.3 Voronoi intensity estimation

In practice, it is often the case that events mainly occur in specific parts of the study region, e.g. that accidents often happen in more crowded streets or on specific parts of a highway, or that trees tend to grow mainly in specific parts of a forest. In other words, there are sharp boundaries between parts with high and low intensities. We argue, similarly to Barr and Schoenberg (2010) and Ogata (2011), that in order not to blur such boundaries, it is preferable to employ an adaptive intensity estimation scheme, which adapts locally to changes in the spatial distribution of the events.

Here we focus on a particular kind of adaptive intensity estimator, the Voronoi estimator, defined as follows.

**Definition 3.1.** *For a point process  $X$  with intensity function  $\lambda(\cdot)$ , the Voronoi intensity estimator of  $\lambda(u)$ ,  $u \in W \subseteq S$ ,  $|W| > 0$ , is given by*

$$\hat{\lambda}^V(u) = \hat{\lambda}^V(u; X, W) = \sum_{x \in X \cap W} \frac{\mathbf{1}\{u \in \mathcal{V}_x\}}{|\mathcal{V}_x|} = \sum_{x \in X \cap W} \frac{\mathbf{1}\{u \in \mathcal{V}_x(X, W)\}}{|\mathcal{V}_x(X, W)|}, \quad u \in W, \quad (3.2)$$

where  $\mathcal{V}_x$  is the Voronoi cell defined in (3.1). If  $X \cap W = \emptyset$  then  $\hat{\rho}^V(u) = 0$ .

It should be noted that the points of  $X$  which lie outside  $W$  may interact with those inside  $W$ . Indeed, due to the way we define the Voronoi cells in expression (3.1), the Voronoi intensity estimator neglects possible edge effects.

The Voronoi intensity estimator, which was introduced by Brown (1965) and Ord (1978) in the context of Euclidean spaces has been also considered by Baddeley (2007), Ogata (2011), Barr and Schoenberg (2010), and van Lieshout (2012). In the context of linear networks, Okabe and Sugihara (2012) discussed a Voronoi based density estimator, which they referred to as the network Voronoi cell histogram, for the purpose of non-parametric density estimation on linear networks. They further discussed geometric properties of Voronoi tessellations on linear networks. Barr and Schoenberg (2010) focused on the planar case and particular statistical properties.

### 3.3 Resample-smoothing of intensity estimators

Barr and Schoenberg (2010) pointed out that when there are abrupt changes in intensity, kernel-based estimators may yield substantial bias and high variance, and they showed that the Voronoi estimator can alleviate these problems. Unfortunately, Voronoi estimators tend to under-smooth in very dense areas surrounded by nearly empty neighbourhoods. This may be said about adaptive estimators in general; there is a tendency of adapting too much to the particular features of the observed point pattern  $\mathbf{x}$ , rather than reflecting the features of the intensity function of the underlying point process  $X$ . To see how the under-smoothing, i.e. the over accentuating of local features of the Voronoi intensity estimator occurs, note that for a pattern  $\mathbf{x}$ , if  $x \in \mathbf{x}$  is located in a very dense part then its Voronoi cell becomes small and, consequently,  $\hat{\lambda}^V(u) = 1/|\mathcal{V}_x|$  becomes very large for  $u \in \mathcal{V}_x$ . A further issue with the Voronoi intensity estimator is that its variance tends to be quite large, thus resulting in quite unreliable estimates.

One may further ask whether there are data-dependent tessellations  $\{\mathcal{C}_i\}$ ,  $\bigcup_i \mathcal{C}_i = W$ , giving rise to estimators  $\hat{\lambda}(u) = \sum_i \beta_i \mathbf{1}\{u \in \mathcal{C}_i\}$ ,  $\beta_i > 0$ , which perform better than the Voronoi intensity estimator. In addition, an advantage of the kernel estimation approach is arguably in that it generates a smoothly varying intensity estimate, at least when using certain kernels, as opposed to the possibly unnatural “jumps” generated by the Voronoi estimator.

As a remedy for these issues, one suggestion is to follow Barr and Schoenberg

(2010) by considering the so-called centroidal Voronoi intensity estimator. A further idea is to introduce a smoothing procedure for  $\widehat{\lambda}^V(\cdot)$ , which would reduce the unnaturally extreme peaks while smoothing out the “jumps”. We next propose such a smoothing procedure, which we refer to as *resample-smoothing*.

### 3.3.1 Definition of Resample-Smoothing

Recall the independent thinning operation in Section 3.2.1. We will here focus on the simple case where  $p(u) \equiv p \in (0, 1]$ ,  $u \in L$ , which is referred to as  $p$ -thinning (Chiu et al., 2013, Section 5.1); we identify the case  $p = 1$  with the unthinned process  $X$ . From Section 3.2.1 we have that

$$\lambda(u) = \frac{\lambda_{th}(u)}{p}, \quad u \in S,$$

where we recall the intensity  $\lambda_{th}(\cdot)$  of the thinned process  $X_p$ . Hence, dividing by  $p$  is exactly what is needed to compensate for the reduced intensity caused by removing points.

We exploit this relationship in the following way. Given a point pattern  $\mathbf{x}$  and an estimator  $\widehat{\lambda}(\cdot)$  of  $\lambda(u)$ ,  $u \in W$ , fix some  $p \in (0, 1]$  and generate  $m \geq 1$  independent random point patterns, each obtained by randomly thinning the original point pattern with the same retention probability  $p$ . This results in the thinned patterns  $\mathbf{x}_p^1, \dots, \mathbf{x}_p^m$ , each for which the intensity is estimated. We now let the average of these  $m$  estimated intensity functions, divided by  $p$ , be reported as the final estimate; note the similarity with the approach considered by Baddeley (2007). The resample-smoothed Voronoi intensity estimator is formally defined as follows (Moradi et al., 2018a).

**Definition 3.2.** Consider a point process  $X \subset S$  with intensity function  $\lambda(\cdot)$ . Given some  $p \in (0, 1]$  and  $m \geq 1$ , the resample-smoothed Voronoi intensity estimator of  $\lambda(u)$ ,  $u \in W \subseteq S$ ,  $|W| > 0$ , is given by

$$\widehat{\lambda}_{p,m}^V(u) = \widehat{\lambda}_{p,m}^V(u; X, W) = \frac{1}{m} \sum_{i=1}^m \frac{\widehat{\lambda}_i^V(u)}{p}, \quad u \in W, \quad (3.3)$$

where

$$\widehat{\lambda}_i^V(u) = \widehat{\lambda}^V(u; X_p^i, W) = \sum_{x \in X_p^i} \frac{\mathbf{1}\{u \in \mathcal{V}_x(X_p^i, W)\}}{|\mathcal{V}_x(X_p^i, W)|}$$

is the Voronoi intensity estimator based on the  $i$ -th thinned process  $X_p^i$  of  $X \cap W$ . Note that when  $p = 1$ ,  $\widehat{\lambda}_{p,m}^V(\cdot)$  reduces to  $\widehat{\lambda}^V(\cdot)$  for any  $m \geq 1$ .

Reflecting on the effect of the thinning procedure, for each thinned version we obtain new Voronoi cells and consequently different locations of the jumps in the corresponding intensity estimate  $\widehat{\lambda}_i^V(\cdot)$ . This is what results in the “smoothing” and it is also the remedy for choosing the specific tiling in a possibly wrong/rigid way. Note also that we in fact simply are considering the average of  $m$  different estimators of  $\rho(\cdot)$ .

### 3.3.2 Theoretical properties

We next look closer at some statistical properties of resample-smoothed Voronoi intensity estimators. We stress that in the case of the restriction  $X \cap W$  of a point process to a bounded region  $W \neq S$ , the Voronoi cells  $\mathcal{V}_x(X, W)$  are different from those when  $W = S$ . Hereby distributional properties of  $\widehat{\lambda}_{p,m}^V(\cdot)$  may be different depending on how  $W$  is chosen.

Turning to the first order properties of  $\widehat{\lambda}_{p,m}^V(\cdot)$ , we note that

$$\int_W \widehat{\lambda}_{p,m}^V(u) du = \frac{1}{mp} \sum_{i=1}^m \sum_{x \in X_p^i} \frac{\int_W \mathbf{1}\{u \in \mathcal{V}_x(X_p^i, W)\} du}{|\mathcal{V}_x(X_p^i, W)|} = \frac{1}{mp} \sum_{i=1}^m N(X_p^i \cap W). \quad (3.4)$$

Hence, when  $p = 1$  we have preservation of mass, i.e.  $\int_W \widehat{\lambda}_{p,m}^V(u) du = N(X \cap W)$ . Taking expectations on both sides in (3.4) results in

$$\mathbb{E} \left[ \int_W \widehat{\lambda}_{p,m}^V(u) du \right] = \frac{1}{m} \sum_{i=1}^m \frac{p \int_W \lambda(u) du}{p} = \int_W \lambda(u) du, \quad (3.5)$$

i.e., for any  $m \geq 1$  and  $p \in (0, 1]$ ,  $\int_W \widehat{\lambda}_{p,m}^V(u) du$  is an unbiased estimator of  $\mathbb{E}[N(X \cap W)]$ .

We now consider the asymptotic scenario where the number of thinned patterns,  $m \geq 1$ , in the estimator (3.3) tends to infinity.

**Lemma 3.3.1.** *Given fixed  $p \in (0, 1]$ , for any point pattern  $\mathbf{x} \subset W \subseteq S$  we have that  $\lim_{m \rightarrow \infty} \widehat{\lambda}_{p,m}^V(u; \mathbf{x}, W)$  a.s. exists.*

*Proof.* Conditionally on  $\mathbf{x}$ , all  $\widehat{\lambda}_i^V(u)$  are non-negative i.i.d. random variables, hence by the (conditional) law of large numbers,  $\widehat{\lambda}_{p,m}^V(u)$  converges a.s. to the conditional

expectation of  $\widehat{\lambda}_i^V(u)$  given  $\mathbf{x}$ . Using (3.5), it yields that the latter conditional expectation is a.s. finite.  $\square$

Noting that  $\mathbb{E}[\widehat{\lambda}_{p,m}^V(u; X, W)] = \mathbb{E}[\widehat{\lambda}^V(u; X_p, W)]/p$  for any  $p \in (0, 1]$  and  $m \geq 1$ , we see that  $\widehat{\lambda}_{p,m}^V(u; X, W)$  is unbiased for the estimation of the intensity of  $X$  if and only if the original Voronoi intensity estimator is unbiased for the estimation of the intensity of an arbitrary thinning  $X_p$ . There is unfortunately not much more to be said without explicitly saying something about the distributional properties of  $X$ .

When  $X$  is stationary, all Voronoi cells have the same distribution and we may speak of the typical Voronoi cell  $\mathcal{V}_0 = \mathcal{V}_0(X)$ , which satisfies  $\mathcal{V}_0 \stackrel{d}{=} \theta_{-x}\mathcal{V}_x(X, S)$  for any  $x \in X$ ; here  $\theta_{-x}$  denotes the transformation/shift such that  $x$  is taken to the origin  $0 \in S$ .

**Theorem 3.3.1.** *For a stationary point process  $X \subset W = S$  with intensity  $\lambda > 0$ , the resample-smoothed Voronoi intensity estimator is unbiased for any choice of  $p \in (0, 1]$  and  $m \geq 1$ .*

*Proof.* A  $p$ -thinning of  $X$  is again stationary with intensity  $p\lambda$ . By Daley and Vere-Jones (2008, Expression (11.3.2)),

$$G_{\theta_v X_p}(\cdot) = G_{(\theta_v X)_p}(\cdot) = G_{\theta_v X}(p \cdot + 1 - p) = G_X(p \cdot + 1 - p) = G_{X_p}(\cdot), \quad v \in S,$$

where  $G_X(\cdot)$  is the generating functional of  $X$ . Using Last (2010, Corollary 8.7) we immediately obtain that  $\mathbb{E}[\widehat{\lambda}_{p,m}^V(u)] = \mathbb{E}[\widehat{\lambda}^V(u; X_p, S)]/p = p\lambda/p = \lambda$ .  $\square$

As our main interest lies in estimating non-constant intensity functions, stationary models are of limited practical interest. We next turn to inhomogeneous Poisson processes in Euclidean spaces.

**Theorem 3.3.2.** *Let  $X \subseteq W = S = \mathbb{R}^d$ ,  $d \geq 1$ , be a Poisson process with intensity function  $\lambda(u)$ ,  $u \in \mathbb{R}^d$ , which satisfies the Lipschitz condition that for some  $\mu_u > 0$ ,  $|\lambda(v) - \lambda(u)| \leq \mu_u \varepsilon$  for  $v \in B(u, \varepsilon)$  and  $\varepsilon > 0$  sufficiently small;  $B(u, \varepsilon)$  denotes the Euclidean ball with centre  $u$  and radius  $\varepsilon > 0$ . Denoting by  $C_u(X)$  the Voronoi cell containing  $u \in \mathbb{R}^d$ , assume further that  $m^\kappa := \sup_{u \in \mathbb{R}^d} \mathbb{E}[|C_u(X)|^{-\kappa}] < \infty$  for some  $\kappa \geq 1 + 1/d$ . Then, for any  $u \in \mathbb{R}^d$ ,  $p \in (0, 1]$  and  $m \geq 1$ ,*

$$\left| \lambda(u) - \mathbb{E} \left[ \widehat{\lambda}_{p,m}^V(u) \right] \right| \leq Cp^{-1}(p\lambda(u))^{-1/d} \log(p\lambda(u))^{2/d}$$



for some  $C > 0$  that depends on the intensity. The right hand side tends to 0 as the intensity tends to infinity.

*Proof.* We denote by  $x_u(X) \in X$  the centre of the Voronoi cell  $C_u(X)$ , the cell containing  $u \in \mathbb{R}^d$ . Let  $\varepsilon > 0$ ,  $\mu = \mu_u$  and  $\lambda_- = \min_{v \in B(u, \varepsilon)} \lambda(v)$ , such that  $\lambda(v)/2 \leq \lambda(v) - \mu\varepsilon \leq \lambda_- \leq \lambda(v)$  on  $B(u, \varepsilon)$ . Let  $X_-$  be obtained by independently removing/adding points at rate  $\lambda_- - \lambda(v)$ ,  $v \in \mathbb{R}^d$ . Note that  $X_-$  is a homogeneous Poisson process with intensity  $\lambda_-$  and  $X_- \subseteq X$  on  $B(u, \varepsilon)$  a.s..

We call Voronoi neighbours in some configuration  $\mathbf{x}$  the centres of cells of  $\mathbf{x}$  which are neighbours of  $C_u(\mathbf{x})$ . Denote by  $R(\mathbf{x})$  the maximal Euclidean distance between  $x_u(\mathbf{x})$  and its Voronoi neighbours. Remark that if  $R(\mathbf{x}) \leq \varepsilon$ , then  $C_u(\mathbf{x}) \subseteq B(u, \varepsilon)$ . One can find a finite number of balls such that if any such ball contains a point of  $\mathbf{x}$ , then  $R(\mathbf{x}) \leq 1$ . Hence, using the void probabilities of  $X$ , we have at the scale  $\varepsilon$  for  $X$  that

$$\mathbb{P}(R(X) \geq \varepsilon) \leq C_d e^{-c_d \lambda_- \varepsilon^d}$$

for some  $C_d, c_d > 0$ .

Now, let  $\Omega$  be the event that  $X$  and  $X_-$  coincide on  $B(u, \varepsilon)$  and  $R(X) \leq \varepsilon$ . Conditionally on  $\Omega$ ,  $C_u(X) = C_u(X_-) \subseteq B(u, \varepsilon)$ . We obtain

$$\begin{aligned} \mathbf{1}_{\{\Omega^c\}} &\leq \mathbf{1}_{\{R(X) > \varepsilon\}} + \sum_{x \in X_- \cap B(u, \varepsilon)} \mathbf{1}_{\{x \text{ eliminated at thinning}\}}, \\ \mathbb{P}(\Omega^c) &\leq \mathbb{P}(R(X) > \varepsilon) + \int_{B(u, \varepsilon)} \mu \varepsilon dx \leq C_d e^{-c_d \lambda_- \varepsilon^d} + c\varepsilon^d \mu \varepsilon. \end{aligned}$$

Let further  $\kappa' = (1 - \kappa^{-1})^{-1} \leq d + 1$ . By Hölder's inequality and Theorem 1 we have that

$$\begin{aligned} \left| \mathbb{E} \left[ \widehat{\lambda}^V(u) \right] - \lambda(u) \right| &\leq \left| \mathbb{E} \left[ \frac{\mathbf{1}_{\{\Omega\}}}{|C_u(X)|} \right] - \lambda(u) \right| + \mathbb{E} \left[ \frac{\mathbf{1}_{\{\Omega^c\}}}{|C_u(X)|} \right] \\ &\leq \left| \mathbb{E} \left[ \frac{\mathbf{1}_{\{\Omega\}}}{|C_u(X_-)|} \right] - \lambda(u) \right| + (\mathbb{E} |C_u(X)|^{-\kappa})^{1/\kappa} \mathbb{P}(\Omega^c)^{1/\kappa'} \\ &\leq \underbrace{\mathbb{E} \left[ \frac{1}{|C_u(X_-)|} - \lambda_- \right]}_{=0} + \mathbf{1}_{\{\Omega^c\}} \frac{1}{|C_u(X_-)|} + |\lambda(u) - \lambda_-| \\ &\quad + m(c_d \mu \varepsilon^d \varepsilon + C_d e^{-c_d \lambda_- \varepsilon^d})^{1/\kappa'} \\ &\leq \mu \varepsilon + 2m(c_d \mu \varepsilon^d \varepsilon + C_d e^{-c_d \lambda_- \varepsilon^d})^{1/\kappa'}. \end{aligned}$$

Setting  $\varepsilon = \lambda_-^{-1/d} \log(\lambda_-)^{2/d}$  and recalling that  $\lambda(u)/2 \leq \lambda_-$ , using that  $\kappa' \leq d + 1$ , proves the result for the original Voronoi intensity estimator.

As a  $p$ -thinning  $X_p$ ,  $p \in (0, 1]$ , of  $X$  is a Poisson process with intensity  $p\lambda(\cdot)$ , we finally note that

$$\begin{aligned} p|\mathbb{E}[\widehat{\lambda}_{p,m}^V(u)] - \lambda(u)| &= \left| \mathbb{E} \left[ \widehat{\lambda}^V(u; X_p, \mathbb{R}^d) \right] - p\lambda(u) \right| \\ &\leq \mu p^{-1} \varepsilon + 2m(c_d \mu p^{-1} \varepsilon^d \varepsilon + C_d e^{-c_d p \lambda(u) \varepsilon^d})^{1/\kappa'}, \end{aligned}$$

since  $\mathbb{E}|C_u(X_p)|^{-\kappa} \leq \mathbb{E}|C_u(X)|^{-\kappa}$ . □

**Remark 3.3.1.** *The moment condition, and the Lipschitz assumption on  $\lambda$  can be relaxed to weaker versions and still have the left hand side go to 0, but the rate would be different.*

*It has been conjectured that the size of the typical cell of a homogeneous Poisson process follows a (generalised) Gamma distribution (see e.g. Chiu et al. (2013)); note in particular Lemma 3.3.2. The moment condition in the statement of the above result, i.e.  $m^\kappa < \infty$ , would be satisfied if this is indeed the case. Under such a conjectured distribution, Barr and Schoenberg (2010) showed that in the planar case the original Voronoi intensity estimator is ratio-unbiased for a given class of intensity functions.*

Regarding the variance of  $\widehat{\lambda}_{p,m}^V(u)$ , the next result shows that by thinning as much as possible we also obtain a variance of the resample-smoothed Voronoi estimator which is close to 0. Hence, for cases where the estimator is unbiased we should, in theory, smooth as much as possible, in combination with choosing  $m$  as large as possible.

**Theorem 3.3.3.** *Consider a point process  $X$  observed in  $W \subseteq S$ , with  $\text{var} \left[ \widehat{\lambda}^V(u) \right] < \infty$  for any  $u \in W$ . For any  $p \in (0, 1]$  and  $m \geq 1$  we have that*

$$\text{var} \left[ \widehat{\lambda}_{p,1}^V(u) \right] / m \leq \text{var} \left[ \widehat{\lambda}_{p,m}^V(u) \right] \leq \text{var} \left[ \widehat{\lambda}_{p,1}^V(u) \right],$$

*and  $\text{var} \left[ \widehat{\lambda}_{p,m}^V(u) \right]$  converges as  $m \rightarrow \infty$  to the covariance between  $\widehat{\lambda}_1^V(u)/p = \widehat{\lambda}^V(u; X_p^1, W)/p$  and  $\widehat{\lambda}_2^V(u)/p = \widehat{\lambda}^V(u; X_p^2, W)/p$ . Moreover, for a fixed  $m \geq 1$  and a bounded  $W$  it follows that  $\lim_{p \rightarrow 0} \text{var} \left[ \widehat{\lambda}_{p,m}^V(u) \right] = 0$ .*

*Proof.* Note first that

$$\begin{aligned} \text{var} \left[ \widehat{\lambda}_{p,m}^V(u) \right] &= \frac{1}{(mp)^2} \sum_{i=1}^m \text{var} \left[ \widehat{\lambda}_i^V(u) \right] + \frac{1}{(mp)^2} \sum_{i \neq j} \text{cov} \left[ \widehat{\lambda}_i^V(u), \widehat{\lambda}_j^V(u) \right] \quad (3.6) \\ &= \frac{1}{m} \text{var} \left[ \widehat{\lambda}_1^V(u)/p \right] + \frac{m-1}{m} \text{cov} \left[ \widehat{\lambda}_1^V(u)/p, \widehat{\lambda}_2^V(u)/p \right] \\ &= \text{var} \left[ \widehat{\lambda}_{p,1}^V(u) \right] \frac{1 + (m-1) \text{corr} \left[ \widehat{\lambda}_1^V(u), \widehat{\lambda}_2^V(u) \right]}{m}, \end{aligned}$$

where  $\text{cov}[\cdot, \cdot]$  and  $\text{corr}[\cdot, \cdot]$  denote covariance and correlation, respectively. Since the variance is non-negative, by (3.6) we obtain that  $\text{corr} \left[ \widehat{\lambda}_1^V(u), \widehat{\lambda}_2^V(u) \right] \geq -1/(m-1)$  for any  $m \geq 1$ , i.e. the correlation is non-negative, whereby

$$\text{var} \left[ \widehat{\lambda}_{p,1}^V(u) \right] / m \leq \text{var} \left[ \widehat{\lambda}_{p,m}^V(u) \right] \leq \text{var} \left[ \widehat{\lambda}_{p,1}^V(u) \right]. \quad (3.7)$$

Also, letting  $m \rightarrow \infty$  in (3.6) we obtain  $\text{cov} \left[ \widehat{\lambda}_1^V(u), \widehat{\lambda}_2^V(u) \right] / p^2$  since  $\text{var} \left[ \widehat{\lambda}_1^V(u) \right] < \infty$ .

Regarding the variance tending to 0, it is sufficient to show it for  $m = 1$  since  $\text{var} \left[ \widehat{\lambda}_{p,m}^V(u) \right] \leq \text{var} \left[ \widehat{\lambda}_{p,1}^V(u) \right]$ . Let  $(X_p)_{p \in (0,1]}$  be a coupling such that  $X_p$  is non-increasing in terms of inclusion: assign independent  $U(0,1)$ -distributed labels to the points of  $X$  and generate  $X_p$  by keeping all points with labels smaller than  $p$ . For a bounded  $W$  there is a.s. some  $p_0 \in (0,1)$  such that  $X_p = \emptyset$  for all  $p \in (0, p_0)$ . Hence,  $\widehat{\lambda}_{p,1}^V(u) = \widehat{\lambda}^V(u; X_p, W)/p = 0/p = 0$  (by definition) for such  $p$ , which means that the limit  $\widehat{\lambda}_{p,1}^V(u) \downarrow 0$  is deterministic. Since there are  $p \in (0, 1]$  such that  $\mathbb{E}[\widehat{\lambda}_{p,1}^V(u)^2] < \infty$ , by the dominated convergence theorem it follows that  $\text{var} \left[ \widehat{\lambda}_{p,1}^V(u) \right] \rightarrow 0$  as  $p \rightarrow 0$ . □

From the proof of Theorem 3.3.1 we have that the  $p$ -thinning  $X_p$  of a stationary point process  $X$  with intensity  $\lambda > 0$  is again stationary, but with intensity  $p\lambda$ . For  $X_p$ , the distribution  $\bar{P}_p(\cdot)$  of the size of the cell that covers  $u$  is the same for any  $u \in S$  and it is given by (see Last (2010, Section 8) and Schneider and Weil (2008, Theorem 10.4.1.))

$$\bar{P}_p(A) = p\lambda \int_A t P_{|\mathcal{V}_0(X_p)|}(dt), \quad A \subseteq [0, \infty), \quad (3.8)$$

where  $P_{|\mathcal{V}_0(X_p)|}(\cdot)$  is the distribution of the typical cell size. Besides giving us the unbiasedness in Theorem 3.3.1, i.e.

$$\mathbb{E}[\widehat{\lambda}_{p,m}^V(u)] = p^{-1} p\lambda \int_0^\infty t^{-1} t P_{|\mathcal{V}_0(X_p)|}(dt) = \lambda,$$

the relationship (3.8) further yields

$$\begin{aligned}\mathbb{E}[\widehat{\lambda}_{p,1}^V(u)^2] &= \frac{1}{p^2} \int_0^\infty \frac{1}{t^2} \bar{P}_p(dt) = \frac{\lambda}{p} \int_0^\infty \frac{1}{t} P_{|\mathcal{V}_0(X_p)|}(dt) = \frac{\lambda}{p} \mathbb{E}_0[1/|\mathcal{V}_0(X_p)|], \\ \text{var} \left[ \widehat{\lambda}_{p,1}^V(u) \right] &= \frac{\lambda}{p} \mathbb{E}_0[1/|\mathcal{V}_0(X_p)|] - \lambda^2.\end{aligned}$$

Through the proof of Theorem 3.3.3 we obtain

$$\text{var} \left[ \widehat{\lambda}_{p,m}^V(u) \right] = \lambda \left( \frac{\mathbb{E}_0[1/|\mathcal{V}_0(X_p)|]}{p} - \lambda \right) \frac{1 + (m-1) \text{corr} \left[ \widehat{\lambda}^V(u; X_p^1, S), \widehat{\lambda}^V(u; X_p^2, S) \right]}{m},$$

Unfortunately, we cannot get much further in the general setup; the problem lies in that  $P_{|\mathcal{V}_0|}(\cdot)$  typically is not known.

There is, however, one particular case where we can say a bit more and that is for Poisson processes on  $\mathbb{R}$ .

**Lemma 3.3.2.** *For a Poisson process on  $\mathbb{R}$  with intensity  $\lambda > 0$ , for any  $p \in (0, 1]$  and  $m \geq 1$  the typical cell size of  $X_p$  follows an Erlang/Gamma distribution with shape and rate parameters 2 and  $2p\lambda$ , respectively, and  $\text{var} \left[ \widehat{\lambda}_{p,m}^V(u) \right] \leq \text{var} \left[ \widehat{\lambda}_{p,1}^V(u) \right] = \lambda^2$ .*

*Proof.* Recall that  $X_p$  is a homogeneous Poisson process with intensity  $p\lambda$ . For a typical point of  $X_p$ , let  $\Delta_-$  and  $\Delta_+$  be the distances to the point's nearest neighbours to the left and to the right, respectively; they are independent and exponentially distributed with mean  $p\lambda$ . Since  $\Delta_-/2$  and  $\Delta_+/2$  are independent and exponentially distributed with mean  $2p\lambda$ , the typical cell size,  $\Delta_-/2 + \Delta_+/2$ , follows an Erlang/Gamma distribution with shape parameter 2 and rate  $2p\lambda$ , whereby the density of  $P_{|\mathcal{V}_0(X_p)|}(\cdot)$  is given by  $f_{|\mathcal{V}_0(X_p)|}(t) = (2p\lambda)^2 t e^{-2p\lambda t}$ . Through (3.8) we now obtain

$$\begin{aligned}\mathbb{E}[\widehat{\lambda}_{p,1}^V(u)^2] &= \frac{\lambda}{p} \mathbb{E}_0[1/|\mathcal{V}_0(X_p)|] = \frac{\lambda}{p} \int_0^\infty \frac{1}{t} (2p\lambda)^2 t e^{-2p\lambda t} dt \\ &= 4p\lambda^3 \int_0^\infty e^{-2p\lambda t} dt = \frac{4p\lambda^3}{2p\lambda} = 2\lambda^2,\end{aligned}\tag{3.9}$$

i.e.,  $\text{var} \left[ \widehat{\lambda}_{p,m}^V(u) \right] \leq \text{var} \left[ \widehat{\lambda}_{p,1}^V(u) \right] = 2\lambda^2 - \lambda^2 = \lambda^2$  by Theorem 3.3.3.  $\square$

Empirically, we have consistently observed that for a large enough  $m$ , the variance of  $\widehat{\rho}_{p,m}^V(u)$  decreases with  $p$ , for  $u \in W$  located a given distance from the boundary of  $W \subseteq S$ . As this is partly justified by Theorem 3.3.3, we are led to the following conjecture.

**Conjecture 3.3.1.** *For an arbitrary point process  $X \subset S$  and a large enough  $m$ , the variance of  $\hat{\rho}_{p,m}^V(u)$  is a decreasing function of  $p \in (0, 1]$ . In particular, if  $\hat{\rho}_{p,m}^V(u)$  is unbiased, this means that MISE is decreasing with  $p$ .*

### 3.3.3 Choosing the smoothing parameters

When using the resample-smoothed Voronoi intensity estimator (3.3) in practice, one needs to specify the smoothing parameters  $m \geq 1$  and  $p \in (0, 1]$  prior to finding the intensity estimate. We next discuss how to obtain proper choices for  $m$  and  $p$ .

#### 3.3.3.1 Choosing the number of thinnings

Lemma 3.3.1 tells us that for fixed  $p \in (0, 1]$ ,  $k \geq 1$  and any point pattern  $\mathbf{x} \subset W \subseteq S$  we have that  $\hat{\lambda}_{p,m}^V(u; \mathbf{x}, W)$  converges a.s. as  $m \rightarrow \infty$ . The question that remains, however, is for which  $m \geq 1$  we are sufficiently close to the limit. In our numerical experiments in Section 3.4 we illustrate that the estimated bias and variance of  $\hat{\lambda}_{p,m}^V(u)$  do not change significantly for  $m \geq 200$ . Hence, we propose to fix  $m = 200$  and proceed by finding a proper choice for  $p \in (0, 1]$ .

#### 3.3.3.2 Choosing retention probability

The selection of  $p \in (0, 1]$  is clearly the more delicate matter here; essentially we are faced with problems similar to those of choosing bandwidths in kernel estimation.

Through our numerical experiments (see Section 3.4) we have found that the choice  $p \in [0.1, 0.3]$  always seems to generate the best intensity estimates in the sense that the variance-bias-tradeoff is taken into account by keeping both the bias and variance relatively small. The lower limit 0.1 is based on our observation that the removal of more than 90% of the points per thinning might generate too flat estimates in some cases; from Section 3.4, Theorem 3.3.3 and Conjecture 3.3.1 it seems that the smaller the  $p$ , the better the estimate. We refer to the choice of  $m = 200$  and  $p \in [0.1, 0.3]$  as our rule-of-thumb.

We also propose a cross-validation approach to select  $p$  when a data-driven approach is preferred to the rule-of-thumb (see (1.12)). However, if the value obtained for  $p$  through the cross-validation would deviate too much from the rule-

of-thumb, we advise to proceed with the rule-of-thumb; see the log-Gaussian Cox process example in Section 3.4 for a situation where this occurs.

### 3.3.4 Large scale data and sparsity

In general, when the number of events,  $n$ , is very large for an observed point pattern  $\mathbf{x} = \{x_1, \dots, x_n\}$ , it is often natural to consider an adaptive intensity estimator as the scales of intensity likely vary a lot.

It may not be computationally feasible to compute  $\widehat{\lambda}_{p,m}^V(\cdot)$ ,  $p \in (0, 1]$ , for an arbitrary  $m \geq 1$  (or any other intensity estimator for that matter). An alternative way of exploiting the proposed setup is to consider  $\widehat{\lambda}_{p_0,m}^V(\cdot)$  for some  $p_0 \in [0.1, 0.3]$  and  $m = 1$ . This means that we would introduce sparsity by only having to generate Voronoi cells for 10–30% of the original number of points. The results in Section 3.4 indicate how good an estimate one would typically obtain. Moreover, if the computation of  $\widehat{\lambda}_{p_0,1}^V(\cdot)$  is reasonably quick, one could generate another estimate  $\widehat{\lambda}_{p_0,1}^V(\cdot)$  and average over these to obtain  $\widehat{\lambda}_{p_0,2}^V(\cdot)$ . One could then continue like this in a stepwise fashion, given a total computation timeframe.

## 3.4 Numerical experiments

As previously pointed out, we assess the performance of our intensity estimation approach numerically, which we choose to do in the Euclidean setting.

In our simulation study, we consider four different types of models with varying degrees of change in intensity and spatial interaction; clustering, spatial randomness and regularity. For each model we use 500 realisations on  $W = [0, 1]^2$  to generate numerical estimates of relevant quantities such as bias, variance, Integrated Variance (IV), Integrated Square Bias (ISB) and Integrated Absolute Bias (IAB) for  $\widehat{\rho}_{p,m}^V(u)$ ,  $u \in W$ ; recall that Mean Integrated Square Error (MISE) is obtained as the sum of IV and ISB. We also provide a comparison with the kernel intensity estimator under the uniform correction (1.8) and using two different bandwidth selection methods (1.12) and (1.13) (Loader, 1999; Cronie and van Lieshout, 2018). The resample-smoothed Voronoi estimators in the two-dimensional plane and on a linear network were implemented in the R language using the package **spatstat** (Baddeley et al., 2015) and will be released publicly in a future version of **spatstat**. Our simulation experiments and figures were generated using this

implementation.

The overall conclusion is that we clearly reduce the estimation errors by resample-smoothing the Voronoi intensity estimator. Moreover, the cross-validation approach to selecting  $p$  on average yields slightly poorer intensity estimates than the rule-of-thumb, in particular if the model is clustered.

### 3.4.1 Homogeneous Poisson process

Here we consider a homogeneous Poisson process  $X \subset W = [0, 1]^2$  with intensity  $\lambda = 60$ . Table 3.1 provides estimates of IAB, ISB and IV for  $\widehat{\lambda}_{p,m}^V(u)$ ,  $u \in W$ ,  $m = 200, 300, 400$ ,  $p = 0.1, \dots, 1$ ; recall that we use 500 realisations of  $X$ . Indeed, the bias seems fairly stable over the range of values for  $p$  and the variance is clearly decreasing with  $p$ ; choosing  $p$  according to the rule-of-thumb keeps MISE small. For illustrational purposes, in Figure 3.1 we provide estimation error plots for one of the realisations, for  $p = 0.2$  and  $p = 1$  with  $m = 200$ , and one can clearly see the gain of the resample-smoothing; note that the under-estimation occurs in the empty regions. In addition, we provide plots of the estimated bias and variance for  $p = 0.1, 0.3, 0.5, 0.7, 0.9, 1$  when  $m = 200$ , together with estimated bias and variance for kernel intensity estimate  $\widehat{\lambda}^U(u)$  considering both bandwidth selection method (1.12) and (1.13), and they essentially confirm what has been observed in Table 3.1. See Figures 3.2 and 3.3.

Turning to the cross-validation approach to selecting  $p$  for  $m = 200$ , based on 100 realisations of the model, we obtain  $\text{IAB} = 4.9$ ,  $\text{ISB} = 30.3$  and  $\text{IV} = 255$  which are in the range of what one obtains when  $p$  is fixed in  $(0.1, 0.3)$ . In Table 3.2 we further provide the 100 selected values for  $p$  and we see that the majority of the selected values for  $p$  fall within the range of our rule-of-thumb. Comparing with kernel estimation under uniform edge correction, using Poisson likelihood cross-validation (1.12) to select the bandwidth, we obtain  $\text{IAB} = 0.24$ ,  $\text{ISB} = 0.11$  and  $\text{IV} = 126.05$ . By instead employing the bandwidth selection method (1.13) (Cronie and van Lieshout, 2018), we obtain  $\text{IAB} = 0.87$ ,  $\text{ISB} = 1.12$  and  $\text{IV} = 688.25$ .

Table 3.1: Estimates of IAB, ISB and IV for  $\widehat{\lambda}_{p,m}^V(u)$ ,  $u \in W = [0, 1]^2$ ,  $m = 200, 300, 400$ ,  $p = 0.1, \dots, 1$ , based on 500 realisations of a homogeneous Poisson process in  $W = [0, 1]^2$  with intensity  $\lambda = 60$ .

$m \backslash p$	IAB			ISB			IV		
	200	300	400	200	300	400	200	300	400
.1	5.7	5.7	5.7	43.5	43.2	43.0	158.4	154.8	152.5
.2	4.6	4.6	4.6	28.4	28.5	28.4	264.1	260.3	257.9
.3	3.9	3.9	3.9	22.5	22.2	22.2	375.3	370.6	368.8
.4	3.5	3.5	3.5	19.7	19.6	19.6	490.6	488.8	487.8
.5	3.2	3.2	3.2	18.1	18.1	18.1	672.0	623.9	622.9
.6	3.0	3.0	3.0	17.1	17.1	17.0	781.9	779.4	779.0
.7	2.9	2.9	2.9	16.5	16.5	16.5	960.0	958.7	958.8
.8	2.9	2.9	2.9	16.0	16.0	16.0	1172.2	1171.8	1171.1
.9	2.9	2.9	2.9	15.8	15.8	15.8	1422.2	1419.6	1418.9
1	2.9	2.9	2.9	15.8	15.8	15.8	1733.2	1733.2	1733.2

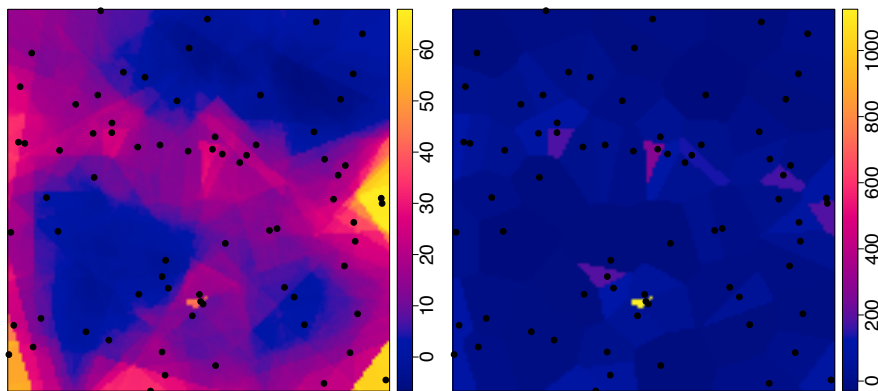


Figure 3.1: Estimation error plots for a realisation of a homogeneous Poisson process  $X$  in  $W = [0, 1]^2$  with intensity  $\lambda = 60$ . *Left:*  $p = 0.2$  and  $m = 200$ . *Right:*  $p = 1$ .



Table 3.2: Cross-validation selections of  $p$  for  $m = 200$  in a geometric sequence, based on 100 realisations of a homogeneous Poisson process in  $W = [0, 1]^2$  with intensity  $\lambda = 60$ .

$p$	0.10	0.13	0.18	0.24	0.33	0.44	0.59	0.80
Frequency	63	15	5	8	3	4	2	0

### 3.4.2 Inhomogeneous Poisson process

More interestingly, we next consider 500 realisations of an inhomogeneous Poisson process  $X \subseteq W = [0, 1]^2$  with intensity  $\lambda(x, y) = |10 + 90 \sin(16x)|$ ; the expected total point count is 58.6. Table 3.3 provides estimates of IAB, ISB and IV for  $\hat{\lambda}_{p,m}^V(u)$ ,  $u \in W$ ,  $m = 200, 300, 400$ ,  $p = 0.1, \dots, 1$ . Moreover, in Figure 3.4 we provide estimation error plots for one of the realisations, for  $p = 0.2$  and  $p = 1$  with  $m = 200$ , and, we also provide plots of the estimated bias and variance for  $p = 0.1, 0.3, 0.5, 0.7, 0.9, 1$  and  $m = 200$ . See Figures 3.5 and 3.6.

Table 3.3: Estimates of IAB, ISB and IV for  $\hat{\lambda}_{p,m}^V(u)$ ,  $u \in W = [0, 1]^2$ ,  $m = 200, 300, 400$ ,  $p = 0.1, \dots, 1$ , based on 500 realisations of an inhomogeneous Poisson process on  $W = [0, 1]^2$  with intensity  $\lambda(x, y) = |10 + 90 \sin(16x)|$ .

		IAB			ISB			IV		
$m$	$p$	200	300	400	200	300	400	200	300	400
.1		25.6	25.6	25.6	892.3	891.8	891.7	154.2	150.1	147.6
.2		25.5	25.5	25.5	882.8	883.2	883.3	249.1	247.3	245.6
.3		25.6	25.5	25.5	881.5	881.5	881.5	360.1	356.3	356.2
.4		25.5	25.5	25.5	878.8	879.0	879.0	479.9	477.2	475.0
.5		25.5	25.5	25.5	872.6	872.5	872.6	609.8	609.6	609.8
.6		25.4	25.4	25.4	862.7	862.7	862.7	762.6	764.3	764.1
.7		25.2	25.2	25.2	849.9	850.0	850.0	952.0	948.3	949.0
.8		25.0	25.0	25.0	835.1	834.8	834.8	1171.9	1172.3	1172.1
.9		24.7	24.7	24.7	817.7	817.6	817.6	1440.1	1440.9	1440.0
1		24.4	24.4	24.4	799.3	799.3	799.3	1783.8	1783.8	1783.8

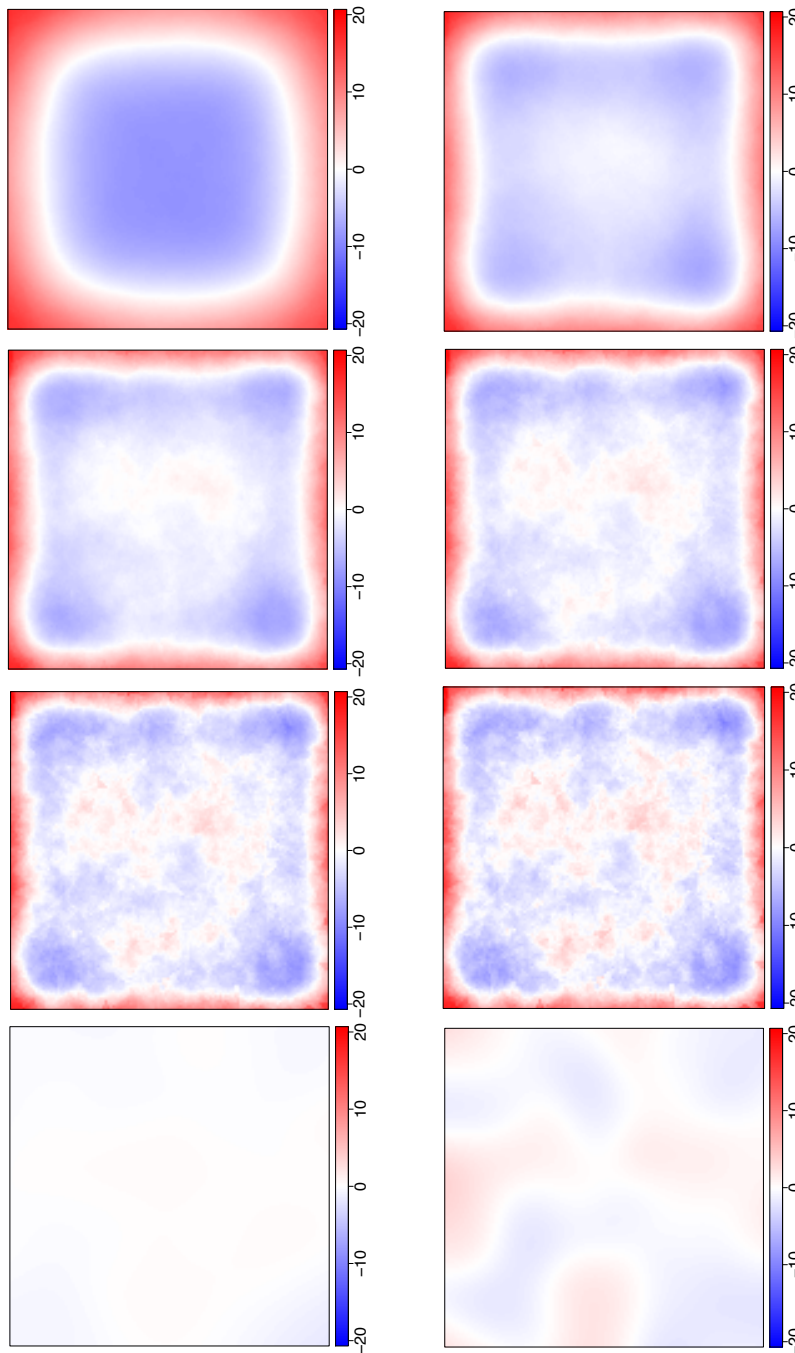


Figure 3.2: Estimated bias for  $\hat{\lambda}_{p,m}^V(u)$  and  $\hat{\lambda}^U(u)$ ,  $u \in W = [0, 1]^2$ ,  $m = 200$ , based on 500 realisations of a homogeneous Poisson process  $X \subset W = [0, 1]^2$  with intensity  $\lambda = 60$ . From top-left to bottom-right:  $\hat{\lambda}_{p,m}^V(u)$  with  $p = 0.1, 0.3, 0.5, 0.7, 0.9, 1$ , and  $\hat{\lambda}^U(u)$  using bandwidth selection (1.12) and (1.13).

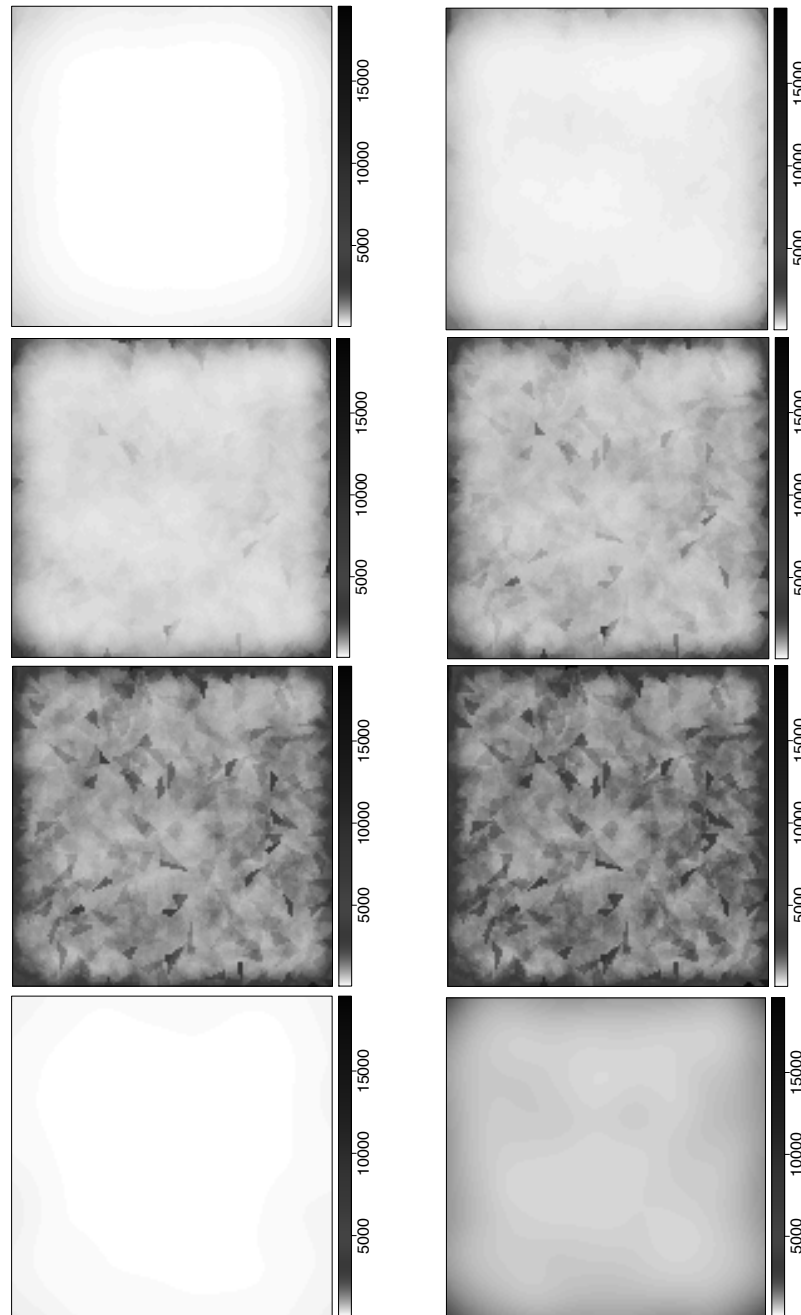


Figure 3.3: Estimated variance for  $\hat{\lambda}_{p,m}^V(u)$  and  $\hat{\lambda}^U(u)$ ,  $u \in W = [0, 1]^2$ ,  $m = 200$ , based on 500 realisations of a homogeneous Poisson process  $X \subset W = [0, 1]^2$  with intensity  $\lambda = 60$ . From top-left to bottom-right:  $\hat{\lambda}_{p,m}^V(u)$  with  $p = 0.1, 0.3, 0.5, 0.7, 0.9, 1$ , and  $\hat{\lambda}^U(u)$  using bandwidth selection (1.12) and (1.13).

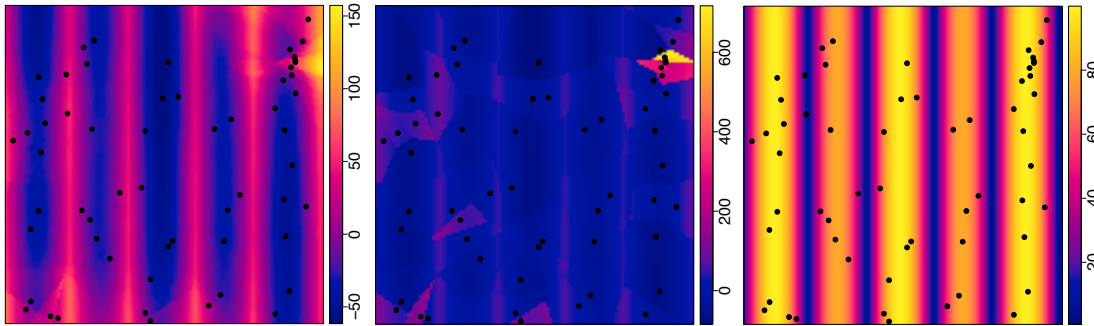


Figure 3.4: True intensity and estimation error plots for a realisation of an inhomogeneous Poisson process on  $W = [0, 1]^2$  with intensity  $\lambda(x, y) = |10 + 90 \sin(16x)|$ . *Left*:  $p = 0.2$  and  $m = 200$ . *Middle*:  $p = 1$ , *Right*: True intensity.

Turning to the cross-validation approach to selecting  $p$  based on  $m = 200$  and 100 realisations of the model, we obtain  $\text{IAB} = 25.5$ ,  $\text{ISB} = 885.2$  and  $\text{IV} = 218.5$ , with the majority of the selected  $p$ 's coinciding with the rule-of-thumb (see Table 3.4). Comparing with kernel estimation under uniform edge correction, using Poisson likelihood cross-validation (1.12) to select the bandwidth, we obtain  $\text{IAB} = 25.16$ ,  $\text{ISB} = 853.24$  and  $\text{IV} = 158.00$ . By instead employing the bandwidth selection (1.13), we obtain  $\text{IAB} = 24.43$ ,  $\text{ISB} = 797.02$  and  $\text{IV} = 636.63$ .

Hence, the conclusions here are essentially the same as for the homogeneous Poisson process in Section 3.4.1, with the main difference arguably being that inhomogeneity enforces slightly harder thinning in the cross-validation.

Table 3.4: Cross-validation selections of  $p$  in a geometric sequence for  $m = 200$ , based on 100 realisations of an inhomogeneous Poisson process in  $W = [0, 1]^2$  with intensity  $\lambda(x, y) = |10 + 90 \sin(16x)|$ .

$p$	0.10	0.13	0.18	0.24	0.33	0.44	0.59	0.80
Frequency	69	15	11	2	0	1	0	2

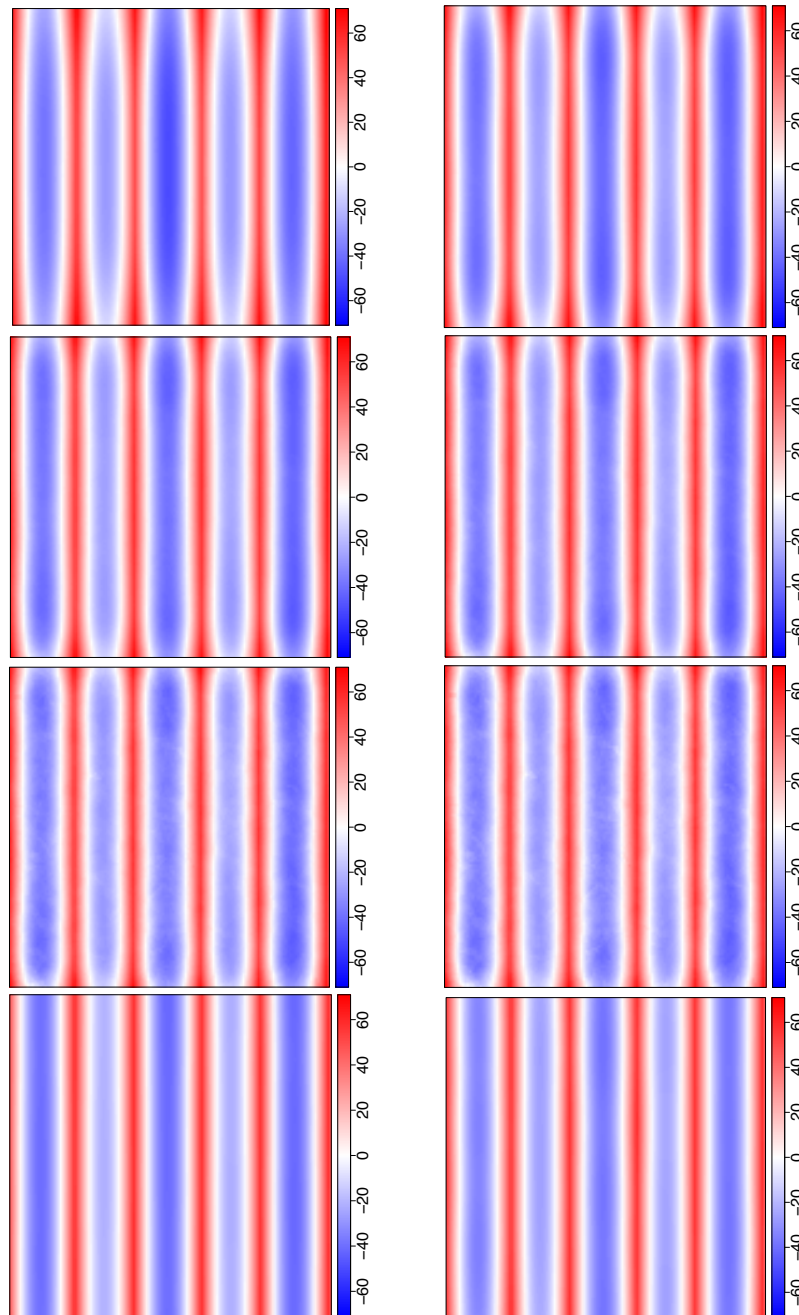


Figure 3.5: Estimated bias for  $\hat{\lambda}_{p,m}^V(u)$  and  $\hat{\lambda}^U(u)$ ,  $u \in W = [0, 1]^2$ ,  $m = 200$ , based on 500 realisations of an inhomogeneous Poisson process  $X \subseteq W = [0, 1]^2$  with intensity  $\rho(x, y) = |10 + 90 \sin(16x)|$ . From top-left to bottom-right:  $\hat{\lambda}_{p,m}^V(u)$  with  $p = 0.1, 0.3, 0.5, 0.7, 0.9, 1$ , and  $\hat{\lambda}^U(u)$  using bandwidth selection (1.12) and (1.13).

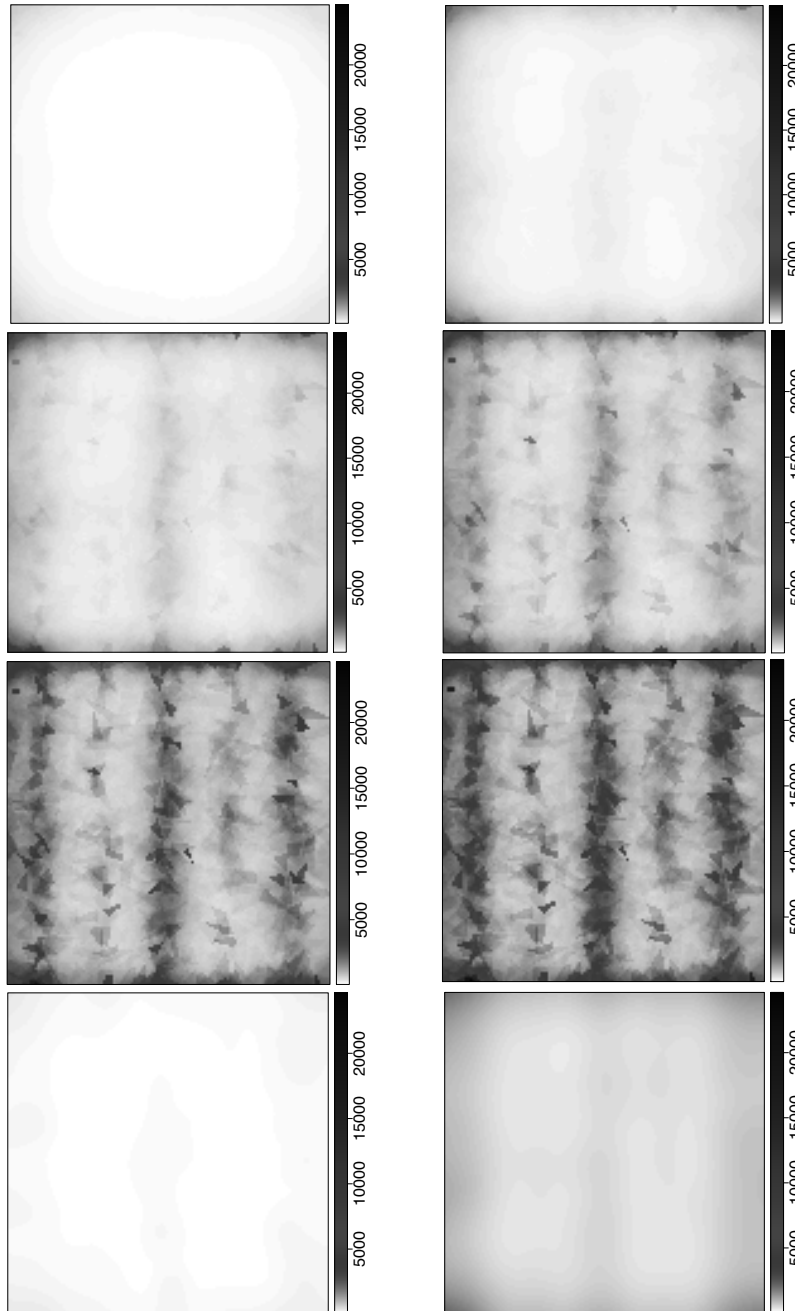


Figure 3.6: Estimated variance for  $\hat{\lambda}_{p,m}^V(u)$  and  $\hat{\lambda}^U(u)$ ,  $u \in W = [0, 1]^2$ ,  $m = 200$ , based on 500 realisations of an inhomogeneous Poisson process  $X \subseteq W = [0, 1]^2$  with intensity  $\rho(x, y) = |10 + 90 \sin(16x)|$ . From top-left to bottom-right:  $\hat{\lambda}_{p,m}^V(u)$  with  $p = 0.1, 0.3, 0.5, 0.7, 0.9, 1$ , and  $\hat{\lambda}^U(u)$  using bandwidth selection (1.12) and (1.13).

### 3.4.3 Log-Gaussian Cox process

Turning to the scenario where the underlying point process exhibits clustering, we next consider 500 realisations of a log-Gaussian Cox process  $X \subseteq W = [0, 1]^2$  where the driving Gaussian random field has mean function  $(x, y) \mapsto \log(40|\sin(20x)|)$  and covariance function  $((x_1, y_1), (x_2, y_2)) \mapsto 2 \exp\{-\|(x_1, y_1) - (x_2, y_2)\|/0.1\}$ . Hereby, the intensity is given by  $\rho(x, y) = 40|\sin(20x)|e^1$ . Table 3.5 provides estimates of IAB, ISB and IV for  $\hat{\rho}_{p,m}^V(u)$ ,  $u \in W$ ,  $m = 200, 300, 400$ ,  $p = 0.1, \dots, 1$  and we see that the rule-of-thumb, i.e.  $p \in [0.1, 0.3]$ , seems to be the preferable choice. In Figure 3.7 we provide estimation error plots for one of the realisations, for  $p = 0.2$  and  $p = 1$  with  $m = 200$ , and plots of the estimated bias and variance for  $p = 0.1, 0.3, 0.5, 0.7, 0.9, 1$  and  $m = 200$  are also provided in the Figures 3.8 and 3.9. Here it becomes visually clear that the resample smoothing is improving the estimation quite significantly.

The cross-validation approach to selecting  $p$  based on  $m = 200$  and 100 realisations of the model yields IAB = 28.4, ISB = 1118.2 and IV = 17207.5, which may be comparable to the choice  $p \approx 0.5$ . In Table 3.6 we further provide the 100 selected values for  $p$ . The phenomenon that too little smoothing tends to be applied ( $p$  is mainly chosen large) is not extremely surprising; as our cross-validation approach is based on a Poisson process likelihood function, it treats a realisation  $\mathbf{x}$  of  $X$  as a realisation of a Poisson process which has the corresponding realisation of the driving (random) intensity field as intensity function. In other words, it tries to perform state estimation, i.e. it tries to reconstruct each realisation of the driving intensity field through  $\mathbf{x}$ . This phenomenon, and that the Poisson process likelihood cross-validation is not performing well for clustered inhomogeneous point processes, has previously been observed in the context of kernel intensity estimation (Cronie and van Lieshout, 2018). Hence, if one suspects that there is clustering in addition to inhomogeneity, or if the cross-validation generates large values for  $p$ , then it is wiser to stick with the proposed rule-of-thumb,  $p \in [0.1, 0.3]$ . In fact cross-validation generated deviations from the rule-of-thumb may be seen as a possible indication of clustering or inhibition.

Comparing with kernel estimation under uniform edge correction, using Poisson likelihood cross-validation (Loader, 1999; Baddeley et al., 2015) to select the bandwidth, we obtain IAB = 27.75, ISB = 1031.03 and IV = 9952.85. By instead employing the bandwidth selection method of Cronie and van Lieshout (2018), we

obtain IAB = 28.97, ISB = 1117.94 and IV = 3856.79.

Table 3.5: Estimates of IAB, ISB and IV for  $\widehat{\rho}_{p,m}^V(u)$ ,  $u \in W = [0, 1]^2$ ,  $m = 200, 300, 400$ ,  $p = 0.1, \dots, 1$ , based on 500 realisations of a log-Gaussian Cox process in  $W = [0, 1]^2$  with mean function  $(x, y) \mapsto \log(40|\sin(20x)|)$  and  $((x_1, y_1), (x_2, y_2)) \mapsto 2 \exp\{-\|(x_1, y_1) - (x_2, y_2)\|/0.1\}$  as covariance function for the driving random field.

		IAB			ISB			IV( $\times 10^2$ )		
$m$ $p$		200	300	400	200	300	400	200	300	400
.1		29.5	29.5	29.5	1181.5	1181.9	1180.9	48.8	48.8	48.7
.2		28.8	28.8	28.8	1127.3	1127.4	1127.3	87.8	87.2	88.0
.3		28.2	28.2	28.2	1081.4	1081.7	1081.6	123.8	122.6	123.1
.4		27.6	27.6	27.6	1038.8	1039.2	1039.4	153.2	153.0	152.6
.5		27.1	27.1	27.1	1000.1	999.6	999.7	181.3	182.2	182.0
.6		26.5	26.5	26.5	963.9	963.7	963.5	212.4	212.5	212.1
.7		26.0	26.0	26.0	930.5	930.4	930.6	243.1	243.0	243.2
.8		25.6	25.6	25.6	901.1	900.6	900.7	278.8	279.2	279.3
.9		25.2	25.2	25.2	874.4	874.3	874.2	321.4	321.5	320.9
1		24.7	24.7	24.7	852.3	852.3	852.3	371.4	371.4	371.4

Table 3.6: Cross-validation selections of  $p$  in a geometric sequence for  $m = 200$ , based on 100 realisations of a log-Gaussian Cox process in  $W = [0, 1]^2$  with mean function  $(x, y) \mapsto \log(40|\sin(20x)|)$  and covariance function  $((x_1, y_1), (x_2, y_2)) \mapsto 2 \exp\{-\|(x_1, y_1) - (x_2, y_2)\|/0.1\}$  for the driving random field.

$p$	0.10	0.13	0.18	0.24	0.33	0.44	0.59	0.80
Frequency	4	4	0	1	8	14	34	35

### 3.4.4 Thinned simple sequential inhibition point process

To study inhomogeneity in combination with inhibition, we consider a simple sequential inhibition point process in  $W = [0, 1]^2$  with a total point count of 450 and



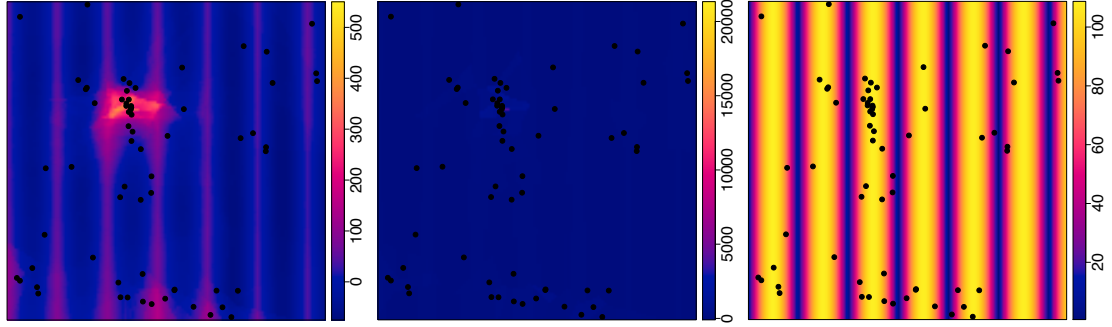


Figure 3.7: True intensity and estimation error plots for a realisation of a log-Gaussian Cox process in  $W = [0, 1]^2$  with mean function  $(x, y) \mapsto \log(40|\sin(20x)|)$  and  $((x_1, y_1), (x_2, y_2)) \mapsto 2 \exp\{-\|(x_1, y_1) - (x_2, y_2)\|/0.1\}$  as covariance function for the driving random field. *Left:*  $p = 0.2$  and  $m = 200$ . *Middle:*  $p = 1$ . *Right:* True intensity.

inhibition distance 0.3, which we thin according the retention probability function  $p(x, y) = \mathbf{1}\{x < 1/3\}|x - 0.02| + \mathbf{1}\{1/3 \leq x < 2/3\}|x - 0.5| + \mathbf{1}\{x \geq 2/3\}|x - 0.95|$ ,  $x, y \in W$ . This results in an inhomogeneous process with intensity  $\rho(x, y) = 450p(x, y)$ , which yields an expected total point count of 53.6. Table 3.7 provides estimates of IAB, ISB and IV for  $\hat{\rho}_{p,m}^V(u)$ ,  $u \in W$ ,  $m = 200, 300, 400$ ,  $p = 0.1, \dots, 1$ . Just as for the previous models, we argue that  $p$  should be chosen within the range of the rule-of-thumb.

In Figure 3.10 we provide estimation error plots for one of the realisations, for  $p = 0.2$  and  $p = 1$  with  $m = 200$ , and plots of the estimated bias and variance for  $p = 0.1, 0.3, 0.5, 0.7, 0.9, 1$  and  $m = 200$  can be found in Figures 3.11 and 3.12. Also here the improvements caused by the resample-smoothing are visually clear.

The cross-validation approach to selecting  $p$  based on  $m = 200$  and 100 realisations of the model yields  $\text{IAB} = 25.1$ ,  $\text{ISB} = 932.9$  and  $\text{IV} = 595.2$ , which is comparable to choosing  $p \approx 0.5$ . Moreover, Table 3.8 lists the selected values for  $p$  and we see that they tend to be either very large or very small. It thus seems that approximately one third of the time the cross-validation chooses  $p$  within our rule-of-thumb, and approximately half of the time it chooses  $p$  too large. Comparing with kernel estimation under uniform edge correction, using Poisson likelihood cross-validation to select the bandwidth, we obtain  $\text{IAB} = 20.5$ ,  $\text{ISB} = 663.94$  and  $\text{IV} = 485.48$ . By instead employing the bandwidth selection method of Cronie and van Lieshout (2018), we obtain  $\text{IAB} = 23.97$ ,  $\text{ISB} = 860.67$  and  $\text{IV} = 308.47$ .

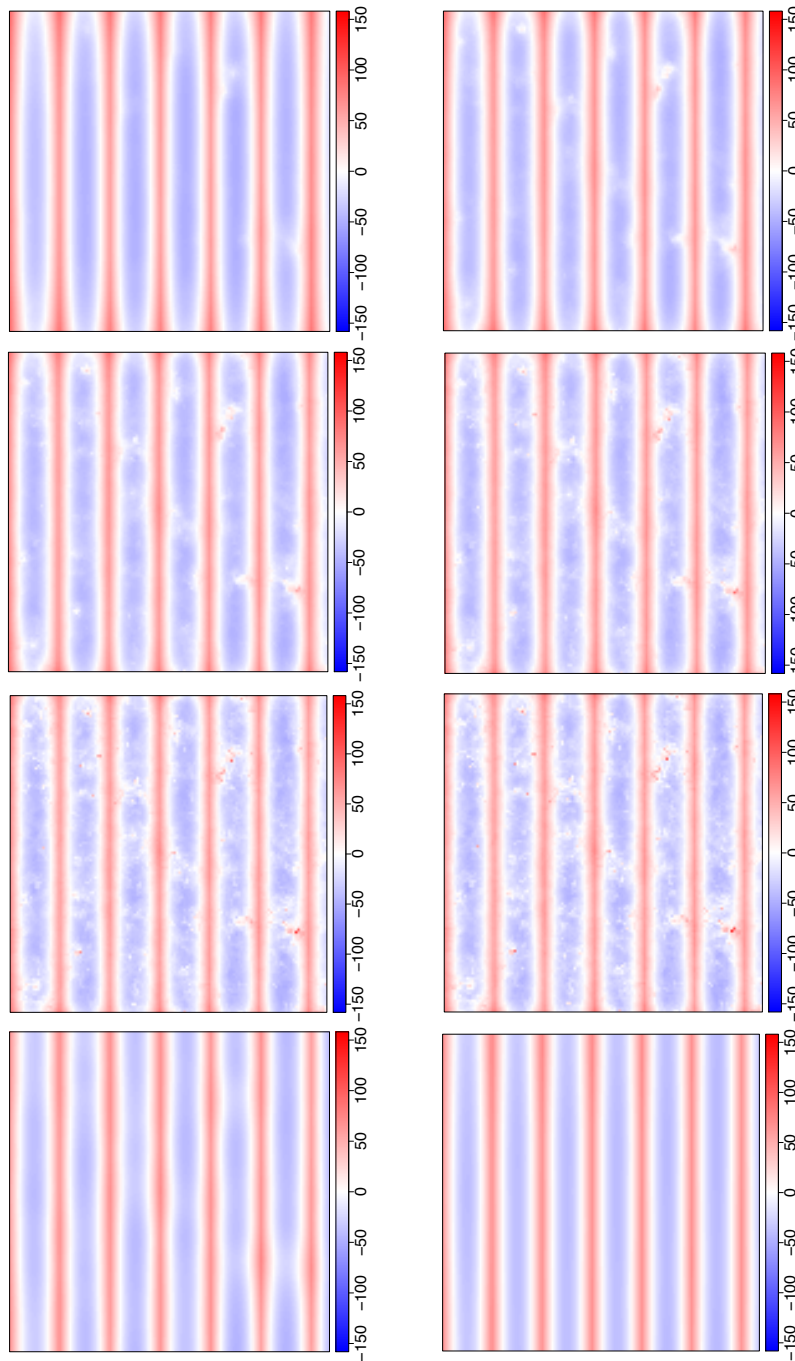


Figure 3.8: Estimated bias for  $\hat{\rho}_{p,m}^V(u)$ ,  $u \in W = [0, 1]^2$ ,  $m = 200$ , based on 500 realisations of a log-Gaussian Cox process  $X \subseteq W = [0, 1]^2$  where the driving Gaussian random field has mean function  $(x, y) \mapsto \log(40|\sin(20x)|)$  and  $((x_1, y_1), (x_2, y_2)) \mapsto 2 \exp\{-\|(x_1, y_1) - (x_2, y_2)\|/0.1\}$  as covariance function. From top-left to bottom-right:  $\hat{\lambda}_{p,m}^V(u)$  with  $p = 0.1, 0.3, 0.5, 0.7, 0.9, 1$ , and  $\hat{\lambda}^U(u)$  using bandwidth selection (1.12) and (1.13).

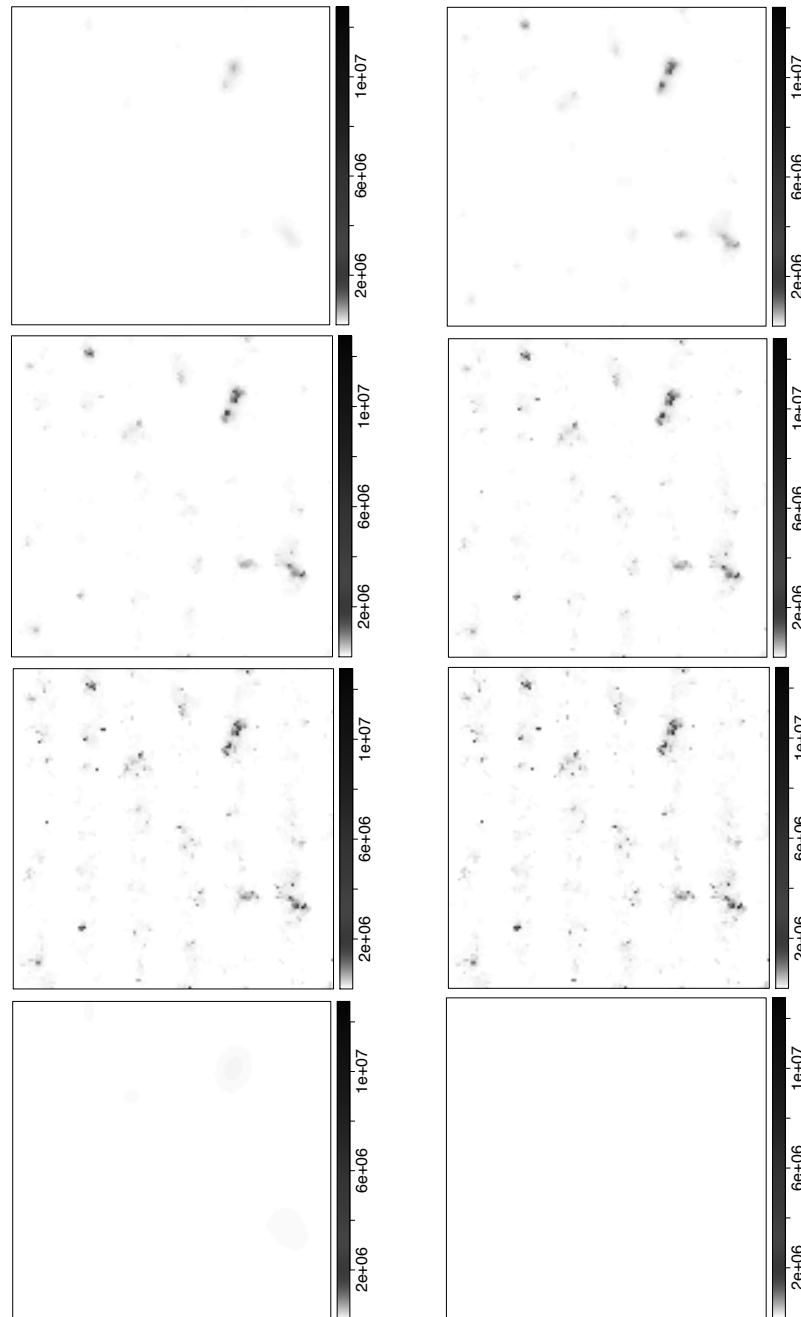


Figure 3.9: Estimated variance for  $\hat{\rho}_{p,m}^V(u)$ ,  $u \in W = [0, 1]^2$ ,  $m = 200$ , based on 500 realisations of a log-Gaussian Cox process  $X \subseteq W = [0, 1]^2$  where the driving Gaussian random field has mean function  $(x, y) \mapsto \log(40|\sin(20x)|)$  and  $((x_1, y_1), (x_2, y_2)) \mapsto 2 \exp\{-\|(x_1, y_1) - (x_2, y_2)\|/0.1\}$  as covariance function. From top-left to bottom-right:  $\hat{\lambda}_{p,m}^V(u)$  with  $p = 0.1, 0.3, 0.5, 0.7, 0.9, 1$ , and  $\hat{\lambda}^U(u)$  using bandwidth selection (1.12) and (1.13).

Table 3.7: Estimates of IAB, ISB and IV for  $\widehat{\rho}_{p,m}^V(u)$ ,  $u \in W = [0, 1]^2$ ,  $m = 200, 300, 400$ ,  $p = 0.1, \dots, 1$ , based on 500 realisations of an independently thinned simple sequential inhibition process in  $W = [0, 1]^2$  with intensity  $\rho(x, y) = 450p(x, y)$ ,  $p(x, y) = \mathbf{1}\{x < 1/3\}|x - 0.02| + \mathbf{1}\{1/3 \leq x < 2/3\}|x - 0.5| + \mathbf{1}\{x \geq 2/3\}|x - 0.95|$ ,  $x, y \in W$ .

$\begin{matrix} m \\ p \end{matrix}$	IAB			ISB			IV		
	200	300	400	200	300	400	200	300	400
.1	32.4	32.4	32.4	1502.2	1502.7	1502.2	109.4	105.9	103.4
.2	31.2	31.2	31.2	1385.7	1385.2	1384.5	176.2	173.8	172.2
.3	29.2	29.2	29.2	1223.6	1223.0	1222.8	253.4	251.2	250.3
.4	27.0	27.0	27.0	1060.4	1060.7	1060.3	348.8	345.3	345.3
.5	25.0	25.0	25.0	919.5	919.8	920.6	457.3	455.6	454.1
.6	23.1	23.1	23.1	803.3	803.3	803.0	584.4	582.7	581.9
.7	21.5	21.5	21.5	707.9	707.7	707.8	734.2	733.9	732.8
.8	20.0	20.1	20.1	628.5	628.9	629.1	916.3	914.2	913.4
.9	18.9	18.9	18.9	567.2	567.5	567.7	1120.5	1118.5	1117.5
1	24.7	24.7	24.7	852.3	852.3	852.3	1382.4	1382.4	1382.4

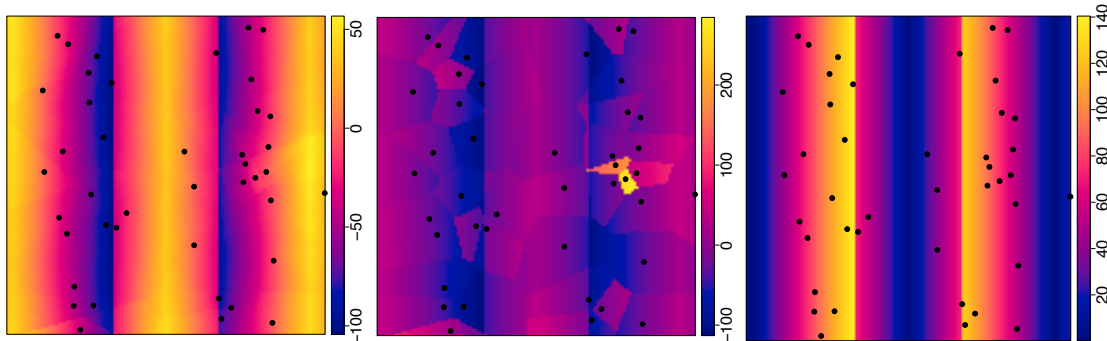


Figure 3.10: True intensity and estimation error plots for a realisation of an independently thinned simple sequential inhibition process in  $W = [0, 1]^2$  with intensity  $\rho(x, y) = 450p(x, y)$ ,  $p(x, y) = \mathbf{1}\{x < 1/3\}|x - 0.02| + \mathbf{1}\{1/3 \leq x < 2/3\}|x - 0.5| + \mathbf{1}\{x \geq 2/3\}|x - 0.95|$ ,  $x, y \in W$ . *Left*:  $p = 0.2$  and  $m = 200$ . *Middle*:  $p = 1$ . *Right*: True intensity.

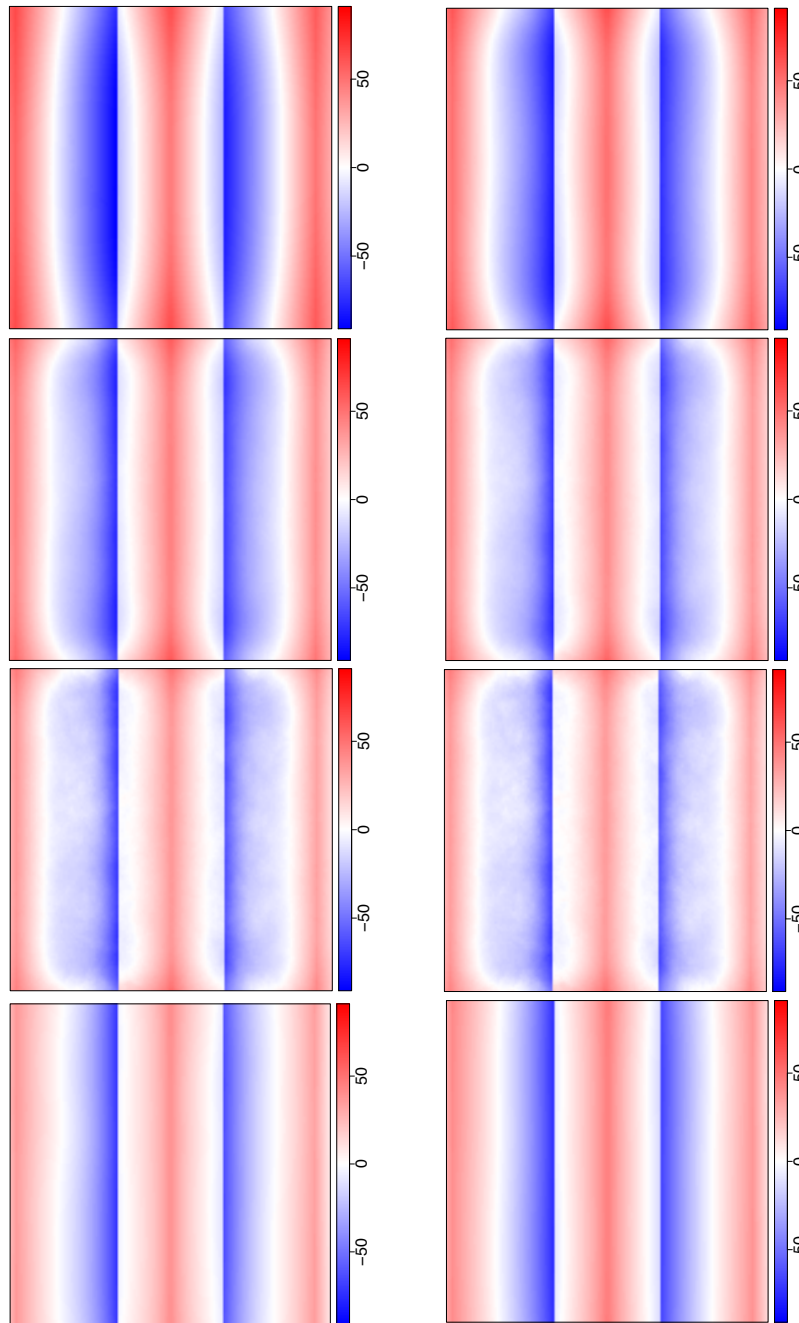


Figure 3.11: Estimated bias for  $\hat{\rho}_{p,m}^V(u)$ ,  $u \in W = [0, 1]^2$ ,  $m = 200$ , based on 500 realisations of an independently thinned simple sequential inhibition process in  $W = [0, 1]^2$  with intensity  $\rho(x, y) = 450p(x, y)$ ,  $p(x, y) = \mathbf{1}\{x < 1/3\}|x - 0.02| + \mathbf{1}\{1/3 \leq x < 2/3\}|x - 0.5| + \mathbf{1}\{x \geq 2/3\}|x - 0.95|$ ,  $x, y \in W$ . From top-left to bottom-right:  $\hat{\lambda}_{p,m}^V(u)$  with  $p = 0.1, 0.3, 0.5, 0.7, 0.9, 1$ , and  $\hat{\lambda}^U(u)$  using bandwidth selection (1.12) and (1.13).

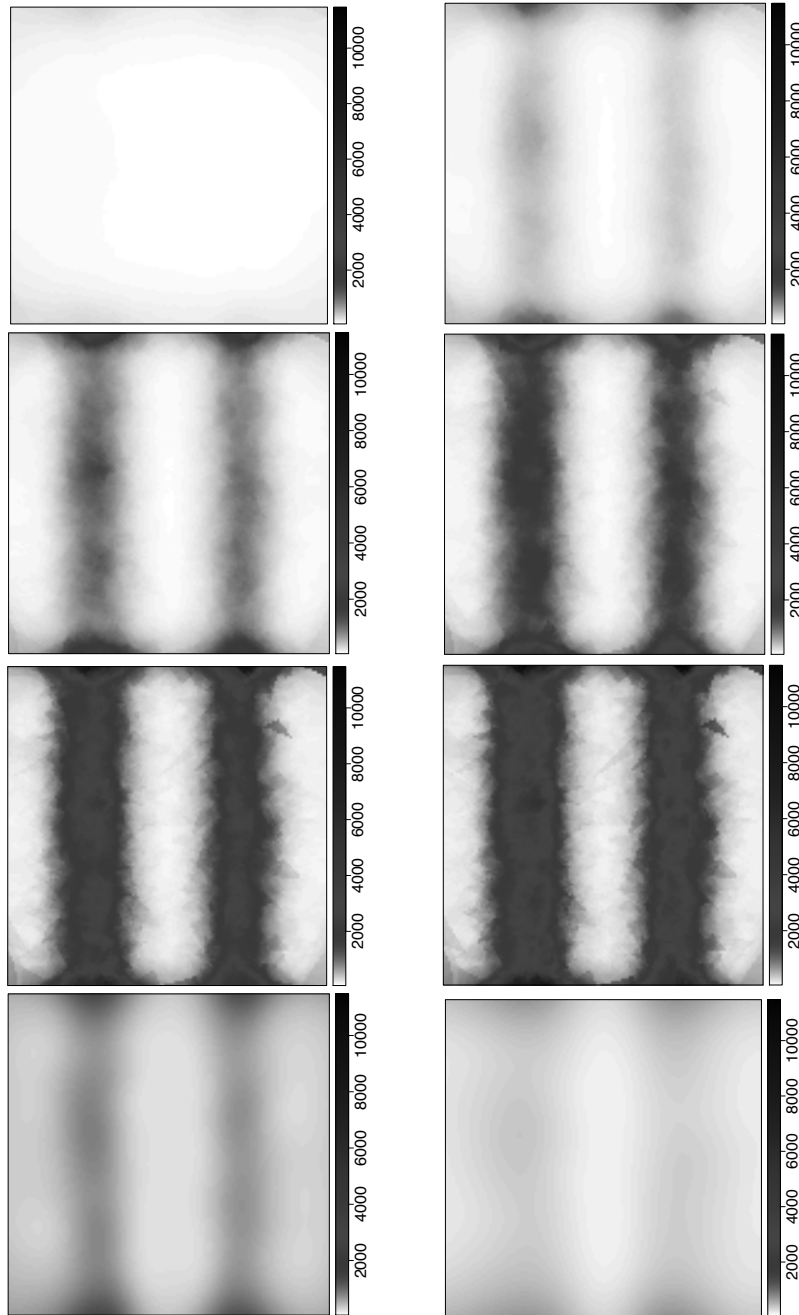


Figure 3.12: Estimated variance for  $\hat{\rho}_{p,m}^V(u)$ ,  $u \in W = [0, 1]^2$ ,  $m = 200$ , based on 500 realisations of an independently thinned simple sequential inhibition process in  $W = [0, 1]^2$  with intensity  $\rho(x, y) = 450p(x, y)$ ,  $p(x, y) = \mathbf{1}\{x < 1/3\}|x - 0.02| + \mathbf{1}\{1/3 \leq x < 2/3\}|x - 0.5| + \mathbf{1}\{x \geq 2/3\}|x - 0.95|$ ,  $x, y \in W$ . From top-left to bottom-right:  $\hat{\lambda}_{p,m}^V(u)$  with  $p = 0.1, 0.3, 0.5, 0.7, 0.9, 1$ , and  $\hat{\lambda}^U(u)$  using bandwidth selection (1.12) and (1.13).

Table 3.8: Cross-validation selections of  $p$  in a geometric sequence for  $m = 200$ , based on 100 realisations of an independently thinned simple sequential inhibition process in  $W = [0, 1]^2$  with intensity  $\rho(x, y) = 450p(x, y)$ ,  $p(x, y) = \mathbf{1}\{x < 1/3\}|x - 0.02| + \mathbf{1}\{1/3 \leq x < 2/3\}|x - 0.5| + \mathbf{1}\{x \geq 2/3\}|x - 0.95|$ ,  $x, y \in W$ .

$p$	0.10	0.13	0.18	0.24	0.33	0.44	0.59	0.80
Frequency	24	3	3	2	6	13	21	28

### 3.5 Houston traffic accident

The dataset consists of motor vehicle traffic accident locations in a given area of Houston, US, during the month of April 1999. The linear network  $L$  describing the street/road network in question (see Figure 3.13) has a total length of 708301.7 feet, and has 187 nodes (road intersections) with a maximum vertex degree of 4, and 253 line segments, i.e. pieces of streets connecting the intersections.

Figure 3.13 (left) shows the reference points of the 249 accidents in question over the street network. The data have been collected by individual police departments in the Houston metropolitan area and have been compiled by the Texas Department of Public Safety. The compiled data have been obtained by the Houston-Galveston Area Council and have been geocoded by N. Levine. Between 1999 and 2001, in the eight-county region considered, there were 252,241 serious accidents, with an average of 84,080 per year. From these accidents, 1,882 were person related. See Levine (2006, 2009) for details.

In Figure 3.13 (right) we also provide the resample-smoothed Voronoi intensity estimate obtained for  $m = 200$  and  $p = 0.15$ . The specific choice  $p = 0.15$  has been motivated by the rule-of-thumb  $p \in [0.1, 0.3]$  and Table 3.9, which shows the selected values for  $p \in (0, 1]$  obtained by carrying out cross-validation for the sequence  $m = 100, 110, \dots, 200$ . We see that most of the selected values for  $p$  are given by 0.15.

Visually, there seems to be a good correspondence between the observed pattern and the obtained estimate. Note that for higher values of  $p$ , in the right panel of Figure 3.13 we would obtain more significant blobs in the parts corresponding to the dense parts in the left panel of Figure 3.13.

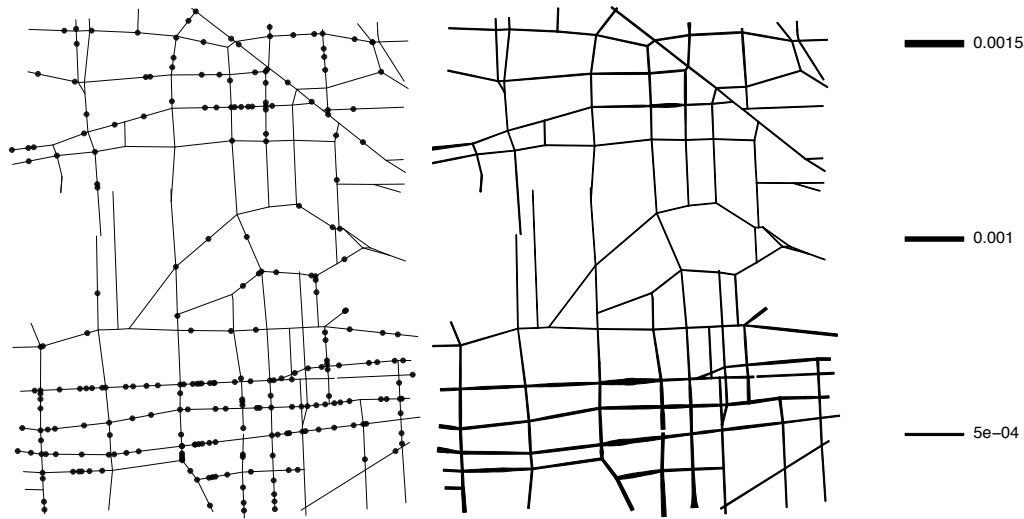


Figure 3.13: *Left*: Motor vehicle traffic accidents in an area of Houston, US, during April, 1999. *Right*: Resample-smoothed Voronoi intensity estimate for  $m = 200$  and  $p = 0.15$ .

Table 3.9: Cross-validation selected values for  $p$ , based on the sequence  $m = 100, 110, \dots, 200$ .

$m$	100	110	120	130	140	150	160	170	180	190	200
$p$	0.15	0.20	0.20	0.20	0.20	0.15	0.15	0.15	0.15	0.15	0.15

## 3.6 Finnish pines

The dataset consists of locations of 126 pine saplings in a Finnish forest, within a rectangular window  $W = [-5, 5] \times [-8, 2]$  (metres), and can be found in the R package **spatstat** (Baddeley et al., 2015). It was recorded by Seppo Kellomaki, Faculty of Forestry, University of Joensuu, Finland, and further processed by Antti Penttinen, Department of Statistics, University of Jyväskylä, Finland.

In Figure 3.14 we illustrate the estimate  $\hat{\rho}_{p,m}^V(u)$ ,  $u \in W$ ,  $m = 200$ , for  $p = 0.2$  and  $p = 0.5$ , together with the locations of the saplings. We also provide the cross-validation results for the sequence  $m = 100, 110, \dots, 200$  in Table 3.10 which motivates the choice  $p = 0.5$ . We argue that  $p = 0.2$  is the preferable choice since it better respects the global features of the data.



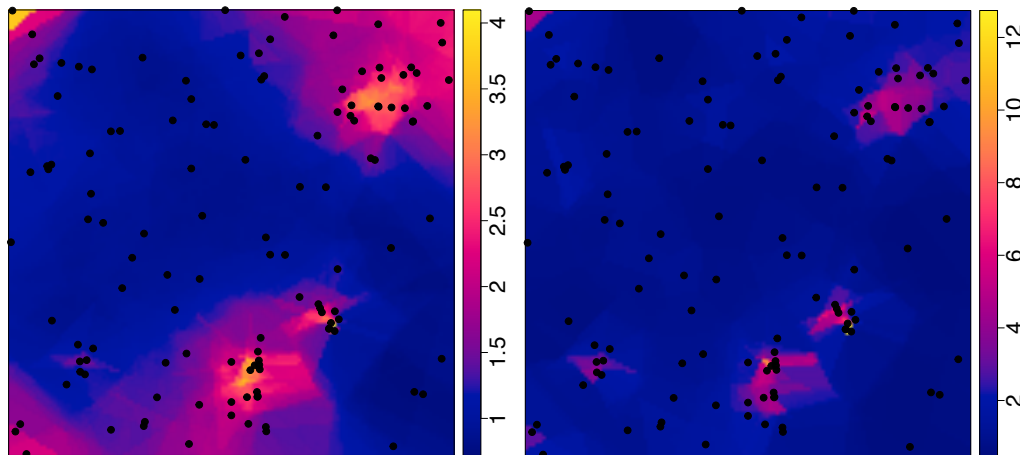


Figure 3.14: The estimate  $\hat{\rho}_{p,m}^V(u)$ ,  $u \in W$ ,  $m = 200$ , for  $p = 0.2$  (left) and  $p = 0.5$  (right), together with the locations of 126 pine saplings in a Finnish forest, within a rectangular window  $W = [-5, 5] \times [-8, 2]$  (metres).

Table 3.10: Cross-validation selected values for  $p$ , based on the sequence  $m = 100, 110, \dots, 200$ .

$m$	100	110	120	130	140	150	160	170	180	190	200
$p$	0.65	0.40	0.50	0.45	0.40	0.50	0.45	0.55	0.50	0.50	0.50

### 3.7 Summary

Although the Voronoi intensity estimator estimates the intensity function by considering the local effects and it does represent the local behaviour better than kernel estimators, it however generates a huge variance. In this chapter, we focused on the Voronoi intensity estimator and provided a new technique to improve its performance in terms of variance.

We have proposed a general approach for resampling/additional smoothing of arbitrary intensity estimators, which is based on averaging over rescaled intensity estimators generated by a set of thinned samples. The point pattern in question is thinned according to some independent  $p$ -thinning. Although such technique can be easily applied to Voronoi intensity estimates regardless of the state space, we considered point processes on  $\mathbb{R}^2$  and linear networks. The proposed estimator

is well defined and its smoothing parameter selection was also discussed. A rule-of-thumb is also provided by considering retention probability  $p$  within the range  $[0.1 - 0.3]$ .

We studied the theoretical behaviour of such intensity estimator such as unbiasedness and variance. A simulation study was drawn to see its performance when dealing with homogeneous/inhomogeneous Poisson processes, Log-Gaussian Cox point processes and thinned simple sequential inhibition processes. It is clearly shown that the variance decreases when retention probability  $p$  decreases. However, the bias (estimation error) of the proposed estimator seems fairly stable.

So far, we have only considered spatial point processes whereas sometimes data points are labelled with the time occurrence of the event. This demands spatio-temporal analysis which will be discussed in the next Chapter.

## CHAPTER 4

---

# Spatio-temporal point patterns on linear networks

---

## 4.1 Introduction

Naturally, most of spatial point patterns are snapshots of spatio-temporal patterns occurring along time. Sometimes (as explained in Chapters 2 and 3), we only consider the spatial domain and analyse point patterns regardless of time while they are inherently happening jointly in space and time. In the case of spatio-temporal point processes on linear networks, the same story holds. Thinking of traffic accidents or street crimes that happen on streets of cities, we easily see that many accidents happen in particular moments of a day, which depends on time. Also some crimes, as robberies, happen in the sidewalk when it is crowded. Knowing that there may be scientific questions that can not be answered by only analysing spatial components (not considering time) (Diggle, 2003), it prompts to consider the time component when analysing point patterns on networks.

The challenge when we think on network events is that we face a change of support, i.e. events do not happen on the entire space. Since the location of events is restricted to a linear network, clearly existing methods for spatio-temporal point processes on  $\mathbb{R}^2 \times \mathbb{R}^+$  can not be appropriate. Considering a network as state-space instead of the entire space requires modifications in the whole story.

Although network events got attention recently (see Chapters 2 and 3), temporal components have not been taken into account yet. Knowing that the behaviour of

spatial data may vary over time since they are spatio-temporal patterns inherently (considering the occurrence time of events), we here aim at considering the temporal component and develop methods to be able to study spatio-temporal point processes on linear networks. In this Chapter, we develop a kernel-based intensity estimator to estimate the intensity function of spatio-temporal point patterns on networks. We then present spatio-temporal second-order summary statistics, the  $K$ -function and the pair correlation function, which are useful to study the correlation between events. They are independent of the geometry of the network and have known values for Poisson point processes. Such summary statistics can be used in model selection as benchmarks due to having known forms under completely spatio-temporal randomness. Non-parametric estimates are also provided for both homogeneous and inhomogeneous processes.

Section 4.2 provides some definitions and preliminaries of spatio-temporal point processes on linear networks. Section 4.3 is devoted to the first- and second-order tools of spatio-temporal point processes on linear networks and presents the corresponding definitions, estimators and properties. In Section 4.4 we consider some applications to traffic accidents and demonstrate our findings from datasets of Houston (US), Medellín (Colombia) and Eastbourne (UK).

## 4.2 Setup

As explained in the Chapters 2 and 3, point processes on linear networks are recently considered to analyse events that are happening on/along a network, such as traffic accidents, car thefts, assaults and so forth. Suppose that the reference locations are labelled by the time of occurrence. We can then consider these events as spatio-temporal point patterns on linear networks. Any spatio-temporal point process  $X$  on a linear network  $L \subset \mathbb{R}^2$  is a random countable subset of  $\mathbb{R}^2 \times \mathbb{R}^+$  in which the location of events is restricted to lying on a network structure  $L$ . As the location of events is limited to the network  $L$  and also, the travel distance is somehow different from the Euclidean distance, such events might not be analysed properly through more classical statistical methods for spatio-temporal point processes in  $\mathbb{R}^2 \times \mathbb{R}^+$ ; note the work in Lu and Chen (2007). Examples of spatial point patterns on linear networks are already analysed in the Chapters 2 and 3. Considering the setup of spatial point processes on linear networks, we directly focus on a spatio-temporal point process  $X$  on the network  $L$  with

no overlapping points  $(x, t_x)$  where  $x \in L$  is the location of an event and  $t_x \in T$  ( $T \subset \mathbb{R}^+$ ) is the corresponding time occurrence of  $x$ . Hence, a realisation of  $X$ , say  $\mathbf{x}$ , with  $n$  events can be represented as  $\mathbf{x} = \{(x_i, t_{x_i}), i = 1, \dots, n\}$  where each  $(x_i, t_{x_i}) \in L \times T$ . Naturally, most spatial processes exhibit a temporal behaviour while it has often been neglected. We here aim at bringing the temporal behaviour on board and engage it in analysing network events. There may be questions that we can not answer by only analysing the spatial behaviour (Diggle, 2003) of events, e.g. the number of traffic accidents may vary over time (hour, day, week, etc). Note that the temporal state-space  $T$  can be either a continuous or a discrete set.

A realisation  $\mathbf{x}$  of a spatio-temporal process  $X$  on a linear network  $L$  and within a bounded temporal period  $T$  is formed by a finite set of distinct points  $(x_i, t_{x_i}) \in L \times T$ . A spatio-temporal disc with centre  $(x_i, t_{x_i}) \in L \times T$ , network radius  $r > 0$  and temporal radius  $t > 0$  is defined as

$$b_{LT}((x_i, t_{x_i}), r, t) = \{(u, t_u) \in L \times T : d_L(x_i, u) \leq r \text{ and } |t_{x_i} - t_u| \leq t\},$$

where  $d_L$  denotes the shortest-path distance on the linear network  $L$  and  $|\cdot|$  is a numerical distance. The cardinality of the subset  $A \subset L \times T$ ,  $N(X \cap A) \in \{0, 1, \dots\}$ , is the number of points restricted to  $A$ , whose expected number is denoted by

$$\mu(A) = \mathbb{E}[N(A)], \quad A \subset L \times T.$$

Assume that  $X$  has an intensity function  $\lambda(\cdot)$  and a second-order product density function  $\lambda_2(\cdot, \cdot)$ , hence,

$$\mathbb{E}[N(X \cap A)] = \int_A \lambda(u, t_u) d_2(u, t_u), \quad A \subset L \times T,$$

where  $d_2(u, s)$  denotes the integration over  $L \times T$ , and

$$\mathbb{E}[N(X \cap A)N(X \cap B)] = \int_A \int_B \lambda_2((u, t_u), (v, t_v)) d_2(u, t_u) d_2(v, t_v), \quad A, B \subset L \times T.$$

For a homogeneous process,  $\lambda(\cdot, \cdot)$  takes a constant value  $\lambda$  for any point in  $L \times T$ , i.e. the events are spread out over  $L \times T$  uniformly. Following Campbell's theorem (1.26), for any non-negative function  $f : (L \times \mathbb{R}^+) \times (L \times \mathbb{R}^+) \rightarrow [0, \infty)$ ,

$$\begin{aligned} & \mathbb{E} \left[ \sum_{\substack{\neq \\ (x, t_x), (y, t_y) \in X}} f((x, t_x), (y, t_y)) \right] \\ &= \int \int f((u, t_u), (v, t_v)) \lambda_2((u, t_u), (v, t_v)) d_2(u, t_u) d_2(v, t_v), \end{aligned}$$

where  $\sum^{\neq}$  means that the sum is over distinct pairs of points (Diggle, 2003). Nevertheless, one can still work with the projection of  $X$  onto the linear network  $L$  and the time interval  $T$ , which are given as

$$X_{\text{net}} = \{x : (x, t_x) \in X\}, \quad X_{\text{time}} = \{t_x : (x, t_x) \in X\},$$

where  $X_{\text{net}}$  defines a spatial point process on  $L$ , and  $X_{\text{time}}$  stands for a temporal point process on  $T$  (Møller and Ghorbani, 2012; Diggle, 2003). Note that  $X_{\text{net}}$  and  $X_{\text{time}}$  may no longer be simple point processes in the sense of having no overlapping points. Then, the corresponding intensity functions of  $X_{\text{net}}$  and  $X_{\text{time}}$  can be obtained as

$$\lambda_{\text{net}}(u) = \int_T \lambda(v, t_v) dt_v, \quad \lambda_{\text{time}}(t_u) = \int_L \lambda(v, t_v) d_1 v \quad (4.1)$$

where  $d_1 u$  denotes one-dimensional integration over the line segment (Federer, 1996).

The Poisson point process model is crucial in defining other more complicated models and can be considered as a benchmark model in exploratory data analysis. Definition of a spatio-temporal Poisson process on  $L \times T$  is similar to that on  $\mathbb{R}^2 \times T$ . Thus, for a spatio-temporal Poisson point process, the points  $(x, t_x)$  on  $L \times T$  need to satisfy the following conditions:

- In any bounded set  $A \subset L \times T$ ,  $N(X \cap A)$  follows a Poisson distribution with expected number  $\int_A \lambda(u, t_u) d_2(u, t_u)$ . For instance, if we assume  $A = A_L \times A_T \subset L \times T$ , then the expected value is the expected number of points in the subnetwork  $A_L$  and within the time interval  $A_T$ .
- For any  $k$  arbitrary disjoint subsets of  $L \times T$ , say  $A_1, \dots, A_k$ , their numbers of points  $N(X \cap A_1), \dots, N(X \cap A_k)$  are independent variables.

Moreover, for any time interval  $S = [s_1, s_2] \subset T$ , the projection of  $X$  onto  $L$  defines a spatial Poisson process on  $L$  with intensity function  $\int_{s_1}^{s_2} \lambda(u, t_u) dt_u$ . Similarly, for any set of segments  $A \subset L$ , the projection of  $X$  onto  $T$  defines a one-dimensional Poisson process with intensity function  $\int_A \lambda(u, t_u) d_1 u$  (Illian et al., 2008). For more details, see Diggle (2003, Chapter 10) and Baddeley et al. (2015, Chapter 17).

We note that here we need to be able to perform integration of functions  $f$  on  $L \times T$ . To do so, we here follow Federer (1996) and Ang et al. (2012), converting the integration over  $L \times T$  to the one on  $\mathbb{R}^+ \times \mathbb{R}^+$ .

For any measurable function  $f : L \times T \rightarrow \mathbb{R}$ ,

$$\int_{L \times T} f(v, t_v) d_2(v, t_v) = \int_0^\infty \int_0^\infty \sum_{(v, t_v) \in L \times T: d_L(u, v) = r \text{ \& } |t_u - t_v| = t} f(v, t_v) dr dt. \quad (4.2)$$

For instance, consider

$$f(v, t_v) = h(d_L(u, v), |t_u - t_v|),$$

then,

$$\int_{L \times T} h(d_L(u, v), |t_u - t_v|) d_2(v, t_v) = \int_0^\infty \int_0^\infty h(r, t) M((u, t_u), r, t) dr dt,$$

where  $M((u, t_u), r, t)$  is the number of points lying exactly at the shortest-path distance  $r \geq 0$  and the time distance  $t \geq 0$  away from  $(u, t_u)$  (Moradi et al., 2018b).

## 4.3 Methodologies

This Section is devoted to first- and second-order characteristics of a spatio-temporal point process  $X$  on  $L \times T$ .

### 4.3.1 First-order characteristics

Similar to spatial analysis, a very first attempt to understand the behaviour of events is to analyse whether the distribution of a spatio-temporal point process  $X$  varies over its corresponding state-space  $L \times T$ . We then firstly need to study the intensity function  $\lambda(\cdot, \cdot)$  so that a good estimate can reveal the more populated parts in both the linear network and the temporal range. We here point out that the intensity function  $\lambda(u, t_u)$  can also be interpreted as a conditional spatial intensity given any time  $t$  or a conditional temporal intensity given any subnetwork of  $L$  (Diggle, 2003).

As a general assumption, and following Gabriel and Diggle (2009), Møller and Ghorbani (2012) and Ghorbani (2013), we consider that the intensity function  $\lambda(u, t_u)$  is separable, i.e.,

$$\lambda(u, t_u) = \bar{\lambda}_1(u) \bar{\lambda}_2(t_u), \quad (u, t_u) \in L \times T \quad (4.3)$$

where  $\bar{\lambda}_1$  and  $\bar{\lambda}_2$  are non-negative functions on  $L$  and  $T$ , respectively. As a result of (4.1) and (4.3),

$$\lambda_{\text{net}}(u) = \bar{\lambda}_1(u) \int_T \bar{\lambda}_2(t_u) dt_u, \quad \lambda_{\text{time}}(t_u) = \bar{\lambda}_2(t_u) \int_L \bar{\lambda}_1(u) d_1 u.$$

Then, the intensity function  $\lambda(u, t_u)$  can be rewritten as

$$\lambda(u, t_u) = \frac{\lambda_{\text{net}}(u)\lambda_{\text{time}}(t_u)}{\int_{L \times T} \lambda(u, t_u) d_2(u, t_u)}. \quad (4.4)$$

Note that if  $X$  is a homogeneous point process then all  $\lambda$ ,  $\lambda_{\text{net}}$  and  $\lambda_{\text{time}}$  take constant values. However, we here do not assume homogeneity, i.e. we assume that the intensity function varies over  $L \times T$ . Taking the count function  $N$  into account, the denominator in (4.4) equals the expected number of points in  $L \times T$ . Suppose we are given  $\hat{\lambda}_{\text{net}}(u)$  and  $\hat{\lambda}_{\text{time}}(t_u)$  so that they result unbiased estimators for the expected number of observed points, say  $n$ , then it is not surprising that

$$\hat{\lambda}(u, t_u) = \frac{\hat{\lambda}_{\text{net}}(u)\hat{\lambda}_{\text{time}}(t_u)}{n}, \quad (u, t_u) \in L \times T, \quad (4.5)$$

provides an unbiased estimator for the expected number of observed points, i.e.

$$\mathbb{E} \left[ \int_{L \times T} \hat{\lambda}(u, t_u) d_2(u, t_u) \right] = \int_{L \times T} \lambda(u, t_u) d_2(u, t_u) = \mathbb{E} [N(L \times T)] = n,$$

meaning that the intensity estimator (4.5) preserves mass. Therefore, and in order to estimate the intensity function, we follow the idea in (4.5), knowing that this kind of estimation has already been proposed in the literature, see Gabriel and Diggle (2009); Møller and Ghorbani (2012); Ghorbani (2013).

In order to estimate  $\lambda_{\text{net}}(u)$  non-parametrically, one can consider equal-split kernel density estimators (2.2) defined by Okabe et al. (2009), a heat-kernel estimator (2.13) introduced by McSwiggan et al. (2017), an adapted Jones-Diggle estimator (2.7) by Moradi et al. (2017), the quick kernel-based method (2.14) or (2.15) proposed by Rakshit et al. (2018) or resample-smoothed Voronoi intensity estimator (3.3) defined by Moradi et al. (2018a). Note also the usability of adaptive kernel intensity estimators. For details of the aforementioned methods, see the Chapters 2 and 3. Here we use the quick kernel-based estimator developed in Rakshit et al. (2018) to estimate  $\lambda_{\text{net}}(u)$  and the usual kernel smoothing technique to estimate  $\hat{\lambda}_{\text{time}}(t_u)$ . The bandwidth for the estimation of  $\lambda_{\text{net}}(u)$  will be selected by Scott's rule of thumb (Rakshit et al., 2018) and the bandwidth to estimate  $\lambda_{\text{time}}(t_u)$  will be chosen using Silverman's rule of thumb (Silverman, 1986).

### 4.3.2 Homogeneous second-order characteristics

Additionally to the intensity function, one may think of second-order summary statistics such as the  $K$ -function and the pair correlation function to get an insight



into the type of interaction between the events. In such analysis, it is often assumed that the point process, a realisation of which is under description, is stationary (pseudostationary). We here begin with second-order pseudostationary point processes, and in Section 4.3.3 we turn to the inhomogeneous case which is usually more suitable for analysing real datasets.

**Definition 4.1.** Assume  $X$  is a spatio-temporal point process on  $L \times T$  with constant intensity function  $\lambda > 0$ . Then, the homogeneous linear  $K$ -function is given by

$$K_L^{ST}((u, t_u), r, t) = \frac{1}{\lambda} \mathbb{E} \left[ \sum_{(x, t_x) \in X} \frac{\mathbf{1}\{0 < d_L(u, x) \leq r, |t_u - t_x| \leq t\}}{M((u, t_u), d_L(u, x), |t_u - t_x|)} \mid (u, t_u) \in X \right] \quad (4.6)$$

for all  $r, t \geq 0$  that satisfy  $M((u, t_u), r, t) > 0$ . Then,  $X$  is called second-order pseudostationary if  $K_L^{ST}((u, t_u), r, t)$  does not depend on the point  $(u, t_u)$ , and we then write  $K_L^{ST}((u, t_u), r, t) = K_L^{ST}(r, t)$  which is an extension of the geometrically corrected  $K$ -function to the context of spatio-temporal point processes on a linear network  $L$ .

For details on the geometrically corrected  $K$ -function for spatial point processes on a linear network  $L$ , see Section 1.3.

**Theorem 4.3.1.** For a homogeneous Poisson point process on  $L \times T$  with constant intensity function  $\lambda$ ,  $K_L^{ST}((u, t_u), r, t) = rt$ .

*Proof.* Note that the expectation in the right-hand side of (4.6) is an expectation with respect to the Palm distribution of the point process  $X$  on  $L \times T$ . For a Poisson process  $X$ , Slivnyak's theorem says that the reduced Palm distribution at any location  $(u, t_u)$  is identical to the distribution of  $X$  (Møller and Waagepetersen, 2003; Baddeley et al., 2015). Then by using the transformation in (4.2) and Campbell's theorem,

$$\begin{aligned} \lambda K_L^{ST}((u, t_u), r, t) &= \mathbb{E} \left[ \sum_{(x, t_x) \in X} \frac{\mathbf{1}\{0 < d_L(u, x) \leq r, |t_u - t_x| \leq t\}}{M((u, t_u), d_L(u, x), |t_u - t_x|)} \mid (u, t_u) \in X \right] \\ &= \mathbb{E} \left[ \sum_{(x, t_x) \in X} \frac{\mathbf{1}\{0 < d_L(u, x) \leq r, |t_u - t_x| \leq t\}}{M((u, t_u), d_L(u, x), |t_u - t_x|)} \right] \\ &= \lambda \int_{L \times T} \frac{\mathbf{1}\{0 < d_L(u, x) \leq r, |t_u - t_x| \leq t\}}{M((u, t_u), d_L(u, x), |t_u - t_x|)} d_2(u, t_u) \\ &= \lambda rt. \end{aligned}$$

□

This can then be used as a benchmark in model selection to see the deviation from the Poisson model. Assume that the second-order product density function  $\lambda_2(\cdot, \cdot)$  exists, then the pair correlation function of a spatio-temporal point process  $X$  with constant intensity function  $\lambda$  on  $L \times T$  is of the form

$$g_L^{ST}((u, t_u), (v, t_v)) = \frac{\lambda_2((u, t_u), (v, t_v))}{\lambda^2}, \quad (u, t_u), (v, t_v) \in L \times T, \quad (4.7)$$

which is the standardised probability function of observing a pair of points from  $X$  occurring jointly in each of two infinitesimally small areas around  $(u, t_u)$  and  $(v, t_v)$  (Gabriel and Diggle, 2009; Møller and Ghorbani, 2012). Note that here a small area around  $(u, t_u)$  is a small segment (sub-network) extended over a short period of time. Following Gabriel and Diggle (2009) and Ang et al. (2012), it can be shown that for Poisson processes and any pairs  $(u, t_u), (v, t_v) \in L \times T$ ,  $g_L^{ST}((u, t_u), (v, t_v)) = 1$  as  $\lambda_2((u, t_u), (v, t_v)) = \lambda^2$ .

Since both the  $K$ -function and the pair correlation function have certain values for Poisson processes, they can be used to reveal any deviation from being Poisson, that is,  $K_L^{ST}(r, t) > rt$  ( $g_L^{ST}(r, t) > 1$ ) shows clustering, whereas  $K_L^{ST}(r, t) < rt$  ( $g_L^{ST}(r, t) < 1$ ) indicates inhibition.

As  $K_L^{ST}$  and  $g_L^{ST}$  are closely connected, the following theorem shows their relationship in the context of spatio-temporal point patterns on linear networks.

**Theorem 4.3.2.** *A spatio-temporal point process  $X$  on  $L \times T$  with constant intensity function  $\lambda > 0$  is second-order pseudostationary if for any pairs  $(u, t_u), (v, t_v) \in L \times T$  with  $d_L(u, v) < \infty, |t_u - t_v| < \infty$ , the pair correlation function  $g_L^{ST}((u, t_u), (v, t_v))$  depends only on the distance vector  $(d_L(u, v), |t_u - t_v|)$ , and in this case*

$$K_L^{ST}(r, t) = \int_0^r \int_0^t g_L^{ST}(r', t') dr' dt', \quad (4.8)$$

where  $r$  denotes the shortest-path distance, and  $t$  is the corresponding temporal distance.

*Proof.* Assume  $X$  is second-order pseudostationary, and

$$g_0(r', t') = \frac{\mathbf{1}\{r' < r, t' < t\}}{M((u, t_u), r', t')}. \quad (4.9)$$

Then,

$$K_L^{ST}(r, t) = \frac{1}{\lambda} \mathbb{E} \left[ \sum_{(x, t_x) \in X} g_0(d_L(u, x), |t_u - t_x|) \mid (u, t_u) \in X \right].$$

By using the properties of Palm distributions, the above equation can be rewritten as follows,

$$\begin{aligned} K_L^{ST}(r, t) &= \frac{1}{\lambda} \int_{L \times T} \frac{\lambda_2((u, t_u), (v, t_v))}{\lambda} g_0(d_L(u, v), |t_u - t_v|) d_2(v, t_v) \\ &= \int_{L \times T} g_L^{ST}((u, t_u), (v, t_v)) g_0(d_L(u, v), |t_u - t_v|) d_2(v, t_v) \\ &= \int_{L \times T} g_L^{ST}(d_L(u, v), |t_u - t_v|) g_0(d_L(u, v), |t_u - t_v|) d_2(v, t_v), \end{aligned}$$

where the last equation comes from the second-order pseudostationary property of  $X$ . Now, let

$$\begin{aligned} h(r', t') &= g_L^{ST}(r', t') g_0(r', t') M((u, t_u), r', t') \\ &= g_L^{ST}(r', t') \mathbf{1}\{r' < r, t' < t\}, \end{aligned}$$

Then,

$$\begin{aligned} K_L^{ST}(r, t) &= \int_{L \times T} \frac{h(d_L(u, v), |t_u - t_v|)}{M((u, t_u), r', t')} d_2(v, t_v) \\ &= \int_0^\infty \int_0^\infty h(r', t') dr' dt' \\ &= \int_0^t \int_0^r g_L^{ST}(r', t') dr' dt'. \end{aligned} \tag{4.10}$$

Moreover, it is clearly seen that  $K_L^{ST}(r, t)$  does not depend on the events.  $\square$

In practice, however, we usually need to estimate the  $K$ -function (4.6) and the pair correlation function (4.7). Following Ang et al. (2012) and Section 1.3.2, we propose a non-parametric estimator of  $K_L^{ST}(r, t)$  as

$$\widehat{K}_L^{ST}(r, t) = \frac{|L| |T|}{n(n-1)} \sum_{i=1}^n \sum_{i \neq j} \frac{\mathbf{1}\{d_L(x_i, x_j) < r, |t_{x_i} - t_{x_j}| < t\}}{M((x_i, t_{x_i}), d_L(x_i, x_j), |t_{x_i} - t_{x_j}|)}, \tag{4.11}$$

where  $|L| > 0$  and  $|T| > 0$  are the total length of the linear network  $L$  and of the time interval  $T$ , respectively. The corresponding estimator for  $g_L^{ST}(r, t)$  is

$$\widehat{g}_L^{ST}(r, t) = \frac{|L| |T|}{n(n-1)} \sum_{i=1}^n \sum_{i \neq j} \frac{\kappa_\epsilon(d_L(x_i, x_j) - r) \kappa_\delta(|t_{x_i} - t_{x_j}| - t)}{M((x_i, t_{x_i}), d_L(x_i, x_j), |t_{x_i} - t_{x_j}|)}, \tag{4.12}$$

where  $\kappa_\epsilon$  and  $\kappa_\delta$  are one-dimensional kernel functions with bandwidth parameters  $\epsilon$  and  $\delta$ , respectively. The properties of the estimators (4.11) and (4.12) are similar to those in the spatial case, which have been studied in Ang et al. (2012) and Section 1.3.2.

### 4.3.3 Inhomogeneous second-order characteristics

Motivated by practical situations where we often face a number of cases that homogeneity is not a realistic assumption (e.g. traffic accidents happen in particular areas and during specific hours), we extend both the  $K_L^{ST}$  and  $g_L^{ST}$  to inhomogeneous processes. In the two-dimensional space, Baddeley et al. (2000) developed a version of Ripley's  $K$ -function (Ripley, 1977) to the case that can handle non-stationary point processes, see Section 1.2. Gabriel and Diggle (2009) extended the inhomogeneous  $K$ -function to spatio-temporal point processes on  $\mathbb{R}^2 \times \mathbb{R}^+$ . Similar to Baddeley et al. (2000) and Gabriel and Diggle (2009), here we also know that if simple exploratory analysis reveals that the pattern in question can not have a constant intensity function, then the homogeneous  $K$ -function (4.6) and the pair correlation function (4.7) can no longer be proper choices. At this stage, we need an analogue of the spatio-temporal  $K$ -function and the pair correlation function defined by Gabriel and Diggle (2009) for the case of linear networks. We now define such second-order summary statistics (Moradi et al., 2018b).

**Definition 4.2.** Assume  $X$  is a spatio-temporal point process on  $L \times T$  with intensity function  $\lambda(u, t_u) > 0$  and second-order product density function  $\lambda_2((u, t_u), (v, t_v))$ . Then, the inhomogeneous pair correlation function is of the form

$$g_{LI}^{ST}((u, t_u), (v, t_v)) = \frac{\lambda_2((u, t_u), (v, t_v))}{\lambda(u, t_u)\lambda(v, t_v)}, \quad (u, t_u), (v, t_v) \in L \times T. \quad (4.13)$$

If  $g_{LI}^{ST}((u, t_u), (v, t_v)) = g_{LI}^{ST}(d_L(u, v), |t_u - t_v|)$ , then  $X$  is called second-order intensity reweighted pseudostationary and isotropic.

**Definition 4.3.** For a second-order intensity reweighted pseudostationary and isotropic spatio-temporal point process  $X$  on  $L \times T$  with intensity function  $\lambda(u, t_u) > 0$ , the inhomogeneous  $K$ -function is given by

$$K_{LI}^{ST}((u, t_u), r, t) = \mathbb{E} \left[ \sum_{(x, t_x) \in X} \frac{\mathbf{1}\{0 < d_L(u, x) < r, |t_u - t_x| < t\}}{\lambda(x, t_x)M((u, t_u), d_L(u, x), |t_u - t_x|)} \mid (u, t_u) \in X \right] \quad (4.14)$$

Under the assumption of Definition 4.3, equation (4.14) can be rewritten as

$$K_{LI}^{ST}((u, t_u), r, t) = K_{LI}^{ST}(r, t) = \int_0^r \int_0^t g_{LI}^{ST}(r', t') dr' dt'.$$

Note that the proof of the above equality is similar to that of Theorem 4.3.2. Moreover, for Poisson processes, and for any  $r, t > 0$ ,  $K_{LI}^{ST}(r, t) = rt$  and  $g_{LI}^{ST}(r, t) = 1$ .

Following Gabriel and Diggle (2009), Ang et al. (2012) and Gabriel (2014),  $K_{LI}^{ST}$  can be non-parametrically estimated by

$$\widehat{K}_{LI}^{ST}(r, t) = \frac{1}{|L| |T|} \sum_{i=1}^n \sum_{i \neq j} \frac{\mathbf{1}\{d_L(x_i, x_j) < r, |t_{x_i} - t_{x_j}| < t\}}{\widehat{\lambda}(x_i, t_{x_i}) \widehat{\lambda}(x_j, t_{x_j}) M((x_i, t_{x_i}), d_L(x_i, x_j), |t_{x_i} - t_{x_j}|)} \quad (4.15)$$

and  $g_{LI}^{ST}$  by

$$\widehat{g}_{LI}^{ST}(r, t) = \frac{1}{|L| |T|} \sum_{i=1}^n \sum_{i \neq j} \frac{\kappa_\epsilon(d_L(x_i, x_j) - r) \kappa_\delta(|t_{x_i} - t_{x_j}| - t)}{\widehat{\lambda}(x_i, t_{x_i}) \widehat{\lambda}(x_j, t_{x_j}) M((x_i, t_{x_i}), d_L(x_i, x_j), |t_{x_i} - t_{x_j}|)} \quad (4.16)$$

where  $\widehat{\lambda}(\cdot, \cdot)$  is an estimate of the intensity function,  $\kappa_\epsilon$  and  $\kappa_\delta$  are one-dimensional kernel functions with bandwidths  $\epsilon$  and  $\delta$ , respectively.

Estimators (4.15) and (4.16) are unbiased when the intensity function is known or has been estimated with low bias and variance (Baddeley et al., 2000; Gabriel and Diggle, 2009; Ang et al., 2012; Gabriel, 2014), but in practice we usually need to estimate the intensity function. Therefore and in order to reduce the bias and variability of estimators (4.15) and (4.16), Moradi et al. (2018d) recommends using the reciprocal of the following normalisation factor

$$D(\mathbf{x}) = \frac{1}{(|L| |T|)^2} \sum_{i=1}^n \sum_{i \neq j} \frac{1}{\widehat{\lambda}(x_i, t_{x_i}) \widehat{\lambda}(x_j, t_{x_j})}. \quad (4.17)$$

Hence, the updated estimates of the inhomogeneous  $K$ -function and the inhomogeneous pair correlation function are as

$$\frac{1}{D(\mathbf{x})} \widehat{K}_{LI}^{ST}(r, t) \quad \text{and} \quad \frac{1}{D(\mathbf{x})} \widehat{g}_{LI}^{ST}(r, t), \quad (4.18)$$

which ensure a minimised bias and variance (Moradi et al., 2018d).

## 4.4 Data analysis

In this Section we consider some applications to traffic accidents and analyse datasets from Houston (US), Medellín (Colombia) and Eastbourne (UK).

### 4.4.1 Traffic accidents in Houston

Figure 4.1 shows the pattern of motor vehicle traffic accidents in an area of Houston (US) near the university of Houston in 2001 which caused non-incapacitating injury such as bump on the head, abrasions or minor lacerations. The data was collected by individual police departments in the Houston metropolitan area and were compiled by the Texas Department of Public Safety. The compiled data were obtained by the Houston-Galveston Area Council and was geocoded by N. Levine, see Levine (2006, 2009) for details. The space-time pattern contains 144 reference points representing the location and corresponding occurrence time of traffic accidents, with a corresponding linear network with total length 144,253.4 feet and maximum node degree 5. The network is built by 610 nodes and 631 segments. The occurrence time of each traffic accident is rounded to an integer number, i.e. all accidents happened between midnight and 1 am are labelled by 0 as occurrence time.

From the right plot in Figure 4.1 we can see some jumps in the number of accidents during the afternoon and evening. The projected pattern onto the network is also displayed in the left plot in Figure 4.1.

We estimated the intensity function using the estimator (4.5), and both the intensities in time and over the network are represented in Figure 4.2. From the left plot in Figure 4.2 we can see an increase in the estimated intensity during the afternoon which might be caused by heavier traffic jams in the afternoon, especially in the commuting hours. A decrease in the intensity during the night is also clearly visible. The right plot in Figure 4.2 shows the estimated intensity over the network in which the wider the segment the higher the intensity. Higher intensity is visible in the main north-south streets. The bandwidth 993.29 feet and 1.76 hours were considered to estimate the intensity function over the network and in time, respectively.

Figure 4.3 represents the estimated inhomogeneous  $K$ -function and pair correlation function for Houston data together with the corresponding pointwise envelopes based on 99 simulations and significance level 5%, from a complete spatio-temporal random model. Both corresponding surfaces of  $K$ -function and pair correlation function are mainly within the envelopes, indicating spatio-temporal randomness, except for some indication of clustering for a range of distances and time lags.

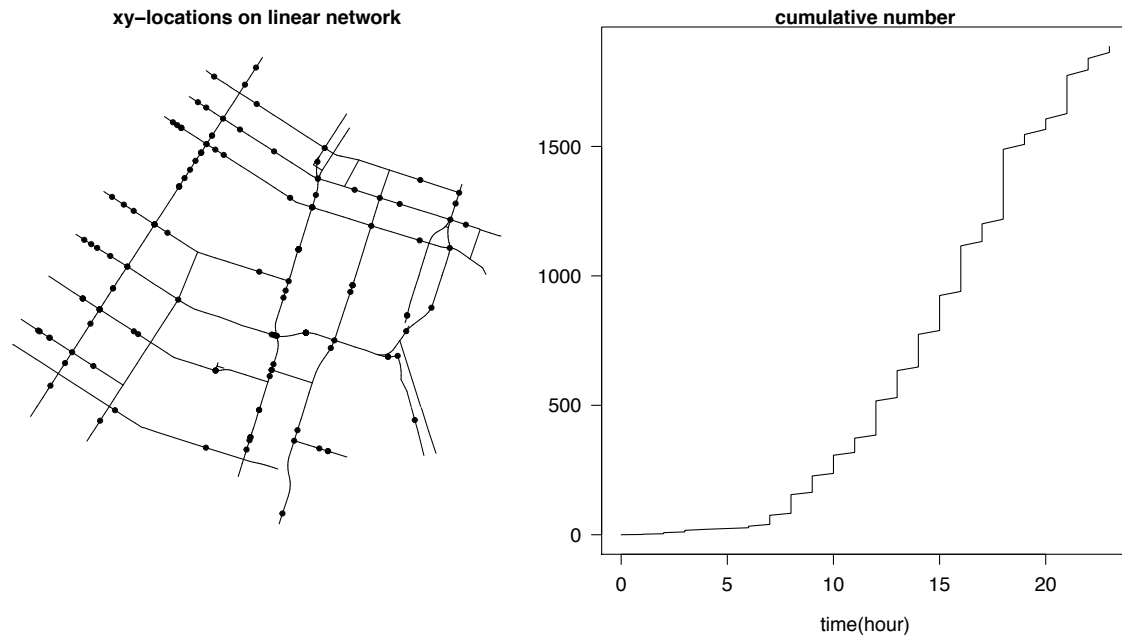


Figure 4.1: The motor vehicle traffic accidents in Houston near the university of Houston in 2001 which caused non-incapacitating injury such as bump on the head, abrasions or minor lacerations. *Left*: The projection of the data onto the network. *Right*: Cumulative number of data points versus time.

#### 4.4.2 Traffic accidents in Medellín

Figure 4.4 shows the traffic accidents in an area near the pontifical bolivarian university in Medellín (Colombia) during 2016. The entire data were published in the OpenData portal of Medellín Town Hall. The represented data in Figure 4.4 has 665 points on a network with total length 29759.42 meters, 643 nodes and 728 segments. The maximum node degree is 6.

From Figure 4.4 we can see that most of the accidents happened near the intersections and also there are some jumps in the cumulative number of accidents during the afternoon.

Using the intensity estimator (4.5), we estimated the intensity function of such data, and both the intensities in time and over network are represented in Figure 4.5. The estimated intensity in time shows a clear decrease during night whereas the intensity starts growing up from morning and it has a peak around 3 pm. The estimated intensity of the projection onto the network is displayed in the right

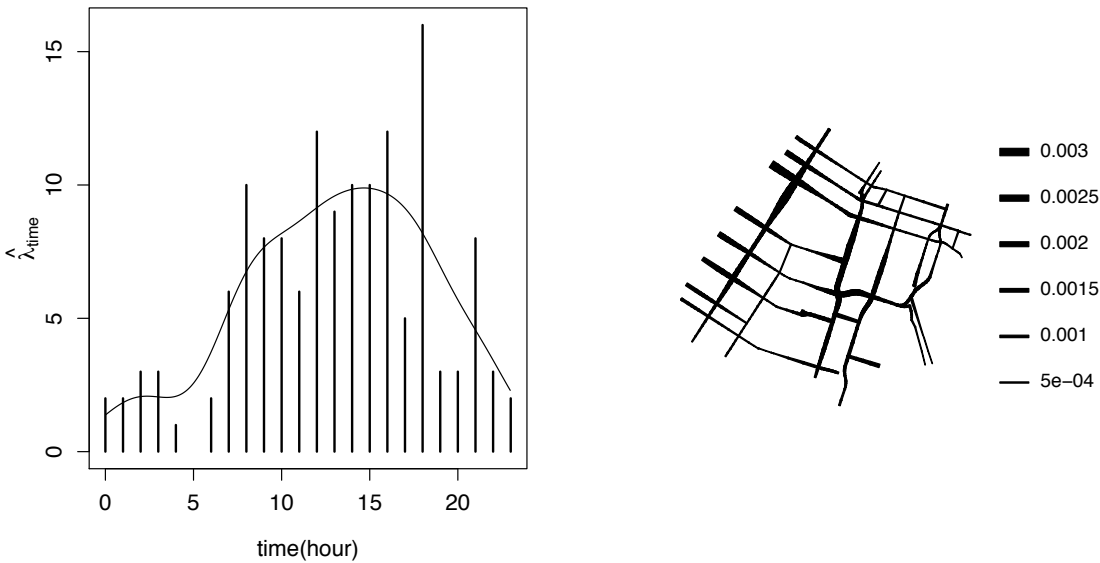


Figure 4.2: Intensity estimates for motor vehicle traffic accidents in Houston. *Left*: Intensity estimate of daily hours together with the frequency of accidents per hour (bar plot). *Right*: Intensity estimate of the projection onto the network.

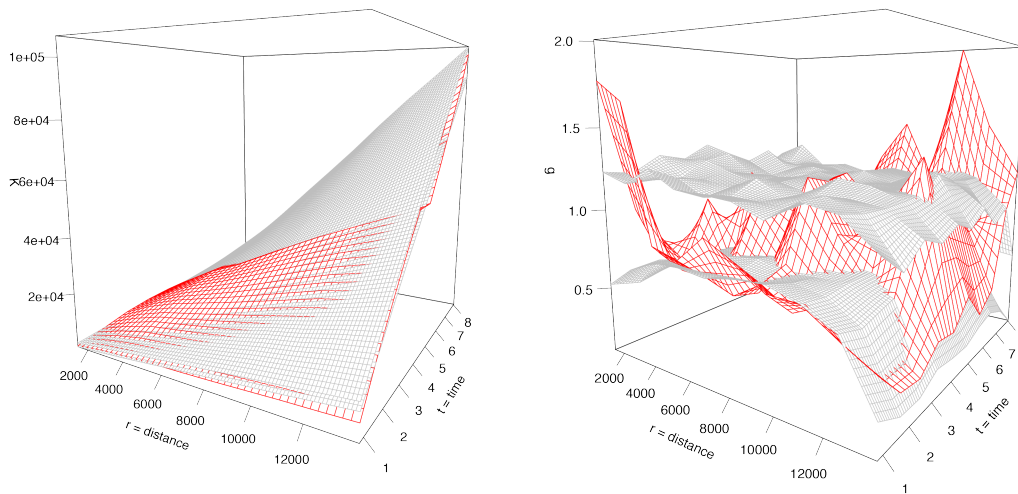


Figure 4.3: Estimated second-order characteristics for motor vehicle traffic accidents in Houston. *Left*: Inhomogeneous  $K$ -function. *Right*: Inhomogeneous pair correlation function. Gray surfaces are envelopes based on 99 simulations and significance level 5% from the complete spatio-temporal randomness.



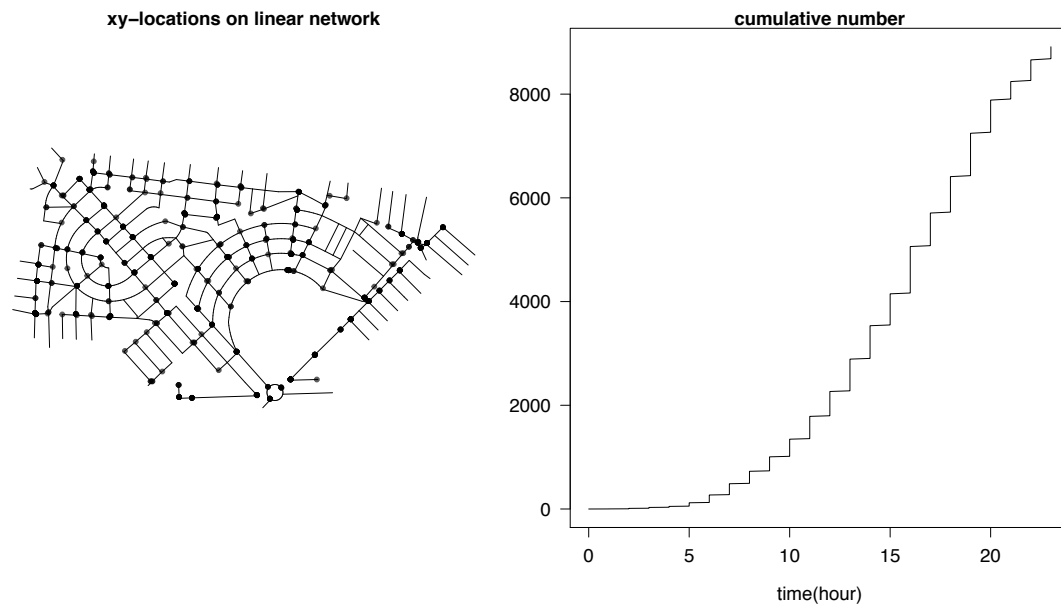


Figure 4.4: Traffic accidents in Medellín during the year 2016. *Left*: The projection of data onto the network. *Right*: Cumulative number of data points versus occurrence time.

plot of Figure 4.5 confirming higher intensity in the eastern part of the network. Bandwidths 87.12 meters and 1.26 hours were used for the estimation of intensities over network and in time, respectively.

Figure 4.6 shows perspective plots of the estimated inhomogeneous  $K$ -function and pair correlation function for Medellín data together with the corresponding envelopes based on 99 simulations and significance level 5% from a complete spatio-temporal random model. According to both plots in Figure 4.6, a spatio-temporal interaction between data points is visible, indicating both clustering and inhibition behaviour within certain distances and periods of time. This behaviour might suggest fitting a multi-scale area-interaction model to such data.

### 4.4.3 Traffic accidents in Eastbourne

Figure 4.7 shows the traffic accidents in the down-town of Eastbourne (UK). The network was provided by “OS OpenData <sup>1</sup>” and is usable under the terms of the OS OpenData license. The traffic locations were collected by the UK Department

<sup>1</sup>[www.ordnancesurvey.co.uk](http://www.ordnancesurvey.co.uk)

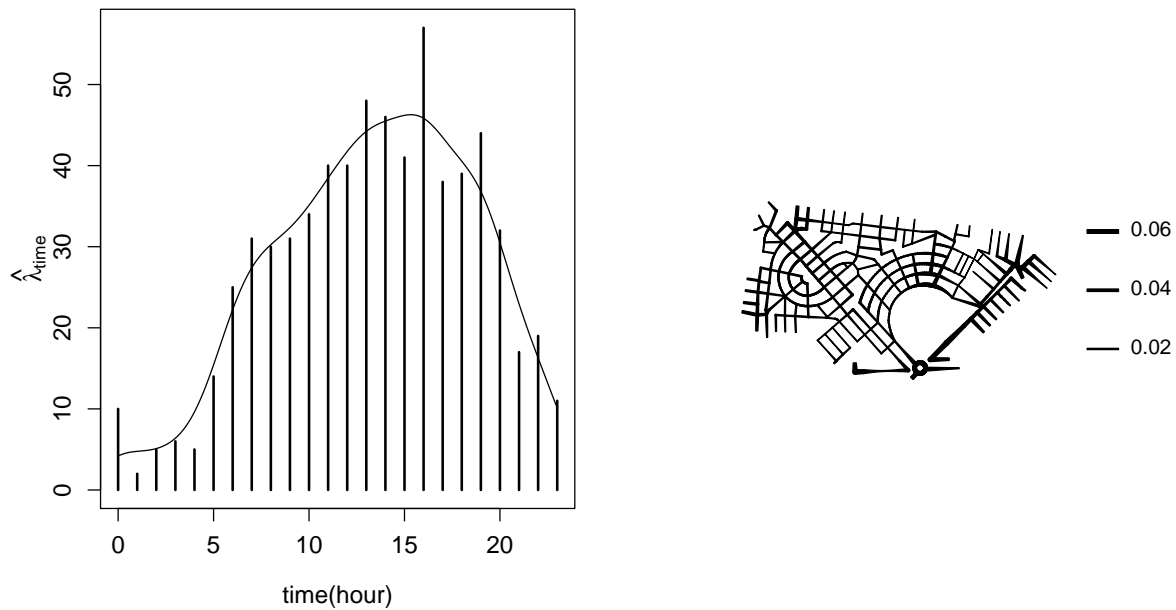


Figure 4.5: Intensity estimates for traffic accidents in Medellín. *Left*: Intensity estimate of daily hours together with the frequency of accidents per hour (bar plot). *Right*: Intensity estimate of the projection onto the network.

for Transport <sup>2</sup> and obtained through kaggle<sup>3</sup>. Accidents happened during the period 2005-2010. The network contains 119 nodes and 153 line segments with a total length of 17270.61 meters. Maximum node degree is also 4. The point pattern contains 163 points, representing the locations of traffic accidents that are labelled by the time occurrence. The point pattern on the left side of Figure 4.7 shows concentration of points around some intersections, it further seems there are more accidents in south-north streets. The right plot in Figure 4.7 shows some jumps in the afternoon which is a hint of happening more accidents during the afternoon.

Figure 4.8 displays the estimated intensities over both linear network and in time. The estimated time intensity confirms a higher intensity during the afternoon and also shows an increase from the morning, a peak around 4 pm and it decreases towards the night. The estimated intensity in the right side of Figure 4.8 shows higher intensities around some intersections. Bandwidths 80 meters

<sup>2</sup>[www.data.gov.uk](http://www.data.gov.uk)

<sup>3</sup>[www.kaggle.com](http://www.kaggle.com)

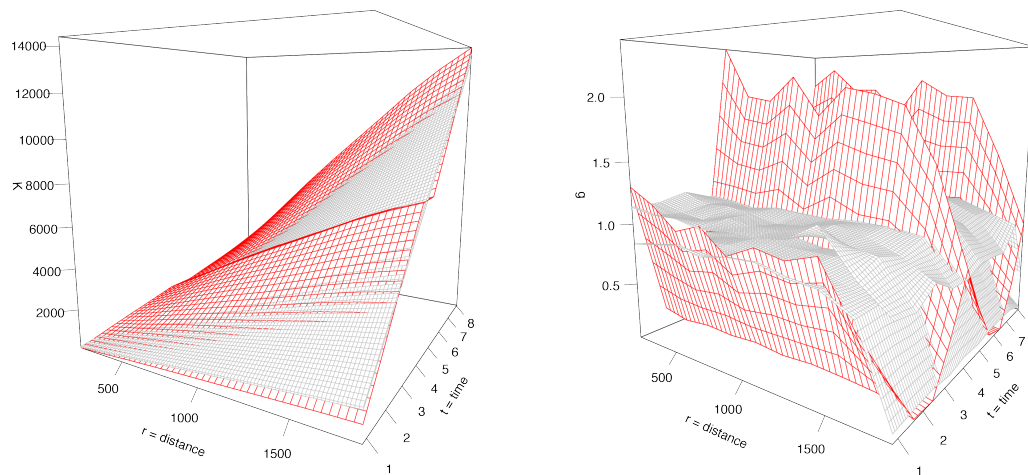


Figure 4.6: Estimated second-order characteristics for traffic accidents in Medellín. *Left*: Inhomogeneous  $K$ -function. *Right*: Inhomogeneous pair correlation function. Gray surfaces are envelopes based on 99 simulations and significance level 5% from complete spatio-temporal randomness.

and 1.21 hours were used to estimated the intensities over network and in time, respectively.

Figure 4.9 shows the estimations of the inhomogeneous  $K$ -function and pair correlation function together with pointwise envelopes based on 99 simulations and significance level 5% from a complete spatio-temporal random model. According to Figure 4.9, there seems to be an evidence of clustering behaviour within some particular distances and time lags.

## 4.5 Summary

We provided preliminaries of spatio-temporal point processes on linear networks. The proposed methodologies allow us to include time component when analysing network point patterns. We introduced a kernel-based intensity estimator which preserves mass. This allows to figure out the higher and lower intensity parts within space and time jointly. We also developed second-order summary statistics, the  $K$ -function and pair correlation function, for spatio-temporal point processes on linear networks to analyse the type of interaction between data points. The defined  $K$ -

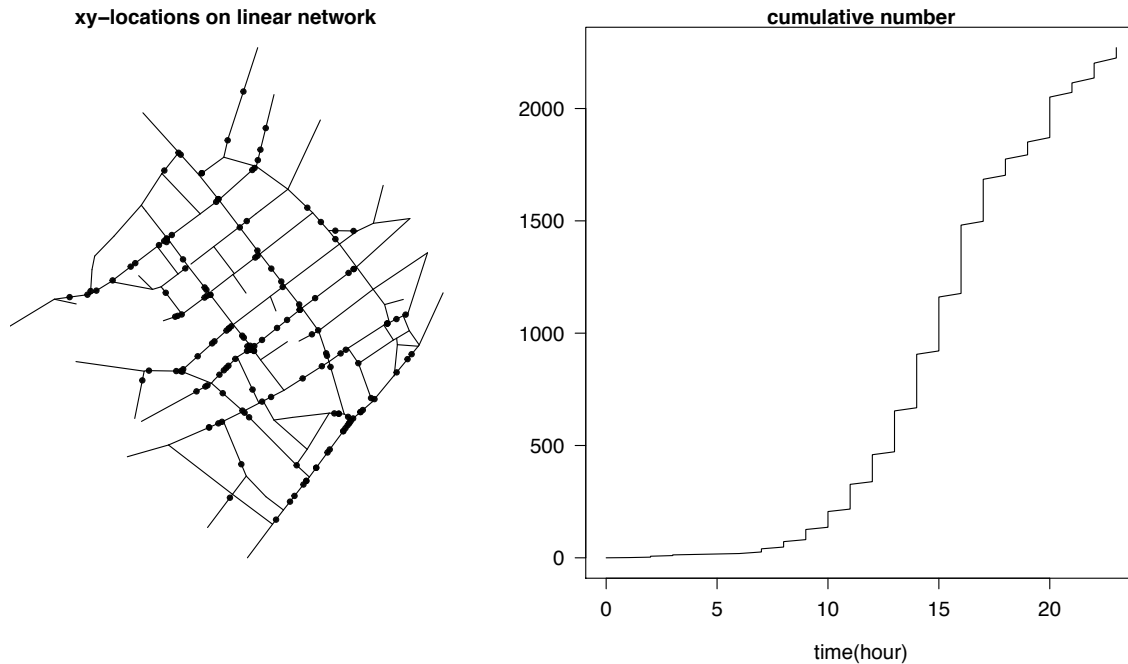


Figure 4.7: Traffic accidents in the down-town of Eastbourne (UK) in . *Left*: The projection of data onto the network. *Right*: Cumulative number of data points versus occurrence time.

function and pair correlation function have known values for Poisson processes that can be used for model selection and to measure deviation from being Poisson. Non-parametric estimators were discussed for both homogeneous and inhomogeneous cases. The estimators are independent of the geometry of the network and it then allows comparing the behaviour of different point patterns on different networks. The introduced  $K$ -function and pair correlation function also maintain the same relationship similar to their analogues in space. Although the computations in Sections 4.4.1, 4.4.2 and 4.4.3 were fast, the current implementation of the  $K$ -function and the pair correlation function might be computationally expensive in large networks.

Three different traffic accidents data from different countries (US, Colombia, UK) were analysed. The estimated time intensities for all the three datasets showed an increase from the morning to the afternoon and a decrease over the night that might be caused due to rush hours. Also some particular streets showed higher intensities. According to the  $K$ -function and the pair correlation function, the three considered datasets showed different types of interaction such as randomness,

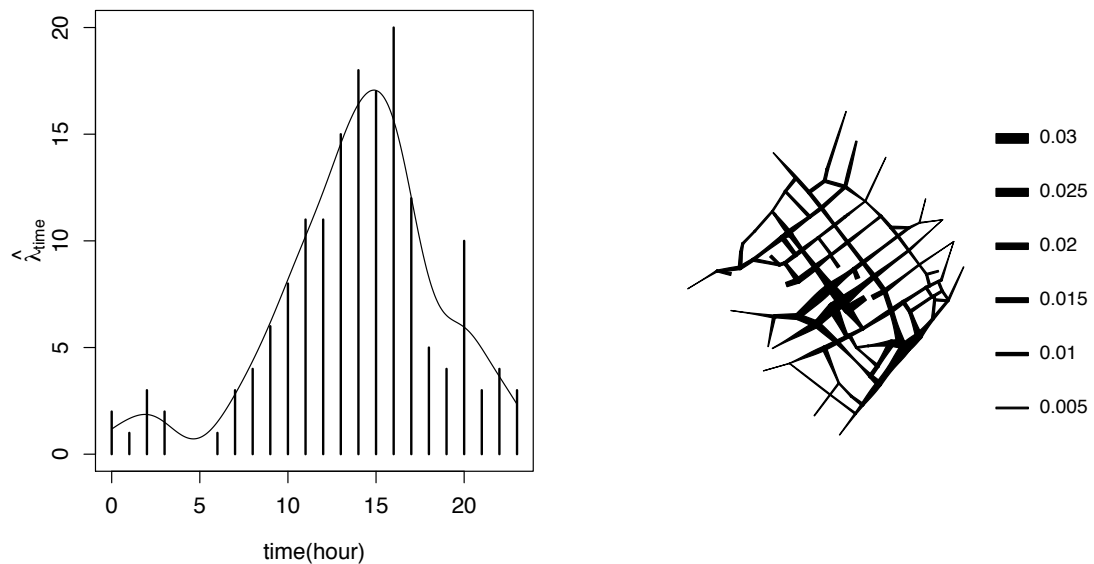


Figure 4.8: Intensity estimates for traffic accidents in the down-town of Eastbourne. *Left*: Intensity estimate of daily hours together with the frequency of accidents per hour (bar plot). *Right*: Intensity estimate of the projection onto the network.

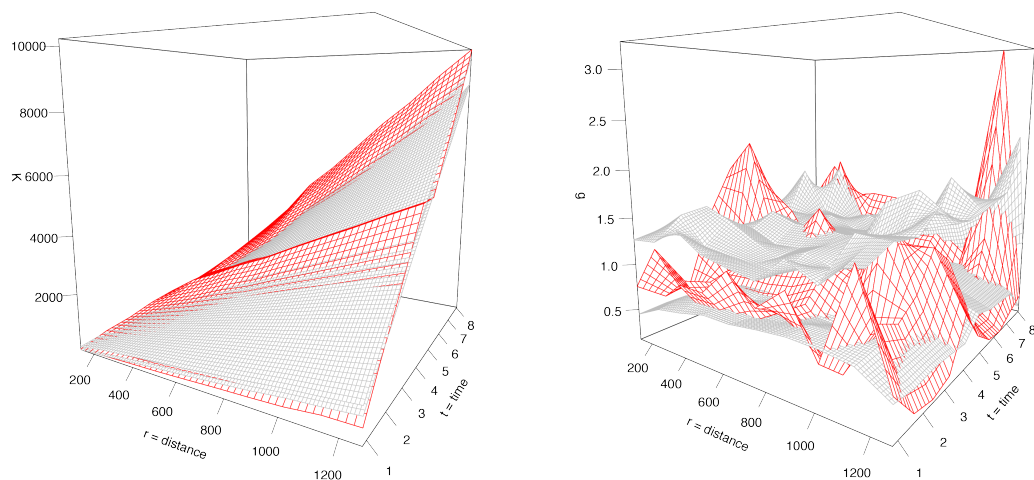


Figure 4.9: Estimated second-order characteristics for traffic accidents in the down-town of Eastbourne. *Left*: Inhomogeneous  $K$ -function. *Right*: Inhomogeneous pair correlation function. Gray surfaces are pointwise envelopes based on 99 simulations and significance level 5% from complete spatio-temporal randomness.

clustering and inhibition within certain distances and periods of time.

## CHAPTER 5

---

# trajectories: Classes and Methods for Trajectory Data

---

### 5.1 Introduction

Human beings, cars, bicycles, animals, etc. are examples of moving objects in space and over time. New techniques allow tracking such objects continuously. Figure 5.1 shows three consecutive movements of 50 taxis in Beijing, China per 20 minutes. From Figure 5.1, it can be seen that e.g. the length/speed of movements over the same period of time depends on space.

In practice, however, the locations of objects are recorded according to regular/irregular time stamps so that there is a set of spatial point patterns on  $\mathbb{R}^2$  within



Figure 5.1: Three consecutive movements of 50 taxis in Beijing, China on Feb 2008 per 20 minutes.

the time period in question. This motivates us to build statistical methodologies to analyse movement data using the literature of spatial point process. Studying the behaviour of moving objects over time and their interaction, either between objects or with the environment, plays a crucial role in understanding how they use space and more importantly how they interact with each other. As moving objects are moving within a particular area over time, they then have two sources of random effect.

We provide some statistical tools to handle, simulate and statistically analyse movement data regardless of the domain, converting a trajectory pattern into a list of point patterns based on regular timestamps. Distance analysis can reveal the behaviour of objects over time and with respect to each other. We also propose different functions to analyse the behaviour of objects through time and see how they use space and also how they interact with each other. The type of interaction between objects may vary over time.

Section 5.2 reviews some classes and methods to handle trajectories. Simulating random trajectories and model fitting is discussed in Section 5.3. In Section 5.4 we provide statistical methodologies to analyse the first- and second-order characteristics of moving objects together with distance analysis and movement smoothing. The new methods are applied to a dataset of taxis movement in Beijing, China during the period of Feb. 2 to Feb. 8, 2008 in Section 5.4.

## 5.2 Classes and methods

In this section, we review different classes of trajectories to handle movement data in R. These classes were initially defined in the R package **spacetime** (Pebesma, 2012). Before moving into the details and start analysing trajectory patterns, we load the package with

```
R> library("trajectories")
```

### 5.2.1 Track

The class ‘Track’ represents a single track followed by a person, animal or an object. Instances of this class are meant to hold a series of consecutive location/timestamps that are not interrupted by another activity. The class contains five slots: @sp to store the spatial points, @time to store the corresponding time,



@endtime to store the end time when having generalised line geometries with one value per attribute for a set of points (otherwise, defaults to the time defined in @time), @data to store the attributes (covariate information) and @connections to keep a record of attribute data between points (e.g., distance, duration, speed and direction). A 'Track' object can be created out of an 'STIDF' object (see Pebesma (2012)) as follows

```
R> set.seed(10)
R> t0 = as.POSIXct(as.Date("2013-09-30", tz="CET"))
R> x = c(7,6,5,5,4,3,3)
R> y = c(7,7,6,5,5,6,7)
R> n = length(x)
R> t = t0 + cumsum(runif(n) * 60)
R> crs = CRS("+proj=longlat +ellps=WGS84") # longlat
R> stidf = STIDF(SpatialPoints(cbind(x,y),crs), t,
+               data.frame(co2 = rnorm(n,mean = 10)))
R> A1 = Track(stidf)
R> A1
```

An object of class Track

7points

bbox:

	min	max
x	3	7
y	5	7

Time period: [2013-09-30 02:00:30, 2013-09-30 02:02:31]

Figure 5.2 shows the plot of track A1 passed by person A. By default, distance, duration, speed and direction are computed as the connections data. Optionally, a data frame containing additional connections data (covariates) and/or a custom function for calculating the data of segments between consecutive points can be passed.

Moreover, and using the function `as.Track`, one can create an object of class 'Track' if spatial coordinates and corresponding times are provided. Additional information can also be passed to the function `as.Track` using an argument `covariate`.

```
R> plot(A1)
```

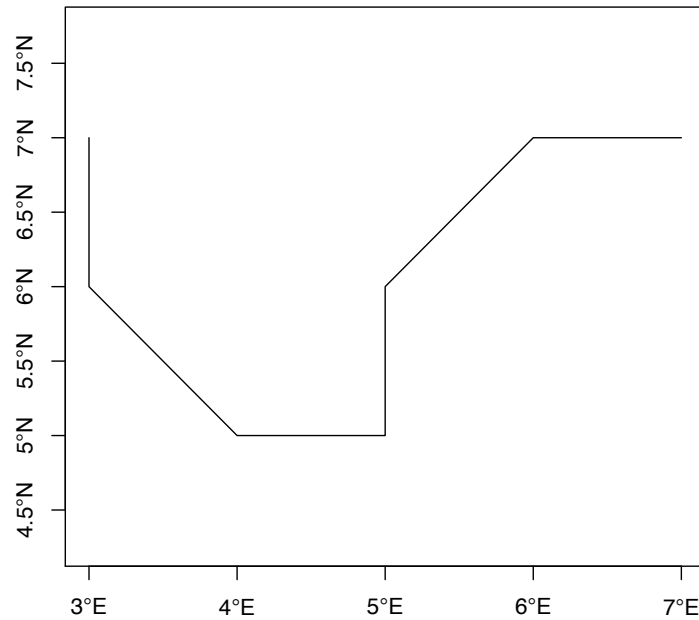


Figure 5.2: Single track A1 passed by person A.

```
R> x <- runif(10,0,1)
R> y <- runif(10,0,1)
R> date <- seq(as.POSIXct("2015-1-1 0:00"), as.POSIXct("2015-1-1 9:00"),
+             by = "hour")
R> records <- as.data.frame(rpois(10,5))
R> as.Track(x,y,date,covariate = records)
```

An object of class Track

10points

bbox:

	min	max
x	0.23958913	0.8382877
y	0.09308813	0.9546536

Time period: [2015-01-01, 2015-01-01 09:00:00]

## 5.2.2 Tracks

The class ‘Tracks’ embodies a collection of tracks followed by a single person, animal or object. The class contains two slots: `@tracks` to store the tracks as objects of class ‘Track’ and `@tracksData` to hold a summary record for each particular track (e.g. minimum and maximum time, total distance and average speed). An object of class ‘Tracks’ can be created by

```
R> x = c(7,6,6,7,7)
R> y = c(6,5,4,4,3)
R> n = length(x)
R> t = max(t) + cumsum(runif(n) * 60)
R> stidf = STIDF(SpatialPoints(cbind(x,y),crs), t,
+               data.frame(co2 = rnorm(n,mean = 10)))
R> A2 = Track(stidf)
R> # Tracks for person A:
R> A = Tracks(list(A1=A1,A2=A2))
R> A
```

An object of class Tracks  
2 tracks followed by a single object

where A1 and A2 are of class ‘Track’. By default, the minimum and maximum coordinates and time, the total number of geometries, the total distance as well as the average speed are computed as the summary information data. As for the ‘Track’ method, a data frame and/or a custom function can be passed to expand the default data.

## 5.2.3 TracksCollection

The class ‘TracksCollection’ represents a collection of tracks followed by many persons, animals or objects. The class contains two slots: `@tracksCollection` to store the tracks as objects of class ‘Tracks’ and `@tracksCollectionData` to hold summary information about each particular person, animal or object (e.g., the total number of tracks per each object). A ‘TracksCollection’ object can be created by

```
R> # person B, track 1:
R> x = c(2,2,1,1,2,3)
```

```

R> y = c(5,4,3,2,2,3)
R> n = length(x)
R> t = max(t) + cumsum(runif(n) * 60)
R> stidf = STIDF(SpatialPoints(cbind(x,y),crs), t,
+               data.frame(co2 = rnorm(n,mean = 10)))
R> B1 = Track(stidf)
R> # person B, track 2:
R> x = c(3,3,4,3,3,4)
R> y = c(5,4,3,2,1,1)
R> n = length(x)
R> t = max(t) + cumsum(runif(n) * 60)
R> stidf = STIDF(SpatialPoints(cbind(x,y),crs), t,
+               data.frame(co2 = rnorm(n,mean = 10)))
R> B2 = Track(stidf)
R> # Tracks for person B:
R> B = Tracks(list(B1=B1,B2=B2))
R> Tr = TracksCollection(list(A=A,B=B))
R> Tr

```

An object of class `TracksCollection`  
 2 collection of tracks followed by 2 object

where A and B are objects of class ‘Tracks’. By default, the total number of tracks as well as the minimum and maximum coordinates, and time are computed as the summary information data. As for the ‘Track’ and ‘Tracks’ methods outlined above, a data frame and/or a custom function can be passed to expand the default data.

### 5.2.4 segments

The class ‘segments’ is written to provide a data structure for storing all the segments of a track, with a segment representing the line between two consecutive points.

Figure 5.3 shows the classes and their connection. We point out that classes ‘STIDF’, ‘STI: irregular’ and ‘STI’ belong to the package **spacetime** inherently.

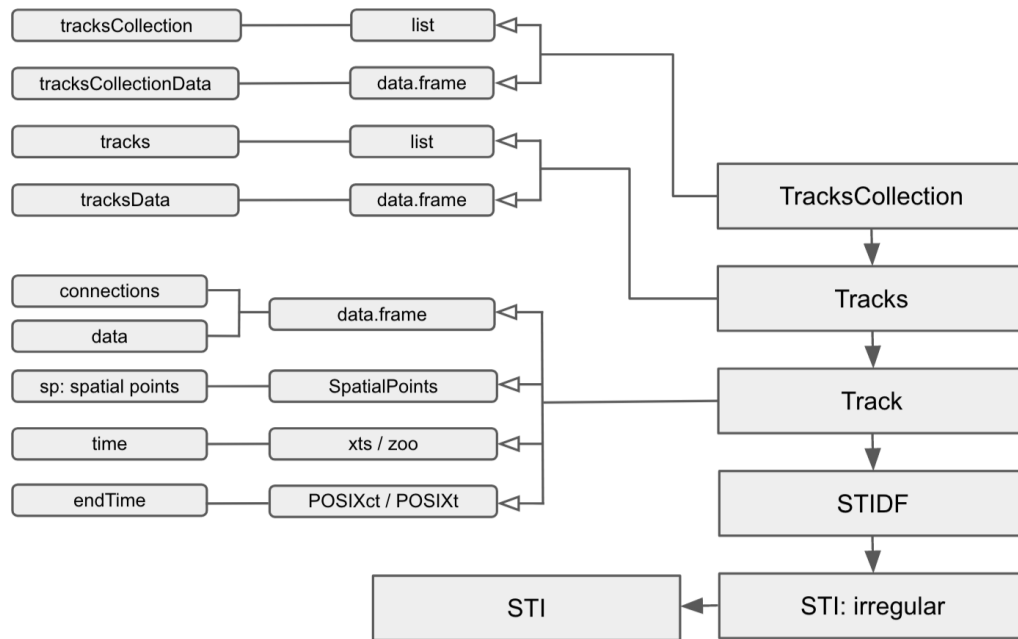


Figure 5.3: Classes for trajectory data in the package **trajectories**. Solid arrows denote inheritance. Arrows show the corresponding slot's class and slot's names are displayed using lines accordingly.

### 5.2.5 Methods

A wealth of methods have been implemented to cover the most frequently used use cases. Table 5.1 lists some of the methods applied to the objects of classes `Track`, `Tracks` and `TracksCollection`. Apart from those listed in Table 5.1, attribute data can be obtained or replaced by using `[]`, `[[[]]`, `@` and `$`.

The use of some methods in Table 5.1 is shown in the following lines of code. Figure 5.4 shows the trellis plot of object `Tr` from class `'TracksCollection'` which is previously created. We point out that we have used slot `data` of the corresponding tracks as attributes to see their changes over time.

```
R> dim(A1)
```

```
geometries
```

```
7
```

```
R> dim(B1)
```

Table 5.1: Methods implemented in the package **trajectories** for objects from class 'Track', 'Tracks' and 'TrackCollection'.

Method	Operation
dim	Returns the number of spatial points of any track
summary	Summarises the internal information
proj4string	Retrieves projection attributes
coordinates	Retrieves the coordinates of spatial locations
coordnames	Retrieves coordinate names of fixes
bbox	The box (window) which contains the objects
stbox	The spatio-temporal box (window) which contains the objects
aggregate	Spatially aggregate track properties (coercing fixes to points)
compare	Compares two 'Track' objects: for the common time period
dists	Compares two 'Tracks'
downsample	Remove fixes from a 'Track'
frechetDist	Compute Frechet distance between two 'Track' objects
stcube	Draw a space-time cube
stplot	Create trellis plot for 'TracksCollection' objects
generalize	Resample 'Track' to lower frequency or minimal distance
cut	Obtain ranges of space and time coordinates

```
geometries
```

```
6
```

```
R> stbox(A1)
```

```
      x y          time
min 3 5 2013-09-30 02:00:30
max 7 7 2013-09-30 02:02:31
```

```
R> downsample(A1,B1)
```

```
An object of class Track
```

```
6points
```

```
bbox:
```

```
min max
x  3  7
y  5  7
Time period: [2013-09-30 02:00:30, 2013-09-30 02:02:31]

R> stplot(Tr, attr = "co2", arrows = TRUE, lwd = 3,
+         by = "IDs", cex.axis=2)
```

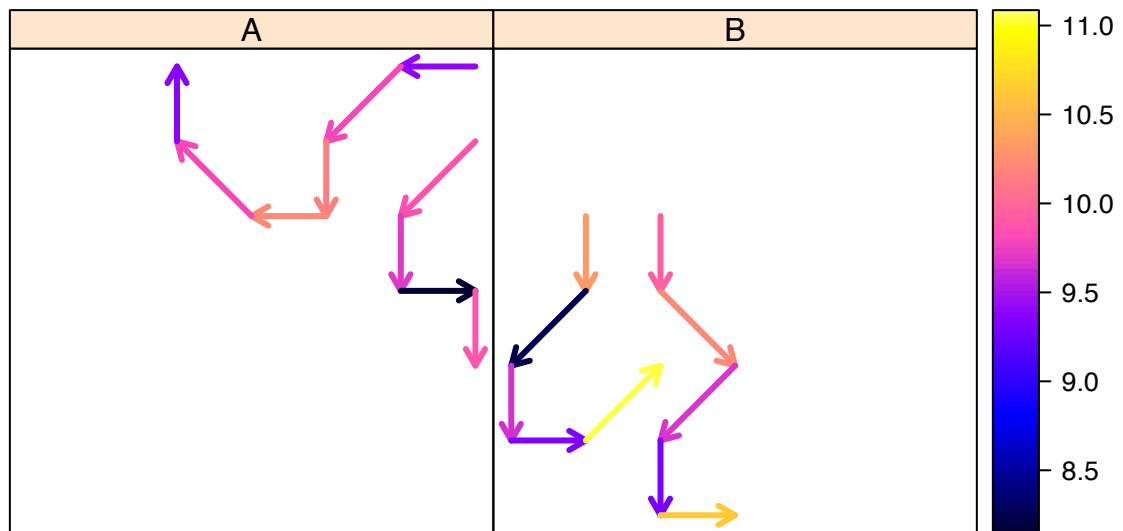


Figure 5.4: Co2 consumption over time.

## 5.3 Simulation and model fitting

### 5.3.1 Trajectory simulation

Simulating trajectory patterns can be a useful tool to imitate true phenomena and understand their behaviour. The package **trajectories** allows simulating tracks using `rTrack`, `rTracks`, `rTracksCollection` where `rTrack()` generates a single track, `rTracks()` simulates a collection of tracks assumed to be passed by a single object and `rTracksCollection` is used to simulate a set of tracks passed by different objects. By default, these functions do not consider any box (or window) for the track to be simulated in and consider `origin=c(0,0)` as the origin of the track. However, one can still restrict the track to a desirable closed box using the

argument `bbox`. If `transform=TRUE` and no `bbox` is given, then `rTrack` transforms the track to the default box  $[0, 1] \times [0, 1]$ , where in this case the origin is a random point in the default box. If a default box `bbox` (e.g. `m` in the following example) is given and `transform=TRUE`, then `origin` is a random point in `bbox` and the final track is also transformed into `bbox`. The function `rTrack` simulates tracks with a predefined number of points per track (indicated as `n` in the code with default 100). However if `nrandom=TRUE` then it simulates a track with a random number of points based on a Poisson distribution with parameter `n`. An example of these functions is the following

```
R> set.seed(10)
R> x <- rTrack();x
```

An object of class `Track`

```
100points
bbox:
      min      max
x -53.38677 -3.232289
y -53.08787 -1.594295
Time period: [1970-01-01, 1970-01-01 01:39:00]
```

```
R> y <- rTrack(transform = T);y
```

An object of class `Track`

```
100points
bbox:
      min max
x    0   1
y    0   1
Time period: [1970-01-01, 1970-01-01 01:39:00]
```

```
R> m <- matrix(c(0,10,0,10),nrow=2,byrow = T)
```

```
R> w <- rTrack(bbox = m,transform = T);w
```

An object of class `Track`

```
100points
bbox:
```



```
      min max
x     0  10
y     0  10
Time period: [1970-01-01, 1970-01-01 01:39:00]

R> z <- rTrack(bbox = m, transform = T, nrandom = T);z
```

```
An object of class Track
108points
bbox:
      min max
x     0  10
y     0  10
Time period: [1970-01-01, 1970-01-01 01:47:00]
```

Figure 5.5 shows four different random tracks:  $x$  is a random track with all defaults,  $y$  is a random track transformed to a unit box,  $w$  is a random track transformed to the box  $[0, 10] \times [0, 10]$ , and  $z$  is a simulated track in a same box as  $w$  but with a random number of points. The number of points in  $w$  is 100 whereas  $z$  is constituted by 108 points.

### 5.3.2 Model fitting

The behaviour of a track might also be studied using available tools for time series modelling. However, obtaining a proper model is extremely important as it highlights the underlying structure of the series, and the fitted model can be used for future forecasting. The R package **trajectories** can fit ARIMA models to movement data. Using R package **forecast**, the function `auto.arima.Track` fits arima models to the spatial coordinates of an object of class 'Track'. Note this is applicable to individuals. See example below. We point out that the dataset Beijing is stored in package **taxidata** and can be installed using the following line of code

```
R> install.packages("taxidata",
+   repos = "http://pebesma.staff.ifgi.de", type = "source")
```

```
R> par(mfrow=c(2,2),mar=rep(2.2,4))
R> plot(x,lwd=2,main="x");plot(y,lwd=2,main="y")
R> plot(w,lwd=2,main="w");plot(z,lwd=2,main="z")
```

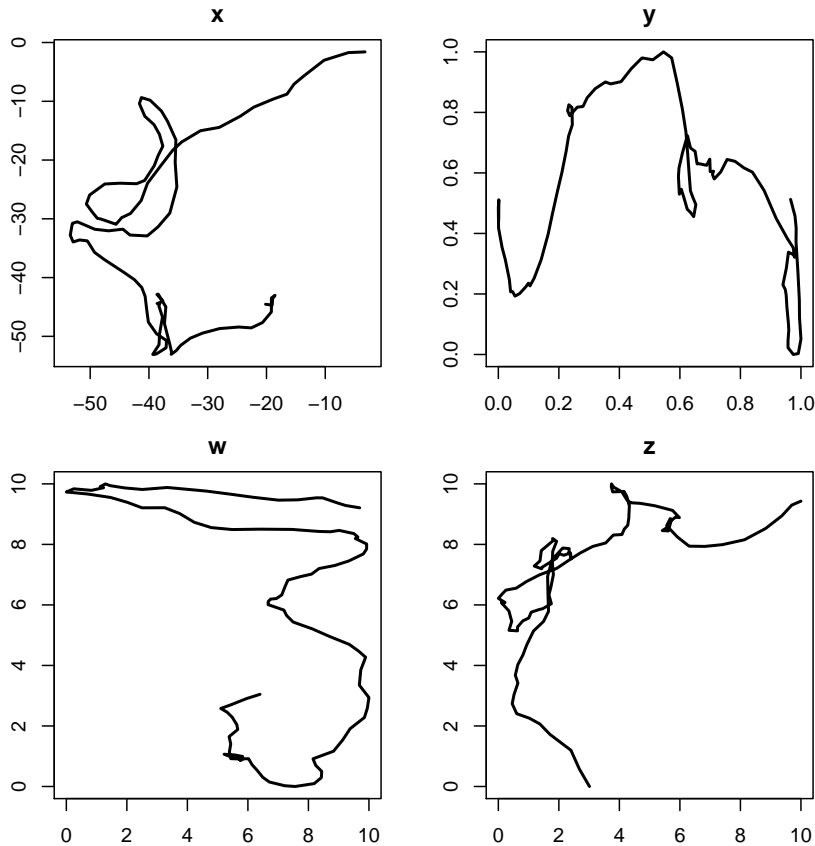


Figure 5.5: Simulated random tracks using `rTrack`. `x` is random track with all defaults. `y` is a random track transformed to a unit box. `w` is a random track transformed to the box  $[0, 10] \times [0, 10]$  and `z` is in a same box as `w` but with random number of points.

```
R> library("forecast")
R> library("taxidata")
R> Beijing <- taxidata
R> auto.arima.Track(Beijing[[5]])
```

Arima model fitted to x-coordinate: ARIMA(1,0,2) with non-zero mean

Arima model fitted to y-coordinate: ARIMA(2,1,2) with drift

## 5.4 Exploratory data analysis

This Section presents some statistical methods, implemented in the R package **trajectories**, to analyse the behaviour of trajectory patterns. A single plot of trajectories pattern might not display interesting information, and if the pattern contains too many tracks, it then needs some analysis to summarise and reveal concealed information. In particular, one may be interested in discovering the more visited streets within a city. Other interesting findings could be the type of interaction between moving objects over time. In short, having a trajectory pattern, we might be interested in answering the following questions:

1. How does the average distance between objects change over time?
2. How is the spatially varying distribution of objects?
3. How do moving objects interact with each other? Does their interaction vary over time?
4. How does the spatially varying distribution of objects vary over time?

### 5.4.1 Data

We considered a sample of the T-Drive trajectory dataset that contains one-week trajectories of 10357 taxis during the period of Feb. 2 to Feb. 8, 2008, within Beijing, China. T-drive is a smart driving direction services based on GPS trajectories of a large number of taxis. The GPS-equipped taxis are mobile sensors probing the traffic flows on road surfaces. So, the taxi trajectories contain the information of both human knowledge of experienced drivers and traffic patterns. The total number of points in this dataset is about 15 million and the total distance of the trajectories reaches up to 9 million kilometres. For more details about the data, see Yuan et al. (2010, 2011).

We here point out some useful information about the dataset:

- 21 taxis have no information recorded.
- Regardless of taxis with no information, there are 4694 taxis with less than 10 recorded locations in at least one day.

- There are tracks with some jumps to the outside of the studied area and it may be caused by lack of GPS accuracy so that wrong locations have been removed. These locations might later be recovered by interpolation.

Considering the aforementioned points, we thus analyse the cleaned dataset which is based on moving data of 5642 taxis. The map of the studied area is displayed in Figure 5.6. It is seen that the metropolitan area of Beijing is almost located in the centre of the map while there are some other townships, airports in the countryside of Beijing.

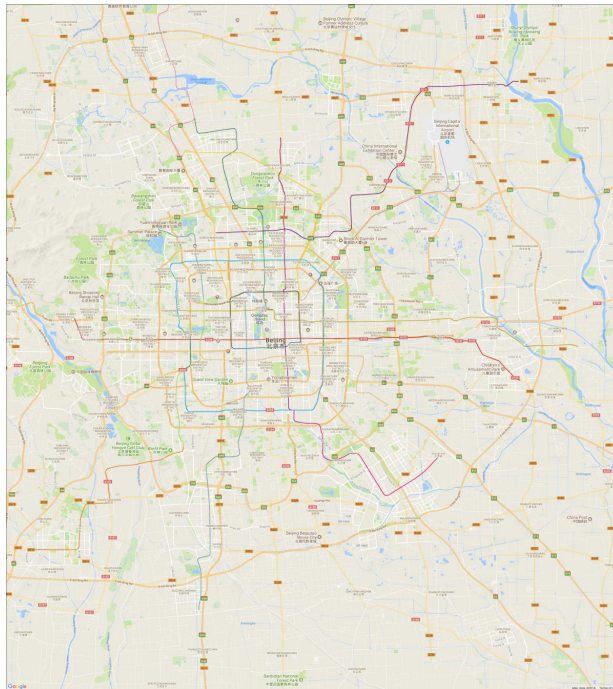


Figure 5.6: Map of the studied area in Beijing, China.

In the following, we present the implemented methods in **trajectories** by applying them to the taxi movement dataset in Beijing, China.

### 5.4.2 Distance analysis

A simple way to get into the nature of movement data is to study the distance between objects. The function `dists` provides users with calculating the distance between a pair of objects of class 'Tracks'. This considers the distance between tracks when they overlap in time. The output is a matrix with distances between

each pair of tracks or 'NA', if they do not overlap in time. A function to calculate distances can be passed to `dists`, such as `mean`, `sum`, `frechetDist`, etc.

```
R> tracks1 <- Tracks(list(Beijing[[1]],Beijing[[2]]))
R> tracks2 <- Tracks(list(Beijing[[3]],Beijing[[4]]))
R> dists(tracks1,tracks2,mean)
```

```
      [,1]      [,2]
[1,] 16854.55  9103.338
[2,] 15025.30 10275.476
```

#### 5.4.2.1 Average distance over time

The distance between objects over time might discover some interesting information. Studying pairwise distances over time can somehow reveal the type of interaction between objects. Having a pattern of tracks, we may be able to see how moving objects interact each other over time. Moreover, this can highlight the crowded hours within a particular period of time. We here propose to look at average pairwise distances over time. To do so, one can imitate the following steps:

1. Based on the time range of all tracks  $s_i$ , create a regular time sequence.
2. Interpolate each track  $s_i$  based on the created time sequence. For this purpose, the function `reTrack` can be used. It reconstructs each track  $s_i$  according to a desirable time sequence.
3. Discretise the trajectory pattern  $S$  to a collection of point patterns  $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_k$ . Note that the number of points in each pattern might be different.
4. For each  $\mathbf{x}_i, i = 1, \dots, k$ , calculate pairwise distances between all data points.
5. Report the average of pairwise distances per each time.

Steps above are implemented in the function `avedistTrack`. In order to use `avedistTrack`, we only need to specify the argument `timestamp`. It then returns the average distance between objects based on the corresponding timestamps.

```
R> par(mfrow=c(1,2))
R> meandist <- avedistTrack(Beijing,timestamp = "20 mins")
R> plot(meandist,type="l",lwd=2,cex.axis=1.7,cex.lab=1.7)
R> distinframe <- data.frame(tsq=attr(meandist,"tsq")
+                           ,dist=meandist)
R> dist3rd <- distinframe[substr(distinframe$tsq,start = 1,stop=10)
+                           == "2008-02-03",]
R> plot(dist3rd$tsq,dist3rd$dist,type="l",xlab="time",
+       ylab="average distance",lwd=2,cex.axis=1.7,cex.lab=1.7)
```

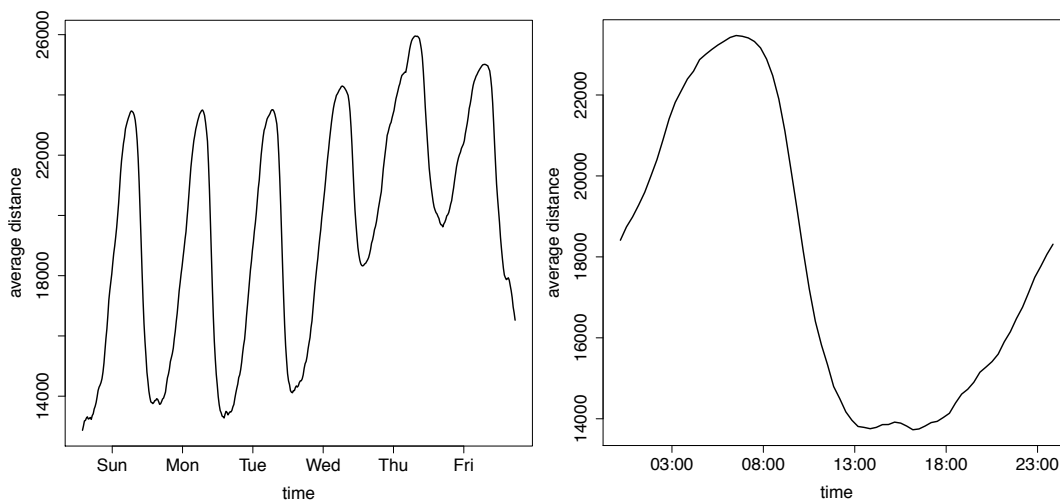


Figure 5.7: Average pairwise distance between taxis in Beijing, China. *Left*: Within the period 2-8, Feb 2008. *Right*: During 3-*rd* of Feb 2008.

Figure 5.7 shows the average distance between taxis in Beijing. The left plot shows the average pairwise distance between taxis for all the week and it is easily seen that there is a daily trend. In order to see the more crowded hours within a day, we show the average pairwise distances during the 3-*rd* of Feb 2008 in the right plot in Figure 5.7. It can be seen that the crowded time comes between midday and 17. Also, it shows how taxis are getting far from each other during night. Note that small average distances between taxis might be a sign of traffic

during the corresponding hours. We point out that the reason of having larger average distances in the last two days might be the Chinese new year holidays.

As an alternative, one can use nearest neighbour distances instead of pairwise distances. But that might not be a good choice when there are several clusters in the generated point patterns surrounded by possible empty neighbourhoods. This might be seen when trajectory pattern contains different sets of tracks concentrated in some particular subregions.

### 5.4.3 Movement smoothing

Aiming at analysing moving objects, it might be of interest to highlight the relationship of movement with space and time. This Section is considered to perform smoothing over the length of movements per each consecutive time. Thus, it might reveal those areas with faster/slower movements. For this purpose, the function `Track.idw` performs inverse-distance weighted smoothing over a trajectory pattern by imitating the following steps:

1. Follow steps 1-3 in Section 5.4.2.
2. Using each consecutive point patterns, say  $\mathbf{x}_o$  and  $\mathbf{x}_d$ , build  $k - 1$  segment patterns.
3. For each segment pattern, find the mid-point of segments, mark it with the length of the corresponding segment.
4. Using the marked mid-points, create  $k - 1$  marked point patterns where each mark represents the length of movement per location.
5. Apply the function `idw` from the package **spatstat** (Baddeley et al., 2015) to each marked point pattern where it does inverse-distance weighted smoothing.
6. Step 5 returns  $k - 1$  maps in which the average of them is the output of function `Track.idw`.

Mathematically speaking, if for each point pattern  $\mathbf{x}_i$ , data points  $\{x_1, x_2, \dots, x_{n_i}\}$  are marked by  $\{l_1, l_2, \dots, l_{n_i}\}$  then the smoothed value at an arbitrary location

$u \in W$  is

$$\bar{g}(u) = \frac{1}{k-1} \left[ \frac{\sum_{j=1}^{n_i} w_j l_j}{\sum_{j=1}^{n_i} w_j} \right], \quad (5.1)$$

where

$$w_j = \frac{1}{(d(u, x_j))^p}, \quad (5.2)$$

in which  $d$  measures the distance between  $u$  and  $x_j \in \mathbf{x}_i$ ,  $l_j$  is the corresponding mark of  $x_j$  that is the length of the corresponding segment to  $x_j$ ,  $n_i$  is the number of points in the  $i$ -th pattern and  $p$  is an integer, being 2 as a default value. For details see Baddeley et al. (2015, Chapter 15).

In order to use `Track.idw`, we only need to set the argument `timestamp`. When collecting data, it may happen to record the location of objects which are actually stopped. Therefore and to not include them in the movement smoothing, there is an argument `epsilon` so that movements with length less than `epsilon` are not included in the computation. If no `epsilon` is set in the function, it then uses all segments. The following code generates an image of class 'im' which reflects the smoothed movement of taxis per 20 minutes.

```
R> b <- Track.idw(Beijing,timestamp = "20 mins",epsilon=1000)
R> plot(b,main="",ribwid=0.04,ribsep=0.02)
```

Figure 5.8 shows the movement smoothing for taxi data in Beijing according to the `timestamp= "20 mins"`. Here, we have not considered movements with length less than 1000 meters. In other words, we assume taxis with the length of movements less than 1000 meters per 20 minutes as stopped. This confirms that moving in the centre is slower than countryside/highways in Beijing, and in particular, it reveals some highways/freeways in which taxis are moving faster.

After smoothing the length of movements over space, we now turn to see the changes in the average of the length of movements over time. The function `avemove` measures the average length of movements passed by a collection of tracks based on a desirable timestamps. Now, we apply this to the taxi data in Beijing as follows:



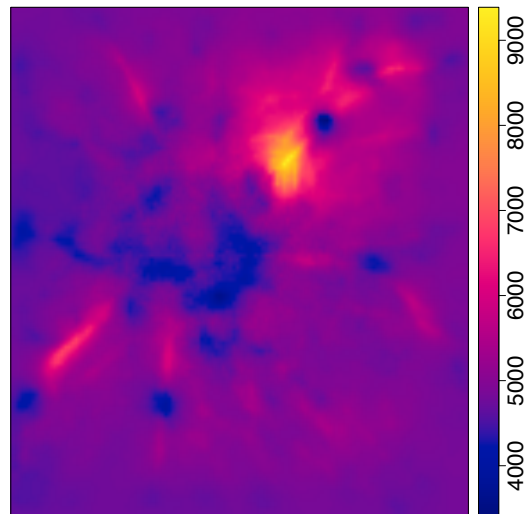


Figure 5.8: Movement smoothing for taxi data in Beijing, China based on `timestamp = "20 mins"` and movements with length longer than 1000 meters.

```
R> q <- avemove(Beijing,timestamp = "20 mins",epsilon=1000)
R> par(mfrow=c(1,2))
R> plot(q,type="l",lwd=2,cex.axis=1.7,cex.lab=1.7)
R> qdata <- data.frame(q,attr(q,"tsq")[-c(1,length(attr(q,"tsq"))
+                               )])
R> colnames(qdata) <- c("dist","startingtime")
R> q3rd <- qdata[substr(qdata$startingtime,start = 1,stop=10)
+               == "2008-02-03",]
R> plot(q3rd$startingtime,q3rd$dist,type="l",xlab="time (hour)"
+       ,ylab="average movement",lwd=2,cex.axis=1.7,cex.lab=1.7)
```

Figure 5.9 shows the average length of movements per 20 minutes by taxis in Beijing. The daily trend can be seen in the left plot, and the right plot shows that between midnight and early morning, the average length of movements is decreasing while from morning till noon there is an increase in the length of movements. In the afternoon, there can be seen a decrease in the average length of movements which might be caused by traffic.

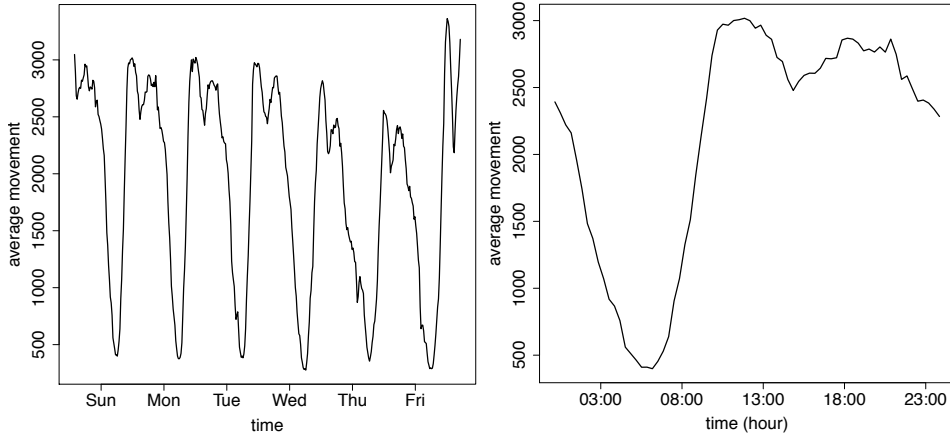


Figure 5.9: Average length of movements by taxis in Beijing, China versus time based on `timestamp = "20 mins"`, and movements with length longer than 1000 meters. *Left*: Within the period 2-8 Feb 2008. *Right*: During the 3-*rd* of Feb 2008.

#### 5.4.4 Intensity function

Being able to convert a trajectory pattern to a list of spatial point patterns as explained in Section 5.4.2, we now present an average intensity function for trajectory patterns as, e.g. using “Jones-Diggle” correction (1.10),

$$\widehat{\lambda}(u) = \frac{1}{k} \sum_{i=1}^k \widehat{\lambda}_i^{JD}(u) = \frac{1}{k} \sum_{i=1}^k \sum_{j=1}^{n_i} \frac{\kappa(u - x_j)}{e_W(x_j)}, \quad u \in W, \quad (5.3)$$

Such estimator can highlight more visited areas based on the tracks of moving objects (e.g. humans, cars) within the time period in question. The intensity estimator (5.3) can be calculated using the following steps:

1. Follow steps 1 – 3 as in Section 5.4.2.
2. For each  $\mathbf{x}_i$ , estimate the intensity function, say,  $\lambda_i$ .
3. The average over all estimated intensity functions  $\widehat{\lambda}_i$  may be considered as an estimated intensity for the trajectory pattern  $S$ .

Intuitively, we interpret  $\widehat{\lambda}(u)$  as the average expected number of points (objects) within the time period  $T$  in a small area around  $u$ . As a simple example, consider

the movements of cars within a city in a particular day. The estimator (5.3) reveals the more dense streets, highways, freeways, etc in that day. The intensity estimator (5.3), can use both edge corrections (1.8) and (1.10), is implemented in the **trajectories** package using the function `density.list`. The function `density.list` builds the point patterns  $x_1, x_2, \dots, x_k$  and pass them to the function `density.pps` in the package **spatstat**.

We next turn to calculate the average estimated intensity of the taxi data using the estimator (5.3).

```
R> d <- density(Beijing,timestamp = "20 mins",bw.ppl)
R> par(mfrow=c(1,2))
R> plot(d,main="",ribwid=0.04,ribsep=0.02)
+ #focus on the center
R> w <- owin(c(440000,455000),c(4410000,4430000))
R> pps <- attr(d,"ppps")
R> npps <- lapply(X=1:length(pps),FUN = function(i){
+   pps[[i]][w]
+ })
+
R> centerimg <- lapply(X=1:length(npps),FUN = function(i){
+   density(npps[[i]],bw.ppl(npps[[i]]))
+ })
+
R> fcenterimg <- Reduce("+",centerimg)/length(centerimg)
+
R> plot(fcenterimg,main="",ribwid=0.04,ribsep=0.02)
```

Figure 5.10 shows the estimated intensity using the uniform edge correction for both Beijing and its metropolitan area. The bandwidth has been selected using a likelihood cross-validation method and the function `bw.ppl` in **spatstat**. Other bandwidth selection methods can also be passed to `density.list`. Figure 5.10 highlights the most well-traveled areas in which those areas in the countryside with higher intensities (left plot) are some townships or airports. The right plot highlights the crowded routes within the centre of Beijing, China.

One may still think of adaptive intensity estimators, see Chapters 2 and 3. We here point out that as the estimator (5.3) is built based on an average of estimated

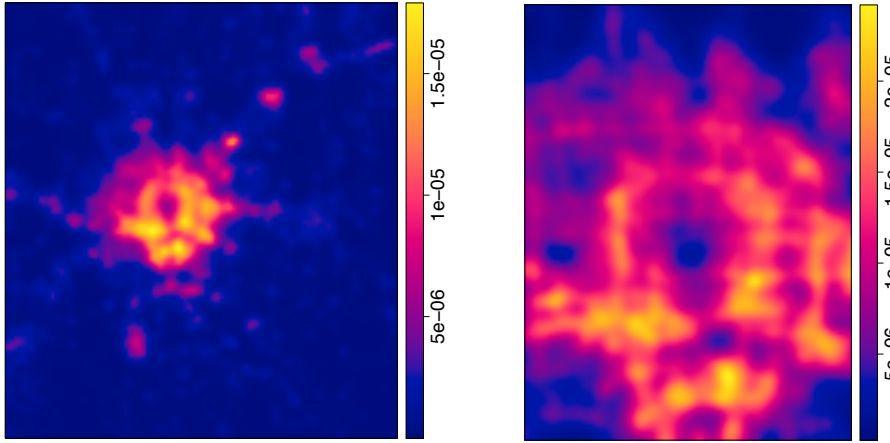


Figure 5.10: Estimated intensity function. *Left*: Beijing. *Right*: Beijing metropolitan area.

intensities of a set of spatial point patterns, one can estimate each of  $\lambda_i$  using adaptive estimators resulting in a final adaptive estimator for the corresponding trajectory pattern.

### 5.4.5 Chi maps

After discretising the trajectory pattern  $S$  to some point patterns and being able to estimate the individual intensity functions  $\lambda_i(\cdot)$ , one may think of discovering the areas with more/less events than the expected number. This motivates us to think of  $\chi^2$  statistics

$$\chi^2 = \frac{o - e}{\sqrt{e}}, \quad (5.4)$$

which measures the discrepancy between the expected number ( $e$ ) and the observed number ( $o$ ). This can be easily applied to the estimated intensity functions  $\hat{\lambda}_1, \hat{\lambda}_2, \dots, \hat{\lambda}_k$  and in any time  $t_i, i = 1, \dots, k$ , and as a result we can see where the estimated intensity differs from the expected intensity. For example, for a fixed time  $t = t_1$ ,

$$e_{t_1}(u) = \frac{\sum_{i=1}^k \hat{\lambda}_i(u) \sum_{v \in W} \hat{\lambda}_1(v)}{\sum_{i=1}^k \sum_{v \in W} \hat{\lambda}_i(v)}, \quad u \in W,$$

is the expected intensity at time  $t = t_1$  and location  $u \in W$ . Doing so for all  $u \in W$  enables us to draw a map of  $\chi^2$  values in a fixed time. The resulting map discloses the areas where the estimated intensity differs from the expected intensity. The function `chimaps` generates a map based on a given timestamp and rank. The argument `rank` is a number between one and the length of the generated time sequence based on the given timestamp, and with default one.

The chi maps of the 3-rd of Feb based on three different ranks are displayed in Figure 5.11. Values of each pixel are calculated by equation (5.4). We show the chi maps for three different times during the day in which changes over time can be seen. The left plot of Figure 5.11 shows the chi map at 06:10:44 so that the estimated intensity is higher than the expected intensity in the countryside. The reason for this might be the movements from countryside to the city center in the early morning. The middle plot of Figure 5.11 shows that the estimated intensity in the city is higher than the expected intensity. This may be caused by heavier traffic in the city during the day than in the countryside. In the right plot of Figure 5.11, although the estimated intensity is still slightly higher than the expected one in the city, we can see that the  $\chi^2$  statistic (5.4) takes values around 0 almost everywhere at night. These three plots together confirm the changes in the values of the  $\chi^2$  statistic (5.4) over time so that the mass is moving to the city in the morning and goes away in the evening. This behaviour may be explained by the movements to the city in the morning and moving back to the countryside in the evening.

```
R> ch <- chimaps(Beijing,timestamp = "20 mins",rank = 200)
R> chall <- attr(ch,"ims")
R> minmax <- lapply(X=1:length(chall),function(i){
+   return(list(min(chall[[i]]$v),max(chall[[i]]$v)))
+ })
R> minmax <- do.call("rbind",minmax)
R> col5 <- colorRampPalette(c('blue','white','red'))
R> color_levels <- 200
R> par(mar=c(0,0,1,1))
R> par(mfrow=c(1,3))
R> plot(chall[[51]],zlim=c(-max(abs(unlist(minmax))),
+   max(abs(unlist(minmax))))),main=attr(ch,"timevec")[51]
+   ,ribwid=0.04,ribsep=0.02,col=col5(n=color_levels))
```

```

+
R> plot(chall[[75]],zlim=c(-max(abs(unlist(minmax))),
+       max(abs(unlist(minmax)))),main=attr(ch,"timevec")[75]
+       ,ribwid=0.04,ribsep=0.02,col=col5(n=color_levels))
+
R> plot(chall[[104]],zlim=c(-max(abs(unlist(minmax))),
+       max(abs(unlist(minmax)))),main=attr(ch,"timevec")[104]
+       ,ribwid=0.04,ribsep=0.02,col=col5(n=color_levels))

```

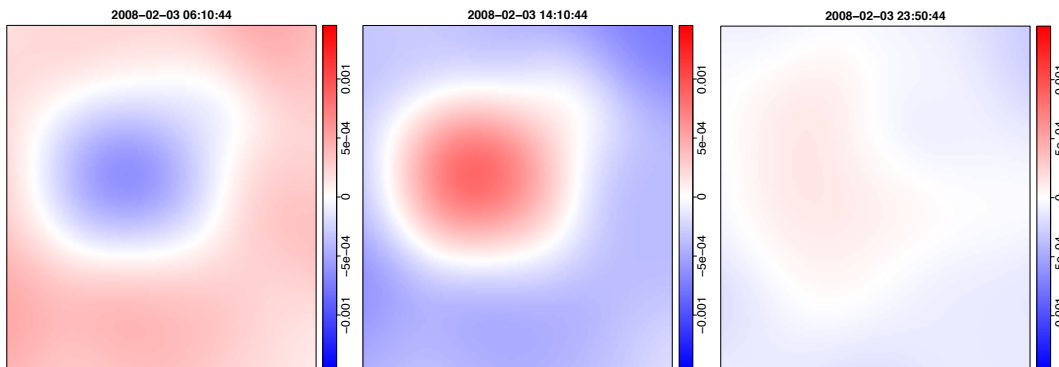


Figure 5.11: Chi maps. *Left*: in the morning, *Middle*: in the afternoon, *Right*: at night. Exact time is reported on top of each plot.

### 5.4.6 Second-order summary statistics

After discretising a trajectory pattern and estimating the intensity function of each single resulted point pattern, we now turn to look at the interaction between the moving objects over time. We are interested in distinguishing whether objects tend to move independently or they show some kind of dependence (e.g. clustering or inhibition). A common way in the point process literature is to use summary statistics such as  $K$ -function and pair correlation function (see Section 1.2.4). Pairwise distances are the hint here: if objects tend to be close to each other, then most of the pairwise distances are going to be small, and if they favour to stand far, then only a few of the pairwise distances are small (Baddeley et al., 2015). Having this in mind, we next turn to use these functions in analysing the trajectory pattern  $S$ . Similar to our proposal for the intensity function and using summary statistics

for point patterns, we here propose a variability area for the  $K$ -function and the pair correlation function as follows:

1. According to regular timestamps, discretise the trajectory pattern  $S$  and build the point patterns  $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_k$ .
2. For all the resulted point patterns  $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_k$ , estimate the  $K$ -function using (1.22).
3. From the estimated  $K$ -functions  $\widehat{K}_1, \widehat{K}_2, \dots, \widehat{K}_k$ , build the pointwise variability area of the  $K$ -function, i.e. for each value of distance argument  $r$ , sort  $\widehat{K}_i(r)$  and then take the lowest and highest value amongst all; doing so for a sequence of  $r$  results in a variability area for the  $K$ -function. This shows how the type of interaction between objects changes over time.

Note that steps above can be applied to the pair correlation function as well. Variability area of  $K$ -function and pair correlation function for trajectory patterns are accommodated in functions `Kinhom.Track` and `pcfinhom.Track` respectively.

In calculating both  $K$ -function and pair correlation function, users can take advantage of the bandwidth selection to first estimate the intensity function and then pass estimated intensities to the function `Kinhom` or `pcfinhom`. The default is to not pass any estimated intensity function to `Kinhom` or `pcfinhom` in which the intensity will be estimated using the ‘leave-one-out’ kernel smoother (Baddeley et al., 2000, 2015). Different edge corrections can be also passed to `Kinhom.Track` and `pcfinhom.Track`.

Finally and taking into account that estimated intensity in Figure 5.10 represents a non-uniform distribution and/or clustering behaviour. We show the variability area of  $K$ -function and pair correlation function over time in Figure 5.12 (considering the “translate” correction, see Gabriel (2014)). The left plot displays the variation of  $K$ -function, showing that for small distances taxis tend to have a clustering behaviour while for larger distances they favour inhibition. The right plot of the variation of the pair correlation function also confirms the same behaviour. Due to the preference of moving within particular zones,  $K$ -function and pair correlation function might result as what is displayed in Figure 5.12. In other words, taxis might prefer to take passengers to close destinations within particular zones rather than

further destinations. Argument  $q$  is to use quantiles of the values of  $K$ -function (or pair correlation function) rather than using maximum and minimum. Default  $q=0$  uses maximum and minimum.

```
R> K <- Kinhom.Track(Beijing,correction = "translate",
+                   timestamp = "20 mins",q=0)
R> par(mfrow=c(1,2))
R> plot(K,cex.axis=1.7,cex.lab=1.5,cex=2)
R> g <- pcfinhom.Track(Beijing,timestamp = "20 mins",q=0)
R> plot(g,cex.axis=1.7,cex.lab=1.5,cex=2)
```

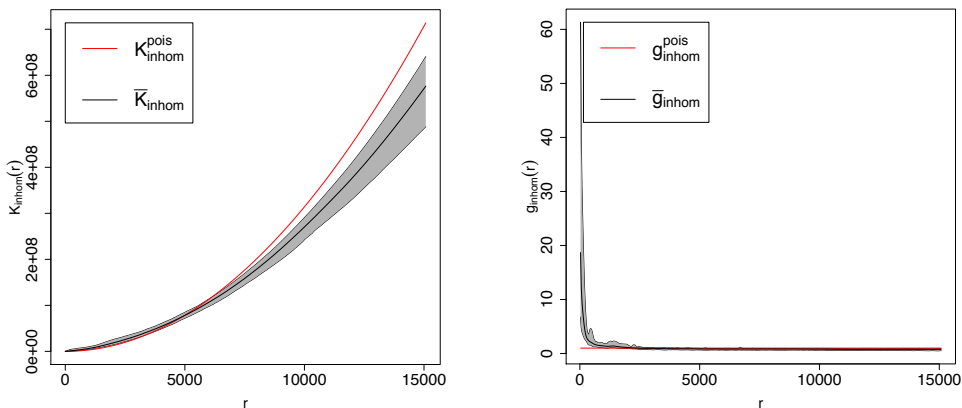


Figure 5.12: Variability area of second-order summary statistics for taxi data in Beijing, China. *Left*:  $K$ -function, *Right*: pair correlation function.

We point out that as this kind of discretising trajectory patterns results in a set of spatial point patterns, it may also drive us to consider them as replicated spatial point patterns. Therefore, one may be able to introduce a single  $K$ -function or pair correlation function for all patterns. Note that in this case, point patterns are not independent. For more details, see Diggle (2003, Section 5.4).

## 5.5 Summary and discussion

Analysing trajectory data is often a challenge as moving objects are living in space whereas moving in time. Such data might be analysed in space or time separately.

In this Chapter, we introduced a set of classes to handle trajectory data and defined a set of methods to summarise trajectory patterns. We have considered



different classes to handle tracks passed by one object or a group of objects. We also aimed at statistically analysing trajectory data in space and see the changes in the behaviour of data over time. To do so, the literature of spatial point processes has been adapted to trajectory patterns. An average intensity function is defined for a trajectory pattern with tracks that overlap in time and it nicely reveals the more visited spots, streets, highways, etc. Pairwise distances between different objects have been also considered in **trajectories** showing how moving objects interact over time. Moreover, we have implemented methods to smooth the length of movements over time which highlights where objects move faster/slower. In addition, the discrepancy between the estimated intensity and the expected intensity per location per time can be studied in **trajectories** which shows how the mass is moving within the studied area over time. We finally presented a variability area for second-order summary statistics such as  $K$ -function and pair correlation function which displays the type of interaction between objects over time, highlighting possible clustering or inhibition.

To perform an exploratory data analysis, we finally applied the developed methodologies to taxi data from Beijing, China. The results nicely highlight the behaviour of taxis over time. We have recognised where the more visited spots are and also at what time data pattern is denser or objects get closer to each other. We smoothed the movements of taxis and discovered where and what time they move faster. Not surprisingly, results confirm that moving within the city is slower than countryside or highways. Also, in the afternoon there is a decrease in the length of movements which might be affected by traffic. Chi maps have been studied showing how the mass is moving within the studied area. They display that objects are moving from countryside to the city in the morning reporting a heavy traffic in the city in the afternoon, as the estimated intensity is higher than the expected one. Chi maps also confirm that the discrepancy between the estimated intensity and the expected one is smaller during the evening than any other time. The variability area of  $K$ -function and pair correlation function show that for small distances, taxis tend to show a clustering behaviour while for longer distances they show repulsion.



## CHAPTER 6

---

# Conclusions and Future work

---

In this thesis we have mainly focused on spatial and spatio-temporal point processes on linear networks and trajectory analysis. Throughout the thesis, we have developed: 1) several intensity estimation methods for network events, with both adaptive and non-adaptive approaches; 2) an intensity estimator and second-order summary statistics for spatio-temporal network data; 3) classes and methods to analyse movement data. Publications associated with the thesis research are listed in Appendix A.

In Chapter 2, we broadly discussed the literature of spatial point processes on linear networks. The current intensity estimators were reviewed and it seemed to be challenging to estimate the intensity function of network events. Two different kernel intensity estimators for spatial point patterns on linear networks are defined and compared with the current intensity estimators. Their theoretical properties such bias, variance, asymptotic results, bandwidth selection, variance estimation, relative risk estimation and adaptive smoothing are also studied. We first introduced a kernel intensity estimator which is an extension of ‘Jones-Diggle’ intensity estimator in  $\mathbb{R}^2$  (equation (1.10)) to the case of linear networks. This estimator provides a continuous function over the linear network (subject to using a continuous kernel function), it is mass preserved and its variance can be estimated numerically. In terms of MISE, it shows a better general performance comparing to the equal-split discontinuous estimator (Okabe et al., 2009; Okabe and Sugihara, 2012). Second, we proposed a kernel-based intensity estimator using 2D convolution. The main motivation for this estimator was computation time.

The proposed estimator is computationally efficient and a statistically principled method for kernel smoothing of point pattern data on a linear network. The point locations, and the network itself, are convolved with a two-dimensional kernel, then combined into an intensity function on the network. This can be computed rapidly using the Fast Fourier Transform, even on large networks and for large bandwidths, and is robust against errors in network geometry. The estimator is consistent and its statistical efficiency is only slightly sub-optimal.

In Chapter 3, we paid particular attention to Voronoi intensity estimators. These estimators are fully non-parametric and reflect the local features nicely. However, the problem with such estimators is visible when in the point pattern there are very dense parts surrounded by possibly empty neighbourhoods. In such situation, the Voronoi cell becomes very small and consequently the intensity becomes huge with flat intensity close by. This generates an intensity estimate with a huge variance. In order to smooth out the sharp boundaries and reduce the variance, we proposed an additional smoothing technique for intensity estimators for point processes in arbitrary metric spaces, which is based on repeated independent thinnings of the point pattern. Through a simulation study we showed that our resample-smoothing technique improves the estimation significantly. In addition, we discussed unbiasedness and variance, and proposed a rule-of-thumb and a data-driven cross-validation approach to choose the amount of thinning/smoothing to apply.

In Chapter 4, we provided preliminaries of spatio-temporal point processes on linear networks. This being said, we allowed time component to be included in analysing spatio-temporal network data. Spatial and temporal component processes were discussed together with their intensity functions. We presented a kernel-based intensity estimator for spatio-temporal point patterns on networks which preserves mass. The second-order summary statistics  $K$ -function and pair correlation function were defined to analyse the type of interaction between the data points. We showed that the  $K$ -function and the pair correlation function are independent of the geometry of network and have known values for spatio-temporal Poisson processes on networks. This then could be used as a benchmark to measure the deviation from being Poisson and also for model selection.

In Chapter 5, we focused on movement data such as taxi movement and discussed the R package **trajectories** which provides different classes and methods to handle and summarise trajectory data in R. Studying the behaviour of moving

objects over time and their interaction, either between objects or with environment, plays a crucial role in understanding how they use space and more importantly how they interact with each other. Objects are moving within a particular area over time, thus a snapshot of a trajectory pattern might be seen as a spatial point pattern. Package **trajectories** provides simulation of tracks and model fitting. Moreover, using the literature of point processes, we developed different methodologies such as estimating intensity functions, distance analysis, second-order summary statistics, Chi maps and movement smoothing for movement data. The proposed statistical methods are accommodated in R package **trajectories** which make them reusable for users. We also provided details of how to make use of the proposed methods in practice.

Regarding future works, there are several lines of research arising from this work which should be pursued. In Chapter 2, we discussed intensity estimators and relative risk non-parametrically. It is our intention to incorporate possible covariates in the play and analyse their effect on the intensity function and relative risk. Working with network events, possible covariates might be distances to particular places such as roundabouts, intersections, parks, cinemas, universities, hospitals, etc. However, parametric model fitting still suffers from model development. Having said this, we also plan to develop area-interaction models for network events. Another interesting line to pursue is to develop higher-order summary statistics for inhomogeneous point processes on linear networks. This however demands developing suitable mathematical transformation methods and statistical justification. The linear network plays an important role when analysing network data, nevertheless, the network may change over time. Therefore, this might also be taken into account when performing spatio-temporal analysis. The idea of resample-smoothing can also be adapted to some other tessellation, generated by the point pattern in question. In terms of trajectory analysis, important features affecting object's movement might include points of interest in cities, weather, pollution, urban design and so forth. However, analysing such effects requires parametric studies that are not included in the current version of R package **trajectories** and are left for the future.



## APPENDIX A

---

# Publications and research visits

---

- **Journal papers with impact factor:**

1. **Mohammad Mehdi Moradi**, Francisco Rodríguez-Cortés and Jorge Mateu. (2017). On kernel-based intensity estimation of spatial point patterns on linear networks. *Journal of Computational and Graphical Statistics*. Volume **27 (2)**, 302–311.
2. Khoi Manh Ngo, Sven Casteleyn, **Mohammad Mehdi Moradi**, and Edzer Pebesma. (2018). Do monetary incentives influence users' behaviour in participatory sensing?. *Sensors*. Volume **18 (5)**, 14–26.
3. Suman Rakshit, Tilman M. Davies, **Mohammad Mehdi Moradi**, Greg McSwiggan, Gopalan Nair, Jorge Mateu, and Adrian Baddeley. (2018). Fast kernel smoothing of point patterns on a large network using 2d convolution. Submitted for publication.
4. **Mohammad Mehdi Moradi**, Ottmar Cronie, Ege Rubak, Raphael Lachize-Rey, Jorge Mateu, and Adrian Baddeley. (2018). Resample-smoothing of Voronoi intensity estimators. Submitted for publication.
5. **Mohammad Mehdi Moradi**, Edzer Pebesma, and Jorge Mateu. (2018). trajectories: Classes and Methods for Trajectory Data. Submitted for publication.
6. **Mohammad Mehdi Moradi**, Jorge Mateu, and Adrian Baddeley. (2018). First and second-order characteristics of spatio-temporal point processes on linear networks. Submitted for publication.

7. **Mohammad Mehdi Moradi**, Nicholas Read, Jorge Mateu, and Adrian Baddeley. (2018). Normalisation of the inhomogeneous  $K$ -function and pair correlation function. Submitted for publication.

- **Conference papers:**

1. **Mohammad Mehdi Moradi**, Francisco Rodríguez-Cortés and Jorge Mateu. (2016). An adapted intensity estimator for linear networks with an application to modeling anti-social behavior in an urban environment. *METMA VIII. The University of Valencia, Valencia – Spain*. ISBN: 978-84-608-8468-2.
2. **Mohammad Mehdi Moradi**, Ottmar Cronie and Jorge Mateu. (2017). The smoothing of Voronoi Intensity estimators and its application to network point pattern data. *Spatial Statistics 2017: One World, One Health, Lancaster University, UK*.

Note: This publication was awarded *the best poster presentation* by the scientific committee of the conference.

- **Workshops:**

1. Fernando Benitez, **Mohammad Mehdi Moradi**, Joaquín Huerta, Pau Arago, Jorge Mateu, Marco Painho and Guiying Du. (2017). Open Data for Open Cities: Re-use and discovery level, applied to the spatial point process on linear networks. *AGILE conference 2017. The University of Wageningen, the Netherlands*.

- **Research visits:**

1. Department of Mathematics and Mathematical Statistics, Umeå University, Sweden. Dr. Ottmar Cronie. 16 September 2016 until 2 October 2016.
2. Institute for Geoinformatics, Muenster University, Germany. Prof. Edzer Pebesma. 7 October 2016 until 28 December 2016.
3. Faculty of Science and Engineering, School of Elec Eng, Comp and Math Sci (EECMS), Curtin University, Australia. Prof. Adrian Baddeley. 21 March 2017 until 3 May 2017.



4. Department of Mathematical Sciences, Aalborg University, Denmark. Dr. Ege Rubak. 5 November 2017 until 11 November 2017.
5. Institute for Geoinformatics, Muenster University, Germany. Prof. Edzer Pebesma. 7 October 2017 until 18 December 2017.

• **Internship:**

1. Policia local de Castellón. 9 January 2017 until 10 February 2017.



## Bibliography

---

- ABRAMSON, I. S. (1982). On bandwidth variation in kernel estimates - A square root law. *The Annals of Statistics*, **10(4)**, pp. 1217–1223.
- ADLER, R. J. (1981). *The Geometry of Random Fields*. Siam.
- ANDERES, E.; MØLLER, J. and RASMUSSEN, J. (2017). Isotropic covariance functions on graphs and their edges. *arXiv preprint arXiv:1710.01295*.
- ANG, Q. W.; BADDELEY, A. and NAIR, G. (2012). Geometrically corrected second order analysis of events on a linear network, with applications to ecology and criminology. *Scandinavian Journal of Statistics*, **39(4)**, pp. 591–617.
- BADDELEY, A. (2007). Validation of statistical models for spatial point patterns. In: J.G. Babu and E.D. Feigelson (Eds.), *Statistical Challenges in Modern Astronomy IV*, volume 371 of *Astronomical Society of the Pacific, Conference Series*, pp. 22–38. Astronomical Society of the Pacific, San Francisco, California, USA.
- BADDELEY, A.; JAMMALAMADAKA, A. and NAIR, G. (2014). Multitype point process analysis of spines on the dendrite network of a neuron. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, **63(5)**, pp. 673–694.
- BADDELEY, A.; MØLLER, J. and WAAGEPETERSEN, R. (2000). Non- and semi-parametric estimation of interaction in inhomogeneous point patterns. *Statistica Neerlandica*, **54(3)**, pp. 329–350.
- BADDELEY, A.; NAIR, G.; RAKSHIT, S. and MCSWIGGAN, G. (2017). “Stationary” point processes are uncommon on linear networks. *Stat*, **6(1)**, pp. 68–78.

- BADDELEY, A.; RUBAK, E. and TURNER, R. (2015). *Spatial Point Patterns: Methodology and Applications with R*. CRC Press.
- BADDELEY, A. and TURNER, R. (2005). Spatstat: an R package for analyzing spatial point patterns. *Journal of Statistical Software*, **12(6)**, pp. 1–42.
- BARR, CH. D. and SCHOENBERG, F. P. (2010). On the Voronoi estimator for the intensity of an inhomogeneous planar Poisson process. *Biometrika*, **97(4)**, pp. 977–984.
- BARRY, R. P. and MCINTYRE, J. (2011). Estimating animal densities and home range in regions with irregular boundaries and holes: A lattice-based alternative to the kernel density estimator. *Ecological Modelling*, **222(10)**, pp. 1666–1672.
- BATTAILE, B. (2014). *TrackReconstruction: Reconstruct animal tracks from magnetometer, accelerometer, depth and optional speed data*. R package version 1.1 <https://cran.r-project.org/package=TrackReconstruction>.
- BERMAN, M. and DIGGLE, P. (1989). Estimating weighted integrals of the second-order intensity of a spatial point process. *Journal of the Royal Statistical Society, Series B*, **51(1)**, pp. 81–92.
- BERMAN, M. and TURNER, R. (1992). Approximating point process likelihoods with GLIM. *Applied Statistics*, **41(1)**, pp. 31–38.
- BITHELL, J. F. (1991). Estimation of relative risk functions. *Statistics in Medicine*, **10(11)**, pp. 1745–1751.
- BORRUSO, G. (2003). Network density and the delimitation of urban areas. *Transactions in GIS*, **7(2)**, pp. 177–191.
- BORRUSO, G. (2005). Network Density Estimation: Analysis of Point Patterns over a Network. In: *Computational Science and its Applications — ICCSA 2005*, Number 3482 in Lecture Notes in Computer Science, pp. 126–132. Springer, Berlin/Heidelberg.
- BORRUSO, G. (2008). Network Density Estimation: A GIS Approach for Analysing Point Patterns in a Network Space. *Transactions in GIS*, **12(3)**, pp. 377–402.

- BOTEV, Z. I.; GROTOWSKI, J. F. and KROESE, D. P. (2010). Kernel density estimation via diffusion. *Annals of Statistics*, **38(5)**, pp. 2916–2957.
- BREIMAN, L.; MEISEL, W. and PURCELL, E. (1977). Variable kernel estimates of multivariate densities. *Technometrics*, **19(2)**, pp. 135–144.
- BROWN, G. S. (1965). *Point Density in Stems per Acre*. Forest Research Institute, New Zealand Forest Service.
- CALENGE, C. (2006). The package “adehabitat” for the R software: a tool for the analysis of space and habitat use by animals. *Ecological Modelling*, **197(3)**, pp. 516–519.
- CAMPBELL, H. A.; WATTS, M.; DWYER, R. G. and FRANKLIN, C. E. (2012). V-Track: software for analysing and visualising animal movement from acoustic telemetry detections. *Marine and Freshwater Research*, **63(9)**, pp. 815–820.
- CHALLA, S.; MORELANDE, M.; MUSICKI, D. and EVANS, R. (2011). *Fundamentals of Object Tracking*. Cambridge University Press.
- CHAUDHURI, P. and MARRON, J. S. (2000). Scale space view of curve estimation. *Annals of Statistics*, **28(2)**, pp. 408–428.
- CHIU, S. N.; STOYAN, D.; KENDALL, W. S. and MECKE, J. (2013). *Stochastic Geometry and its Applications*. John Wiley & Sons.
- CRONIE, O. and VAN LIESHOUT, M. N. M. (2018). A non-model-based approach to bandwidth selection for kernel estimators of spatial intensity functions. *Biometrika*, **105(2)**, pp. 455–462.
- DALEY, D. J. and VERE-JONES, D. (2003). *An Introduction to the Theory of Point Processes: Volume I: Elementary Theory and Methods*. Springer-Verlag New York.
- DALEY, D. J. and VERE-JONES, D. (2008). *An Introduction to the Theory of Point Processes: Volume II: General Theory and Structure*. Springer-Verlag New York.
- DAVIES, T. M. (2013). Jointly optimal bandwidth selection for the planar kernel-smoothed density-ratio. *Spatial and Spatio-Temporal Epidemiology*, **5**, pp. 51–65.

- DAVIES, T. M. and BADDELEY, A. (2018). Fast computation of spatially adaptive kernel estimates. *Statistics and Computing*, **28(4)**, pp. 937–956.
- DAVIES, T. M. and HAZELTON, M. L. (2010). Adaptive kernel estimation of spatial relative risk. *Statistics in Medicine*, **29(23)**, pp. 2423–2437.
- DAVIES, T. M.; JONES, KH. and HAZELTON, M. L. (2016). Symmetric adaptive smoothing regimens for estimation of the spatial relative risk function. *Computational Statistics & Data Analysis*, **101**, pp. 12–28.
- DAVIES, T. M.; MARSHALL, J. C. and HAZELTON, M. L. (2018). Tutorial on kernel estimation of continuous spatial and spatiotemporal relative risk. *Statistics in Medicine*, **37(7)**, pp. 1191–1221.
- DIGGLE, P. (1985). A kernel method for smoothing point process data. *Applied Statistics*, **34(2)**, pp. 138–147.
- DIGGLE, P. (2003). *Statistical Analysis of Spatial and Spatio-Temporal Point Patterns*. CRC Press.
- DIGGLE, P.; ZHENG, P. and DURR, P. (2005). Non-parametric estimation of spatial segregation in a multivariate point process: bovine tuberculosis in Cornwall, UK. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, **54(3)**, pp. 645–658.
- DUYCKAERTS, CH. and GODEFROY, G. (2000). Voronoi tessellation to study the numerical density and the spatial distribution of neurones. *Journal of Chemical Neuroanatomy*, **20(1)**, pp. 83–92.
- DUYCKAERTS, CH.; GODEFROY, G. and HAUW, J. (1994). Evaluation of neuronal numerical density by Dirichlet tessellation. *Journal of Neuroscience Methods*, **51(1)**, pp. 47–69.
- EBELING, H. and WIEDENMANN, G. (1993). Detecting structure in two dimensions combining Voronoi tessellation and percolation. *Physical Review E*, **47(1)**, p. 704.
- ELIEZER, G. (2014). *bcpa: Behavioral change point analysis of animal movement*. R package version 1.1 <https://cran.r-project.org/package=bcpa>.

- FARRELL, E. D. and FUIMAN, L. (2013). *animalTrack: Animal track reconstruction for high frequency 2-dimensional (2D) or 3-dimensional (3D) movement data*. R package version 1.0.0 <https://cran.r-project.org/package=animalTrack>.
- FEDERER, H. (1996). *Geometric Measure Theory*. Springer-Verlag.
- FREITAS, C. (2012). *argosfilter: Argos locations filter*. R package version 0.63. <https://CRAN.R-project.org/package=argosfilter>
- FRICK, H. and KOSMIDIS, I. (2017). trackeR: Infrastructure for Running and Cycling Data from GPS-Enabled Tracking Devices in R. *Journal of Statistical Software*, **82(7)**, pp. 1–29. ISSN 1548-7660. <https://www.jstatsoft.org/v082/i07>.
- FRIGO, M. and JOHNSON, S. G. (2005). The Design and Implementation of FFTW3. *Proceedings of the IEEE*, **93(2)**, pp. 216–231. Special issue on “Program Generation, Optimization, and Platform Adaptation”.
- GABRIEL, E. (2014). Estimating second-order characteristics of inhomogeneous spatio-temporal point processes. *Methodology and Computing in Applied Probability*, **16(2)**, pp. 411–431.
- GABRIEL, EDITH and DIGGLE, PETER (2009). Second-order analysis of inhomogeneous spatio-temporal point process data. *Statistica Neerlandica*, **63(1)**, pp. 43–51.
- GELFAND, A. E.; DIGGLE, P.; GUTTORP, P. and FUENTES, M. (2010). *Handbook of Spatial Statistics*. CRC press.
- GHOORBANI, MOHAMMAD (2013). Testing the weak stationarity of a spatio-temporal point process. *Stochastic Environmental Research and Risk Assessment*, **27(2)**, pp. 517–524.
- GÜTING, R. H. and SCHNEIDER, M. (2005). *Moving Objects Databases*. Elsevier.
- HALL, P. and MARRON, J. S. (1988). Variable window width kernel estimates of probability densities. *Probability Theory and Related Fields*, **80(1)**, pp. 37–49.
- HANKS, E. M.; HOOTEN, M. B. and ALLDREDGE, M. W. (2015). Continuous-time discrete-space models for animal movement. *The Annals of Applied Statistics*, **9(1)**, pp. 145–165.

- HAZELTON, M. L. (2008). Kernel estimation of risk surfaces without the need for edge correction. *Statistics in Medicine*, **27(12)**, pp. 2269–2272.
- HAZELTON, M. L. and DAVIES, T. M. (2009). Inference based on kernel estimates of the relative risk function in geographical epidemiology. *Biometrical Journal*, **51(1)**, pp. 98–109.
- HOOTEN, M. B. and JOHNSON, D. S. (2017). Basis function models for animal movement. *Journal of the American Statistical Association*, **112(518)**, pp. 578–589.
- ILLIAN, J.; PENTTINEN, A.; STOYAN, H. and STOYAN, D. (2008). *Statistical Analysis and Modelling of Spatial Point Patterns*. John Wiley & Sons.
- JEFF, A. T.; JAMES, SH.; JUN, ZH.; SINKOVTS, R.; CHOURASIA, A.; LOCKWOOD, G. and FISHER, R. N. (2014). *mkde: 2D and 3D movement-based kernel density estimates (MKDEs)*. R package version 0.1 <https://cran.r-project.org/package=mkde>.
- JONES, M. CH. (1993). Simple boundary correction for kernel density estimation. *Statistics and Computing*, **3(3)**, pp. 135–146.
- JONES, M. CH.; MARRON, J. S. and SHEATHER, S. (1996). A brief survey of bandwidth selection for density estimation. *Journal of the American Statistical Association*, **91(433)**, pp. 401–407.
- JUN, Y. and VLADIMIR, P. (2016). *smam: Statistical Modeling of Animal Movements*. R package version 0.3-0 <https://cran.r-project.org/package=smam>.
- KARIM, R. (2017). *fftwtools package*. R package version 0.9-8 <https://cran.r-project.org/package=fftwtools>.
- KELSALL, J. E and DIGGLE, P. (1995a). Kernel estimation of relative risk. *Bernoulli*, **1(1–2)**, pp. 3–16.
- KELSALL, J. E. and DIGGLE, P. (1995b). Non-parametric estimation of spatial variation in relative risk. *Statistics in Medicine*, **14(21-22)**, pp. 2335–2342.
- LAST, G. (2010). Stationary random measures on homogeneous spaces. *Journal of Theoretical Probability*, **23(2)**, pp. 478–497.



- LAWSON, A. B. and WILLIAMS, F. L. R. (1993). Applications of extraction mapping in environmental epidemiology. *Statistics in Medicine*, **12(13)**, pp. 1249–1258.
- LEVINE, N. (2006). Houston, Texas, Metropolitan Traffic Safety Planning Program. *Transportation Research Record: Journal of the Transportation Research Board*, **1969(1)**, pp. 92–100.
- LEVINE, NED (2009). A Motor Vehicle Safety Planning Support System: The Houston Experience. In: Stan Geertman and John Stillwell (Eds.), *Planning Support Systems Best Practice and New Methods*, pp. 93–111. Springer Netherlands, Dordrecht. ISBN 978-1-4020-8952-7.
- LOADER, C. (1999). *Local Regression and Likelihood*. Springer, New York.
- LOFTSGAARDEN, D. O. and QUESENBERRY, C. P. (1965). A non-parametric estimate of a multivariate density function. *Annals of Mathematical Statistics*, **36(3)**, pp. 1049–1051.
- LU, YONGMEI and CHEN, XUWEI (2007). On the false alarm of planar K-function when analyzing urban crime distributed along streets. *Social Science Research*, **36(2)**, pp. 611–632.
- MARK, A. F. and ESLER, A. E. (1970). An assessment of the point-centred quarter method of plotless sampling in some New Zealand forests. *Proceedings (New Zealand Ecological Society)*, **(17)**, pp. 106–110.
- MARSHALL, J. C. and HAZELTON, M. L. (2010). Boundary kernels for adaptive density estimators on regions with irregular boundaries. *Journal of Multivariate Analysis*, **101(4)**, pp. 949–963.
- MCSWIGGAN, G.; BADDELEY, A. and NAIR, G. (2017). Kernel density estimation on a linear network. *Scandinavian Journal of Statistics*, **44(2)**, pp. 324–345.
- MICHELOT, T.; LANGROCK, R. and PATTERSON, T. A. (2016). moveHMM: an R package for the statistical modelling of animal movement data using hidden Markov models. *Methods in Ecology and Evolution*, **7(11)**, pp. 1308–1315.
- MØLLER, J. and SCHOENBERG, F.P. (2010). Thinning spatial point processes into Poisson processes. *Advances in Applied Probability*, **42(2)**, pp. 347–358.

- MØLLER, J. and WAAGEPETERSEN, R. (2003). *Statistical Inference and Simulation for Spatial Point Processes*. CRC Press.
- MØLLER, JESPER and GHORBANI, MOHAMMAD (2012). Aspects of second-order analysis of structured inhomogeneous spatio-temporal point processes. *Statistica Neerlandica*, **66(4)**, pp. 472–491.
- MORADI, M. M.; CRONIE, O.; RUBAK, E.; LACHIEZE-REY, R.; MATEU, J. and BADDELEY, A. (2018a). Resample-smoothing of Voronoi intensity estimators. *Submitted for publication*.
- MORADI, M. M.; MATEU, J. and BADDELEY, A. (2018b). First and second-order characteristics of spatio-temporal point processes on linear networks. *Submitted for publication*.
- MORADI, M. M.; PEBESMA, E. and MATEU, J. (2018c). trajectories: Classes and Methods for Trajectory Data. *Submitted for publication*.
- MORADI, M. M.; READ, N.; MATEU, J. and BADDELEY, A. (2018d). Normalisation of the inhomogeneous  $K$ -function and pair correlation function. *In preparation*.
- MORADI, M. M.; RODRIGUEZ-CORTES, F. and MATEU, J. (2017). On kernel-based intensity estimation of spatial point patterns on linear networks. *Journal of Computational and Graphical Statistics*, **27(2)**, pp. 302–311.
- NADARAYA, E. A. (1964). On estimating regression. *Theory of Probability and its Applications*, **9(1)**, pp. 141–142.
- NADARAYA, E. A. (1989). *Nonparametric Estimation of Probability Densities and Regression Curves*. Kluwer, Dordrecht.
- NIELSON, R. M.; SAWYER, H. and McDONALD, T. L. (2013). *BBMM: Brownian bridge movement model*. R package version 3.0 <https://cran.r-project.org/package=BBMM>.
- NIU, M.; BLACKWELL, P. G. and SKARIN, A. (2016). Modeling interdependent animal movement in continuous time. *Biometrics*, **72(2)**, pp. 315–324.
- O'DONNELL, D.; RUSHWORTH, A.; BOWMAN, A. W.; MARIAN. SCOTT, E. and HALLARD, M. (2014). Flexible regression models over river networks. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, **63(1)**, pp. 47–63.

- OGATA, Y. (2011). Significant improvements of the space-time ETAS model for forecasting of accurate baseline seismicity. *Earth, Planets and Space*, **63(3)**, p. 6.
- OKABE, A.; BOOTS, B.; SUGIHARA, K. and CHIU, S.N. (2000). *Spatial Tessellations: Concepts and Applications of Voronoi Diagrams*. Wiley, second edition.
- OKABE, A. and SATOH, T. (2006). Uniform network transformation for points pattern analysis on a non-uniform network. *Journal of Geographical Systems*, **8(1)**, pp. 25–37.
- OKABE, A.; SATOH, T. and SUGIHARA, K. (2009). A kernel density estimation method for networks, its computational method and a GIS-based tool. *International Journal of Geographical Information Science*, **23(1)**, pp. 7–32.
- OKABE, A. and SUGIHARA, K. (2012). *Spatial Analysis along Networks: Statistical and Computational Methods*. John Wiley & Sons.
- OKABE, A. and YAMADA, I. (2001). The  $K$ -function method on a network and its computational implementation. *Geographical Analysis*, **33(3)**, pp. 271–290.
- ORD, J. K. (1978). How many trees in a forest? *Mathematical Sciences*, **3**, pp. 23–33.
- PEBESMA, E. (2012). spacetime: Spatio-Temporal data in R. *Journal of Statistical Software*, **51(7)**, pp. 1–30.
- PINSKY, M. A. (2002). *Introduction to Fourier Series and Wavelets*. American Mathematical Society.
- RADINGER, J. and WOLTER, CH. (2014). Patterns and predictors of fish dispersal in rivers. *Fish and Fisheries*. R package “fishmove”, version 0.3-3 <http://dx.doi.org/10.1111/faf.12028>.
- RAKHIMBERDIEV, E.; SAVELIEV, A.; PIERSMA, T. and KARAGICHEVA, J. (2017). FLightR: an R package for reconstructing animal paths from solar geolocation loggers. *Methods in Ecology and Evolution*.
- RAKSHIT, S.; DAVIES, T. M.; MORADI, M. M.; MCSWIGGAN, G.; NAIR, G.; MATEU, J. and BADDELEY, A. (2018). Fast Kernel Smoothing of Point Patterns on a Large Network using 2D Convolution. *Submitted for publication*.

- RAKSHIT, S.; NAIR, G. and BADDELEY, A. (2017). Second-order analysis of point patterns on a network using any distance metric. *Spatial Statistics*, **22(1)**, pp. 129–154.
- RIPLEY, B. D. (1977). Modelling spatial patterns. *Journal of the Royal Statistical Society. Series B (Methodological)*, **39(2)**, pp. 172–212.
- RIPLEY, B. D. (2005). *Spatial Statistics*. John Wiley & Sons.
- RÜSCHENDORF, L. (1994). Wasserstein metric. In: M.Hazewinkel (Ed.), *Encyclopedia of Mathematics*, Kluwer, Dordrecht, NL.
- RUSSELL, J. C.; HANKS, E. M and HARAN, M. (2016). Dynamic models of animal movement with spatial point process interactions. *Journal of Agricultural, Biological, and Environmental Statistics*, **21(1)**, pp. 22–40.
- SCHAAP, WILLEM EGBERT (2007). *DTFE: the Delaunay Tessellation Field Estimator*. Ph.D. thesis, University of Groningen.
- SCHNEIDER, R. and WEIL, W. (2008). *Stochastic and Integral Geometry*. Probability and its Applications. Springer, Dordrecht.
- SCOTT, D. W. (1992). *Multivariate Density Estimation: Theory, Practice, and Visualization*. John Wiley & Sons.
- SILVERMAN, B. W. (1982). Kernel density estimation using the fast Fourier transform. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, **31(1)**, pp. 93–99.
- SILVERMAN, B. W. (1986). *Density Estimation for Statistics and Data Analysis*. Routledge.
- SOM, N. A.; MONESTIEZ, P.; VER HOEF, J. M.; ZIMMERMAN, D. L. and PETERSON, E. E. (2014). Spatial sampling on streams: principles for inference on aquatic networks. *Environmetrics*, **25(5)**, pp. 306–323.
- SUGIHARA, K.; SATOH, T. and OKABE, A. (2010). Simple and unbiased kernel function for network analysis. In: *ISCIT 2010 (International Symposium on Communication and Information Technologies)*, pp. 827–832. IEEE.

- SUMNER, M. D. (2016). *trip: Tools for the Analysis of Animal Track Data*. R package version 1.5.0 <https://cran.r-project.org/package=trip>.
- SUMNER, M. D.; WOTHERSPOON, S. J. and HINDELL, M. A. (2009). Bayesian estimation of animal movement from archival and satellite tags. *PLoS One*, **4(10)**, p. e7324.
- TIERNEY, L. (2001). Compiling R: A Preliminary Report. In: K.Hornik and F.Leisch (Eds.), *DSC 2001: Proceedings of the Second International Workshop on Distributed Statistical Computing*, . ISSN 1609-395X.
- TOOHEY, K. (2015). *SimilarityMeasures: Trajectory Similarity Measures*. R package version 1.4 <https://cran.r-project.org/package=SimilarityMeasures>.
- VAN LIESHOUT, M. N. M. (2000). *Markov Point Processes and their Applications*. World Scientific.
- VAN LIESHOUT, M. N. M. (2011). A J-function for inhomogeneous point processes. *Statistica Neerlandica*, **65(2)**, pp. 183–201.
- VAN LIESHOUT, M. N. M. (2012). On Estimation of the Intensity Function of a Point Process. *Methodology and Computing in Applied Probability*, **14(3)**, pp. 567–578.
- VAN LIESHOUT, M. N. M. (2017). Nearest-neighbour Markov point processes on graphs with Euclidean edges. *arXiv preprint arXiv:1710.07502*.
- VER HOEF, J. M. and PETERSON, E. (2010). A moving average approach for spatial statistical models of stream networks. *Journal of the American Statistical Association*, **105(489)**, pp. 6–18.
- VER HOEF, J. M.; PETERSON, E. and THEOBALD, D. (2006). Spatial statistical models that use flow and stream distance. *Environmental and Ecological statistics*, **13(4)**, pp. 449–464.
- WAND, M. P. and JONES, M. CH. (1994). *Kernel Smoothing*. Crc Press.
- WATSON, G. S. (1964). Smooth regression analysis. *Sankhyā: The Indian Journal of Statistics, Series A*, **26(4)**, pp. 359–372.

- XIE, ZH. and YAN, J. (2008). Kernel density estimation of traffic accidents in a network space. *Computers, Environment and Urban Systems*, **32(5)**, pp. 396–406.
- YAMADA, I. and THILL, J. (2004). Comparison of planar and network K-functions in traffic accident analysis. *Journal of Transport Geography*, **12(2)**, pp. 149–158.
- YANG, S. L. (2014). *BayesianAnimalTracker: Bayesian Melding of GPS and DR Path for Animal Tracking*. R package version 1.2 <https://cran.r-project.org/package=BayesianAnimalTracker>.
- YUAN, J.; ZHENG, Y.; XIE, X. and SUN, G. (2011). Driving with knowledge from the physical world. In: *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 316–324. ACM.
- YUAN, JING; ZHENG, YU; ZHANG, CHENGYANG; XIE, WENLEI; XIE, XING; SUN, GUANGZHONG and HUANG, YAN (2010). T-drive: driving directions based on taxi trajectories. In: *Proceedings of the 18th SIGSPATIAL International conference on advances in geographic information systems*, pp. 99–108. ACM.