SCUOLA DI DOTTORATO
UNIVERSITÀ DEGLI STUDI DI MILANO-BICOCCA

Department of
Statistics and Quantitative Methods

PhD program: **Statistics and Mathematical Finance** Cycle: **33**
Curriculum: **Statistics**

Statistical Models and Data Structures for Spatial Data on Road Networks

Surname: **Gilardi** Name: **Andrea**

Registration number: **762781**

Tutor: **Prof. Riccardo Borgoni**

Supervisor: **Prof. Riccardo Borgoni**

Co-Supervisor: **Prof. Jorge Mateu**

Coordinator: **Prof. Rosazza Gianin Emanuela**

ACADEMIC YEAR 2019/2020

University of Milan - Bicocca

Doctoral Thesis

Statistical Models and Data Structures for Spatial Data on Road Networks

Author:
Andrea Gilardi

Supervisor:
Prof. Riccardo Borgoni
Prof. Jorge Mateu

A thesis submitted in fulfillment of the requirements for the degree of Doctor of Philosophy

Department of Statistics and Quantitative Methods

February 28, 2021

Statistical Models and Data Structures for Spatial Data on Road Networks

Andrea Gilardi

Department of Statistics and Quantitative Methods
University of Milan - Bicocca
Via Bicocca degli Arcimboldi, 8, 20126 - Milano (IT)

ABSTRACT

In the last years, we observed a surge of interest in the statistical analysis of spatial data lying on or alongside networks. Car crashes, vehicle thefts, bicycle incidents, roadside kiosks, neuroanatomical features, and ambulance interventions are just a few of the most typical examples, whereas the edges of the network represent an abstraction of roads, rivers, railways, cargo-ship routes or nerve fibers.

This type of data is interesting for several reasons. First, the statistical analysis of the events presents several challenges related to the complex and non-homogeneous nature of the network, which creates unique methodological problems. Several authors discussed and illustrated the common pitfalls of re-adapting classical planar spatial models to network data. Second, the rapid development of open-source spatial databases (such as Open Street Map) provides the starting point for creating road networks at a wide range of spatial scales. The size and volume of the data raise complex computational problems, while common geometrical errors in the network's software representations create another source of complexity. Third, at the time of writing, the most important software routines and functions (mainly implemented in `R`) are still in the process of being re-written and re-adapted for the new spatial support.

This manuscript collects four articles presenting data structures and statistical models to analyse spatial data lying on road networks using point-pattern and network-lattice approaches.

The first paper reviews classes, vital pre-processing steps and software representations to manipulate road network data. In particular, it focuses on the `R` packages `stplanr` and `dodgr`, highlighting their main functionalities, such as shortest paths or centrality measures, using a range of datasets, from a roundabout to a complete network covering an urban city.

The second paper proposes the adoption of two indices for assessing the risk of car crashes on the street network of a metropolitan area via a dynamic zero-inflated Poisson model. The elementary statistical units are the road segments of the network. It employs a set of open-source spatial covariates representing the network's structural and demographic characteristics (such as population density, traffic lights or crossings) extracted from Open Street Map and 2011 Italian Census.

The third paper demonstrates a Bayesian hierarchical model for identifying road segments

of particular concern using a network-lattice approach. It is based on a case study of a major city (Leeds, UK), in which car crashes of different severities were recorded over several years. It includes spatially structured and unstructured random effects to capture the spatial nature of the events and the dependencies between the severity levels. It also recommends a novel procedure for estimating the MAUP (Modifiable Areal Unit Problem) for network-lattice data.
Finally, the fourth paper summarises a set of preliminary results related to the analysis of spatio-temporal point patterns lying on road networks using inhomogeneous Poisson processes. It focuses on the ambulance interventions that occurred in the municipality of Milan from 2015 to 2017, developing two distinct models, one for the spatial component and one for the temporal component. The spatial intensity function was estimated using a network readaptation of the classical non-parametric kernel estimator.
The first two appendices briefly review the essentials of INLA methodology and the supplementary materials related to the fourth chapter, while the third appendix introduces an `R` package, named `osmextract`, that was developed during the three years of my PhD and focuses on Open Street Map data.
The sixth chapter concludes the manuscript, summarising the main contributions and emphasising future research developments.

ABSTRACT (ITALIAN VERSION)

Negli ultimi anni si è sviluppato un interesse sempre crescente per l'analisi statistica di dati spaziali aventi supporto di network. Gli esempi più classici di questa tipologia di eventi sono, ad esempio, gli incidenti stradali, i furti di auto (o, più in generale, i crimini), e gli interventi delle ambulanze, mentre le linee (o *edge*) che compongono la network rappresentano tipicamente le strade, i fiumi, i binari della ferrovia, le rotte delle navi cargo oppure le terminazioni nervose.
L'analisi di questi fenomeni è interessante sotto diversi punti di vista. Innanzitutto, i modelli statistici presentano diverse problematiche legate al supporto spaziale. Per questo motivo, negli ultimi anni sono stati pubblicati diversi paper che mostrano le difficoltà principali legate alla natura stessa della network. Inoltre, il recente sviluppo di database spaziali open source (quali Open Street Map) ha permesso il download e la creazione di dataset che coprono le reti stradali e marittime di quasi tutto il mondo. L'enorme mole di dati e gli (inevitabili) errori geometrici presenti nei database di Open Street Map rappresentano due problematiche ulteriori. Infine, dato che al momento la maggior parte dei pacchetti `R` per l'analisi di dati su network sono ancora in fase di sviluppo, esistono anche diverse difficoltà computazionali e problemi nell'implementazione di metodologie nuove.
Questo lavoro di tesi riassume quattro articoli che presentano strutture dati e metodologie statistiche per l'analisi di dati spaziali aventi supporto di network, considerando sia un approccio di tipo *lattice* che un approccio di tipo *point-pattern*.

Il primo paper presenta una revisione dei pacchetti `R` che implementano classi e funzioni per l'analisi di network stradali, concentrandosi in particolare su `stplanr` e `dodgr`. Vengono introdotte le principali routines legate al calcolo di *shortest paths* e *centrality measures* utilizzando dataset via via più complessi.
Il secondo lavoro presenta un modello di Poisson Dinamico Zero Inflated per la stima di due indici di rischiosità relativi agli incidenti stradali avvenuti nel network di Milano dal 2015 al 2017. L'unità statistica elementare è rappresentata dal singolo segmento di strada, mentre la variabile risposta misura il numero di incidenti avvenuti in ognuno dei tre anni. Viene impiegato un insieme di covariate demografiche e strutturali (come, ad esempio, la densità di popolazione o la presenza di semafori e incroci) estratte da Open Street Map e dai dati del censimento italiano avvenuto nel 2011.
Il terzo paper introduce un modello Bayesiano gerarchico multivariato per la stima della rischiosità stradale tramite un approccio di tipo network-lattice. Ci si è concentrati sul network stradale della città di Leeds (UK) e su due diverse tipologie di incidenti. La componente spaziale è stata modellata tramite un errore casuale di tipo Multivariate CAR, mentre le correlazioni residue sono state catturate tramite un errore casuale non strutturato. Infine, si è sviluppata una metodologia per l'analisi di MAUP su dati di tipo network-lattice.
Per concludere, il quarto articolo presenta alcuni risultati preliminari relativi all'analisi spazio-temporale di point pattern su network tramite processi di Poisson non-omogenei. In particolare, si è analizzata la distribuzione degli interventi in emergenza delle ambulanze nel comune di Milano tra il 2015 ed il 2017, sviluppando un modello a fattori latenti per la componente temporale ed uno stimatore kernel non-parametrico per l'intensità spaziale, riadattato nel caso di dati su network.
La tesi si compone anche di tre appendici. Le prima riassume le caratteristiche di base della metodologia INLA, la seconda presenta i materiali addizionali legati al quarto capitolo, mentre la terza introduce un pacchetto `R` chiamato `osmextract` che può essere utilizzato per manipolare dati estratti da Open Street Map.
Il sesto capitolo chiude la tesi, riassumendo i risultati principali e introducendo alcuni sviluppi futuri.

Keywords: Bayesian Hierarchical Models, INLA, Network Lattice, Open Data, Point Pattern on Networks, Spatial and Spatio-temporal Statistics, Street Networks

Ringraziamenti

Il primo ringraziamento non può che andare al Prof. Riccardo Borgoni per il suo sostengo durante questi anni di dottorato. È stata una lunghissima epopea, partita dall'incubo dei primi esami e conclusa con il panico delle ultime settimane prima della consegna. Semplicemente grazie per la pazienza e la disponibilità che hai dimostrato e che dimostri tutt'ora. So benissimo di essere una persona estremamente pigra, testarda, e lenta a scrivere, e non so ancora come hai fatto a non arrabbiarti mai una volta durante questo periodo. Inoltre, non posso fare a meno che ringraziare il Prof. Jorge Mateu ed il Dott. Robin Lovelace per il supporto che mi avete dato durante questi anni, a partire dai periodi trascorsi a Castelló (ES) e Leeds (UK). Il visiting a Leeds è stata una delle mie prime esperienze da solo all'estero ed è stato semplicemente incredibile, soprattutto grazie alle persone che ho conosciuto ed i posti che ho visitato. Ricordo spesso alcuni degli eventi accaduti a Leeds, e non posso fare a meno che scoppiare a ridere per la gioia e la nostalgia. Il visiting a Castelló è stato chiaramente più complicato per i motivi che conosciamo tutti. Ciò nonostante, non vedo l'ora di poterci ritornare per vivere meglio la città ed il mare. Per concludere, ci tengo anche a ringraziare il Prof. Montes ed il Prof. Gómez Rubio per i loro preziosi suggerimenti che mi hanno permesso di migliorare e correggere diverse parti di questa tesi.

Ringrazio di cuore anche i miei genitori, mio fratello, e gli zii che mi hanno supportato (e sopportato) in questi anni (anche se con troppe ansie nell'ultimo periodo e mentre ero in spagna). Mi spiace non essere mai riuscito a spiegarvi di cosa mi occupo durante tutte le ore che passo in università, ma sono abbastanza sicuro di essere riuscito a trasmettervi la passione che ho per il mio lavoro.

Un doveroso ringraziamento va a tutti i colleghi con cui ho condiviso scrivanie, esami e, soprattutto, ansie e frustrazioni. Pur lavorando tutti in ambiti piuttosto eterogenei, credo di essere sempre riuscito ad andare d'accordo con voi, ridendo e scherzando più o meno su tutto. Ci tengo particolarmente a ringraziare Anna ed Andrea. Con voi ho condiviso pause caffè, lunghissime giornate in U7, e momenti a dir poco incredibili (la summer school a Catania, la SIS a Palermo, ed i tweet su eRum, solo per nominarne alcuni). Sono più che sicuro che senza di voi avrei abbandonato il dottorato pochi mesi dopo i primi esami.

Non posso non nominare anche tutti gli amici di Olginate, Villa, Calolzio e, più in generale, del lecchese. Devo ringraziarli per tutte le serate (ultimamente solo su Skype), i pranzi, le cene, le gite, e, più in generale, i momenti passati insieme (tutti conditi da numerosissimi insulti e minacce). So di avervi stressato molto ultimamente con la tesi ed il nuovo lavoro, e sono sicuro che continuerò a tormentarvi con sempre nuove ansie e lamentele.

Per concludere, voglio ringraziare anche tutti i colleghi che ho conosciuto durante la laurea

triennale e quella magistrale. In particolare, Valentina, Fabio, i *Dottori in Stat* e i *Markov Unchained* che, nonostante tutto, sopportano i miei silenzi, le mie follie, e le mie paranoie da molti anni. Mi avete aiutato a sorridere nei momenti più tristi e complicati del dottorato.
Vorrei chiudere questa sezione riportando alcune frasi prese dal monologo finale[1] di una delle mie serie TV preferite, *Scrubs*: *Una fine non è mai facile. Me la immagino così tanto nella mia testa che non potrà mai soddisfare le mie aspettative e finirò sempre per rimanere deluso. Non sono nemmeno sicuro del perché mi importi di come finirà tutto. Immagino che sia perché tutti crediamo che quello che facciamo sia molto importante, che le persone pendano dalle nostre labbra, che diano importanza a quello che pensiamo. La verità è che devi considerarti fortunato se anche solo di tanto in tanto fai sentire qualcuno, chiunque, un po' meglio.* Questo solo per dire che dopo aver lavorato alla tesi per tantissime ore e lunghissime notti, penso di aver capito che la cosa più importante sono le persone con cui ho condiviso questi tre anni.

Acknowledgements

Dear Jorge and Robin, I'm sorry but I decided to write the "emotional" parts of the thesis in Italian. Nevertheless, this is the translation of the first paragraph.

Dear Jorge and Robin, thank you very much for your support and assistance during my visiting periods in Castelló (ES) and Leeds (UK). I'm so happy that we keep working on the projects we started in your universities, chatting more or less every week.
I loved every minute that I spent in Leeds. I met several lovely people, outstanding researches, and I had great experiences (that I cannot thoroughly list here). I had a great time, and I still laugh when I think about the days that I spent with Julie, Andy, Robin, Katy, and the other researchers at ITS. I'm not a transport researcher, but I hope to return to Leeds sometimes during the next years.
For obvious reasons, the visiting period in Castelló was much more complicated. Nevertheless, I want to thank Prof. Jorge Mateu for his statistical and emotional support during those difficult times. I plan to go back to Castelló as soon as possible and live all the opportunities and events that I missed the last year.
Finally, I need to thank Prof. Montes and Prof. Gómez Rubio for their valuable suggestions that helped me correct several mistakes in this manuscript.

[1] La nona stagione facciamo finta che non sia mai esistita.

Contents

List of Figures

List of Tables

Chapter 1
Introduction

In the last years, we observed a surge of interest in the statistical analysis of spatial data lying on or alongside networks (Baddeley, Nair, et al., 2020; Okabe and Sugihara, 2012; Ziakopoulos and Yannis, 2020). Car crashes, vehicle thefts, bicycle incidents, roadside kiosks, neuroanatomical features, and ambulance interventions are just a few of the most typical examples, whereas the edges of the network represent an abstraction of roads, rivers, railways, cargo-ship routes or nerve fibers.

The most important characteristic of this type of events is undoubtedly the fact that they are constrained to lie on a specific spatial domain, which clearly cannot be ignored for proper model development. For example, H. Huang et al. (2016) compared zonal and network lattice models for road safety research. They conclude saying that the road segments approach should be preferred since it has *better overall fit and predictive performance, provides better insights about the micro factors that closely contribute to crash occurrence, and leads to more direct countermeasures.* Indeed, as detailed in Chapters 3 and 4, a network-lattice approach returns more detailed results than the zonal counterpart, proving to be more useful from a social and policy perspective. Analogously, Yamada and Thill (2004) and Y. Lu and X. Chen (2007) presented the risks of spurious analysis and false-positive detection associated with the application of point-pattern methods designed for a planar space to analyse spatial data on road networks.

Besides, a road network is not a homogeneous space, which implies that readaptations of classical techniques must be adjusted to take into account the new spatial support. The network-version of the kernel intensity estimator represents a relevant example. The first proposals can be tracked back to Borruso (2003, 2005, 2008), and they were introduced without any proper statistical or theoretical justification. Xie and Yan (2008) suggested a variation of the planar kernel intensity estimator (P. Diggle, 1985) in which the Euclidean metric is replaced by the shortest-path distance on the network. Unfortunately, as reported by Okabe and Sugihara (2012), their proposal turned out to be erroneous and extremely biased since the direct readaptation of the planar technique does not converse mass. Okabe, Satoh, and Sugihara (2009) and Sugihara, Satoh, and Okabe (2010) presented *equal-split continuous* and *equal-split discontinuous* estimators, which have desirable properties, but can be costly to compute for large networks. Finally, in the last years, we observed several proposals that combine rigorous theoretical and statistical justifications with effective implementations (McSwiggan, Baddeley, and Nair, 2017; M. M. Moradi, Rodríguez-Cortés, and Mateu, 2018). For example, the approach adopted in Chapter 5 is based on a recent paper by Rakshit, Davies, et al. (2019) that proposes a computationally efficient method that combines the classical planar kernel estimator with a convolution of the kernel on the network. Ang, Baddeley, and Nair (2012) represents another relevant example to illustrate fallacies and methodological errors linked with direct readaptations of planar techniques. More precisely, the authors introduced geometrically corrected K and pair correlation functions (Ripley, 1977) that do not depend on the network's topology, modifying the previous attempts.

The rising interest in road network data can almost certainly be linked with two factors:

1. the rapid development of high-quality street network objects, available to people worldwide thanks to open-access databases such as Open Street Map (OpenStreetMap contributors, 2017);
2. the open-access policy adopted by several private or national agencies (like the Italian Statistical Institute or United Kingdom's Department for Transport, see Chapters 3 and 4) to share more and more datasets related to road networks events such as car crashes or petty crimes.

The size and the volume of Open Street Map network data raise complex computational problems that require ad-hoc solutions. The analysis of network data using a lattice approach (see Chapters 3 and 4) benefits from innovative statistical techniques, such as INLA methodology, that permit the analysis of city-wide road networks much faster than classical MCMC methods (Håvard Rue, Martino, and Chopin, 2009; Håvard Rue, Riebler, et al., 2017). Moreover, considering that Open Street Map data can present geometrical errors that create invalid or erroneous network supports, such as missing junctions or fictitious links, several authors focused on developing spatial and geographical operations that manipulate the network, fixing these types of problems (Cooper and Chiaradia, 2020; Van der Meer et al., 2021).

This monograph collects four articles introducing data structures and statistical models to analyse spatial data lying on road networks using point-pattern and network-lattice approaches. In particular, the rest of the thesis is organised as follows.

Chapter 2 presents methods, pre-processing steps and software implementations for the spatial analysis of street networks. The first two sections briefly overview several `R` packages that define class systems for representing road networks, with a particular focus on two libraries that offer analytical capabilities difficult to implement using more general approaches: `stplanr` and `dodgr` (Lovelace and Ellison, 2018; Padgham, 2019). The former proposes the S4 class `sfNetwork` , while the latter defines the S3 class `dodgr_streetnet`. In both cases, the starting point is typically an `sf` object (E. J. Pebesma, 2018), which represents the spatial dimension of the network. `stplanr` wraps the `igraph` package (Kolaczyk and Csárdi, 2014) to handle the graph component, while `dodgr` defines its own functions. We demonstrate that street networks have particular properties that make them unsuitable to be analysed using traditional graph-related software. Moreover, the pros and cons of

the two implementations are compared using more and more complex street networks extracted from Open Street Map. Finally, we report several practical examples that display algorithms and functions to perform common street network analysis tasks, such as shortest and fastest paths or centrality measures, which are repeatedly applied in the other articles.
Chapter 3 shows how a range of information collected from open data sources concerning socio-demographic (e.g. population density) and structural (e.g. highways types, traffic lights, and pedestrian crossings) covariates can be harmonised to develop a road safety model. More precisely, we analysed the car crashes that occurred in the road network of Milan (IT) from 2015 to 2017 considering a network-lattice approach. The spatial support was downloaded from Open Street Map, and its road segments represent the elementary statistical units. We developed a Dynamic Zero-Inflated Poisson (ZIP) regression model (Lambert, 1992) and proposed the adoption of two indices to assess the risk of car crashes on the street network of the city. The first index, derived from the counting component of the ZIP model, measures the road risk. The other, derived from the zero component, represents an estimate of the likelihood of each segment not to be exposed to crashes. The socio-demographic variables, that were measured for each census tract of Milan during the 2011 Census, were summarised using a geographically-weighted principal component analysis (Fotheringham, Brunsdon, and Charlton, 2003). Moreover, since the two sources of information are spatially misaligned, they were merged using an overlay operation. We found that the most relevant determinant of road risk proneness is crash history and that the structural characteristics are much more relevant than the demographic variables. Finally, we show how this information can be visualised to produce maps of crash risk and forecast the values of the indices in the future.

Chapter 4 proposes a Multivariate Bayesian Hierarchical model with spatially structured and unstructured random effects to identify street sections with anomalously high car crashes rates considering a network-lattice approach. It is motivated by a case study of a major city, Leeds (UK), in which car crashes of different severities were recorded over an eight-year period (2011-2018). The spatial support was downloaded from Ordnance Survey, a web provider of street network data for the United Kingdom, and it was manipulated and simplified using several procedures (described in Chapter 2). The spatially structured random effects were modelled using a Multivariate Proper Conditional AutoRegressive (CAR) prior (Besag, 1974; Gelfand and Vounatsou, 2003; Mardia, 1988), with a separate coefficient for each severity level and a first-order binary adjacency matrix calculated on the segments of the network. The unstructured random effects were modelled as multivariate Gaussian errors. An offset component, proportional to an estimate of commuting flows that pass through each segment of the network was included in the Bayesian model. The results underline a strong correlation between the two severity levels in the unstructured error and an even stronger dependence in the spatial component. Moreover, we produced a series of maps that highlight several roads in the north-east, north-west and south-east of the city centre as being more prone both to severe and slight crashes. The Modifiable Areal Unit Problem (MAUP) (Openshaw, 1981) was investigated, proposing a novel procedure to test the presence of MAUP for count models developed on a network-lattice. Finally, an extensive sensitivity analysis has been performed to assess the robustness of models to a range of assumptions, such as different hyperpriors or adjacency matrices.

Chapter 5 introduces a spatio-temporal point process for analysing the distribution of ambulance interventions that occurred in the road network of Milan from 2015 to 2017. It represents the first attempt to estimate a non-separable first-order spatio-temporal intensity function for points on networks. We considered all events that were managed by the regional Emergency Medical System (EMS) and required the dispatch of one or more ambulances. The spatial support was downloaded from Open Street Map, and it was simplified using several methods (analogous to the pre-processing steps described in Chapters 2 and 4). The preliminary analysis revealed that the temporal evolution of the interventions presents several seasonalities due to hourly, daily, and weekly patterns, while the spatial representations showed that the points are located along the segments of a network and tend to be clustered near popular and busy areas. We noticed the presence of space-time interactions in the hourly spatial distribution of ambulance dispatches. Hence, we treated the events as a point pattern on a linear network. We divided the EMS data into discrete intervals of one hour, and we assumed that, independently for each time unit, the interventions could be modelled as an Inhomogeneous Poisson Process with a non-separable intensity function, which is assumed to be decomposable into two terms. The first term captures the temporal dynamics, while the second one models the spatial effects and the space-time interactions. A dynamic latent factor model with deterministic covariates was defined for the temporal component (David S Matteson et al., 2011), while the spatial dimension was estimated using a non-parametric Gaussian network kernel function (Rakshit, Baddeley, and Nair, 2019) combined with a set of weights to capture the spatio-temporal interaction (Zhou and David S. Matteson, 2015). The approach was exemplified by estimating the spatio-temporal intensity function for two future time units, and the results show the effectiveness of our proposal, displaying the spatial, temporal and spatio-temporal dimensions.

Chapter 6 concludes the manuscript, summarising the most important findings and the key ideas for future developments.

Three appendices are also included. The first one reports a short introduction to INLA methodology, briefly reviewing the basic ideas and the key components. The second appendix contains the supplementary materials for Chapter 4, which include the pseudo-code related to the algorithm used to test Bayesian Hierarchical model's predictive accuracy and several tables summarising the sensitivity analysis. Finally, the third appendix introduces an `R` package that was developed during the three years of my PhD, named `osmextract`. It defines functions and methods used for matching, downloading, converting and reading Open Street Map data obtained by external providers such as Geofabrik. It wraps several GDAL functions (GDAL/OGR contributors, 2020), creating an efficient workflow for manipulating large road networks covering a city, a metropolitan area, a country or even a continent.

Chapter 2

Data structures and methods for reproducible street network analysis: overview and implementations in R

> *Maybe I'll become a theoretician. Nobody expects you to maintain a theorem.*
>
> Doug Bates, *lme4* author, 2013

Based on: *Gilardi, A., Lovelace, R., Padgham, M. Data structures and methods for reproducible street network analysis: overview and implementations in R. osf preprint.* URL: https://doi.org/10.31219/osf.io/78yub

2.1 Introduction

Spatial networks, such as power grids, railways or rivers, are entities that can simultaneously be represented as spatial and graph objects (Barthélemy, 2011). From a spatial perspective, they consist of features embedded in a (two or three-dimensional) spatial domain while, from a graph perspective, they consist of a set of vertices connected by a set of edges. Both edges and vertices are associated with spatial geometries, typically points, lines, or polygons. *Street networks* represent a particular type of spatial network, with notable distinctive traits. Their abstract representation can be created in several ways (see, e.g., S. Marshall et al. (2018)), but, in this Chapter, we will focus on the most common approach: each road is associated to the edges of a graph, while the vertices[1] correspond to the intersections, typically at road junctions, although potentially also in between.

The first examples of road network analysis can be traced back to Euler's solution to the Seven Bridges of Königsberg problem (Euler, 1741) and John Snow's map of Cholera in London (Snow, 1855), two seminal studies representing the foundation of *graph theory*

[1] The examples that we present in the following Sections are based on *Open Street Map* (OSM) data. The basic structure of OSM data is composed of three *elements* named *nodes*, *ways* and *relations* (OpenStreetMap contributors, 2017). For this reason, when we use the term *node* we are referring to the OSM elements, while the term *vertices* is used for the generic graph element. See also Appendix C for more details.

and *epidemiology*, respectively. Starting from the 1960s, several authors investigated the characteristics of spatial networks using a more systematic approach (Chorley and Petercoed Haggett, 1967; Peter Haggett and Chorley, 1969), while, more recently, other authors focused on the comparisons and the analysis of structural characteristics of urban street networks (Cardillo et al., 2006; Crucitti, Latora, and Porta, 2006; Jiang, 2007; Lämmer, Gehlsen, and Helbing, 2006).

This chapter demonstrates the particular properties of street networks and explains the problems that make them unsuitable to be analysed using traditional network analysis software. We will present the main functionalities, the classes and the methods of two `R` packages that focus on street network analysis: `stplanr` and `dodgr` (Lovelace and Ellison, 2018; Padgham, 2019). The techniques introduced here are extensively applied in the next Chapters.

Rather than focusing on specialised GIS platforms, such as `QGIS` (QGIS Development Team (2020), 2020) or `GRASS` (GRASS Development Team, 2017), we decided to present data-structures and pre-processing operations implemented in `R`, an increasingly popular programming language for geographical data and network analysis (R Core Team, 2020). Although other languages, including `Python` (Van Rossum and Drake, 2009), `C++`, and `Julia` (Zappa Nardelli et al., 2018), have emerging projects for spatial network analysis, such as `osmnx` (Boeing, 2017), `networkx` (Hagberg, Schult, and Swart, 2008), and `networkit` (Staudt, Sazonovs, and Meyerhenke, 2016), the choice of `R` was based on the authors' experience and the wide range of statistical methods implemented in the language. Moreover, even if the procedures shown in this chapter are specific to `R`, the concepts and the approaches demonstrated here can be implemented using any other software. `stplanr` and `dodgr` may enable reproducible research, similar to the studies mentioned before, and within a popular open-source command-line drive computational environment.

As described at the outset, the two packages offer functions and analytic capabilities which are difficult to implement using more general software, including most equivalent packages in `python`. `stplanr` and `dodgr`, along with their respective class systems, are introduced in Sections 2.2 and 2.3, in the context of `R`'s evolving capabilities for handling spatial networks. The particularities of street network data are outlined in Sections 2.4 and 2.5, which cover Open Street Map data and illustrate concrete examples of street network entities, from roundabouts to city-wide networks. Section 2.6 demonstrates how street networks can be analysed using the two `R` packages, and the final section discusses the strengths and limitations of each approach, with a view to informing future development and research efforts.

2.2 `R` packages and classes for spatial networks

Spatial networks can be represented using various approaches in `R` and, in this section, we try to present the most important ones. In theory, any package that is capable of representing

Listing 2.1: R code used to generate Figure 2.1. The five matrices store several pairs of coordinates.

```r
street_net_matrix_list <- list(
  matrix(c(0, 0, 1, 0, 2, 0, 3, 0, 4, 0, 5, 0), ncol = 2, byrow = TRUE),
  matrix(c(5, 0, 5, -1, 5, -2, 5, -3, 5, -4, 5, -5), ncol = 2, byrow = TRUE),
  matrix(c(5, 0, 5, 1, 5, 2, 5, 3, 5, 4, 5, 5), ncol = 2, byrow = TRUE),
  matrix(c(5, 5, 6, 5, 7, 5, 8, 5, 9, 5, 10, 5), ncol = 2, byrow = TRUE),
  matrix(c(5, 0, 6, 0, 7, 0, 8, 0, 9, 0, 10, 0), ncol = 2, byrow = TRUE)
)
street_net_df <- as.data.frame(do.call(rbind, street_net_matrix_list))
plot(street_net_df , xlab = "",ylab = "",pch = as.character(rep(1:5, each = 6)))
```

spatial coordinates and graphs can also represent street networks, and we do not attempt to cover all options. Instead, this section focuses on packages that can represent the spatial and graph components of street networks *simultaneously*. Before describing some of the key approaches, however, it is worth considering that base-R already supports `lists` and `matrices`, two of the essential data structures needed for representing street networks, as demonstrated in Listing 2.1 (which generates Figure 2.1, a crude representation of the same street network depicted in Figures 2.2 and 2.3).

The object `street_net_matrix_list` is a `list` composed of five `matrices` representing, schematically and straightforwardly, the streets of a minimal urban network. Each street is represented using a sequence of points, and each pair of coordinates define one point.

The *list-of-matrices* representation of street networks, demonstrated with stylised data in the code chunk above and converted into a data frame for plotting, is of limited use. Such base-R representations are impractical because they lack a formal class system for performing commonly needed operations such as plotting, subsetting, network analysis and shortest-path calculations. One could build additional components on the structure represented in `street_net_matrix_list`. However, for most tasks, it is likely that using pre-existing representations — encoded in class definitions of several R packages that support street networks — will be more effective. A selection of such packages, in ascending order of their first release on CRAN, is outlined below.

spatstat: First released in 2002, it was developed for analysing spatial point patterns. It was not initially designed with street networks in mind but, because point patterns on a linear network represent a more and more stimulating application with its problems and solutions (Baddeley, Nair, et al., 2020), several methods for working with linear networks have been developed in the package since `spatstat` version `1.22-0`, released in 2011. The authors defined the class `linnet` for representing *a connected network of line segments, such as a road network*, and several ad-hoc functions for Kernel Density Estimation (KDE) and other statistical techniques (Baddeley, Rubak, and Turner, 2015). This work has recently been extended in the package `spatstat.Knet`, which

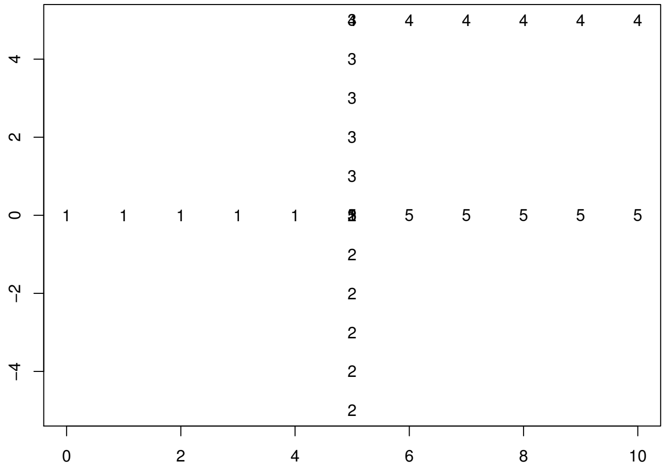


Figure 2.1: Plot of a simple street network stored as a list of matrices in base-`R`. It was generated using the code documented in Listing 2.1. Each matrix store several pairs of coordinates, and each number from 1 to 5 represents a distinct segment. There is an overlap of labels on junction points

improves the computational efficiency of statistical methods, such as the K function, on route networks (Rakshit, Baddeley, and Nair, 2019). The main limitation of this approach is its flexibility since `spatstat` was developed to perform spatial statistical analysis, and it offers limited support for routing and other street network operations outside of the `spatstat` ecosystem.

shp2graph: Before the release of the `R` package `sf` in 2016, the `Spatial` class system was the most prominent approach to handling spatial data in `R`, with the package `sp` that defined it, and imported by 73 other packages by 2013 (R. S. Bivand, E. Pebesma, and Gómez-Rubio, 2013). Building on the `Spatial` class system, `shp2graph`, first published in 2014, provides functions for converting `SpatialLinesDataFrame` objects into a list representing the components of a spatial network including nodes, edges, weights and other edge attributes. The package enables a variety of common street network operations, including shortest-path calculations and graph connectivity (B. Lu et al., 2018). The main disadvantage of this package is its reliance on `sp`, which has become largely superseded by `sf`. Moreover, it has not been updated for more than two years.

stplanr: This is a package designed to support evidence-based transport planning, with a focus on geographic *desire lines* and route data. The package also provides functions for working with street network data, including `overline2()`, which converts overlapping lines into a non-overlapping network of lines, and `SpatialLinesNetwork()`, which creates an S4 spatial network object comprising spatial (sp or sf) and graph (igraph) sub-objects (Lovelace and Ellison, 2018).

dodgr: This R package focuses on routing and distances, with a primary focus on directed graphs, and many functions dedicated to analyses of street networks. It performs efficient calculation of many-to-many pairwise distances on dual-weighted directed graphs, aggregation of flows throughout networks, and highly realistic routing through street networks (including time-based routing considering incline, turn-angles and surface quality). It defines classes and functions to represent and manipulate street networks (Padgham, 2019).

A couple of packages that have not been published on CRAN are also worthy of mention. spnetwork represents an alternative approach to convert Spatial objects into igraph objects. sfnetworks is a recently developed package that uses tidygraph as the basis for the graph manipulations. The version 0.1 is described in detail in a blog post on the subject hosted at r-spatial.org, while the current version (0.4 at the time of writing) can be explored starting from the package's website.
The remainder of this chapter focuses on street network representations in the packages stplanr and dodgr, which provide distinct data structures for the representation and analysis of street networks.

2.3 sfNetwork and dodgr_streetnet objects

As outlined in the Introduction, a defining feature of street networks is their duality: they are simultaneously spatial and graph objects. As spatial objects they are embedded in (typically two dimensional) space; as graphs, their vertices and edges correspond to geographical elements, such as roads or junctions. This section expands on this broad definition and describes the specific data structures that enable street networks to be represented in stplanr and dodgr.
The stplanr representation of a street network is typically created starting from an sf object, which constitutes the spatial dimension of the network (E. J. Pebesma, 2018). More precisely, the sf objects extend the data.frame class, defining a list-column named sfc (acronym for simple feature column) that store the spatial information. The elements of sfc are called sfg (acronym for simple feature geomery) and are usually characterised using an attribute that describes their type. The most common types are called POINT (e.g. zero-dimensional objects representing a geometrical point), LINESTRING (e.g. an ordered list of points creating straight, non-self-intersecting segments), and POLYGON (e.g. a sequence of points that form a closed, non-self-intersecting polygon with positive area). We refer to (OGC) Open Geospatial Consortium Inc (2011) and Lovelace, Nowosad, and Muenchow (2019) for more details. The printing method for sf objects is displayed in the top-right

part of Figure 2.2. On the other hand, the graph structure is inferred using an algorithm which is detailed below, and it is stored as an igraph object (Csardi, Nepusz, et al., 2006). The SpatialLinesNetwork() function combines the two worlds defining an S4 object with class sfNetwork. More precisely, the input given to SpatialLinesNetwork() is an sf object[2] with LINESTRING geometry ((OGC) Open Geospatial Consortium Inc, 2011), while the output is an S4 object having four slots named:

sl: the sf object that was passed as input with an additional column, named length, which measures the geographical length of each LINESTRING geometry.

g: an igraph object. The slots sl and g represent, respectively, the spatial and the graph dimensions.

nb: a list that summarises the connectivity of the graph;

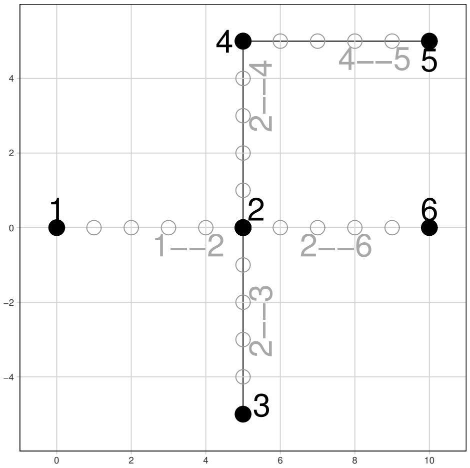
weightfield: a character vector that identifies the weighting profile. The default weights are the lengths of each road segment, but they can be modified using the function weightfield().

The other two parameters of SpatialLinesNetwork() function, i.e. uselonglat and tolerance, are used to 1) control the Coordinate Reference System (CRS) of the input object and 2) set a numerical value indicating a tolerance threshold to be used when creating the graph structure.

The igraph component of an sfNetwork object is created using the following algorithm (which is also illustrated in Figures 2.2 and 2.4a): the edges of the network are the LINESTRING geometries that were passed as input, while the vertices of the graph are the first and last points of each geometry (removing the duplicates with identical coordinates if necessary). The connectivity of the graph is determined as follows: two vertices are connected if they belong to the same LINESTRING geometry and, by the same reasoning, two edges are connected if they share one boundary point. Moreover, by construction, the igraph object has several attributes, named x, y, n and weight, that store the spatial coordinates (i.e. x, y), the connectivity of each vertex, and the weight associated to each edge (which is equal to the length of the corresponding spatial line). This algorithm, despite being natural and intuitive, has a few pitfalls that are typical of street network data. In this Chapter we present four examples that exhibit how and why it can fail.

On the other side, dodgr adopts a different philosophy for representing street networks since it merges the spatial and the graph dimensions, defining a unique object of class dodgr_streetnet as an extension of regular data.frame. Therefore, the authors of the package defined several functions and methods for working with dodgr_streetnet objects, while stplanr package adopts igraph as a backend for managing its graph structure. They also coded several ways to translate dodgr_streetnet objects into and from other formats, like sf or igraph, which enhances its interoperability with other packages.

[2] Note that SpatialLinesNetwork() can also take SpatialLinesDataFrame objects from the sp package. In this manuscript, we focus on the sf representations because the sp method will no longer be updated.



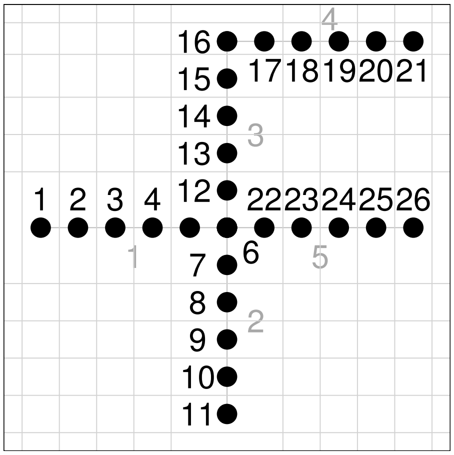
Spatial Component

```
Simple feature collection with 5 features
geometry type: LINESTRING
dimension: XY
1 LINESTRING (0 0, 1 0, ...)
2 LINESTRING (5 0, 5 −1, ...)
3 LINESTRING (5 0, 5 1, ...)
4 LINESTRING (5 5, 6 5, ...)
5 LINESTRING (5 0, 6 0, ...)
```

Graph Component

```
6 vertices
[1] 1 2 3 4 5 6
5 edges
[1] 1−−2 2−−3 2−−4 4−−5 2−−6
```

Figure 2.2: Graphical example of the data structure used by `stplanr` for representing street networks in the `sfNetwork` class. *Left*: Map of a street network. The black dots represent the starting and ending points of each `LINESTRING` geometry. The grey dots represent the internal points, which are ignored (but they will be important for the `dodgr` representation). *Right*: Summary of the spatial and graph dimensions of the street network.



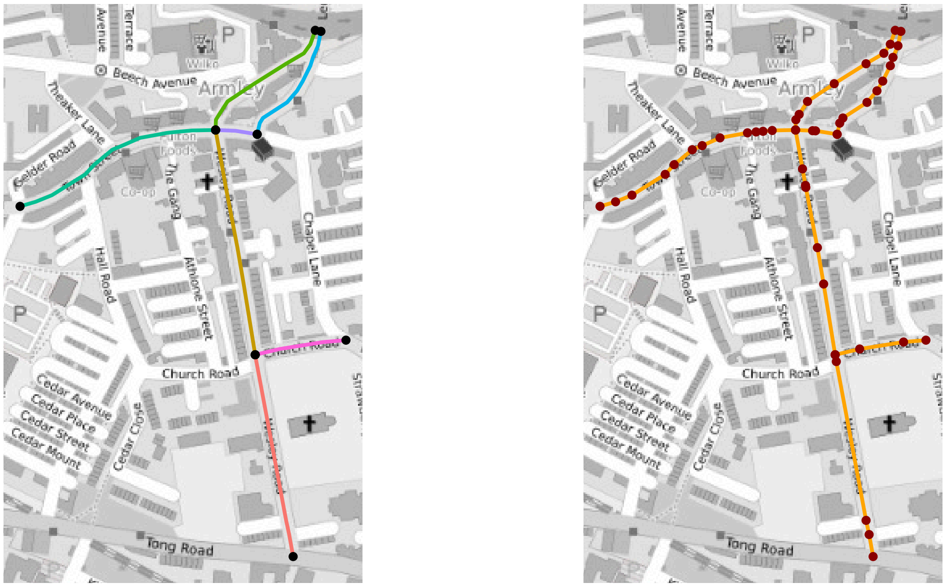
geom_num	edge_id	from_id	to_id	...
1	1	1	2	...
1	2	2	1	...
1	3	2	3	...
1	4	3	2	...
....	....	....	....	...
5	47	24	25	...
5	48	25	24	...
5	49	25	26	...
5	50	26	25	...

Figure 2.3: Graphical example of the algorithm used by `dodgr` for the creation of a `dodgr_streetnet` object. *Left*: Geographical map of the input data. The black dots are the `POINTS` (or *nodes* in OSM jargon) composing each `LINESTRING`. *Right*: Conversion into a `dodgr_streetnet` object. Each row of the new data frame is an edge of the network and it is linked with a pair of vertices. We created an undirected graph, so each edge is repeated two times. See Section 2.5 for more details.

The main function for creating a `dodgr_streetnet` object is `weight_streetnet()`, and the only mandatory input parameter is an `sf` data structure having a `LINESTRING` geometry, with coordinates expressed using the WGS84 reference ellipsoid (EPSG code 4326). Data expressed using other CRS must first be transformed to `EPSG:4326` (using, for example, the `sf` function `st_crs()`) before submitting to `weight_streetnet`. One of the primary advantages of `dodgr` is its support for *dual-weighted directed graphs*, which are typical of street networks and also necessary to generate realistic routes reflecting mode-specific preferences: pedestrians prefer quiet walkways removed from busy roads, cyclists prefer dedicated bicycle infrastructure, and car drivers prefer fast routes along motorways and large roads. Each edge in a dual-weighted graph has two distances, one representing the *true* geodesic distance, and the other representing a weighting preference (or profile) such that, for example, the weighted length for a pedestrian along an edge of a multi-lane motorway would be considerably longer than the actual distance. The `wt_profile` parameter can be used to select the preferred mode of transport when building a new street network with `weight_streetnet()`. The default value is `bicycle`, with other possibilities including `foot` and `motorcar`, as described in the help page of `weighting_profiles()`, which also includes details on how to implement custom weighting profiles. The dual weights are also determined by a character label that defines the characteristics of each segment of the network, such as *motorway*, *primary*, *residential* or *pedestrian*[3]. All road segments are mapped to a set of coefficients according to the chosen mode of transport, and these coefficients determine the dual-weights that define all `dodgr` functions. Their usefulness is exemplified in Section 2.6.

One further important advantage of `dodgr` package is its ability to *contract* a road network down to only those edges connecting street junctions through the `dodgr_contract_graph()` function. Road networks are commonly represented by points which are effectively arbitrarily located such that, for example, a curved way between two junctions might be represented by ten intermediate points, while a straight way might not have any intermediate points. These intermediate points are arguably representational artefacts, rather than intrinsic components of the network geometry (Karduni, Kermanshah, and Derrible, 2016; Open Street Map, 2020). It is particularly important for analyses of networks (such as shortest paths or network centrality, described below) to removes these artefacts throughout contracting a network down to only those edges directly connecting junctions, which is precisely what the `dodgr_contract_graph()` function does. The computation efficiency of routing on street networks increases non-linearly with N, the numbers of vertices, with efficiencies commonly scaling as $O(N^2)$). Graph contraction is the single most important step necessary for efficient routing. The contracted version of the network considered in the final examples of this Chapter has over 3 times fewer vertices, with typical values generally ranging between 5 and 10. Moreover, graph contraction and `dodgr_contract_graph()` represent the starting

[3] We refer to Appendix C for a more detailed description of the highway types and the corresponding Open Street Map *tags*.



(a) `stplanr` approach.

(b) `dodgr` approach.

Figure 2.4: *Left*: `stplanr`-representation of a network. The black dots represent the nodes and the colored lines the edges. *Right*: `dodgr`-representation of a network. The red dots represent the nodes and the orange lines the underlying street lines. The nodes and the edges correspond to several different street segments located in the Armley District of Leeds (UK).

point for the Modifiable Area Unit Problem (MAUP) analysis proposed in Chapter 4. We refer to the help page of `dodgr_contract_graph()` for more details.
Although the `stplanr` and `dodgr` packages may start from the same spatial objects, they diverge in the construction of street networks. More precisely, `dodgr` divides each `LINESTRING` geometry into its minimal components (also called `Line Segments`, as documented in (OGC) Open Geospatial Consortium Inc (2011)), creating a vertex for each node and an edge for every subsequent pair of points belonging to the same `LINESTRING` geometry.
The output of `weight_streetnet()` function is a `dodgr_streetnet` dataframe of edges with several columns including:

1. a unique ID for each `LINESTRING` geometry, called `geom_num`;
2. a unique ID for each edge, called `edge_id`, and for its corresponding vertices, called `from_id` and `to_id`. Duplicated vertices always share the same ID. If the input `sf` data is created using the `osmdata R` package (see below), then `dodgr` checks the uniqueness of the vertices by comparing their Open Street Map ID(s) instead of the spatial coordinates. This is a peculiar characteristic of street networks, and we will

see in Section 2.5 why this feature may be relevant;

3. four columns named `from_lon`, `from_lat`, `to_lon` and `to_lat` that represent the coordinates of each vertex using the WGS84 reference ellipsoid;
4. the spatial length, in meters, of each edge;
5. the weighted length of each edge, estimated using the weighting profile.

We include an exemplification of `dodgr` algorithm in Figure 2.3, where, for simplicity, we report only the first columns, and we omit the dual-weights, typical of `dodgr` objects. Another example is reported in Figure 2.4b. Some of the missing columns are reported in Table 2.1, while the dual-weights will be extensively covered in Section 2.6.

2.4 Open Street Map

Both packages can work with any type of street data (as long as they are coded in the right way) but, in this manuscript, we are going to focus on Open Street Map (OSM) data, briefly mentioned also in Chapter 1 (OpenStreetMap contributors, 2017). Open Street Map is the largest, openly available source of spatial network data, and it provides a continuously evolving and future proof basis for research (Barrington-Leigh and Millard-Ball, 2017). It's rapid evolution certainly helped the development of street network analysis (Anderson, Sarkar, and Palen, 2019).

There are several `R` packages for downloading data from Open Street Map servers but, at the moment, the most important one is probably `osmdata` (Padgham et al., 2017), which is also perfectly integrated with `dodgr`. The authors of `dodgr` created the `dodgr_streetnet()` function (which is a wrapper around several other routines defined in `osmdata`) to download and format OSM road data for a given location, which can be expressed either as a string (and processed by *Nominatim geocoding servers*) or a numeric matrix of coordinates that define a bounding box. Then, the output of `dodgr_streetnet()` can be passed to `weight_streetnet()` to create a `dodgr_streetnet` object that represents the street network of a particular area.

The main benefits of this integration are the following:

1. The `osmdata` functions retain the Open Street Map ID(s) of each node hidden within the actual `LINESTRING` structure[4], which enables vertex identification through ID(s) instead of coordinates. We present an example of why this distinction is important in the Section 2.5.2.
2. The column used for estimating the dual weights is automatically detected and formatted according to a pre-specified set of weights, as explained in the help page of `weighting_profiles()`.

[4] The `osmdata::unname_osmdata_sf()` function can be used to remove these ID(s), which can be problematic for some plotting routines. We refer to the following github issue for more details: https://github.com/ropensci/osmdata/issues/188.

Table 2.1: First five rows and first eight columns of `dodgr` representation of the spatial network depicted in Figure 2.4b. The missing columns represent other characteristics of the edges like their highway-type or the weighted lengths.

	geom_num	edge_id	from_id	from_lon	from_lat	to_id	to_lon	to_lat
1	1	1	0	-1.5886	53.7973	1	-1.5885	53.7969
2	1	2	1	-1.5885	53.7969	0	-1.5886	53.7973
3	1	3	1	-1.5885	53.7969	2	-1.5885	53.7968
4	1	4	2	-1.5885	53.7968	1	-1.5885	53.7969
5	1	5	2	-1.5885	53.7968	3	-1.5885	53.7967

3. There are several ancillary functions in `osmdata` that can be used to modify the street network objects in `dodgr`, including `trim_osmdata()`, which can trim a network within a bounding polygon rather than a simple rectangle, and the `osm_poly2line()` function which convert all `POLYGON` objects into `LINESTRING` objects suitable for routing.

The previous list describes three problems, i.e. node identification, dual weights and spatial filters, that are typical of street networks and require ad-hoc solutions that are difficult to implement using a generic graph software. Finally, we note that the `R` package `osmextract` represents an alternative approach for downloading and reading OSM data. It is extensively documented in Appendix C.

2.5 Street network data types

Street networks are complex phenomena with a range of sizes, shapes and interrelations, ranging from a simple network of mud paths in a remote village to a complex city containing dozens of separate but touching roads for walking, cycling, and motorised modes. In this context, it is important that software for representing and analysing street networks can handle a range of data types. To that end, this section introduces four scenarios that highlight common issues encountered when working with spatial networks representing transport systems. The first three examples are a *roundabout* (which is represented as a circular geometry), an *overpass* (in which intersecting streets are not connected due to a vertical grade of separation) and a *oneway road* (in which vehicles are prohibited from travelling in one direction). They are simple, but they highlight tricky problems that should be taken into account when working with street network data. The final example is a citywide graph containing multiple instances of each of the previous entities, approximating objects that are encountered in applied research and showing how `stplanr` and `dodgr` can work on real-world datasets.

Before proceeding with these examples, it is worth taking a step back to consider the minimal requirements that input datasets must meet before they can be classified as street networks. These requirements apply to data from any source, but, as we said in the previous section, we focus on Open Street Map, and we present a small part of its mature and open set of guidelines. The first requirement is that two or more intersecting streets, or *ways* in OSM nomenclature, must share at least one point, or *node* in OSM nomenclature, otherwise the corresponding edges can not be considered to intersect. A more subtle assumption is that streets that are not truly intersecting due to a vertical degree of separation, such as overpasses and underpasses, should not share any point with identical ID (even though those points may share identical geographical coordinates). These minimal requirements are documented in the *Editing Standards and Conventions page on the Open Street Map wiki*.

These are not strict assumptions, and we never found a situation where they are not easily met. There are also several tools for checking these hypotheses, such as `OSM Inspector` (`https://wiki.openstreetmap.org/wiki/OSM_Inspector`), and fixing the problems, such as `v.clean` tool in `GRASS`[5] (GRASS Development Team, 2017). We do not add more details on spatial networks preprocessing, and we refer to Cooper and Chiaradia (2020) and Section 8 of sDNA open-source manual, hosted at sDNA webpage. Nevertheless, the importance of *cleaning* the data before building the road network should never be overlooked.

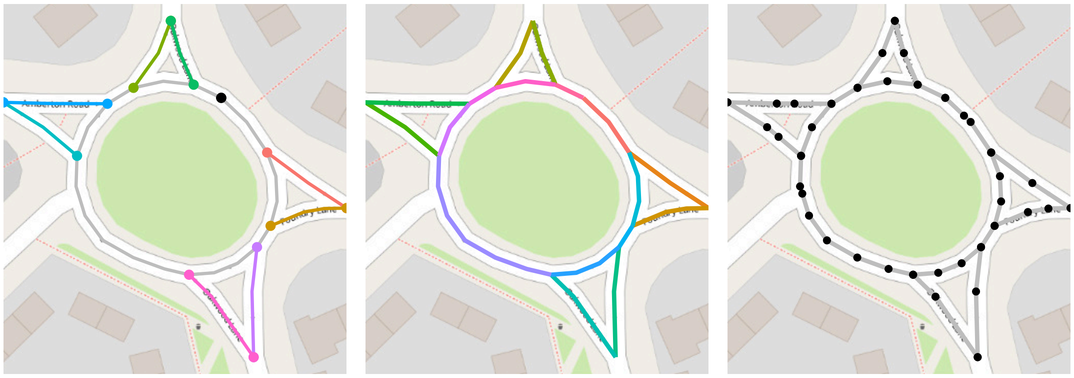
2.5.1 Roundabouts

Roundabouts are saved by Open Street Map as circular geometries composed by one or more connected `LINESTRING`. In the previous sections, we introduced the algorithm used by `stplanr` for inferring the graph structure of an `sf` data and we said that the connectivity of the graph is determined according to the presence or absence of shared boundary points in the `LINESTRING` geometries. This algorithm implies that the `stplanr`-representation of a roundabout may be unroutable since circular geometries have only one boundary point, which is not always shared with another `LINESTRING`. We present an example of this problem in Figure 2.5a. The grey line represents the roundabout, and the black dot is its boundary point. The coloured points are the boundaries of the other streets, which, according to `stplanr` algorithm, are not connected to the roundabout.

This problem can be solved by splitting the circular `LINESTRING`, and this procedure is implemented in the `rnet_breakup_vertices()` function. We refer to its help page for a more detailed description of the algorithm, which is similar to the procedures illustrated in Karduni, Kermanshah, and Derrible (2016). The result is a routable street network, illustrated in Figure 2.5b.

The `dodgr` approach immediately solves this problem by decomposing each `LINESTRING` into its minimal segments, while OSM itself ensures that each junction is represented by a

[5] The `GRASS` tools can be accessed from `R` via *bridges* (Lovelace, Nowosad, and Muenchow, 2019) such as `qgisprocess` (Dunnington, 2020b) and `rgrass7` (R. Bivand, 2019).



(a) Circular `LINESTRING`. (b) Routable `LINESTRING`. (c) `dodgr` approach.

Figure 2.5: *Left*: The network is unroutable using `stplanr` since the black dot, which is the unique boundary point of the roundabout, is not shared with any other street. *Center*: The network is routable since every boundary point is also a node of the network. *Right*: The `dodgr` approach bypasses any problem dividing each `LINESTRING` into its underlying segments.

shared vertex. This is clear looking at Figure 2.5c.

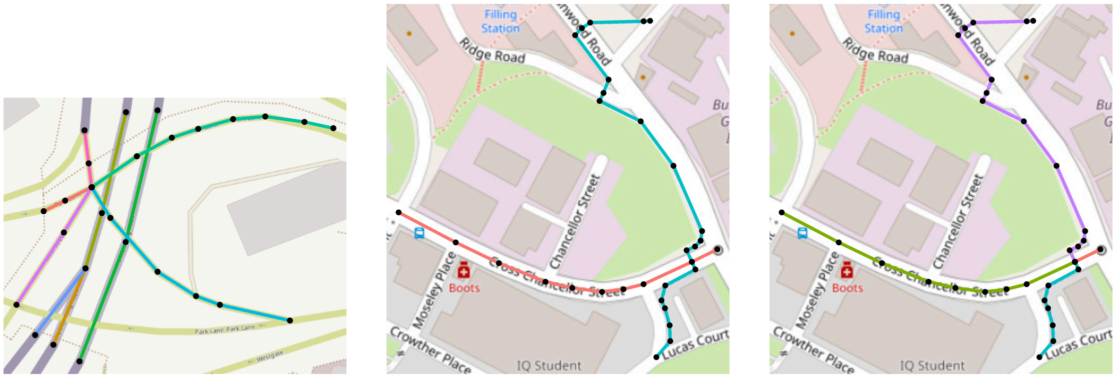
2.5.2 Overpasses, underpasses and other types of intersections

The second problem that we analyse is strictly related to roundabouts and it concerns overpasses, underpasses and other types of street intersections.

Overpasses and underpasses could create a challenge for street network software since they represent a crossing of two highways located at different heights, where clearance to traffic on the lower level is obtained by elevating the higher level. From a routing perspective, this particular structure of the network must be taken into account since, even if we are always working in a two-dimensional space, it should never be possible to pass from one street-level to another. The algorithms behind `stplanr` and `dodgr` were designed taking care of this problem. An example of an overpass, located in the south of Leeds (UK), is reported in Figure 2.6a. We overlayed a coloured map of the corresponding `stplanr` graph representation where there are connections only between streets belonging to the same level.

Another problem, strictly related to overpasses and roundabouts, is the following. There exist some streets in Open Street Map data that intersect each other, lie at the same level, but do not share any point in their boundaries. This implies that, from `stplanr` perspective, those streets are not connected. See, for example, Figure 2.6b. This problem can also be solved (see Figure 2.6c) using `rnet_breakup_vertices()` function.

The `dodgr` approach for representing street networks immediately solves both problems. Moreover, if the input `sf` data is built using `osmdata` package, then the assumption about



(a) **stplanr** representation of an overpass.

(b) **stplanr** representation of an unroutable junction.

(c) The result of breaking up the street network. Each edge is depicted with a different colour.

Figure 2.6: *Left*: `stplanr`-representation of one overpass where there are connections only between roads at the same level. *Center*: `stplanr`-representation of an unroutable junction. *Right*: The result obtained after splitting the junction.

the absence of shared nodes between roads at different levels can be removed because, as we said in the previous sections, the comparisons between the vertices of the network are performed according to their Open Street Map ID(s), which are always unique, even when they share identical coordinates.

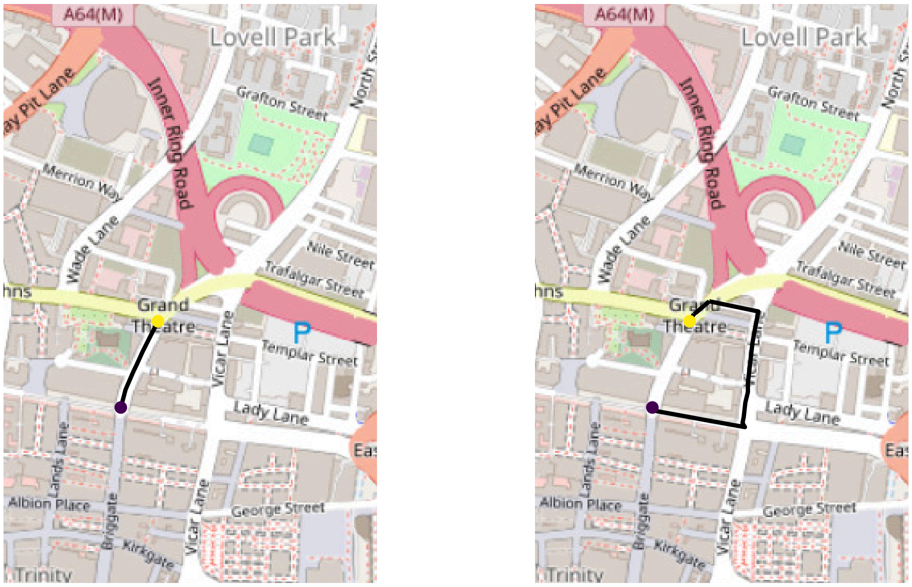
2.5.3 Oneway streets

Oneway streets prohibit certain transport modes (typically only motor traffic) from travelling in one direction. They are used in many cities to free up space for other land uses or dedicated cycleways or walkways, with a textbook example being Torrington Place in London, where one carriageway was converted into a bidirectional cycleway[6]. Oneway streets pose a challenge from a routing perspective since, by definition, they allow vehicles to travel only in one direction.

At present, support for oneway streets has not been implemented in `stplanr`, meaning that for every two vertices in the network, the shortest path between them is symmetrical, even if that implies going against traffic in the real world. Oneway streets are supported by `dodgr`.

To illustrate the real-world consequences of this, Figures 2.7a and 2.7b demonstrate shortest

[6] A consultation report by Camden City Council proposing a oneway street on the Torrington Place / Tavistock Place Corridor, which has been implemented, provides a detailed introduction to oneway streets from a transport planning perspective. See `https://www.camden.gov.uk/documents/20142/3452947/Consultation+leaflet+FINAL.pdf/f628d6e8-c47b-82f1-cb40-8e24db78bea6` for details.



(a) Path in the contra-flow direction

(b) Path in the right direction

Figure 2.7: Examples of routing with oneway street, located in New Briggate, Leeds (UK). Figure (a) presents a path in the contra-flaw direction, estimated by `stplnar`. Figure (b) reports the right path, estimated with `dodgr`.

paths calculated between the end points of an important oneway street in Leeds (UK). The path represented in the first figure, based on a shortest path calculation on the `sfNetwork` class, is shorter but against the law. The path represented in the second figure takes a longer route but follows the law. To activate oneway routing in `dodgr_streetnet` objects, there must be a column in the input `sf` object named *oneway*, typically derived from the *oneway tag* in Open Street Map. We refer to Appendix C for more details. This column must be a logical vector, containing `TRUE` for streets that are oneway only and `FALSE` otherwise. This feature can be deactivated (e.g. when working with pedestrian or bicycle routing, where directionality is generally unimportant) by removing the *oneway* column from the input data and then rebuilding the `dodgr_streetnet` object.

2.5.4 A transport network

Having seen several types of street network components in the previous examples and the corresponding peculiar problems, this final example demonstrates what a citywide street network looks like. We report `R` code to create `sfNetwork` and `dodgr_streetnet` objects starting from an `sf` object with a `LINESTRING` geometry named `chapeltown_leeds`. We

took highway data from *Geofabrik*[7] servers, considering a 5 km radius of the Chapeltown neighbourhood of Leeds (UK), an area the authors are familiar with, as the basis for the following examples.
The conversion between the pure sf format and the stplanr and dodgr representations is performed using the following commands:

```
street_network_stplanr <- SpatialLinesNetwork(
  rnet_breakup_vertices(chapeltown_leeds)
)
Warning: Graph composed of multiple subgraphs, consider cleaning it.
street_network_dodgr <- weight_streetnet(
  chapeltown_leeds,
  wt_profile = "bicycle"
)
```

A few notes:

1. The `rnet_breakup_vertices()` function was applied to the input sf data in the stplanr representation for the reasons explained in the previous examples. It fixes several possible routing problems, and it runs in approximately 3 seconds considering a road network of 12000 segments. Starting from stplanr version 0.8.2, it should scale quite efficiently for larger network.
2. The `SpatialLinesNetwork()` function returns a warning message, and we are going to explain its meaning in the next Section.
3. The `wt_profile` parameter in `weight_streetnet` is used to specify the preferred weighting profile, which is going to determine the dual weights.

Neither packages provides explicit tools for visualisation, other than a basic `plot()` method to show the geometry for `sfNetwork` objects. However, a wide range of visualisation packages can be used on the spatial component, which can be exported from the `@sl` slot (for `sfNetwork` objects), or via the `dodgr_to_sf()` function (for `dodgr_streetnet` objects). See, for example, Figure 2.8. All the examples shown in the previous subsections were also taken from within this case study area.

2.6 Analysing street networks

In this Section, we present how to perform a few common operations on street networks with stplanr and dodgr packages. The examples are based on the road network built in the last example. These operations are repeatedly applied in the next Chapters.

[7] Geofabrik is a website that provides extracts of Open Street Map data that are updated daily. It is one of osmextract's providers, and it is presented in Appendix C.

2.6.1 Connectivity

The first functions that we introduce are related to the identification of the *components* of a street network. The *connectivity* of a network is usually defined as follows: a graph G is said to be *connected* if there exists a sequence of edges going from each vertex to every other vertex. The *components* of the network are its sub-graph(s) formed by all connected vertexes (Kolaczyk and Csárdi, 2014).
The package dodgr defines a function, named dodgr_components(), which is used to identify the clusters of connected edges in a dodgr_streetnet object. The output is the same graph object in input with an additional column, called component, that identifies the component of each edge, sequentially numbering them starting from one (the ID of the largest component).
On the other side, stplanr package does not define any explicit function for a direct identification of the components of a street network. Nevertheless, the function sln_clean_graph() can be used for an automatic selection of all the edges belonging to the largest component, creating a fully connected graph. Moreover, as we saw with the previous example, SpatialLinesNetwork() may raise a warning every time its output is formed by two or more components, suggesting the adoption of sln_clean_graph() function. The same result can be obtained using the dodgr approach by filtering all the edges where the *component* column is equal to one. As we can see from Figure 2.8, there are several small clusters of roads in the Chapeltown road network, and we can exclude them with the following commands:

```
street_network_stplanr <- sln_clean_graph(street_network_stplanr)
street_network_dodgr <- street_network_dodgr[
street_network_dodgr$component == 1,
]
```

The connectivity analysis is linked with Conditional Autoregressive prior and kernel methods, as detailed in several papers (Besag and Kooperberg, 1995; Rakshit, Baddeley, and Nair, 2019) and also mentioned in Chapters 4 and 5.

2.6.2 Shortest paths

We present now several functions that can be used for estimating shortest (in terms of geographical distance) and fastest (according to the dual weights set by dodgr package) paths between two locations: Monk Bridge, which lies in the south-west of Leeds City Center, and Chapeltown neighbourhood, located in the north-east. The first step is the geo-coding of the coordinates of the two points through stplanr::geo_code() via the *Nominatim* service (Open Street Map, 2017):

```
leeds_monk_bridge <- geo_code("leeds monk bridge")
chapeltown <- geo_code("Leeds Chapeltown")
```

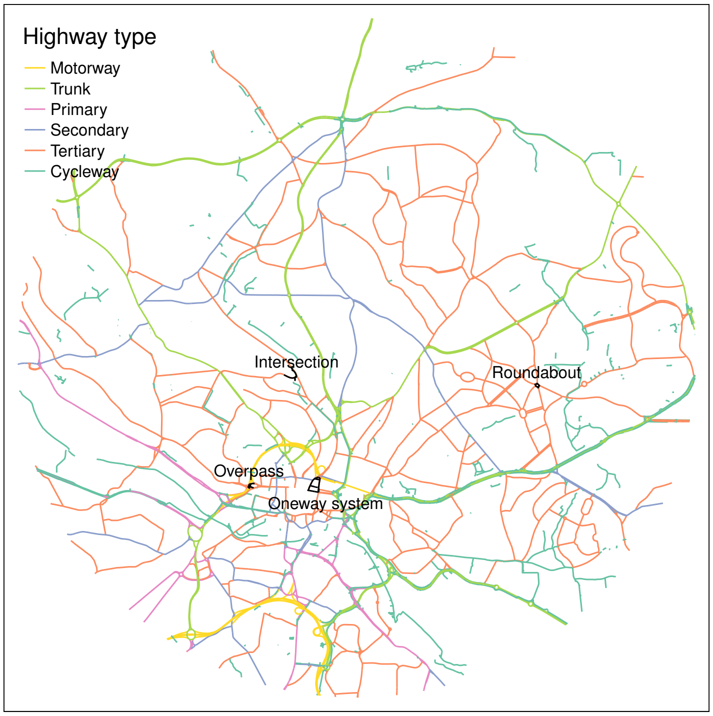


Figure 2.8: Illustration of a citywide street network, in Leeds (UK). For simplicity, we report only the most important highways. The previous examples were taken from within this case-study area.

The `route_local()` function can be used for calculating the shortest path according to the `stplanr` approach. The inputs of the function are an `sfNetwork` object, representing the street network, and the coordinates of the start and end points, either as numeric values or text strings. The following command estimates the shortest path between Monk Bridge and Chapeltown, given the road network built in the previous examples.

```
stplanr_shortest <- route_local(
  sln = street_network_stplanr,
  from = leeds_monk_bridge,
  to = chapeltown
)
```

The output of `route_local()` is an `sf` object, which is created as a subset of the original input data, containing only the edges connecting the two points through the shortest path. The `dodgr` package offers multiple options for estimating shortest and fastest paths on a street network. The main function is called `dodgr_paths()` and it works as follows:

```
dodgr_fastest_ids <- dodgr_paths(
  street_network_dodgr, leeds_monk_bridge, chapeltown, vertices = FALSE
)
```

The inputs are a `dodgr_streetnet` object, representing the street network, and two matrices (or vectors) of numeric coordinates for the start and end points. The output is a list of paths tracing the connections between the points, either as a list of vertices or edges ID(s) (according to the `vertices` parameter). `dodgr` functions are optimized for many-to-many pairwise distances on dual-weighted graphs, so the element `dodgr_fastest_ids[[i]][[j]]` contains the path from the ith starting point to the jth ending point. The examples presented in this manuscript are based on just two points, so the indexes of the vertices composing the best route between Monk Bridge and Chapeltown can be extracted with the following command: `dodgr_fastest_ids[[1]][[1]]`. The fastest path can be reconstructed as a `dodgr_streetnet` dataframe of edges as follows:

```
dodgr_fastest_path <- street_network_dodgr[dodgr_fastest_ids[[1]][[1]], ]
```

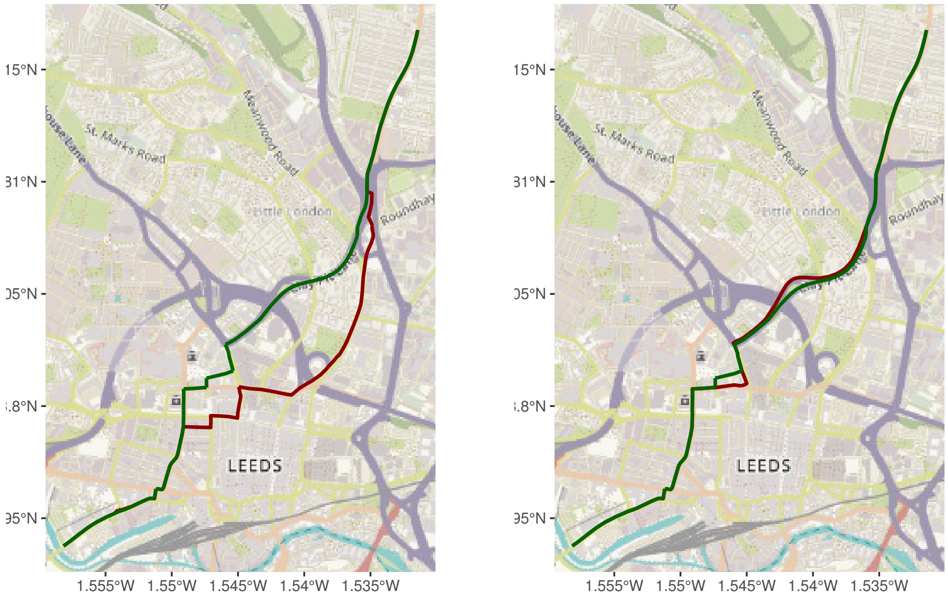
Both routes are reported in Figure 2.9a. The shortest path suggested by `route_local()` is coloured in dark-green, and it is different from the fastest path suggested by `dodgr_paths()` and coloured in dark-red. We can see that the route chosen by `stplanr` is going through several trunks and motorways, while `dodgr` path prefers tertiary roads and cycleways. This difference is due to the fact that the fastest way is optimised for bicycle routing using a dual-weights system.

More precisely, as we mentioned in the previous sections, the real advantage of `dodgr` approach is the ability to calculate paths on *dual-weighted graphs*. The fastest path is calculated according to one set of weights, while the resultant distances are calculated by accumulating a different set of weights along the resultant path. This is particularly important for realistic transport routing, where different kinds of ways are more or less suitable for different kinds of transport. The true shortest path for a pedestrian may be along an eight-lane highway, but they are never likely to actually traverse that way. Instead, that eight-line highway should be weighted to yield an effective length that is longer than the actual geographical length. `dodgr` offers a range of *weighting profiles*, listed using the following `R` command, and detailed in the help page of the included `dodgr::weighting_profiles` data set.

```
wp <- weighting_profiles
unique(wp$weighting_profiles$name)
#> "foot" "horse" "wheelchair" "bicycle" "moped" "motorcycle" "motorcar"
#> "goods" "hgv" "psv"
```

Street networks can be weighted for transport of any of the associated types, by entering the name in the `wt_profile` parameter. The following code demonstrates what one profile actually looks like.

```
head(wp$weighting_profiles [wp$weighting_profiles$name == "foot", ])
#>     name  way          value   max_speed
#> 1   foot  motorway       0.0          NA
```



(a) Shortest and fastest paths according to a bicycle weighting profile.

(b) Shortest and fastest paths according to a motorcar weighting profile.

Figure 2.9: Graphical example of shortest paths between Monk Bridge and Chapeltown (UK). The dark green route is the path suggested by `stplanr`, while the dark red route is the path suggested by `dodgr`. The red route on the left is optimized according to a bicycle mode of transport, while the red route on the right is optimized according to a motorcar transport.

```
#> 2    foot    trunk           0.4          NA
#> 3    foot    primary         0.5           5
#> 4    foot    secondary       0.6           5
#> 5    foot    tertiary        0.7           5
#> 6    foot    unclassified    0.8           5
```

We can check the differences between two weighting profiles by estimating the fastest path between Monk Bridge and Chapeltown according to a *motorcar* type of transport. We do not repeat the commands since they are analogous to the previous example, but the fastest path is reported in Figure 2.9b. We can see that the routes suggested by `stplanr` and `dodgr` are almost identical, but for a few minor differences in the City Center due to the fact that the shortest path estimated by `route_local()` is going against the flow.

The same ideas can be tested using the `dodgr_dists()` function. The following code calculates the geographical distance between Monk Bridge and Chapeltown according to a bicycle weighting profile using the fastest path:

```
dodgr_dists(
  graph = street_network_dodgr,
  from = leeds_monk_bridge,
  to = chapeltown,
  shortest = FALSE
)
#> 8023
```

or the shortest path:

```
dodgr_dists(
  graph = street_network_dodgr,
  from = leeds_monk_bridge,
  to = chapeltown,
  shortest = TRUE
)
#> 3704
```

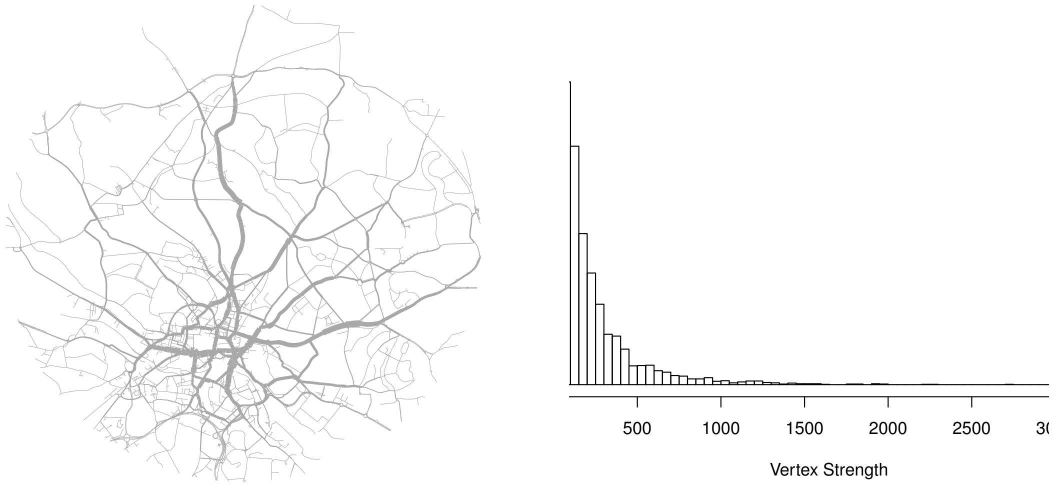
dodgr also offers the ability to incorporate elevation data, and to take account of the effect of elevation changes on travel times. While elevation has little or no effect on motorised transport, it has very important effects on human-powered modes of transport (walking and bicycling). Elevation effects can currently only be incorporated when the underlying network is represented in is represented in silicate (sc) format (Sumner and Padgham, 2020), and we refer to the corresponding R package repository for more details. This effectively only involves replacing the dodgr_streetnet() function with dodgr_streetnet_sc(). The final dodgr network will appear largely the same but will incorporate effects of elevation changes, as well as waiting times at traffic lights, and time penalties for waiting to turn across oncoming traffic. Time-based routing using these dodgr networks will reflect highly realistic transit behaviour.

Finally dodgr also offers the functions dodgr_isodists() and dodgr_isochrones() to calculate the respective isocontours from a given set of starting points. Like all other dodgr functions, these are computationally optimised for highly efficient parallel calculation of isocontours from large numbers of starting points.

2.6.3 Graph metrics

A strength of the stplanr approach to street networks analysis is that its sfNetwork class contains an igraph object, opening-up a wide range of algorithms provided in the igraph package (Kolaczyk and Csárdi, 2014). A basic characteristic of any graph is whether or not it is simple, meaning all nodes are connected by only one edge (or either one or two edges for directed graphs). We can use the igraph function is.simple() to test the street_network_stplanr citywide street network:

```
is.simple(slot(street_network_stplanr, "g"))
#> FALSE
```



(a) igraph: edge betweenness centrality metric.

(b) igraph: vertices strength metric.

Figure 2.10: Graphical summary of two network metrics estimated using `igraph` algorithms. *Left*: We can see how the most important highways in Leeds are highlighted by the betweenness centrality measure. *Right*: Histogram of vertex strength. Several vertices are linked to only a few short edges.

The graph representation of a street network is typically not simple. Another example is provided in the following command, which estimates the edge betweenness measure of centrality that indicates the number of shortest paths that pass through the edges, considering all nodes or some subset of them:

```
edge_betweenness(street_network_stplanr@g)
```

Other `igraph` functions can calculate other network characteristics related to the vertices, such as their *closeness centrality* (which is a measure of the distance from a vertex to all others) or their *degree* (the number of edges incident to each vertex). The following command, to provide another example, estimates the vertex *strength* (defined as the sum of the lengths of all edges incident to a given vertex). Note that, by default, the weights of the edges are the lengths of the corresponding `LINESTRING` objects.

```
strength(street_network_stplanr@g)
```

At present, there is no functionality in `stplanr` for linking such vertex metrics to the corresponding spatial coordinates (an issue we plan to address in future work). Still, vertex metrics can be summarised visually, to provide an overview of the structure of the street network that can be compared with street networks from other cities, to explore concepts such as interdependence and resilience (Morris and Barthelemy, 2014). A graphical summary of the edge betweenness and vertex strength metrics is reported in Figures 2.10a and 2.10b.

The authors of `stplanr` recently extended the integration between `igraph` and `sfNetworks` objects, defining a new function named `rnet_group` that can be used to explore the spatial distribution of several algorithms for *community detection* (Fortunato, 2010). The following code can be used to estimate the *community* structure of the spatial network according to an algorithm defined in Blondel et al. (2008).

```
rnet_group(street_network_stplanr, igraph::cluster_louvain)
```

The output is an `sfNetwork` object with an extra column in the `sl` slot named *rnet_group* that summarizes the partitions.

`dodgr` also supports network analysis. The function `dodgr_centrality()` calculates betweenness centrality, with an implementation that can be computed using parallel processing capabilities of modern computers and which enables centrality to be estimated with respect to weighted measures of distance or time. The following command calculates edge betweenness centrality, resulting in a `dodgr_streetnet` object that has an additional column called `centrality`.

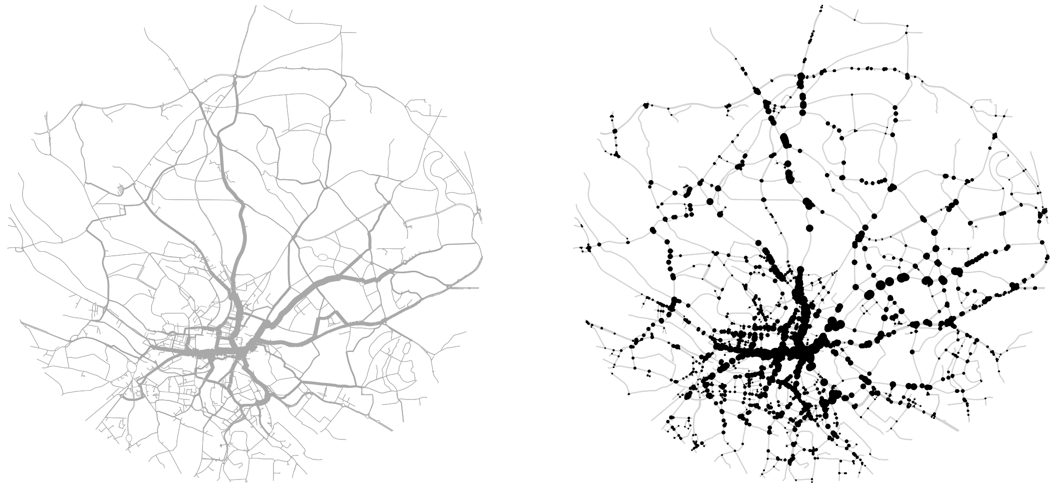
```
dodgr_edges_betweennes <- dodgr_centrality(street_network_graph)
```

Vertex betweenness centrality measures can be calculated by setting the `edges` argument to `FALSE` in the `dodgr_centrality()` function. The output is a `data.frame` object with a column called `centrality`. A graphical comparison of edge and vertex betweenness is reported in Figures 2.11a and 2.11b.

A characteristic of Open Street Map data is that vertex locations are, to some extent, arbitrary, which can lead to overestimates of centrality near road segments with arbitrarily high numbers of vertices per unit distance. `dodgr` addresses this issue by enabling betweenness centrality measures to be calculated with *contracted* street networks using the `contracted` argument (which defaults to `TRUE`). We refer to the help page of `dodgr_centrality()` for details, including the `dist_threshold` argument which constrains path lengths on which centrality estimates are based. Passing suitable values to `dist_threshold` (and the equivalent `cutoff` argument in `igraph` functions) can improve the computational efficiency of street network centrality calculations.

2.7 Discussion and Conclusion

This Chapter has demonstrated that street networks, a particular type of spatial network representing transport systems, can be encoded in classes that build on pre-existing data structures that are available in the statistical programming language `R`. We explored the representation of a range of street network features in `stplanr` and `dodgr`, two recently



(a) dodgr: edge betweenness centrality metric.

(b) dodgr: vertices centrality metric..

Figure 2.11: Graphical summary of edge and vertex betweenness centrality measure estimated using `dodgr`. We can see that both metrics highlight the most important roads of Leeds city area.

developed `R` packages that provide explicit support for street networks building on preexisting geographical/graph and simple data frame class definitions, respectively. Overall, we found that both approaches can be used for analysing street networks, with support for shortest path calculation, characterisation of networks, nodes and edges, and the ability to modify networks. Such capabilities can be (and to some extent are already being) implemented in other languages for interactive data analyisis, as demonstrated by the established `Python` package `osmnx` (Boeing, 2017) and the recently developed `Julia` package `OpenStreetMap.jl`.

Perhaps more important than languages of implementation are the underlying concepts and real-world applications. We found that concepts of network pre-processing, including the vital stage of breaking-up linestrings at junction intersections for approaches based on geographic data structures and weighting profiles/contraction, are vital for effective use of street network data, regardless of language of implementation or the data structures used to represent the street network. Although the approaches have been applied on datasets designed to highlight the techniques rather than to answer applied research questions, it seems clear that both approaches can help answer important research questions, including:

- What is the relative circuity (divergence factor) for motorised and non-motorised modes in different areas?

- What are the relationships between street network characteristics and travel behaviour (e.g. are some network forms associated with more walking and cycling)?
- What are the optimal places to intervene on the road network to improve transport system performances, e.g. based on environmental, health or journey time performance metrics?

Answers to these and many other research questions have real-world applications, particularly in transport planning. Perhaps key test of the packages will therefore be whether they see widespread uptake, beyond niche applications (Lovelace, Goodman, et al., 2017), of the type that spatial R packages such as sf have seen.

stplanr and dodgr each have strengths and limitations that make them more or less appropriate for different tasks. Building on established spatial and graph packages, stplanr's sfNetwork can be analysed using a wide range of spatial and graph functions. Its graph structure can be linked and analysed with several algorithms implemented in igraph. A downside of stplanr, at the time of writing, is that it lacks support for routing features such as in-built weighting profiles for different modes of transport and the representation of one-way streets. Adding these features, either directly in stplanr or other packages that build on the geographical/graph data structures it uses, could represent a promising direction of future development that would address this limitation. dodgr's class system, by contrast, is more specifically focussed on Open Street Map data and routing, with support for a variety of modes and oneway streets being key strengths. The examples presented in Sections 2.5 and 2.6 exhibit the relevance of dual-weights for realistic routing. These relative strengths and weaknesses raise the question of priorities for future development, which could go in various directions including better support for routing in stplanr to integration with other spatial classes in dodgr (which can already be translated to established graph and spatial classes).

An article on the *r-spatial* blog reports an alternative way of representing dual sf/igraph objects that uses tidygraph as the basis for a data frame-like representation of spatial networks (Pedersen, 2020). This approach offers some potential advantages in terms of usability and could be adopted as the basis of future street network classes, but it is not yet mature enough to compare with stplanr and dodgr. A range of alternative approaches, as yet unexplored, may also be advantageous, for example using R as an interface to high performance graph libraries such as sDNA (Cooper and Chiaradia, 2020). While such options remain relatively unexplored, stplanr and dodgr have been tested, are actively maintained, and provide class structures that open-up street network analysis to reproducible analysis.

Acknowledgements

The ggspatial (Dunnington, 2020a) maps are created using tiles from © OpenStreetMap Contributors Tiles.

Data and Codes Availability Statement

The data and codes that support the findings of this Chapter are available with the identifier(s) at the private links: data and code.

Chapter 3

Assessing the Risk of Car Crashes in Road Networks

> *Cosa cazzo serve uno che ti dice che a Caropepe Valguarnera alle 18.45 sulla Tangenziale di Milano c'è il traffico? A un cazzo!*
>
> Giuseppe Cruciani - La Zanzara, 2019-10-18

Based on: *Borgoni, R., Gilardi, A. and Zappa, D., 2020. Assessing the Risk of Car Crashes in Road Networks. Social Indicators Research, pp.1-19.*

3.1 Introduction

Worldwide, thousands of people die annually in highway-related crashes and millions are injured. By 2030, car crashes are predicted to be the 5th leading cause of death in the world according to the World Health Organization (F. L. Mannering and Bhat, 2014). In the United States, traffic incidents are the first cause of workplace fatalities and the leading cause of death for children aged 1–19 (Cunningham, Walton, and Carter, 2018). Underprivileged people, such as those living in poor socio-economic areas, may be vulnerable because of their lack of access to information and to safe roads and vehicles, while adolescents are more likely to be involved in a car or motorcycle accident because of their inexperience. Thus, road crashes have relevant social impacts. Crashes may also have high social costs. For example, a deficient road safety policy implies that hospital beds will be occupied by victims of traffic accidents, increasing the burden on the health system. In low- and middle-income countries, a direct link is found between road safety and poverty (Shah et al., 2018). Research in Bangladesh[1] has shown that 50% of the people involved in a road crash fall into poverty. This is due to losing the ability to generate income and/or high out of the pocket expenses for hospitals and medication.

The statistical analysis of crash data has historically been fundamental to estimate the risk of accidents and develop road-safety policies aimed at saving lives and reducing the severity of injuries. The European Transportation Safety Council defines a Safety Performance

[1] See https://www.safe-crossings.org.

Indicator (SPI) of accident risk as *a measurement that is causally related to accidents or injuries, used in addition to a count of accidents or injuries in order to indicate a safety performance or understand the process that leads to accidents.*

Driving-related data (such as acceleration and braking) and crash-related data (such as those made available from vehicle black-boxes) would help to identify cause and effect relationships, but they are rarely recorded. Hence, a large majority of research has addressed the problem in terms of understanding the factors that affect the frequency of crashes, i.e. the number of crashes occurring in some geographical space over some specified time periods. A few papers (F. Mannering, 2018; F. L. Mannering and Bhat, 2014) provide a comprehensive review of current methodological approaches for studying crash frequencies, stressing the relevance of data quality, the great difficulty in accessing accident related data and the role of spatial information. In order to estimate a driver's risk of crash, for instance, it is customary to use car characteristics, claims history or qualitative and quantitative information of the region where the driver lives (i.e. the population density of the postal code or the town of the driver). However, these covariates are often proprietary and limited to same places or periods. Additional information to infer crash causes can also be obtained by mining exogenous databases or scraping the web. We refer to the use of individual/company-based credit scoring databases and car black boxes to monitor driving habits; moreover, the latent propensity of some categories to be involved in an accident can be ascertained by studying the text of accident policy reports or the "sentiment" that can be inferred from the tweets published by customers, etc (Dugas et al., 2003; Guo, 2003; Zappa et al., 2019). However, these data are typically difficult to collect.

In this Chapter, we show how to monitor accident exposure in geographical space using data taken from open archives. Open data sources, presented in Section 3.2 and already mentioned in Chapter 2, have become more accessible in recent years thanks to the rapid growth of software and hardware capabilities, which can easily manage a huge amount of spatial data and disseminate it via the web. Open data archives often provide relevant information to support local authorities in allocating resources and making political decisions to mitigate accident risk. The case study presented in this Chapter considers the problem of modelling car accident frequency in the municipality area of Milan (Italy). Using open source datasets provided by the Italian national office of statistics (ISTAT), which records information on the location of all car accidents that resulted in fatalities or injuries of at least one person, we projected crashes on the road network of the city, counted the number of occurrences in each street segment and augmented the dataset using several geocoded open data sources. To estimate the risk of accidents, we employed a Zero Inflated Poisson (ZIP) regression model (Lambert, 1992) to account for the excess of zeros that occur in this dataset. This allows us to define two indices that model different aspects of accident risk for every street segment that composes the road network. More specifically, the first index, derived from the counting component of the zero inflated Poisson model, measures how prone the segment is to car crashes. The other, derived by the zero component of the zero inflated model, represents a measure of the likelihood of the segment not to be exposed to crashes.

We show how this secondary information is useful to measure the impact of the road structure (e.g. presence of traffic lights and/or pedestrian crossings, type of highways, etc.) and the role of local socio-demographic features (e.g. population density, concentration of families, housings etc.) to estimate the risk of accidents. In our analysis, we did not consider traffic density measures since they are not available for every street of the network. Although a few existing projects share open traffic data for several cities[2], they typically cover just small portions of the street network.

We are aware that the data considered in this Chapter, although being the largest and most omni-comprehensive open-source geocoded dataset available on car crashes in Italy, may possibly provide only a partially complete image of the entire set of events that actually occurred in the area, since it excludes all accidents not reported to the police. From a social perspective however, these accidents are expected to be of lesser concern than those that resulted in injuries or fatalities because they have less impact on people's lives and the health care system. Hence, this under-reporting is not expected to severely affect our results because the events that led to serious adverse outcomes are unlikely to remain unreported.

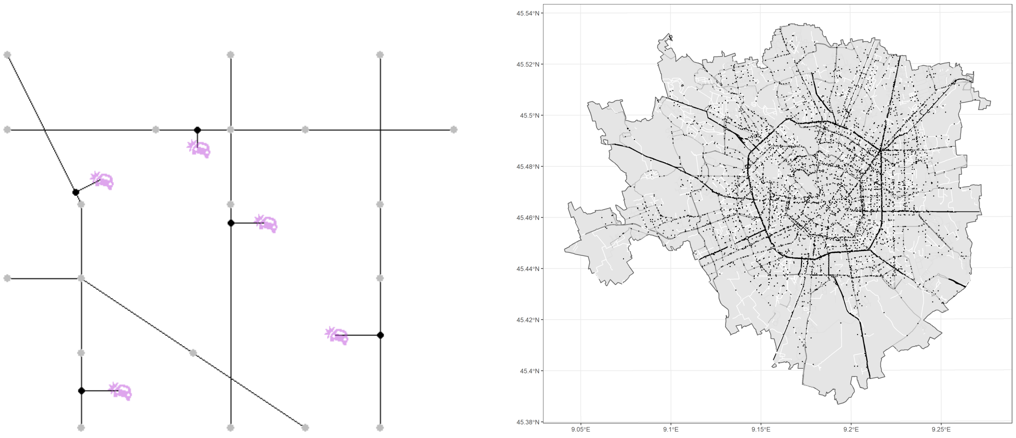
From a methodological and applied perspective, modelling crash frequency to define appropriate safety indexes of territorial units, either areas or streets, is not new. Many papers (Bao et al., 2012; Egilmez and McAvoy, 2013; Gitelman, Doveh, and Hakkert, 2010; Hermans, Van den Bossche, and Wets, 2008, 2009; Rosić et al., 2017; Rosolino et al., 2014) estimate safety measures at a national or regional level using precompiled indices such as economic growth, the number of highways, the number of registered cars, the domain knowledge of experts, etc. To the best of our knowledge, however, this Chapter summarises an original attempt to develop a road safety index modelling accident data on an entire road network covering thousands of kilometres of an extensive and heterogeneous urban territory using a large amount of geocoded data taken from administrative records as well as from different sources that are entirely open. In this sense, the approach discussed herein can be straightforwardly replicated in all areas/countries where similar open source information is available.

The rest of the Chapter is organized as follows. In Section 3.2, the data sources are described in detail. In Sections 3.3 and 3.4, the statistical methods adopted in this Chapter are briefly considered, namely the ZIP model and the geographically weighted principal components, respectively. In Section 3.5, we comment the results of the ZIP regression model used to estimate road risk and safety indices. Conclusions and discussion are in Section 3.6.

3.2 Open Data Sources for Car Crashes

We consider all car accidents occurring between the 1st of January 2015 and the 31st of December 2017 in Milan that required police intervention and resulted in fatalities or

[2] See https://github.com/graphhopper/open-traffic-collection.



(a) Projecting car crashes

(b) Car crashes and road network

Figure 3.1: *Left:* Projecting car crashes to the nearest segment. The grey dots delimit the street segments; *Right:* Spatial representation of car crashes in the road network of Milan during 2015.

injuries of at least one person[3]. The address of every accident was recorded by the police, and we geocoded it with UTM coordinates using the R (R Core Team, 2020) package `googleway` (Cooley, 2020) and the Google geocoding API (Application Program Interface). The original sample included 26,223 events. We exclude from the dataset all those car accidents with an unknown or incomplete civic address that makes geocoding unfeasible or uncertain. The final sample includes 24,948 events. Of these accidents, 8341 occurred in 2015, 8506 in 2016 and 8101 in 2017. The road network is built using data from Open Street Map. Open Street Map is a project that aims at building a free and editable map of the World with an open-content license. The basic components of OpenStreetMap data are called *elements* and they are divided in: *nodes*, which represent points on the earth's surface; *ways*, which are ordered lists of nodes; and *relations*, which are lists of nodes, ways and other relations where each member has additional information that describes its relationship with the other elements. Bus routes, railways and administrative boundaries are classical examples of relations. Every physical object in the landscape is represented by these three elements and its attributes are stored using a *tag*, which is simply a pair of items that identify a category, a *key*, and the corresponding *value* (e.g. *street = "motorway"* or *name = "Park Avenue"*). We point out that, by definition of OpenStreetMap ways, all streets of our network are internally stored as the union of a set of segments defined by their nodes (the grey dots in Figure 3.1a) and the regression model presented later on in

[3] The data are available at `https://www.istat.it/it/archivio/87539`.

the Chapter is based on this particular segmentation.

We use the R package osmdata (Padgham et al., 2017) to download the data to create the street network. It includes every primary, secondary, tertiary, unclassified and residential highway for a total of 25,675 segments having different lengths, which are calculated as the sum of the Euclidean distances between all subsequent nodes that identify each street segment.

All car crashes are projected to the nearest point belonging to the linear network (as in Figure 3.1a), and the final result is reported in Figure 3.1b. Secondary data used as covariates in the model described later on in the Chapter are also collected from the web. More specifically, demographic variables representing population (6 variables), family (3 variables) and building characteristics (5 variables) are taken from 2011 Census Data. These variables are described in detail in Table 3.1.

The locations of traffic lights and pedestrian crossings are downloaded from Open Street Map and projected onto the nearest segment of the network using the same procedure described above. Two binary indicators are defined recording the presence of traffic lights and pedestrian crossings in every street of the network. Finally, the number of roads touching each street segment[4] is calculated to provide a proxy measure of the traffic flow. The higher the number of touching segments, the higher the traffic volume is expected to be.

3.3 Modelling Car Crashes: Methodology

The response variable Y of the regression model considered hereafter is the total number of car crashes projected onto each segment of the road network, and we aim at modelling Y as a function of a set of non-random covariates, say $\boldsymbol{x}$.

Since the response variable takes non-negative integer values, the Poisson regression model serves as a basis to assess the roles of those factors that potentially influence the crash frequencies (Agresti, 2015). This assumption is widely accepted in the literature on road safety and actuarial[5] field (De Jong, Heller, et al., 2008; F. L. Mannering and Bhat, 2014). Let $Y \sim \text{Poisson}(\lambda)$, the loglinear Poisson regression model is specified by setting

$$\log \lambda = \boldsymbol{x}'\boldsymbol{\beta}$$

where $\boldsymbol{x}$ is a q-vector of explanatory variables and $\boldsymbol{\beta}$ is a q-vector of unknown coefficients. Hence, the expected value of Y is modelled on the scale of the canonical link. Standard maximum likelihood procedure is used to find the estimate of $\boldsymbol{\beta}$.

[4] Two street segments are touching each other when the only points in common lie in the union of their boundaries.

[5] How crashes are distributed within a population is very relevant for insurance companies where the accident frequency is used to determine policy premiums, i.e. the cost of the driving insurance contracts often compulsory to drive a car.

Table 3.1: Demographic variables used in the analysis. For reader's convenience, the name of the GWPCA transformation described in Section 3.4 is also reported in the last column.

Macro-area	Description	Short-name
Population variables	Total residential population Total working population (age ≥ 15) Total number of students (age ≥ 15) Total population of daily commuters within the municipality Total population of daily commuters outside of the municipality Total number of foreign inhabitants	*GWPCA_pop*
Building variables	Total number of occupied dwellings Total number of unoccupied dwellings Total number of buildings Total number of occupied buildings Total number of residential buildings	*GWPCA_build*
Family variables	Total number of families with a rented house Total number of families with a proprietary house Total number of families	*GWPCA_fam*

Often, the expected value of Y is proportional to an index τ, which stands for different exposures of the statistical units. For instance, τ represents the exposure with respect to a different amount of time, a different population size or a different size of spatial areas. In these circumstances, one is actually interested in modelling the sampling rate Y/τ, which implies that the loglinear model is changed in

$$\log\left(\frac{\lambda}{\tau}\right) = \boldsymbol{x}'\boldsymbol{\beta}$$

or equivalently $\log\lambda = \boldsymbol{x}'\boldsymbol{\beta} + \log\tau$. The adjusting term $\log\tau$ is called *offset.* In the case study discussed later in the Chapter, the length of a road segment is used as an offset.
Count data are termed *zero inflated* when the number of 0 frequencies in the sample is much larger than what the Poisson model can explain. Following Lord and F. Mannering (2010) *zero inflation* occurs for crash data when the sampling frame is highly detailed, and low crash counts are then expected. These conditions are both present in our case: in the period 2015 - 2017, the crash frequency in Italy was around 0.056 and we are dealing with a very large road network of more than 25,000 segments, many of which having a very short length. This resulted in a high frequency of zero crash segments (approximately 85%), much larger than expected under standard Poisson models. As noticed by Lord, Washington, and

Ivan (2007), segments with zero crashes cannot be considered fully safe. The aim of this model is, in fact, to estimate an index for the risk of accidents, even for those segments where a zero frequency is observed.

Technically, zero inflated models explicitly account for a high frequency at zero by mixing discrete distributions with a degenerate distribution with point mass of one at zero. Given the high number of road segments without accidents, the number of car crashes is modelled using a *Zero Inflated Poisson* (ZIP) model (Lambert, 1992). More specifically, the ZIP model assumes that

$$Y \sim \begin{cases} 0 & \text{with probability } \phi \\ \text{Poisson}(\theta) & \text{with probability } 1-\phi. \end{cases}$$

Hence, the unconditional distribution turns out to be

$$\mathbb{P}(Y=y) = \begin{cases} \phi + (1-\phi)e^{-\theta} & \text{if } y=0 \\ (1-\phi)e^{-\theta}\frac{\theta^y}{y!} & \text{if } y=1,2,\dots, \end{cases}$$

where $0 \leq \phi \leq 1$ and $\theta > 0$. The mean and the variance of this distribution are respectively $\mathbb{E}[Y] = (1-\phi)\theta$ and $\text{var}[Y] = (1-\phi)\theta(1+\phi\theta)$. Since $\text{var}[Y] > \mathbb{E}[Y]$, the model also allows for overdispersion in the count distribution. In a regression framework, the parameters ϕ and θ can be modelled as a function of a set of explanatory variables by assuming $\log\theta = \boldsymbol{x}'\boldsymbol{\beta}$ and $\log\frac{\phi}{1-\phi} = \boldsymbol{z}'\boldsymbol{\gamma}$, where $\boldsymbol{x}$ and $\boldsymbol{z}$ are two vectors of m and q predictors[6] and $\boldsymbol{\beta}$ and $\boldsymbol{\gamma}$ are unknown parameters.

Many generalizations are available. Lord and F. Mannering (2010) and F. L. Mannering and Bhat (2014) provide detailed review of the key issues associated with crash-frequency data as well as the strengths and the weaknesses of the various methodological approaches.

3.4 Geographically Weighted Principal Components Analysis

The fourteen demographic variables, grouped into three groups as described in Table 3.1, are recoded for every census tract of Milan. All variables in the same group are correlated to each other and, for this reason, we summarize them by a dimension reduction technique with the minimum information loss.

A standard approach would be to perform a *principal component analysis* (PCA) (Jolliffe, 2002) on each group of variables separately and retain the first principal component (PC) to represent the corresponding macro-demographic dimension. Instead, since in this study we are concerned with spatial data, a spatial variant of PCA is adopted. Ignoring the spatial characteristics of the data and applying a standard PCA does not seem appropriate since the spatial effects are expected to be relevant and can potentially provide a more complete understanding of the considered process. A *geographically weighted PCA* (GWPCA)

[6] The two sets of predictors are not subjected to constrains and they may be either the same or different.

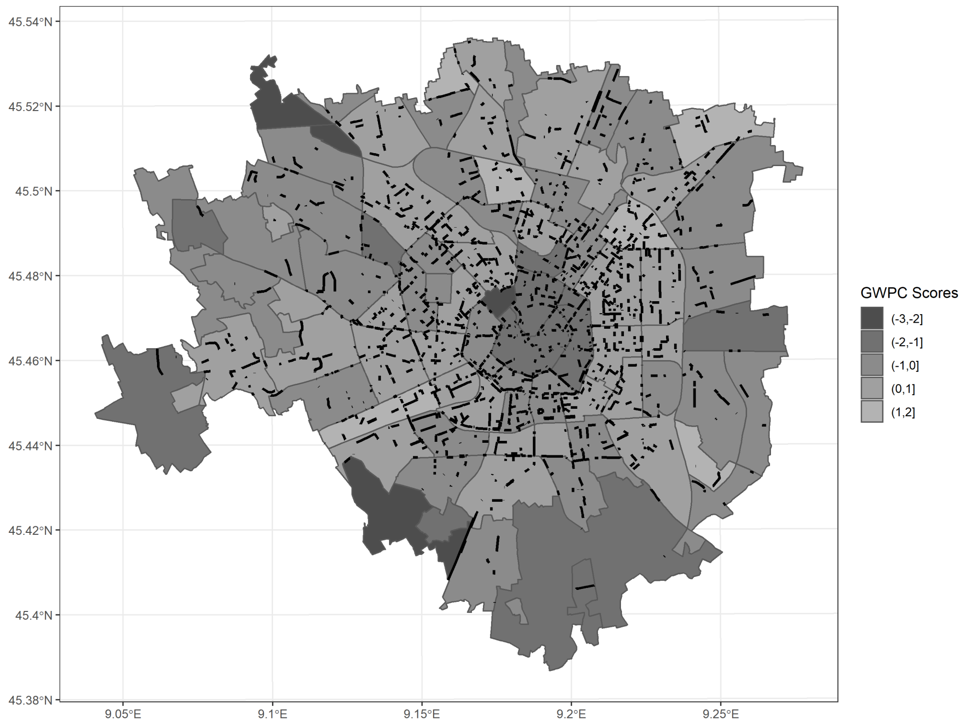


Figure 3.2: GWPC scores for the first group of demographic variables. The black lines represent the street segments where the number of observed car crashes during 2015-2017 is greater than 0.

(Fotheringham, Brunsdon, and Charlton, 2003) is implemented to account for a potential spatial heterogeneity in the data (see Harris, Brunsdon, and Charlton (2011) for a further discussion).
To estimate the geographically weighted PC scores, it is necessary to first calculate the sample geographically-weighted variance and covariance matrix conditional to each sample unit. More precisely, let k be the number of variables to be processed in the PCA and let $\boldsymbol{X}$ be the $n \times k$ matrix of their sample values, n being the sample size. Then, the GW variance–covariance matrix with respect to the unit j of the sample is defined by $\boldsymbol{S}_j = \boldsymbol{X}'\boldsymbol{W}_j\boldsymbol{X}$, where $\boldsymbol{W}_j$ is a $n \times n$ diagonal matrix of geographical weights estimated using some kernel function that depends on the distance d_{jl} between unit j and every other unit l in the geographical space. In this Chapter, we adopt a bi-square kernel function, i.e.

$$w_{ll}(j) = \begin{cases} (1 - (d_{jl}/h)^2)^2 & \text{if } d_{jl} \leq h \\ 0 & \text{otherwise,} \end{cases}$$

where h represents the bandwidth parameter, that is set equal to 2500m after testing the

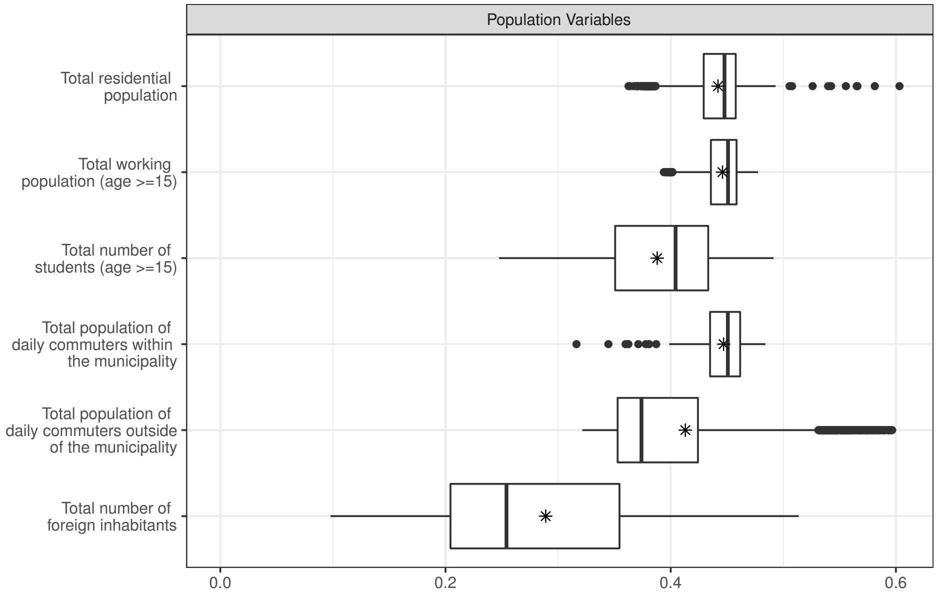


Figure 3.3: Boxplots of GW loadings for every census tract; grey stars represent the classical PC loadings for the first principal component.

predictive performance of the model described in Section 3.3 on a grid of different values. Since the units are administrative areas that are geometrically represented by polygons, the distances d_{jl} are calculated as the minimum Euclidean distance between the two polygons in metres. This implies that in the estimation of the local variance–covariance matrix $\boldsymbol{S}_j$ we exclude those polygons that are farther than 2500 meters from the jth census tract.
The singular value decomposition of $\boldsymbol{S}_j$ provides GW eigenvalues and GW eigenvectors. The product of the j-th row of $\boldsymbol{X}$ with the GW eigenvector associated to the biggest eigenvalue provides the score of the first GWPC at location j. The procedure to calculate the GWPCs has been implemented in the R software. It should be noted that this is a quite computationally intensive algorithm since it requires the estimation of a different weight matrix for each census tract. The whole procedure took slightly more than 3 hours on our laptops (processor I7-6800HQ, ram 8 GB).
Figure 3.2 shows a graphical representation of GWPC analysis applied to the first group of demographic variables, i.e. the population variables in Table 3.1. The map of those segments where at least one car accident occurred during the years 2015 – 2017 is superimposed on the thematic map of the scores by neighbourhood. Note that for this particular plot, we reported the NIL (*Nuclei di Identità Locale*) neighbourhoods instead of the census tracts that, being extremely small, would have made the map extremely confused and difficult to understand.

We calculate the value of the GWPC score for every NIL by averaging the GWPC score of each census tract intersecting that NIL. From Figure 3.2 it appears that there is a spatial relationship between the GWPC scores and the observed number of car crashes since there are far more segments in brighter areas than darker areas. It is not possible to obtain a unique interpretation of the GWPC scores since they are estimated locally for every census tract. For this reason, we summarize them in Figure 3.3, where each boxplot depicts the spatial distribution of the first GWPC loadings. The star represents the ordinary PC loadings for the first principal component. We can see that there is some sort of variability in the geographically weighted estimates of PC loadings, especially for `ST1` variable, i.e. the total number of foreign inhabitants. This spatial behaviour would not have been properly modelled using a classical PCA. Moreover, since the GWPC loadings for all census tracts are strictly greater than zero, we can conclude that high observed numbers of car crashes seem to be related with high values of Population variables. We obtain similar results for the other two groups of demographic variables. For further research, this procedure could also be used to estimate social indexes to link demographic variables to other urban aspects such as crime or ambulance dispatch.

The GWPC scores obtained by GWPCA pertain to the census tracks of Milan. This information is spatially misaligned with the street segments. For the regression model described in Section 3.3, we need to match the GWPCs with the segments of the road network. Hence, we overlaid the map of the census tracts (available as a shape file at the ISTAT website) on the spatial network and the PC score of the most overlapping tract polygon is finally assigned to each road segment. The three GWPCAs are named *GWPCA_pop*, *GWPCA_build*, *GWPCA_fam*, respectively, as summarised in Table 3.1.

3.5 Empirical Results

In this Section, the statistical model presented in Section 3.3 is adopted to analyse the frequency of car accidents occurring in Milan from 2015 to 2017. The covariates included in the regression model represent structural characteristics of the road network as well as sociodemographic dimensions. More specifically, the three groups of socio-demographic variables mentioned in Section 3.2 are summarised using a geographically weighted principal component analysis (see Section 3.4) and the first principal component of each group is retained in the model. As far as the structural variables are concerned, we consider the number of roads touching a street segment and two binary variables indicating the presence of traffic lights and pedestrian crossings in each street segment. We also include the OSM classification of each highway segment[7].

Since we have data collected for three consecutive years, $t = 2015, 2016, 2017$, on a network that is fixed over time, we check for the presence of serial dependence between the crash counts observed in two consecutive years. We display in Figure 3.4 two scatterplots showing

[7] Highway classifications: primary, secondary, tertiary, residential, unclassified.

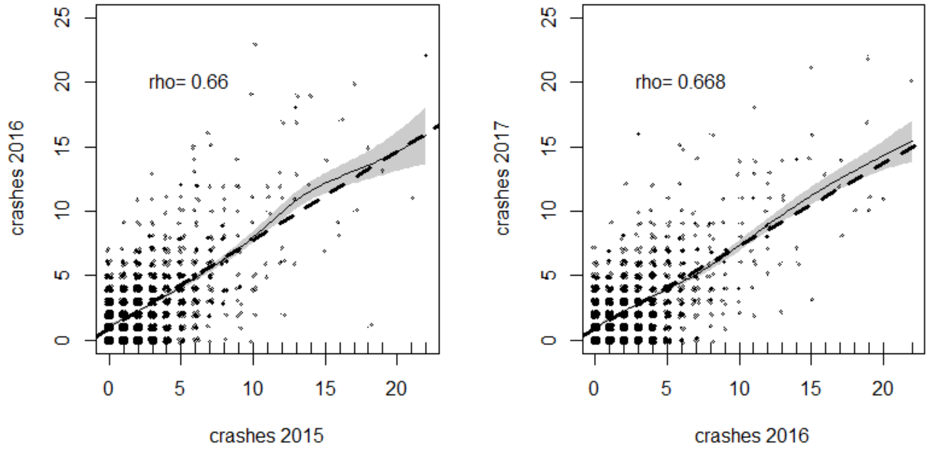


Figure 3.4: Scatterplots of the number of car crashes that occurred in each street segment in 2015 versus 2016 (left) and 2016 versus 2017 (right).

the relationship between the number of car accidents per segment that occurred during 2016 and 2017 and the counts of the same segments in the previous year. We can see a positive association between the frequency of car crashes in two subsequent years, which suggests that car crashes tend to reoccur (or not to occur) in those street segments where crashes occurred (or did not occur) one year earlier. We superimpose on the graph a cubic spline interpolation of the scatter plot with its confidence region (Wood, 2017). This interpolation shows that the relationship between the counts of two consecutive years is reasonably linear, the correlation coefficient[8] being as high as, roughly, 0.66 in both cases.

Hence, since the number of crashes that occurred in a given year is related to the number of crashes occurring in the same segment the year before, the lagged response variable, Y_{t-1}, is also included in the predictor set of the ZIP model. The models for counts and zeros are then specified as a dynamic lagged regression that writes as

$$\begin{cases} \log \theta_t = \beta_0 + \boldsymbol{\beta}_1' \boldsymbol{x} + \boldsymbol{\beta}_2' \boldsymbol{w} + \beta_3 y_{t-1} + \log \tau_c \\ \log \frac{\phi_t}{1-\phi_t} = \gamma_0 + \boldsymbol{\gamma}_1' \boldsymbol{x} + \boldsymbol{\gamma}_2' \boldsymbol{w} + \gamma_3 y_{t-1} + \log \tau_z \end{cases} \tag{3.1}$$

where $\boldsymbol{x}$ is the set of structural covariates of the street, $\boldsymbol{w}$ is the set of GWPC extracted by the three socio demographic macro dimensions in Table 3.1, $\boldsymbol{\beta} = (\boldsymbol{\beta}_1, \boldsymbol{\beta}_2, \boldsymbol{\beta}_3)$ and $\boldsymbol{\gamma} = (\boldsymbol{\gamma}_1, \boldsymbol{\gamma}_2, \boldsymbol{\gamma}_3)$ are unknown parameters to be estimated from the data, and $\log \tau_c$ and $\log \tau_z$

[8] We removed the coordinate $(0, 0)$ in the computation to reduce a polar effect due to the large number of observations in that position. If it was included, correlations would have been 0.6717, 0.6722 respectively.

are offsets for the count and zero component, respectively. The model in Equation (3.1) will be indicated by ZIP (1) in the rest of the chapter.

The results of the estimation procedure for the count and the zero model are reported in Table 3.2, respectively in the third and fifth column. The other columns of Table 3.2 will be explained later on in this Section.

The coefficient associated with the lagged counts in the Poisson component of model ZIP (1) is positive and highly significant, while it is negative and highly significant in the zero component. The coefficients of the logistic regression for the zero component represent the effect of each covariate included in the linear predictor on the log-odds of the probability that no crash occurs in a street segment. In particular, -1.231 (see last raw and last column of Table 3.2) is the decrease of the log-odds for a unit increase in the number of crashes in the previous year, keeping all other variables in the model fixed.

Being $(1-\phi)\theta$ the expected number of car crashes, we can also conclude that this value at time t tends to increase as the number of crashes at time $t-1$ gets larger since both the zero and the count components point towards the same direction. This result is consistent with our exploratory analysis.

The joint interpretation of the overall effect of each variable included in the ZIP model is, however, not easy (Agresti, 2015, Ch. 7), in particular when a covariate affects ϕ and θ in an opposite directions, as for *Pedestrian Crossing*. Following Lambert (1992), another way to address the interpretation problems of the ZIP model is to average the estimated means over all combination of covariates that share the same level of a factor and then compare the averages. Table 3.3 shows the marginal average of $(1-\hat{\phi})\hat{\theta}$ for the network covariates in model ZIP (1). Each value jointly considers both crash averages and the probability of extra zeros. It is found that *Primary highway* is riskier than the other road types whereas the *Residential Highway* is the safest street type. As it was somehow expected, the greater the number of touching segments, the greater the risk of crash. Streets with traffic lights or pedestrian crossings are riskier than roads without them.

To assess the fit of the ZIP model, we consider the Pearson residuals, which are defined by $\hat{r}_i = \frac{y_i-\hat{\mu}_i}{\sqrt{\hat{\omega}_i}}$, where $\hat{\mu}_i = (1-\hat{\phi}_i)\hat{\theta}_i$ and $\hat{\omega}_i = \hat{\mu}_i(1+\hat{\phi}\hat{\theta})$ (Cameron and Trivedi, 2013). The residuals range from -5.925 to 16.205, their mean is -0.022, the median -0.240 and the variance 0.8364, whereas 5th and 95th percentile are -0.679 and 1.818 respectively. These values suggest that, on average, the expected values and the observed counts are close to each other and more than 90% of total residuals in modulus are smaller than 2.

We checked our decision to adopt a ZIP model that includes all variables both in the count and in the zero component comparing the estimated coefficients of ZIP (1) with two other models: a) standard GLM loglinear Poisson model and b) ZIP model, named $\text{ZIP}_{\text{const}}$, which is equal to ZIP (1) for the count component but it includes no covariates in the zero component. The estimates of the GLM model are reported in the first column of Table 3.2, while the estimates of the $\text{ZIP}_{\text{const}}$ model are reported in the second and fourth columns. All three models adopted the length of each street segment as an offset. We can see that

Table 3.2: Estimates of Poisson GLM, ZIP$_{\text{const}}$, and ZIP (1). Dependent variable is the number of car crashes per segment.

	Poisson-GLM	ZIP$_{\text{const}}$	ZIP (1)	ZIP$_{\text{const}}$	ZIP (1)
	Count component			Zero component	
Constant	-6.209***	-5.259***	-5.132***	0.268***	-3.344***
	(0.034)	(0.039)	(0.042)	(0.023)	(0.112)
Unclassified highway	-0.328***	-0.259***	-0.199***		0.239**
	(0.029)	(0.031)	(0.035)		(0.098)
Tertiary highway	0.144***	0.064**	0.034		-0.044
	(0.028)	(0.030)	(0.032)		(0.099)
Secondary highway	0.174***	0.179***	0.212***		0.322***
	(0.030)	(0.031)	(0.034)		(0.106)
Residential highway	-0.437***	-0.370***	-0.294***		0.152
	(0.028)	(0.031)	(0.035)		(0.094)
Number of touching segments	0.188***	0.179***	0.180***		0.096***
	(0.007)	(0.007)	(0.008)		(0.021)
Pedestrian crossings	-0.118***	-0.250***	-0.481***		-1.320***
	(0.018)	(0.019)	(0.023)		(0.054)
Traffic signals	0.211***	0.190***	0.220***		0.193**
	(0.023)	(0.025)	(0.027)		(0.083)
$GWPCA\_pop$	-0.052***	-0.059***	-0.052***		-0.001
	(0.008)	(0.009)	(0.011)		(0.026)
$GWPCA\_build$	0.021***	0.007	-0.014**		-0.086***
	(0.004)	(0.005)	(0.006)		(0.016)
$GWPCA\_fam$	0.107***	0.111***	0.101***		-0.009
	(0.012)	(0.014)	(0.017)		(0.038)
Y_{t-1}	0.234***	0.175***	0.170***		-1.231***
	(0.002)	(0.002)	(0.002)		(0.038)
Log likelihood	-36762.01	-34380.84	-33008.36		
Δ Log likelihood		2381.17	1372.48		

* $p < 0.1$; ** $p<0.05$; *** $p < 0.01$

Table 3.3: Marginal average of the response for each network covariate.

Covariate	Level	Marginal Average
Highway	Residential highway	0.233
	Unclassified highway	0.224
	Tertiary highway	0.411
	Secondary highway	0.505
	Primary highway	0.575
Traffic signals	Absence - 0	0.280
	Presence - 1	0.616
Pedestrian crossings	Absence - 0	0.210
	Presence - 1	0.416
Number of touching segments	0	0.109
	1	0.123
	2	0.206
	3	0.212
	4	0.261
	≥ 5	0.469

the estimates of the coefficients for the count model are found stable and, with the only exception of *GWPCA_build*, we do not observe changes in the sign of the estimates.
At the bottom of Table 3.2, we report the Δ log likelihood, i.e. the additional amount of likelihood when passing from GLM loglinear to ZIP$_{\text{const}}$ model and from ZIP$_{\text{const}}$ to ZIP (1) model. We can conclude that a ZIP regression is more appropriate to model the distribution of the car crash counts with respect to simpler models (such as GLM or ZIP$_{\text{const}}$).
We can see from Table 3.2 that almost all the estimates of coefficients are significantly different from zero. To further support this result, we adopt a statistical learning technique to investigate the complexity of the model (i.e. to select relevant variable) and possibly improve the interpretation. More specifically, we use the LASSO methodology for zero inflated data (P. Banerjee et al., 2018; Z. Wang, S. Ma, and C.-Y. Wang, 2015). The model estimates are obtained by solving the penalized likelihood problem

$$\underset{\lambda_1,\ \lambda_2}{\operatorname{argmin}} -L(\boldsymbol{\beta}, \boldsymbol{\gamma}) + \lambda_1 \sum_{j=1}^{m} |\beta_j| + \lambda_2 \sum_{r=1}^{q} |\gamma_r|,$$

where λ_1 and λ_2 are penalty coefficients for the count and zero components. This strategy aims at shrinking towards zero the coefficients of the covariates that are supposed to be not significant in a validation perspective. Figure 3.5 shows the trajectory of the estimates of the count component as the shrinkage λ_1 increases.
From Figure 3.5, only Y_{t-1} is persistently far from zero as $\log \lambda_1 \to 0$ (i.e. as $\lambda_1 \to 1$),

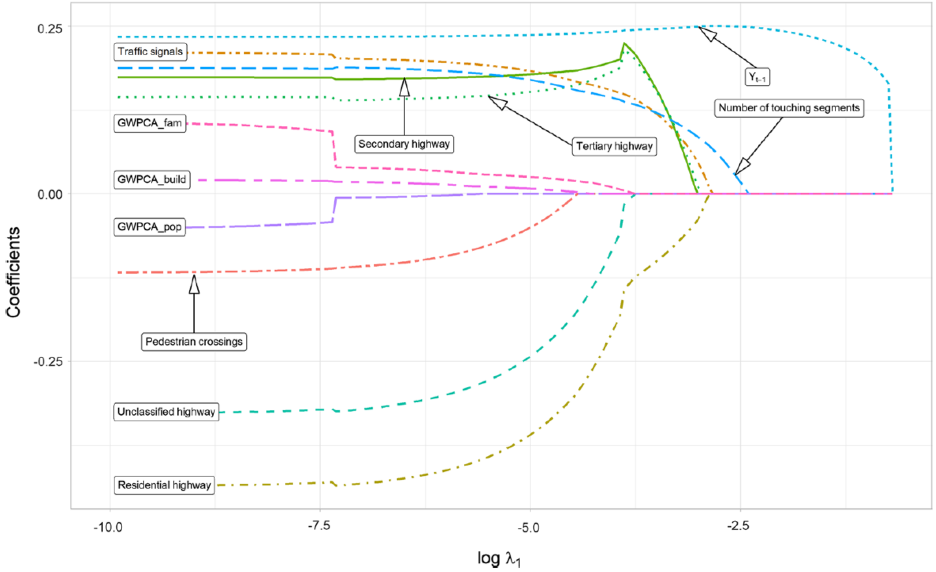
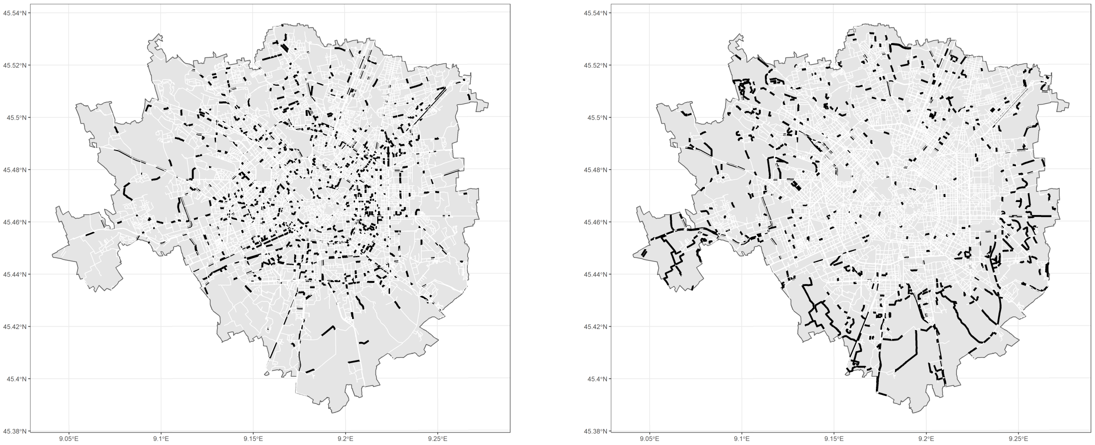


Figure 3.5: Lasso estimates of the count coefficients by varying λ_1 (on log scale); on the left of the picture the tiniest shrinkage, on the right the uppermost shrinkage

while the variables that quickly shrink towards zero as the penalty coefficient increases are the principal components related to population, the number of buildings and the number of families. This result further reinforces the thesis that one of the most striking elements that impacts on crashes are the crash history of a segment, the structure of the network and, to a lesser extent, the density of buildings, population, and families.

3.6 Discussion

In this Chapter, we showed how to monitor the accident exposure in geographical space using geocoded data taken from open archives and a dynamic zero inflated Poisson regression model. This approach allows us to estimate two parameters, θ and ϕ, for every segment of the network. The first estimated quantity, $\hat{\theta}$, represents a *road risk index*: the greater the value of $\hat{\theta}$, the riskier the road segment is, keeping the other parameter fixed. The estimate of the second quantity, $\hat{\phi}$, represents the probability that no car accident occurs in a year for a given road. We consider it as a *road safety index*: the higher the value of $\hat{\phi}$, the safer the street segment is, keeping the value of the road risk index as fixed. It should be pointed out that the interpretation of the road safety index is much easier than its risk counterpart since $0 \leq \phi \leq 1$ while $\theta > 0$.
We represent in Figure 3.6 the results of the estimation of the two indices for the road network of Milan during 2017. In Figure 3.6a we highlight those segments where the



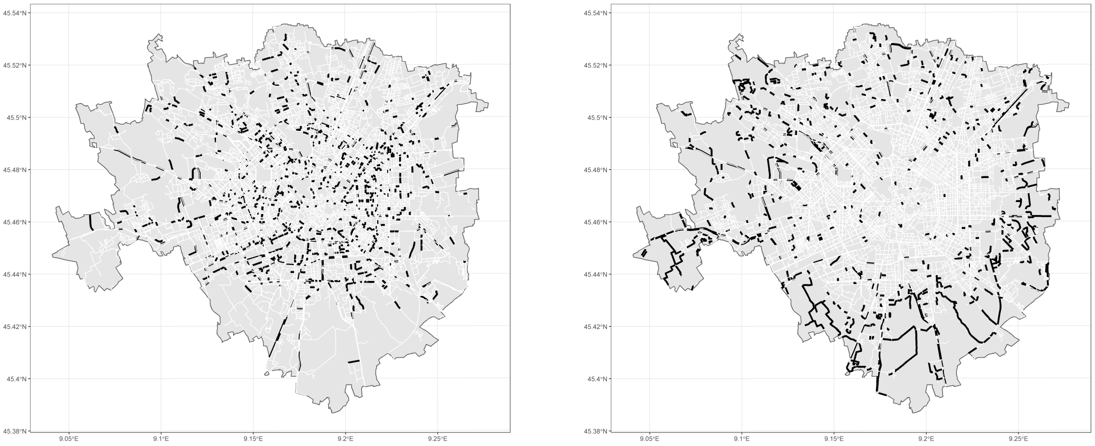
(a) Road Risk Index

(b) Road Safety Index

Figure 3.6: *Left: Road Risk Index.* We represented in solid black the segments where $\hat{\theta} > 1$, while the other segments are in depicted in white. *Right: Road Safety Index.* We represented in solid black the segments where $\hat{\phi} > 0.9$, while the other segments are in depicted in white.

estimated *risk index* is greater than 1, i.e. those street segments that are more exposed to crash occurrences. It should be pointed out that the segments composing our road network are very small (their mean length is approximately 70 m and the median length is slightly less than 50 m), which implies that more than one car crash in one year represents an extremely severe condition. In Figure 3.6b, we highlight those segments whose *safety index* is greater than 0.9. These are the segments where the estimated probability of no car accident during 2017 is greater than 0.9. It turns out from the maps that the safer segments are mostly located at the boundary of the metropolitan area and the areas with a large risk are concentrated around the centre of Milan.

To test the predictive performance of the model, we implemented the following cross-validated procedure. First, we calculated the correlation coefficient between the predicted values $(1 - \hat{\phi})\hat{\theta}$ and the observed number of events for every segment in the road network in 2017. The correlation coefficient was found as high as 0.46. Then, in order to obtain a more robust estimate of the predictive performances of our model, we repeated the same procedure 100 times dividing the dataset into two subsets, which were used to test and train the model. In each replicate, we trained the model using 80% of road segments (chosen at random), we calculated the two estimates $\hat{\phi}$ and $\hat{\theta}$, and the correlation coefficient between $(1 - \hat{\phi})\hat{\theta}$ and y on the remaining 20% of segments. The simulated 95% confidence interval



(a) Road Risk Index

(b) Road Safety Index

Figure 3.7: We highlight in solid back those segments where the road risk index is greater than 1 (left) or the road safety index is greater than 0.9 (right).

ranges within $(0.38, 0.55)$, with the median value equals to 0.46, suggesting that our model is reasonably robust to overfitting.

Exploiting the longitudinal component of the ZIP (1) model and assuming the road network and its structural characteristics as fixed, we can predict the road safety index and the road risk index for every segment in the network for 2018 using the same procedure detailed above. The maps in Figure 3.7 show the result. Given that we are dealing with a high complex network with tens of thousands of nodes and edges, Figure 3.7 does not apparently show any strong difference from Figure 3.6. For this reason, we highlight in Figure 3.8 only those segments where

$$[(1-\hat{\phi})\hat{\theta}]_{2018} - [(1-\hat{\phi})\hat{\theta}]_{2017} > 0 \text{ such that } [(1-\hat{\phi})\hat{\theta}]_{2017} \geq 1$$

that is, where we estimate an increase in the expected number of car crashes. This procedure returns a map of segments that should be monitored since we predict they will be more dangerous in 2018 with respect to the previous year.

To conclude, we summarise the results obtained in this Chapter:

- We produced an analysis of the determinants of crash frequencies in an extended and heterogeneous urban area. This was done using open data sources, which implies that a similar procedure can be easily replicated for any other city with similar information. The whole algorithm could also be enriched by the use of proprietary data, if available;

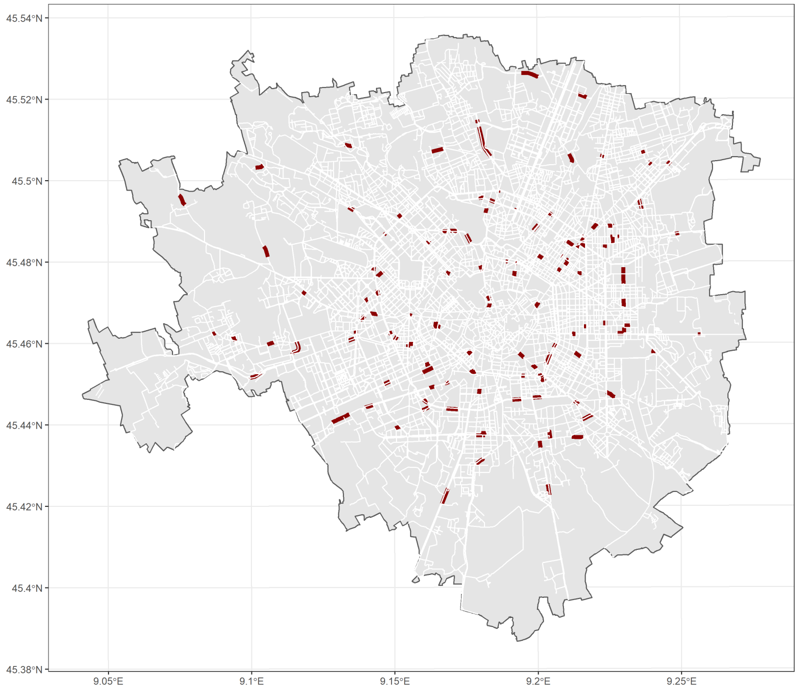


Figure 3.8: We highlight in solid red those segments where the road risk index is expected to increase when passing from 2017 to 2018.

- By spreading socio-demographic features over the territory of interest using geographically weighted principal components, we were able to compare the effect of the structural characteristics of the road network and socio-demographical aspects, and assess which components are more relevant in determining car accident frequency;
- The extremely detailed road network downloaded from OpenStreetMap allowed us to estimate two indexes that provide local information of the street riskiness at the segment level, which is, by far, more interesting than aggregated measures of the proneness to car crashes calculated at areal level (such as municipalities or postal codes).

Concerning the later point, in actuarial studies, the so-called *frequency of accidents*, defined by the ratio between the number of accidents and the total exposure in an area, is customarily used as an index of riskiness. In the present context, the frequency of accidents can be reformulated as the ratio between the number of accidents, *#accidents*, and the number of segments, *#roads*, representing the actual exposure to the risk of the network. This index can be decomposed as

$$\frac{\#\text{accidents}}{\#\text{roads}} = \frac{\#\text{accidents}}{\#\text{roads with} \geq 1 \text{ accident}} \frac{\#\text{roads with} \geq 1 \text{ accident}}{\#\text{roads}}$$

The second term of the product above represents a measure of repeatability, i.e. how many road segments had accidents given that at least one accident actually occurred. It might be interesting to assess the impact of road characteristics on crash frequency, and refit the model of crash occurrences conditional to the roads with at least one accident. Following this route, the zero inflated model is no longer appropriate, and other approaches, such as a truncated Poisson or hurdle models, would be more suitable instead (Agresti, 2015). This way to proceed would provide different tools to monitor riskiness of the road network. However, these approaches are beyond the scope of the present manuscript and are not discussed here.

The computational effort required for this kind of analysis is not negligible. GWPCA or LASSO for count data are the most time demanding tasks and took several hours to run on our laptops. Retrieving data from the open source database and spatially aligning the information are other challenging tasks. However, the computational burden is somehow a minor cost if one considers the high complexity of the adopted network and the detailed information this approach actually provides.

Improvements are obviously possible. Most of them depend on the availability of further information and variables i.e. traffic density, driver behaviour, local well-being indices, economic factors and so on. Model extensions are also possible e.g. the possibility to account in the model for potential dependencies in the network components or for prior information on crash causes. The model presented in Chapter 4 is much more complex and flexible, albeit it considers a simpler and smaller road network (due to the computational costs of the Bayesian models). Notwithstanding, we believe that the approach suggested in this Chapter is relevant, and can contribute to the definition of a methodology to monitor road safety and improve the quality of life of citizens.

Supplementary Materials

We created a Shiny App (Appelhans, 2020; Chang et al., 2020) to better display the results of our model. It shows an interactive map of the street network of Milan where every segment is coloured according to its classification as a risky or safe segment. You can check the app and browse the code behind it at the following links: app and repository.

Chapter 4

Multivariate hierarchical analysis of car crashes data considering a spatial network lattice

> *People can come up with statistics to prove anything, Kent. Forty percent of people know that!*
>
> Homer Simpson, Homer the Vigilante (5x11)

Based on: *Gilardi, A., Mateu, J., Borgoni, R. and Lovelace, R., 2020. Multivariate hierarchical analysis of car crashes data considering a spatial network lattice. arXiv preprint.* URL: https://arxiv.org/abs/2011.12595

4.1 Introduction

Road casualties have been described as a global epidemic (MacKay, 1972; Nantulya and Reich, 2002), representing the leading cause of death among young people worldwide. Car crashes and other types of collisions are responsible for more than 1 million deaths each year (1250000 in 2015, 17 deaths per 100000 people), as reported by World Health Organization (2018). In high income nations, such as the United Kingdom (UK), the roads are safer than the global average, but car crashes are still the cause of untold suffering. According to the statistics published by the UK's Department for Transport (DfT) in the *Annual report on Road Casualties in Great Britain* (Department for Transport, 2020), 153158 road traffic collisions resulting in casualties were recorded in 2019, 5% lower than 2018 and the lowest level since records began, in 1979.

Nevertheless, the DfT estimates that approximately 33648 people were killed or seriously injured (KSI) in 2019[1], and while this number is slightly lower than in 2018, the decaying

[1] The methodology for classifying the severity level of a car crash has been modified starting from 2016, adopting the injury-based systems called *CRASH* and *COPA* (Braunholtz and Elliott, 2019). All police forces are gradually adopting these new reporting systems in England, and the Office for National Statistics (ONS) developed a logistic regression model to compare the severity levels between different years and classification systems. The data used in this application have been adjusted using the procedures developed by ONS (Department for Transport, 2020, pp. 38–41).

rate has been getting lower and lower starting from 2010. These figures are worrisome considering that car occupant fatality rates are particularly high in the 17-24 age band (Department for Transport, 2020, p. 17).

To tackle the flattening trend in the KSI rate over the past decades, a range of interventions are needed, and analytical approaches can help prioritise them. This Chapter presents a statistical model to identify street sections with anomalously high car crashes rates, and to support police responses and cost-effective investments in traffic calming measures (PACTS, 2020).

Statistical models of road crashes have become more advanced over time. In the 1990s, models tended to consider only the discrete and heterogeneous nature of the data (Miaou, 1994; Miaou and Lum, 1993; Shankar, F. Mannering, and Barfield, 1995), omitting spatial characteristics. More recent statistical models of crash data include consideration of crash location in two-dimensional space, with three main advantages for road safety research (El-Basyouny and Sayed, 2009). First, consideration of space allows estimating appropriate measures of risk (such as expected counts, rates or probabilities) at different levels of resolution and the subsequent ranking of geographical areas to support local interventions. Second, spatial dependence can be a surrogate for unknown, potentially unmeasured (or unmeasurable) covariates; adjusting for geographic location can reduce model misspecification (N. A. Cressie, 1993; Dubin, 1988). Third, the spatial dimension can be used to take advantage of autocorrelation in the relevant variables, borrowing strength from neighbouring sites and improving model parameter estimation.

Road crash datasets, which are typically available on a single accident basis, can be spatially aggregated in two main ways: administrative zones (such as cantons, census wards, or regions) or street network features (either as contiguous segments or divided into corridors and intersections). In both cases, the spatial support is a lattice, i.e. a countable collection of geometrical units (polygons or lines, respectively), possibly supplemented by a neighbourhood structure. Several papers addressed the statistical modelling of crash frequencies at the areal level, based on available zoning systems in the study region (e.g. Aguero-Valverde and Jovanis (2006), Boulieri et al. (2017), Miaou, Song, and Mallick (2003), and Noland and Quddus (2004)). The second approach has gained in popularity in recent years, with a number of papers analysing road crash events aggregated to the street level (e.g. Aguero-Valverde and Jovanis (2008), Miaou and Song (2005), and C. Wang, Quddus, and Ison (2009)). We refer to the following review papers for more details: Lord and F. Mannering (2010), Savolainen et al. (2011), and Ziakopoulos and Yannis (2020). In reference to street level data, we note that there has been a recent surge of research for spatial point patterns living on networks (Baddeley, Nair, et al., 2020; Cronie, M. Moradi, and Mateu, 2020; Rakshit, Davies, et al., 2019). The statistical model introduced in Chapter 5 represents a relevant example.

Both zone and network level approaches have advantages, notably computational requirements for the former and spatially disaggregated results for the latter. Given that computational resources are less of a constraint in the 2020s than they were in previous decades, and the fact that it is the nature of roads (not zones) that is responsible for crashes, we argue that road segments are the more appropriate aggregation units for the analysis of road crash data. Network analysis can be used to bring attention to specific segments, and, for these reasons, the models presented in the next sections were developed considering a network lattice.

Aggregation, e.g. number of crashes per road segment, enables comparison between different road segments. However, spatial aggregation also leads to a well-known problem in geographical analysis, the Modifiable Areal Unit Problem (MAUP), firstly described in Openshaw (1981): the size of the spatial units impacts on the statistical analysis, influencing, and possibly biasing, modelling choices and results. Hence, conclusions drawn at one scale of spatial aggregation might not necessarily hold at another scale or be somehow different. The MAUP has been mainly ignored in the road safety literature and, as reported by Xu, H. Huang, and Dong (2018) and Ziakopoulos and Yannis (2020, p. 21), it is mentioned only in a handful of recent papers (Abdel-Aty et al., 2013; Briz-Redón, Martínez-Ruiz, and Montes, 2019b; Ukkusuri et al., 2012; Zhai et al., 2019), which explore the impact of changing the areal zoning system (e.g. TAZ, block groups and census tracts) on parameter estimates, significance and hotspot detection. Only one early paper (Thomas, 1996) could be found exploring the impacts of MAUP on road crash data at the network lattice level, albeit only in terms of summary statistics of aggregated counts. To assess the MAUP effect on network data modelling, we employed an algorithm to modify the structure of a road network, merging contiguous segments in the same corridor and preserving the geometrical properties of the network (Padgham, 2019). This is analogous to the *contraction* procedures introduced in Chapter 2. Then, we compared the results obtained with the two different network configurations. To the best of our knowledge, this is the first attempt at exploring and estimating the presence and the magnitude of MAUP in statistical models that consider a network lattice.

Finally, we note that systems of collision classification present a multivariate nature (Braunholtz and Elliott, 2019; Kirk, Cavalli, and Brazil, 2020). The occurrences of different severity degrees can be correlated to each other, and their spatial dynamics can be potentially interdependent. Hence, it is necessary to account for correlations between crashes counts at different levels of severity. We consider two types of accidents: *slight* and *severe*. The severe class is very sparse in the dataset at hand, hence modelling both types of accidents simultaneously allows to borrow strength from the existing correlations and improves estimates.

Following ideas introduced in Barua, El-Basyouny, and Islam (2014), we consider a range of competing models, developed in a full hierarchical Bayesian paradigm. This approach allows one to encompass complex structures of spatial dependence in a quite natural way. Spatially structured random effects are defined using both Intrinsic Multivariate Conditional Autoregressive (IMCAR) and Proper Multivariate Conditional Auto-regressive (PMCAR) priors (Besag, 1974; Mardia, 1988; Martínez-Beneito and Paloma Botella-Rocamora, 2019; Palmi-Perales, Gomez-Rubio, and Miguel A. Martinez-Beneito, 2019). We also propose to consider a generalisation of PMCAR models unexplored so far in the road safety literature, and firstly defined for polygonal lattice data in Gelfand and Vounatsou (2003).

The case study is the metropolitan area of Leeds (population 800000) in North England. We accessed Ordnance Survey data on major roads (3661 segments, total length 450 km), creating a spatial network substantially larger than previous studies, many of which report findings on only a few roads. The model presented in Chapter 3 represents a notable exception albeit with a simpler statistical and spatial structure. We present results for an entire metropolitan area, approximating more closely the level at which road policing activities and investment in road safety interventions are prioritised. The scale of the case study presented several computational challenges, and, in terms of Bayesian parameter estimation, we used the computationally efficient Integrated Nested Laplace Approximation (INLA) approach instead of Markov chain Monte Carlo (MCMC) sampling Lindgren, Håvard Rue, et al., 2015; Håvard Rue, Riebler, et al., 2017.

The rest of the Chapter is organised as follows. In Section 4.2, the data sources are described. In Section 4.3, the statistical methodology adopted in this Chapter is discussed in detail. In Section 4.4, the main results of the Chapter are presented whereas, model criticism and further model discussion, such as MAUP analysis, are provided in Section 4.5. Conclusions, in Section 4.6, end the Chapter.

4.2 Data

The datasets analysed in this Chapter came from several different sources and required a number of preprocessing steps before they could be made into a structure suitable for a statistical analysis. The study region was defined as the Middle Super Output Area (MSOA) zones within the local authority of Leeds. The City of Leeds was selected because it is a car-dependent city with a large network of major roads that approach the city centre (the city was dubbed the *motorway city of the 70s*) and would therefore be expected to be a place where road safety could be improved. Leeds is part of West Yorkshire and accounts for approximately 40% of all car crashes in the region. Origin-destination data from the 2011 UK Census were used to estimate traffic volumes, to provide an estimate of exposure, with traffic volumes used as the denominator in the statistical models presented in Section 4.3. The road network was obtained from Ordnance Survey, covering all major roads in Leeds. We matched the network and the MSOAs using an overlay operation. Finally, we associated all car crashes that occurred in the city of Leeds from 2011 to 2018 with the nearest point on the road network, counting the occurrences in each street segment. The previous two steps are analogous to the procedure detailed in Chapter 3.

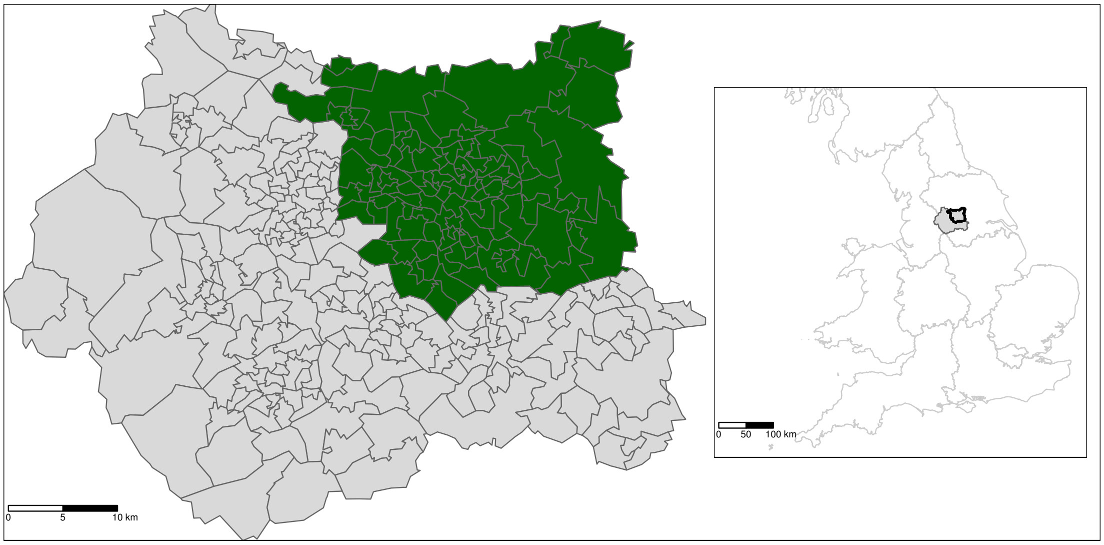


Figure 4.1: The grey polygons show the MSOAs in West-Yorkshire region, while the dark-green area highlights the city of Leeds. The inset map locates the position of the study-area with respect to England.

MSOA zones

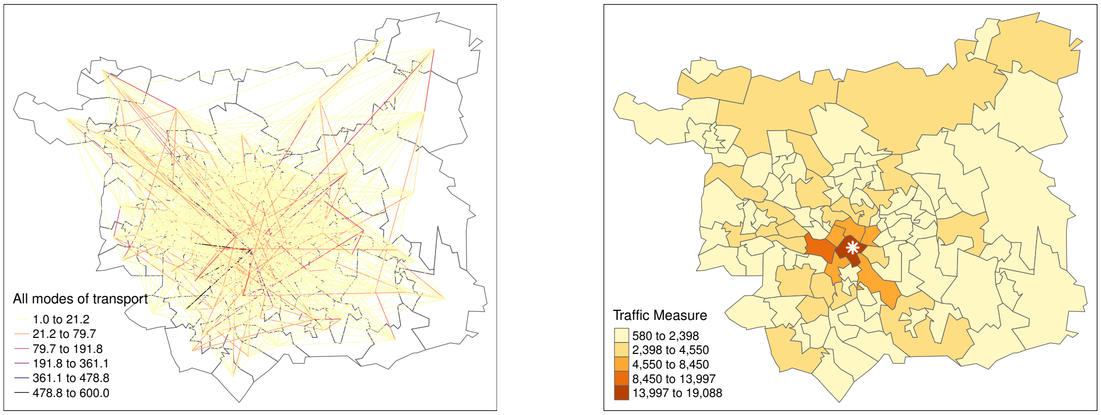
There are 6791 MSOAs in England, 299 of which belong to the West-Yorkshire region. These were accessed from the github-page[2] of Propensity Cycle Tool (Lovelace, Goodman, et al., 2017), focussing, in particular, on the City of Leeds (107 areas). The MSOAs represent the starting point for all the following steps, and they are mapped in Figure 4.1 as grey polygons for the West-Yorkshire, and as dark-green polygons for the City of Leeds. The inset map is used to locate the study-area in the British territory.

Traffic flow

The *traffic flow* data represent the *commuting journeys* from home to workplace using several modes of transport, such as train, bus, bike and motorcycle. The data were collected during the 2011 Census at the individual level, and then aggregated at the MSOA level. The UK Data Service shares the flow data through the *WICID* interface as cross-tables reporting the flows between all pairs of a predefined set of MSOAs (UK Data Service Census Support, 2014). We considered the commuting flows in the region of Leeds for all possible modes of transport. Figure 4.2a shows a random sample of 1000 *traffic flows*[3] between the centroids

[2] Last access on 06/2020.

[3] We downloaded 10536 traffic flows, which is slightly less than $107^2 = 11449$ since there are no commuters between certain MSOAs.



(a) Random sample of one thousand Origin-Destination data representing the daily flows between MSOAs in Leeds.

(b) Estimates of the new traffic measure. The white star represents Leeds City Centre.

Figure 4.2: Raw and modified traffic flows in the area of Leeds. The map on the right highlights several contiguous MSOAs that correspond to the arterial thoroughfares that are used to reach the City Centre.

of the MSOAs in Leeds, coloured according to the number of daily commuters.
Raw WICID data, however, ignore that people may travel to their workplace through several MSOAs. For this reason, we calculated a new traffic measure using the following procedure. Starting from the MSOAs, we defined a graph where the vertices are the centroids of each area, and the edges connect neighbouring areas. Then, we estimated the shortest path for all commuting journeys downloaded from WICID and assigned to each MSOA a value that is equal to the sum of all raw traffic measures going through the area. These values represent the new traffic measures and are displayed in Figure 4.2b. A similar approach was also adopted in Boulieri et al. (2017), and we refer to the references therein for more details. The raw data flows and the MSOAs polygons were downloaded using the R package *pct* (Lovelace, Goodman, et al., 2017).

Road network

The *road network* was built using data downloaded from Ordnance Survey (OS)[4], an agency that provides digital maps and other services for location-based products (Ordnance Survey, 2020). We downloaded the *Vector OpenMap Local* data for the *SE* region[5], selected the *Roads* and *Tunnels* layers, and filtered the streets that belong to the City of Leeds.

[4] Last access on 21-05-2020

[5] The United Kingdom has been divided into several squares of approximately $100km^2$. The SE region is an area centred around Leeds. See here for a list of maps displaying all OS areas.

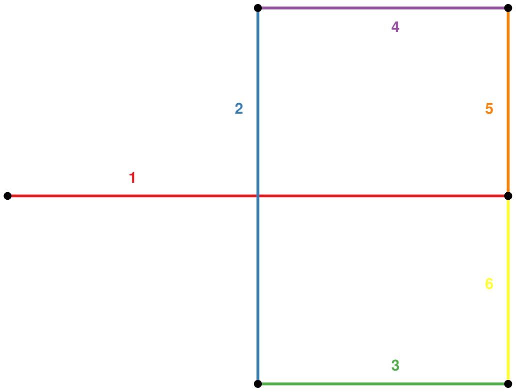
Ordnance Survey represents all the streets of a road network as the union of a finite set of segments, and it includes additional fields such as the road name or the street type. These segments represent the elementary units for the statistical analysis described in Section 4.3. The road network downloaded from OS is composed of approximately 50000 segments. The OS network was simplified using the following procedure. We first selected only the major roads, such as the *motorways*, *primary roads* and *A roads*. They represent less than 10% of the total road network, but more than 50% of all car crashes registered during 2011-2018 occurred in their proximity. The output of this procedure is a road network composed by a big cluster of connected streets, displayed in Figure 4.4a, and several small isolated groups of road segments (which are also called *islands*), created by the exclusion of their links to the other roads.

These small clusters can be problematic from a modelling perspective since they produce a not-fully-connected network (see also Freni-Sterrantino, Ventrucci, and Håvard Rue (2018) and Hodges, Carlin, and Fan (2003), and the properties of ICAR and MICAR distributions explained in Section 4.3), so we implemented an algorithm to further simplify the road network and remove them. This algorithm is based on the dual representation of a road network as a geographical entity, composed by points and lines, and a graph object, with nodes and edges (S. Marshall et al., 2018; Porta, Crucitti, and Latora, 2006).

More precisely, we created a graph whose vertices correspond to the street segments of the road network, and we defined an edge for each pair of spatial units sharing a point at their boundaries. This graph uniquely determines a (sparse) adjacency matrix amongst the spatial units (i.e. the road segments), that summarises the graph dimension of the road network. We sketched a toy example in Figure 4.3, representing the idea behind the dual representation of a road network and the definition of the adjacency matrix. Then, starting from the graph and the adjacency matrix, we calculated its *components*[6], and we excluded all small clusters of road segments that did not belong to the biggest *component*. It should be stressed that this procedure creates a fully connected network, which has relevant consequences on the rank-deficiency problem of the ICAR models described in Section 4.3. This algorithm is just a simplified version of the functionalities presented in Chapter 2 and, in particular, Section 2.6.1.

In the end, the road network is composed of approximately 3600 units, and it is shown in Figure 4.4a, where the segments are coloured according to their road types. Moreover, since the street network and the MSOAs are spatially misaligned, they were matched using an overlay operation: each road segment was assigned to the MSOA that intersects the largest fraction of the segment. This procedure allows us to assign a traffic estimate to each road segment, which will be used as an exposure parameter in the statistical models considered below.

[6] A graph is said to be *connected* if there exist a sequence of edges going from each vertex to every other vertex. The *components* of a graph are sub-graphs formed by all connected vertices Kolaczyk and Csárdi, 2014.



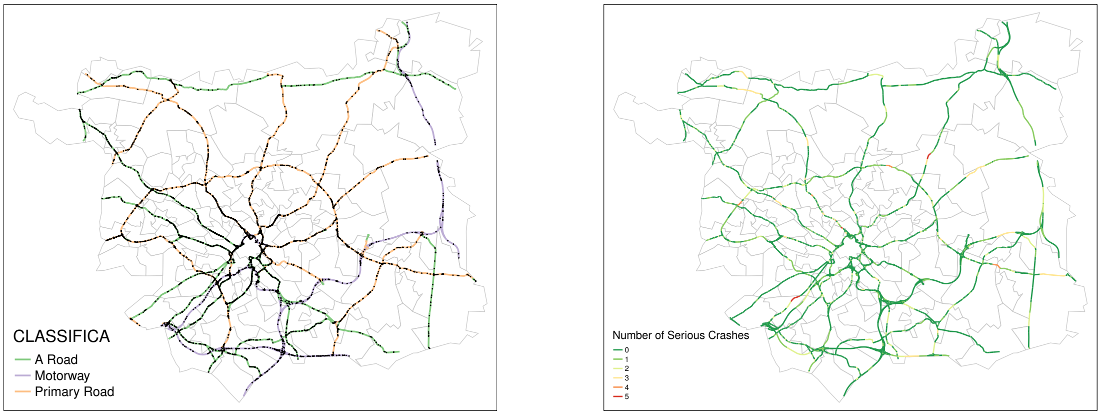
$$\begin{array}{c|cccccc} & 1 & 2 & 3 & 4 & 5 & 6 \\ \hline 1 & \cdot & \cdot & \cdot & \cdot & 1 & 1 \\ 2 & \cdot & \cdot & 1 & 1 & \cdot & \cdot \\ 3 & \cdot & 1 & \cdot & \cdot & \cdot & 1 \\ 4 & \cdot & 1 & \cdot & \cdot & 1 & \cdot \\ 5 & 1 & \cdot & \cdot & 1 & \cdot & 1 \\ 6 & 1 & 1 & 1 & \cdot & 1 & \cdot \end{array}$$

Figure 4.3: Graphical example showing the dual nature of a road network. *Left*: map showing the geographical dimension. Each segment is coloured and labelled using a different ID and colour respectively. *Right*: adjacency matrix of the graph associated with the road network. Each vertex corresponds to a segment whereas an edge connects two vertices if they share one boundary point. For example, segments 1 and 2 are not neighbours since they do not share any point at the boundaries, even if they intersect each other. This situation may occur at bridges or overpasses.

Car crashes data

We analysed all *car crashes* that occurred between the 1st of January 2011 and the 31st of December 2018 in the MSOAs pertaining to the City of Leeds, which involved personal injuries, occurred on public roads and became known to the Police forces within thirty days of the occurrence. The data were downloaded from the UK's official road traffic casualty database, called *STATS19*, using the homonym R package (Lovelace, Morgan, et al., 2019). We excluded all car crashes that occurred farther than ten meters from the closest segment in the simplified road network since they are probably related to other streets. STATS19 data also report the severity level of each casualty using one of three possible categories: *fatal*, *serious* or *slight*[7]. We harmonised the severity levels for different years and police forces using the CRASH methodology (Braunholtz and Elliott, 2019; Department for Transport, 2020), and we decided to aggregate *serious* and *fatal* levels since *fatal* crashes represent approximately 1% of the total number of car accidents. Henceforth, we will refer to *serious* or *fatal* crashes as *severe* accidents. The final sample is composed of 5862 events, and they are reported as black dots in Figure 4.4a. Then, we projected all crashes to the nearest point on the road network, and we counted the number of *slight* or *severe* occurrences on all street segments. We decided to ignore the temporal dimension since

[7] An accident is classified as *fatal* if it involved a human casualty whose injuries caused his death less than thirty days after the accident; *severe* if at least one person was hospitalised after the accident or recorded a particular type of injury (like concussions or severe cuts), and *slight* in all the other cases.



(a) Road network and car crashes in Leeds.

(b) Choropleth map of *severe* counts.

Figure 4.4: The map on the left represents the road network in Leeds. Each segment is coloured according to its OS classification. The black dots represent the car crashes. On the right, we report a choropleth map displaying *severe* car crashes counts.

severe crashes counts present an extreme sparsity, with more than 80% of zero counts during 2011-2018. Moreover, 40% of all segments registered no car crashes during the study period, while another 40% reported two or more crashes. These numbers highlight a common temporal trend between the eight years, and we refer to Appendix B for a space-time representation. The map in Figure 4.4b shows the spatial distribution of *severe* crashes counts.

4.3 Statistical methodology

We first focus on the definition of a three-level hierarchical model structure, which is shared among all the alternative specifications considered below. Then, we introduce two baseline models that serve as benchmarks and starting points for the other specifications. Thereafter, five different extensions to the baseline models are introduced. Finally, some techniques used for model comparison are discussed. The common theme behind all the seven alternatives is the presence of spatially structured and unstructured multivariate random effects.

Let $Y_{ij},\ i = 1, \ldots, n,$ represent the number of car crashes that occurred in the i-th road segment with severity level $j,\ j = 1, \ldots, J.$ In this Chapter, we consider two possible severity levels, a car crash being either *severe*, $j = 1$, or *slight*, $j = 2$.

In the first stage of the hierarchy, we assume that

$$Y_{ij}|\lambda_{ij} \sim \text{Poisson}\left(E_i\lambda_{ij}\right), \tag{4.1}$$

where E_i is an exposure parameter and λ_{ij} represents the car crashes rate in the ith road segment for severity level j. At the second stage of the hierarchical model, a log-linear structure of λ_{ij} is specified. We assume that

$$\log\left(\lambda_{ij}\right) = \beta_{0j} + \sum_{m=1}^{M} \beta_{mj} X_{ijm} + \theta_{ij} + \phi_{ij}, \tag{4.2}$$

where β_{0j} represents a severity-specific intercept, the vector $\{\beta_{mj}\}_{m=1}^{M}$ is a set of coefficients, $(X_{ij1}, \ldots, X_{ijM})$ is a collection of M covariates, ϕ_{ij} is a spatially structured random effect and θ_{ij} represents a normally distributed error component. The third stage that completes the hierarchical model is the specification of prior and hyperprior distributions. We assigned a vague $N(0, 1000)$ prior to $\beta_{mj},\ m = 0, \ldots M$. The two random effects, namely θ_{ij} and ϕ_{ij}, represent the structured and unstructured spatial components and are defined differently in different models as discussed below. Hereafter, we follow the notation used in Martínez-Beneito and Paloma Botella-Rocamora (2019, Ch 4, 6 and 8).

4.3.1 Baseline models: independent spatial and unstructured effects

The two baseline models are defined considering multivariate spatial and unstructured random effects with independent components. More precisely, a bivariate Gaussian prior with independent components is assigned to $(\theta_{i1}, \theta_{i2})$ for both baseline models:

$$(\theta_{i1}, \theta_{i2}) \sim N_2\left(\mathbf{0}, \begin{bmatrix} \sigma^2_{\theta_1} & 0 \\ 0 & \sigma^2_{\theta_2} \end{bmatrix}\right), \qquad i = 1, \ldots, n. \tag{4.3}$$

We assigned a Gamma hyperprior with parameters 1 (shape) and 0.00005 (inverse scale) to the inverse of $\sigma^2_{\theta_1}$ and $\sigma^2_{\theta_2}$, i.e. the precisions.

The spatially structured term in the first baseline model was defined using an *Independent Intrinsic Multivariate Conditional Auto-regressive* (IIMCAR) prior, whereas, for the second model, we adopted an *Independent Proper Multivariate Conditional Auto-regressive* (IPMCAR) prior. The IIMCAR and IPMCAR distributions are briefly introduced hereafter, starting from their classical univariate counterparts, namely the ICAR and PCAR distributions.

Univariate spatial random effects are traditionally modelled using a prior that belongs to the family of *Conditional Auto-regressive* (CAR) distributions (Besag, 1974). Given a random vector $\boldsymbol{\phi} = (\phi_1, \ldots, \phi_n)$, the *Intrinsic Conditional Auto-Regressive* (ICAR) distribution, which is a particular case of the CAR family, is usually defined through a set of conditional distributions (Besag and Kooperberg, 1995):

$$\phi_i | \{\phi_{i'}, i' \in \partial_i\}; \sigma^2 \sim N\left(m_i^{-1} \sum_{i' \in \partial_i} \phi_{i'}, \frac{\sigma^2}{m_i}\right),\ i = 1, \ldots, n, \tag{4.4}$$

where ∂_i and m_i denote, respectively, the indices and the cardinality of the set of neighbours for spatial unit i. These quantities are defined through a sparse binary symmetric neighbourhood matrix $\boldsymbol{W}$ with dimensions $n \times n$, that summarises the spatial relationships in the region of study. We built it taking advantage of the dual representation of a road network as a spatial and a graph object (see Chapter 2 and Section 4.2). More precisely, $\boldsymbol{W}$ is the adjacency matrix of a graph whose vertices correspond to the street segments of the road network and the edges identify a shared point at the boundaries of two spatial units. This procedure defines a *First Order* neighbourhood matrix. *Second* and *Third Order* neighbourhood matrices are defined iteratively in the same way.

It is possible to prove that the prior defined by (4.4) suffers from rank-deficiency problems, that are usually fixed by imposing a set of sum-to-zero constraints on the vector $\boldsymbol{\phi}$, one for each *component* in the graph of the road network (Hodges, Carlin, and Fan, 2003). In this application, we deal with a fully connected road network (see the pre-processing procedures detailed in Section 4.2), so we always had to fix only one set of constraints.

The *Proper Conditional Auto-Regressive* (PCAR) distribution is another member of the CAR family and it is usually defined as follows:

$$\phi_i | \{\phi_{i'}, i' \in \partial_i\}; \sigma^2, \rho \sim N\left(\rho\left(m_i^{-1} \sum_{i' \in \partial_i} \phi_{i'}\right), \frac{\sigma^2}{m_i}\right), \; i = 1, \ldots, n, \tag{4.5}$$

where ∂_i and m_i are defined as for the ICAR distribution and ρ is a parameter controlling the strength of spatial dependence, usually called *spatial autoregression coefficient* (N. A. Cressie, 1993). It is possible to prove that the joint distribution defined by (4.5) is proper if $|\rho| < 1$, hence there is no need to set any sum-to-zero constraint in this case. The ICAR prior can be seen as a limit case of the PCAR distribution with $\rho \to 1$, analogously to the relationship between Auto-Regressive and Random-Walk models in time series models (Botella-Rocamora, Lopez-Quilez, and M. Martinez-Beneito, 2013).

The family of *Multivariate Conditional Auto-regressive* (MCAR) distributions was firstly introduced by Mardia (1988), extending the ideas of Besag (1974) to the multivariate case. Given a random matrix $\boldsymbol{\Phi} = (\phi_{ij})$, which is defined for $i = 1, \ldots, n$ units and $j = 1, \ldots, J$ levels, the *Intrinsic Multivariate Conditional Auto-regressive* (IMCAR) distribution is a particular case of the MCAR family, defined through a set of multivariate conditional distributions (Martínez-Beneito and Paloma Botella-Rocamora, 2019):

$$\boldsymbol{\Phi}_{i \cdot} | \operatorname{vec}\left(\boldsymbol{\Phi}_{-i \cdot}\right); \Omega \sim N_J\left(m_i^{-1} \sum_{i' \in \partial_i} \boldsymbol{\Phi}_{i' \cdot}^T; m_i^{-1} \Omega^{-1}\right). \tag{4.6}$$

The terms $\boldsymbol{\Phi}_{i \cdot}$ and $\boldsymbol{\Phi}_{-i \cdot}$ denote, respectively, the ith row of $\boldsymbol{\Phi}$ and the matrix obtained by excluding the ith row from $\boldsymbol{\Phi}$. The vec operator is used for row-binding the columns of a matrix, meaning that $\operatorname{vec}\left(\boldsymbol{\Phi}_{-i \cdot}\right) = \left(\boldsymbol{\Phi}_{-i1}^T, \ldots, \boldsymbol{\Phi}_{-iJ}^T\right)^T$. The elements m_i and ∂_i are defined as before, through the adjacency matrix $\boldsymbol{W}$ of the graph associated to the road network, and they represent the spatial dimension of the IMCAR distribution. The $J \times J$ precision matrix Ω is used to model the associations between pairs of levels in the same road segment i, and it acts as a multivariate extension of the parameter σ^2 in (4.4).

This distribution suffers from the same rank-deficiency problems as its univariate counterpart, which are usually solved by imposing appropriate sum-to-zero constraints. The number of restrictions is equal to the number of *components* in the graph of the road network times the number of levels in the multivariate setting. The pre-processing operations that we performed on the network data (see Section 4.2) imply that we always have to set only J sum-to-zero constraints.

The IIMCAR distribution is a particular case of (4.6), which is obtained by setting $\Omega^{-1} = \text{diag}(\sigma^2_{\phi_1}, \dots, \sigma^2_{\phi_J})$. More precisely, if we assume $J = 2$ as we do in this application, then IIMCAR is defined by the following set of multivariate conditional distributions:

$$\boldsymbol{\Phi}_{i\cdot} | \text{vec}\left(\boldsymbol{\Phi}_{-i\cdot}\right); \sigma^2_{\phi_1}, \sigma^2_{\phi_2} \sim N_2 \left(m_i^{-1} \sum_{i' \in \partial_i} \boldsymbol{\Phi}^T_{i'\cdot}; m_i^{-1} \begin{bmatrix} \sigma^2_{\phi_1} & 0 \\ 0 & \sigma^2_{\phi_2} \end{bmatrix} \right). \tag{4.7}$$

In equation (4.7) we are assuming independence between the 2 levels, and this implies that the IIMCAR distribution is equivalent to two independent ICAR distributions, one for each level.

Analogously to the univariate case, the *Proper Multivariate Conditional Auto-regressive* (PMCAR) distribution is a particular case of the MCAR family, characterised by the following set of multivariate conditional distributions:

$$\boldsymbol{\Phi}_{i\cdot} | \text{vec}\left(\boldsymbol{\Phi}_{-i\cdot}\right); \rho, \Omega \sim N_J \left(m_i^{-1} \rho \sum_{i' \in \partial_i} \boldsymbol{\Phi}^T_{i'\cdot}; m_i^{-1} \Omega^{-1} \right). \tag{4.8}$$

The strength of the spatial dependence is controlled by ρ (as for the univariate PCAR distribution) and all the other parameters are defined as before. It can be proved that the joint distribution defined by equation (4.8) is proper if $|\rho| < 1$, although we restricted ourself to $\rho \in (0, 1)$ to avoid some counter-intuitive behaviour of the PMCAR distribution (Miaou and Song, 2005; Wall, 2004).

The IPMCAR distribution is defined as a particular case of equation (4.8) with $\Omega^{-1} = \text{diag}(\sigma^2_{\phi_1}, \dots, \sigma^2_{\phi_J})$. More precisely, if we assume $J = 2$, then IPMCAR is defined through the following set of multivariate conditional distribution:

$$\boldsymbol{\Phi}_{i\cdot} | \text{vec}\left(\boldsymbol{\Phi}_{-i\cdot}\right); \rho, \sigma^2_{\phi_1}, \sigma^2_{\phi_2} \sim N_2 \left(m_i^{-1} \rho \sum_{i' \in \partial_i} \boldsymbol{\Phi}^T_{i'\cdot}; m_i^{-1} \begin{bmatrix} \sigma^2_{\phi_1} & 0 \\ 0 & \sigma^2_{\phi_2} \end{bmatrix} \right). \tag{4.9}$$

For the same reasoning as in equation (4.7), the IPMCAR distribution is equivalent to J independent PCAR distributions.

Now we can characterise the random effects for the two baseline models. The first model was defined by considering unstructured random effects with a bivariate independent Gaussian prior (4.3), and spatial random effects with an IIMCAR prior (4.7). The second one was defined analogously to the first baseline model, but assuming an IPMCAR distribution for the spatial random effects (4.9). These models assume independence between the two levels both in the spatial and unstructured components, so they were used as benchmarks. In the next sections, we will also refer to the two baseline models using, respectively, the codes (A) and (B). We assigned an improper prior to σ_1^2 and σ_2^2, the variances in Ω, defined on $\mathbb{R}^+$, and a Uniform$(0, 1)$ prior to ρ.

Hereafter we introduce three increasingly complex sets of extensions that generalise the baseline models. The first one is characterised by the removal of the independence assumption from the spatially structured random effects, whereas, in the second set of extensions, we also relax the independence assumption from the unstructured random effects. The third and last set of extensions is characterised by a generalisation of PMCAR distribution that introduces a separate spatial autoregression coefficient, ρ_j, for each level in $\mathbf{\Phi}$.

4.3.2 Model extensions

First set of extensions

Starting from the baselines, we defined two new models replacing the IIMCAR and IPMCAR priors with their non-independent multivariate counterparts, the generic IMCAR and PMCAR defined above. If we assume $J = 2$, then the variance-covariance matrix Ω^{-1} in (4.6) and (4.8) can be written as

$$\Omega^{-1} = \begin{bmatrix} \sigma_{\phi_1}^2 & \rho_\phi \sigma_{\phi_1} \sigma_{\phi_2} \\ \rho_\phi \sigma_{\phi_1} \sigma_{\phi_2} & \sigma_{\phi_2}^2 \end{bmatrix},$$

where $\sigma_{\phi_1}^2$ and $\sigma_{\phi_2}^2$ represent the conditional variances and ρ_ϕ represents the correlation coefficient between the two levels in the same spatial unit. These models represent a generalisation of the baselines since we are now taking into account the correlations between different levels in the same road segment. We will also refer to them using, respectively, the codes (C) and (D). Following Palmi-Perales, Gomez-Rubio, and Miguel A. Martinez-Beneito (2019), we assigned a Wishart hyperprior to Ω^{-1} with parameters 2 and $\boldsymbol{I}_2$, i.e. the identity matrix of size two. The prior distributions on the unstructured random effects were left unchanged with respect to the baselines.

Second set of extensions

In these models, the independence assumption of the spatially unstructured random effects is removed. More precisely, assuming $J = 2$, we assign a generic bivariate Gaussian prior to the unstructured random effects

$$(\theta_{i1}, \theta_{i2}) \sim N_2 \left(\mathbf{0}, \begin{bmatrix} \sigma^2_{\theta_1} & \rho_\theta \sigma_{\theta_1} \sigma_{\theta_2} \\ \rho_\theta \sigma_{\theta_1} \sigma_{\theta_2} & \sigma^2_{\theta_2} \end{bmatrix} \right).$$

Parameters $\sigma^2_{\theta_1}$ and $\sigma^2_{\theta_2}$ represent the marginal variances of the unstructured random error, whereas ρ_θ represents their correlation. These models will also be identified using the codes (E) and (F). We assigned a Wishart hyperprior to the variance-covariance matrix with parameters 2 and $\boldsymbol{I}_2$.

Third set of extensions

It is possible to prove (Mardia, 1988) that the set of conditional normal random variables defined by equation (4.8) induce a multivariate joint distribution for vec($\mathbf{\Phi}$) that can be written as

$$\text{vec}(\mathbf{\Phi}) \sim N_{nJ}(\mathbf{0}, [\mathbf{\Omega} \otimes (\boldsymbol{D} - \rho \boldsymbol{W})]^{-1}) \tag{4.10}$$

where $\boldsymbol{D}$ is a $n \times n$ diagonal matrix whose elements d_{ii} are equal to m_i, the number of neighbours for spatial unit i, and $\otimes$ denotes the Kronecker product. The precision matrix in Equation (4.10) is given by the Kronecker product of two quantities: $\mathbf{\Omega}$, that models the variability between the J levels in the same road segment i, and $(\boldsymbol{D} - \rho \boldsymbol{W})$, that models the spatial variability between different road segments for the same level j.

In the third set of extensions, following Gelfand and Vounatsou (2003), we generalised the PMCAR distribution, defining a model which introduces a different spatial autocorrelation coefficient ρ_j for each level in $\mathbf{\Phi}$. More precisely, we set

$$\text{vec}(\mathbf{\Phi}) \sim N_{nJ} \left(\mathbf{0}, \left[\text{bdiag}(\boldsymbol{R_1}, \dots, \boldsymbol{R_J})(\mathbf{\Omega} \otimes \boldsymbol{I}_n) \text{bdiag}\,(\boldsymbol{R_1}, \dots, \boldsymbol{R_J})^T \right]^{-1} \right) \tag{4.11}$$

where $\boldsymbol{R}_j, j = 1, \dots, J$ is the Cholesky decomposition of $\boldsymbol{D} - \rho_j \boldsymbol{W}$, i.e. $\boldsymbol{R}_j \boldsymbol{R}_j^T = \boldsymbol{D} - \rho_j \boldsymbol{W}$, and bdiag$(\boldsymbol{R}_1, \dots, \boldsymbol{R}_j)$ denotes a block-diagonal matrix with elements $\boldsymbol{R}_1, \dots, \boldsymbol{R}_J$ and dimension $nJ \times nJ$.

For example, assuming $J = 2$, the precision matrix in equation (4.11) can be written as

$$\begin{bmatrix} \boldsymbol{R_1} & 0 \\ 0 & \boldsymbol{R_2} \end{bmatrix} (\Omega \otimes \boldsymbol{I}_2) \begin{bmatrix} \boldsymbol{R_1} & 0 \\ 0 & \boldsymbol{R_2} \end{bmatrix}^T.$$

It is possible to prove that the joint distribution defined by equation (4.11) is proper if $|\rho_j| < 1 \; \forall j$ and, following the same ideas as before, we assigned a Wishart hyperprior to Ω^{-1} with parameters 2 and $\boldsymbol{I}_2$, and a Uniform$(0, 1)$ prior to ρ_1 and ρ_2. We will refer to this last model using the code (G).

We summarised in Table 4.1 the prior distributions adopted for the random effects in the two baselines and their extensions. We also included the IDs that will be used to identify each model in subsequent Tables and Sections.

ID	Model	Unstructured Effect	Spatial effect
(A)	Baseline 1	Independent Gaussian	Independent IMCAR
(B)	Baseline 2	Independent Gaussian	Independent PMCAR
(C)	Extension 1 - Model 1	Independent Gaussian	IMCAR
(D)	Extension 1 - Model 2	Independent Gaussian	PMCAR
(E)	Extension 2 - Model 1	Correlated Gaussian	IMCAR
(F)	Extension 2 - Model 2	Correlated Gaussian	PMCAR
(G)	Extension 3	Correlated Gaussian	PMCAR - Multiple ρ

Table 4.1: Summary of the prior distributions assigned to the random effects in the models introduced in Section 4.3.

4.3.3 Model comparison

The models proposed in the previous paragraphs were compared using Deviance Information Criterion (DIC) (David J Spiegelhalter et al., 2002) and Watanabe–Akaike Information Criterion (WAIC) (Gelman, Hwang, and Vehtari, 2014; Watanabe, 2010). These criteria represent a measure for the adequacy of a model, penalised by the number of effective parameters. In both cases, the lower is the value of the index, the better is the fitting of the model.

4.4 Results

We estimated the models previously described using the software *INLA* (Gómez-Rubio, 2020; Lindgren, Håvard Rue, et al., 2015; Håvard Rue, Riebler, et al., 2017), interfaced through the homonymous R package (Lindgren, Håvard Rue, et al., 2015; R Core Team, 2020). We used the *Simplified Laplace* strategy for approximating the posterior marginals and the *Central Composite Design* strategy for determining the integration points. The INLA methodology is briefly reviewed in Appendix A.
The code behind multivariate ICAR and PCAR random effects is defined in the package *INLAMSM* (Palmi-Perales, Gomez-Rubio, and Miguel A. Martinez-Beneito, 2019), and we wrote the functions used to estimate the spatial random effects in model (G). They are reported in Appendix B. It took approximately 30 - 45 minutes to estimate each model using a virtual machine with an Intel Xeon E5-2690 v3 processor, six cores, and 32GB of RAM.

Fixed effects

The models listed in Table 4.1 share a common structure for the fixed effect component, with a severity-specific intercept and a set of covariates, representing some characteristics of the road segments. We considered two severity-specific covariates: the road type (either *Motorway*, *Primary Road* or *A Road*, according to OS definition), and the edge betweenness centrality measure, which reflects the number of shortest paths traversing each segment (Kolaczyk and Csárdi, 2014). It can be considered as a proxy for the average vehicle miles traveled (VMT) (Briz-Redón, Martínez-Ruiz, and Montes, 2019a) and it was estimated as reported in Section 2.6.3. Following the suggestions in C. Wang, Quddus, and Ison (2009), the exposure parameter, E_i, is given by the product of two quantities: the segment's length and the estimate of traffic flow (see Section 4.2).

Table 4.2 shows the posterior means and standard deviations for the fixed effects. We first notice that the estimates are stable among the models. The intercept for severe car crashes, β_{01}, is found slightly smaller than β_{02}. This is not surprising since severe accidents are rarer than the slight ones. The coefficients of edge betweenness centrality measures are found close to zero for all models, and their 95% credible interval (not reported in the table) always include the value zero. Road type parameters represent relative differences with respect to the reference category (i.e. *A Roads*), hence *Motorways* are found less prone to severe and slight car crashes than *A roads*. A similar finding was also reported by Boulieri et al. (2017) for UK data. An analogous interpretation applies to *Primary Roads*.

Random effects

The posterior means and standard deviations for all hyperparameters are reported in Table 4.3, which reflects the models' nested structure, also summarised in Table 4.1. We started from two baselines, (A) and (B), with independent random effects, and generalised them until model (G), that presents multiple spatial autocorrelations between the severity levels.

Models from (A) to (D), which assume independent unstructured random effects, exhibit a degenerate posterior distribution of $\sigma^2_{\theta_1}$, i.e. the variance of severe random component, and this is possibly due to the severe car crashes sparseness. This problem gets mitigated once the correlation parameter between the two severity levels is included in the model, suggesting that the estimation procedure benefits from the inclusion of a multivariate structure that allows borrowing strength from less rare events.

The estimates of $\sigma^2_{\theta_2}$ and ρ_θ are stable among the models, and the correlation parameter is estimated as high as 0.40, suggesting a positive and mildly strong relationship between the two random components.

The posterior means for hyperparameters ρ in models (B), (D), (F) and (G), are always very close to one, which is not uncommon for this type of models (Carlin, S. Banerjee, et al., 2003). In particular, the spatial autocorrelation coefficient of model (G) is found higher for slight car crashes than its severe counterpart, indicating a greater spatial homogeneity, something that can also be related with the sparse nature of severe car crashes.

The estimates of the posterior distributions for the two conditional variances, $\sigma^2_{\phi_1}$ and $\sigma^2_{\phi_2}$, are found less stable compared to the unstructured errors. The credible intervals of the two

ID	Severe Crashes				Slight Crashes			
	β_{01}	Betweenness	Motorways	Primary Roads	β_{02}	Betweenness	Motorways	Primary Roads
(A)	−14.325	0.018	−0.828	0.330	−12.834	−0.057	−0.112	0.601
	(0.090)	(0.056)	(0.173)	(0.146)	(0.061)	(0.034)	(0.116)	(0.102)
(B)	−14.430	0.003	−0.825	0.344	−12.856	−0.065	−0.143	0.580
	(0.156)	(0.056)	(0.179)	(0.146)	(0.134)	(0.034)	(0.119)	(0.102)
(C)	−14.342	−0.065	−0.746	0.378	−12.816	−0.084	−0.104	0.578
	(0.090)	(0.056)	(0.179)	(0.147)	(0.061)	(0.034)	(0.119)	(0.104)
(D)	−14.463	−0.098	−0.738	0.487	−12.820	−0.069	−0.187	0.549
	(0.116)	(0.050)	(0.174)	(0.127)	(0.098)	(0.034)	(0.122)	(0.101)
(E)	−14.375	−0.013	−0.787	0.415	−12.816	−0.061	−0.120	0.584
	(0.089)	(0.055)	(0.176)	(0.142)	(0.061)	(0.034)	(0.116)	(0.102)
(F)	−14.460	−0.058	−0.767	0.458	−12.833	−0.066	−0.163	0.556
	(0.133)	(0.052)	(0.175)	(0.133)	(0.119)	(0.034)	(0.120)	(0.102)
(G)	−14.475	−0.063	−0.772	0.469	−12.832	−0.068	−0.164	0.558
	(0.126)	(0.052)	(0.175)	(0.131)	(0.118)	(0.034)	(0.120)	(0.102)

Table 4.2: Estimates for the posterior means and standard deviations, in round brackets, of the fixed effects included in the models described in Table 4.1.

ID	$\sigma^2_{\theta_1}$	$\sigma^2_{\theta_2}$	ρ_θ	ρ		$\sigma^2_{\phi_1}$	$\sigma^2_{\phi_2}$	ρ_θ
(A)	0.0001	0.782				0.148	0.131	
	(0.000 1)	(0.038)				(0.021)	(0.015)	
(B)	0.0001	0.670		0.998		0.268	0.253	
	(0.000 1)	(0.049)		(0.0006)		(0.041)	(0.038)	
(C)	0.0001	0.597				0.315	0.270	0.905
	(0.000 1)	(0.047)				(0.039)	(0.031)	(0.019)
(D)	0.0001	0.358		0.987		0.752	0.725	0.904
	(0.000 1)	(0.055)		(0.004)		(0.105)	(0.102)	(0.019)
(E)	0.573	0.757	0.414			0.141	0.147	0.804
	(0.083)	(0.047)	(0.008)			(0.024)	(0.019)	(0.041)
(F)	0.462	0.632	0.402	0.997		0.291	0.300	0.838
	(0.077)	(0.048)	(0.013)	(0.001)		(0.042)	(0.035)	(0.035)
(G)	0.440	0.633	0.402	0.995	0.997	0.341	0.302	0.842
	(0.084)	(0.049)	(0.013)	(0.002)	(0.001)	(0.066)	(0.042)	(0.031)

Table 4.3: Estimates for the posterior means and standard deviations, in round brackets, of hyperparameters included in the models described in Table 4.1.

ID	DIC	WAIC
(A)	14472.38	14479.97
(B)	14429.88	14445.44
(C)	14271.16	14296.64
(D)	14160.68	14173.21
(E)	14136.85	14118.19
(F)	14114.91	14094.39
(G)	14113.36	14093.90

Table 4.4: Estimates of DIC and WAIC values for the models described in Section 4.3.

hyperparameters overlap in all models, indicating a similar spatial structure between the two kinds of severities. The posterior mean of ρ_ϕ, the correlation coefficient between the two severity levels, is found approximately equal to 0.85, indicating a strong multivariate nature for the spatial random component.

These results suggest that car crash data have a complex latent structure being the severity levels strongly correlated, and the spatially structured and unstructured effects statistically relevant.

Model comparisons

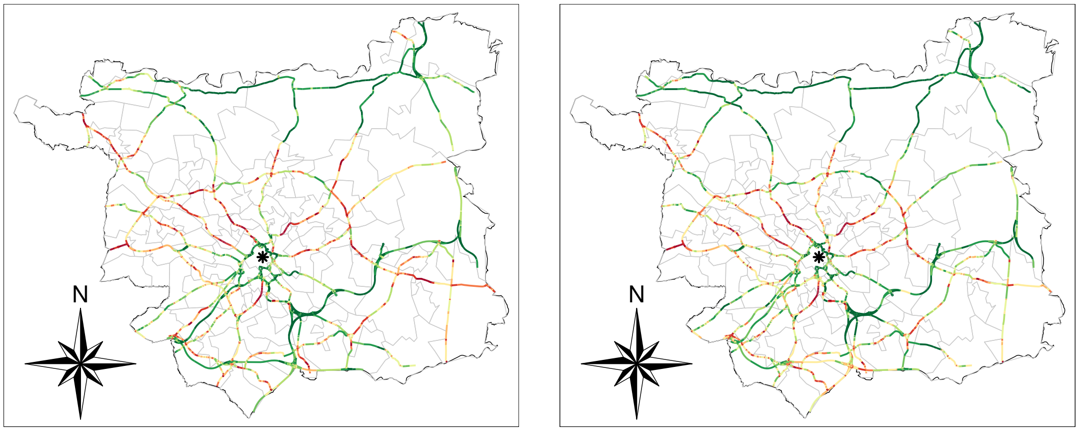
We compared the models listed in Table 4.1 using DIC and WAIC criteria. The results are reported in Table 4.4.

PMCAR models (i.e. (B), (D) and (F)) are found to perform always better than their Intrinsic counterparts in terms of goodness of fit. They are somewhat unexplored in the road safety literature on spatial networks, Miaou and Song (2005) being the only paper we found that analyse the importance of a spatial autocorrelation parameter. However, our results suggest that PCAR distribution and its generalisations should deserve more attention.

Moving from (A) to (G) the model performance improves, indicating one more time the benefits of considering a correlated multivariate structure for the spatial and the unstructured components. In particular, model (G), which includes a specific spatial autocorrelation parameter for each severity level, is the best one according to both criteria. Hence, hereafter, we focus on this model.

Car crashes rate

Figure 4.5 displays the posterior means of car accident rates, λ_{ij}, estimated using model (G) both for severe and slight crashes. The colours of the road segments were generated by dividing the predicted values of each severity level into ten classes based on quantiles,



(a) Severe Car Crashes.

(b) Slight Car Crashes.

Figure 4.5: Maps representing the posterior means for severe and slight car crashes rates, estimated using model (G). The colours go from red (higher quantiles) to green (lower quantiles). The black star represents Leeds City Centre.

ranging from red (highest quantile) to dark green (lowest quantiles). The black star in the middle of the map denotes Leeds City Centre. The two maps show similar patterns, but some roads in the southern part of the city (especially M621) look more prone to severe car crashes. The city of Leeds appears to be divided into several areas. The northern and eastern part of the city are associated with lower car accident rates compared to other suburbs. The areas located in the north, north-west and south-east of the city centre seem to be associated with the highest levels of car crashes rates, especially severe ones. Finally, we note that the roads closer to the city centre are the safest part of the city network.

4.5 Model criticism and sensitivity analysis

DIC and WAIC criteria were never intended to be absolute measures of model fit, and they cannot be used for *Model Criticism*. Hence, we tested the adequacy of model (G) using two strategies.

4.5.1 First strategy for criticism

The classical criterion for criticism of a Bayesian hierarchical model is the Probability Integral Transform (Held, Schrödle, and Håvard Rue, 2010; E. Marshall and Spiegelhalter, 2003), typically adjusted in case of a discrete response variable (such as car crashes counts) using a continuity correction. Unfortunately, these adjustments do not seem to work ap-

		Predicted Counts	
		Zero	One or more
Actual Counts	Zero	A	B
	One or more	C	D

$$\begin{aligned}
\text{Sensitivity} &= \tfrac{A}{A+B}\\
\text{Specificity} &= \tfrac{D}{C+D}\\
\text{Accuracy} &= \tfrac{A+D}{A+B+C+D}\\
\text{Balanced Accuracy} &= \tfrac{1}{2}\left(\tfrac{A}{A+B}+\tfrac{D}{C+D}\right)
\end{aligned}$$

Table 4.5: Left: Confusion matrix showing the observed and predicted counts, binned in two classes. Right: Definition of accuracy measures.

propriately when modelling sparse count data, such as severe crashes, since the correction is not adequate. We refer to Appendix B for more details.

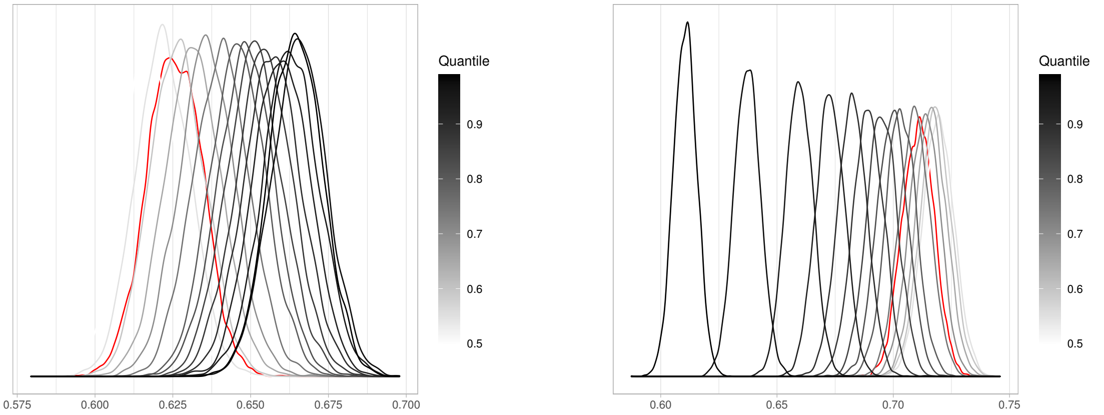
Therefore, hereafter, we followed a different strategy. We binned the observed and predicted counts into two classes: *Zero* and *One or more* car crashes[8]. Then, we built a confusion matrix and evaluated the model's performances via some accuracy measures that are summarised in Table 4.5. A similar correction for sparse count data was also presented in X. Ma, S. Chen, and F. Chen (2017).

The *accuracy* measure, usually adopted for evaluating the predictive performance of a model, is typically biased and overly-optimistic in case of unbalanced classes (such as *Zero* and *One or more* severe car crashes per road segment), since, even in the worst case, it is as high as the percentage of observations in the more frequent class (Brodersen et al., 2010). The *balanced accuracy*, firstly introduced by Brodersen et al. (2010), is defined as the average of Sensitivity and Specificity, and it overcomes this drawback since it represents an average between the predictive performances on each class.

The output of a Bayesian hierarchical model is an estimate of the posterior distribution of predicted values, while the procedure reported in the previous paragraph can only be applied to binary data. For this reason, we simulated n Poisson random variables (one for each road segment) with mean equal to the mean of each posterior distribution. Then, we binned the observed and sampled counts into two classes, i.e. Zero and One or more car crashes, and we compared the two values, obtaining a single estimate of balanced accuracy. Its distribution was finally approximated by repeating this procedure $N = 5000$ times. This strategy is suboptimal since it involves computations based on the posterior marginal distributions of the Gaussian random field instead of the joint posterior distribution. Nevertheless, we compared the two approaches and we found identical results. Hence, for simplicity, we will report only the results based on the marginal distributions.

Moreover, we calculated several quantiles of the posterior distribution of each predicted value, and we run the same steps as in the previous paragraph sampling from a Poisson

[8] We decided to adopt one as a threshold to dichotomise the variables since more than 80% of road segments registered no severe car crash during 2011-2018. The procedure proposed here can be extended to three or more classes, defined using a set of different thresholds (such as *Zero*, *One*, and *Two or more* road crashes).



(a) Severe Crashes

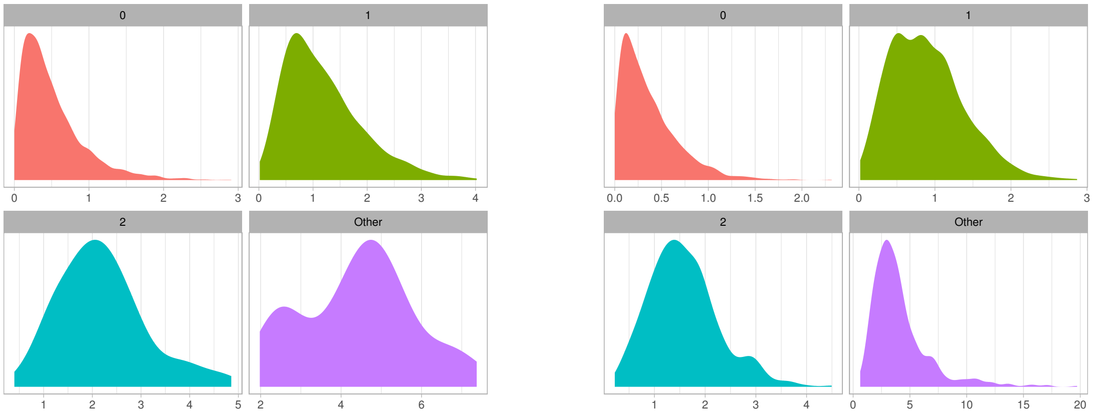
(b) Slight Crashes

Figure 4.6: Distribution of balanced accuracy for severe crashes (right) and slight crashes (left), considering a binary classification using the posterior mean and a set of quantiles. The red curve represents the mean.

distribution with mean equal to each of those quantiles. Lastly, being severe and slight car crashes potentially quite different processes, this algorithm was applied independently for the two severity levels. We reported in Appendix B the pseudo-code for this procedure, whereas results are displayed in Figure 4.6a (severe cashes) and Figure 4.6b (slight crashes). In both cases, the red curve represents the distribution of balanced accuracy obtained by a binary classification based on the posterior means, whereas the other curves represent the same distribution obtained using the set of quantiles. It looks like the optimal threshold for binary classification of severe car crashes is given by the 0.975-quantile, where the balanced accuracy distribution is concentrated around 0.66. The optimal threshold for binary classification of slight car crashes is given by the median, and the distribution of balanced accuracy is centred around 0.72. These plots remark the differences between the two severity levels in terms of sparsity, suggesting the adoption of a higher quantile for the prediction of the more sparse events. However, using the appropriate cut-off(s), Model (G) seems to perform reasonably well in both cases with slightly better performance for slight car crashes.

4.5.2 Second strategy for criticism

Following the results illustrated before, we estimated the 0.975-quantile of λ_{i1} (severe crashes) and the median of λ_{i2} (slight crashes), and we multiplied them by the corresponding offset values, i.e. E_i. Then, we created a sequence of histograms of predicted values, grouped by the observed counts categorised in four levels: 0, 1, 2, and 3 or more. The re-



(a) Severe Crashes

(b) Slight Crashes

Figure 4.7: Histogram of posterior 0.975-quantile (left) and posterior median (right), grouped by the corresponding observed counts. Other means "Three or more."

sults are summarised in Figure 4.7a and Figure 4.7b. Both graphs show a good agreement between predicted and observed number of crashes, since the distributions corresponding to higher observed counts progressively move more and more to the right. Moreover, Figure 4.7a shows the importance of our previous analysis and the pitfalls of predicting severe car crashes counts using the posterior means.

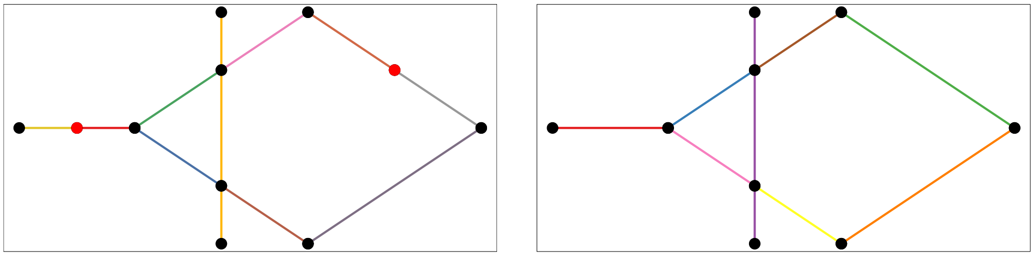
4.5.3 Sensitivity analysis and the modifiable areal unit problem

Finally, we performed a sensitivity analysis evaluating the robustness of model (G) under different specifications for 1) the hyperprior distributions, 2) the adjacency matrix, and 3) the definition of the segments in the road network.

The models described in Section 4.3 considered a Wishart hyperprior for precision matrix $\boldsymbol{\Omega}$ with rank equal to 2 and scale matrix equal to $\boldsymbol{I}_2$. We repeated the analysis using more vague and more informative Wishart distributions, setting the scale matrix equal to $\text{diag}(2, 2)$ and $\text{diag}(0.5, 0.5)$. We did not find any noticeable differences amongst alternative specifications. Hence results are not reported hereafter, but we refer to Appendix B.

We compared different definitions for the adjacency matrix $\boldsymbol{W}$, testing second and third order neighbours and distance-based spatial neighbours[9]. Also in this case, we did not

[9] In this case two road segments are considered neighbours if the euclidean distance between their centroids is smaller than a certain threshold. In particular, we used $25m$, $50m$, $100m$, $250m$ and $500m$ as alternative thresholds. In case a network segment was longer than twice the threshold, we consider first order neighbours to avoid the creation of several islands (see Section 4.2). This problem is more pronounced for smaller values of the threshold.



(a) Road Network with redundant vertices (in red)

(b) Contracted road network

Figure 4.8: Sketching the algorithm used for contracting the road network. Red points on the left represent redundant vertices.

find any noticeable differences as far as the estimation of the fixed effects is concerned, whereas only small differences were found in the posterior distributions of the random effects (especially for $\sigma^2_{\phi_1}$ and $\sigma^2_{\phi_2}$ when we considered a spatial adjacency matrix with a threshold equal to $500m$). However, worse DIC and WAIC values were found for models using alternative definitions of $\boldsymbol{W}$ matrix, and we refer to Appendix B for more details. Similar findings are also reported by Aguero-Valverde and Jovanis (2008), Alarifi et al. (2018), and X. Wang et al. (2016).

Finally, we explored the influence of a particular configuration of the network segments on our results. In fact, the location of the vertices (and, hence, the edges) in a road network created with OS data is essentially arbitrary (although some minimal consistency requirements must be satisfied, see Karduni, Kermanshah, and Derrible (2016) and Section 2.5), which implies that there is no unique and unambiguous way of defining the lengths and relative positions of the road segments. We, therefore, considered an alternative network configuration reshaping and contracting the road network using an algorithm implemented in Padgham (2019). This algorithm manipulates a network by excluding all *redundant* vertices, i.e. those vertices that connect two contiguous segments without any other intersection (Padgham, 2019). More details were reported in Chapter 2.

A toy example representing the ideas behind the contraction of a road network is sketched in Figure 4.8. The red dots in Figure 4.8a represent redundant vertices since they can be removed without tampering the shape or the routability of the network[10]. The goal is to remove all redundant vertices and merge the corresponding edges, creating a graph which looks identical to the original one but with fewer edges. Figure 4.8b shows the results of the contraction operations applied to the toy network sketched in Figure 4.8a. We can see that the redundant vertices were removed, combining the road segments that touched them.

[10] We say that the algorithm in Padgham (2019) preserves the *routability* of a road network since it does not remove any vertex that could add a new *component* to the graph.

	Severe Crashes	Slight Crashes
β_0	-15.335	-13.694
	(0.233)	(0.198)
Betweenness	-0.008	-0.041
	(0.055)	(0.037)
Motorways	-0.872	-0.298
	(0.180)	(0.140)
Primary Roads	0.413	0.496
	(0.132)	(0.117)

Table 4.6: Means and standard deviations for the posterior distributions of the fixed effects in the model estimated after contracting the road network.

	$\sigma^2_{\theta_1}$	$\sigma^2_{\theta_2}$	ρ_θ	ρ_1	ρ_2	$\sigma^2_{\phi_1}$	$\sigma^2_{\phi_2}$	ρ_ϕ
mean:	0.307	0.500	0.388	0.995	0.993	3.970	4.767	0.944
sd:	(0.103)	(0.100)	(0.020)	(0.003)	(0.003)	(0.630)	(0.622)	(0.016)

Table 4.7: Means and standard deviations for the posterior distributions of the hyperparameters in the model estimated after contracting the road network.

Applying this algorithm produced a contracted road network with, approximately, 2700 segments (instead of the original 3661). Following the same procedures detailed in Sections 4.2 and 4.4, we calculated the number of severe and slight car crashes that occurred in each road segment, the traffic volumes (which are used as an offset), and the edge betweenness centrality measures. The road type was automatically determined since the algorithm did not merge two road segments with different classifications. Finally, we estimated model (G) and reported a summary of means and standard errors of the posterior distributions of fixed and random effects in Tables 4.6 and 4.7, respectively.

As far as the fixed effects are concerned, the new network configuration influences results quite mildly, since the estimates of the coefficients did not change in sign, order of magnitude or significance. As one could expect, the impact of network reshaping is slightly more pronounced for the random effects, in particular for $\sigma^2_{\phi_1}$ and $\sigma^2_{\phi_2}$, two of the five hyperparameters in the PMCAR prior. The model trained on the contracted network presents a greater spatial uncertainty than model (G), but similar posterior distributions for car crashes rates. We refer to Appendix B, where we also report the maps of predicted rates using the new network configuration.

Network reshaping and contraction is a network-readaptation of the classical Modifiable Area Unit Problem (MAUP), only recently explored by a handful of authors in the literature of road safety models for areal data (see, for example, Xu, H. Huang, and Dong (2018) for an introduction, and Briz-Redón, Martínez-Ruiz, and Montes (2019b) for an extensive application). The main conclusion of these papers is that MAUP severely impacts car crashes models, affecting the magnitude and significance of the estimates for both fixed and random effects, hence it should never be ignored for a reliable road safety analysis.

Our results tell a somewhat different story. The statistical analysis is found quite robust to MAUP when carried out on a network lattice, possibly because the road network has a physical geometrical meaning and, hence, a lower degree of arbitrariness than administrative boundaries. Hence, we suggest not to ignore the network structure of the data whenever it is available when analysing car crashes data or other phenomena that naturally occur on a network. The only paper that performs a descriptive analysis of the influence of road segment configurations on car crashes counts is Thomas (1996), and, to the best of our knowledge, this is the first attempt to explore the robustness of a statistical model for lattice network data to MAUP.

4.6 Conclusions

This Chapter investigated the spatial distribution of road crashes in a major city using new methods for network analysis. The relationship between crashes of different severity levels, either *slight* or *severe*, were modelled using a range of multivariate models to explore their spatial dynamics. Key to the approach was constraining crash locations to the road network, a one-dimensional linear network composed of segments representing a spatial lattice.

We found that the best model includes a multivariate spatially unstructured random effect and a multivariate spatially structured PMCAR random effect with a different autocorrelation parameter for each severity level. The results, which are summarised in Tables 4.2 and 4.3, suggest that the Motorways are less prone to severe and slight car crashes than A-roads, which are also less dangerous than Primary roads. The posterior distributions of the hyperparameters point out a strong between-level correlation in the unstructured errors and an even stronger dependence in the spatial component.

The approach proposed in this Chapter allows the estimation of a severity specific car crashes rate for every single segment in the road network, which can be visualised using an appropriate choropleth map. Figure 4.5 exemplifies this process by showing two maps that display the posterior means of car crash rates for severe and slight accidents. They highlight several roads, or portions of a road, mainly located in the north-east, north-west and south-east of Leeds city centre, which are associated with higher car crashes rates. The two maps illustrate that adopting a network-based approach allows the identification of dangerous streets more precisely than using an areal-based aggregation. These results could represent the starting point for further analysis, linking car crashes rates with structural aspects of the network such as roundabouts, street junctions or pedestrian crossings, hence, in turn, identifying and evaluating potential levers to intervene on. This extension was ignored in the current manuscript since it requires a different methodological approach, and OS data do not include a precise database of roundabouts or street junctions.

We evaluated the sensitivity of our modelling approach to different hyperprior specifications and adjacency configurations of the components of the lattice network, showing that the statistical model presents substantial robustness in this respect. We finally considered the impact of MAUP when modelling data collected on a spatial network. An algorithm was proposed to assess the magnitude of MAUP effect in the estimates and model predictions. Differently from several previous studies that considered the MAUP for various areal partitions of the spatial region of interest, we found that our results are quite robust under an alternative configuration of the road network. This can be related to the fact that road networks have a physical meaning, hence they are expected to suffer MAUP less, a further advantage of the network approach. Robustness of results is fundamental for the development of a reliable model that can be used to support the implementation of new policies. Nevertheless, further research, possibly in different fields, is definitely necessary to better understand the impact of MAUP on network lattice data.

The ideas presented in this Chapter could be extended in several directions. A first step forward could be focused towards the development of a spatio-temporal extension of model (G), following the suggestions in Boulieri et al. (2017), X. Ma, S. Chen, and F. Chen (2017), Miaou and Song (2005), and C. Wang, Quddus, and Ison (2011). We point out, however, that this is not straightforward (and, to the best of our knowledge, it was pursued only by X. Ma, S. Chen, and F. Chen (2017) using a single road divided into a few segments) given the extreme sparse spatio-temporal nature of severe car crashes on a metropolitan road network. Indeed, for the dataset at hand, more than 95% of all car crashes registered no fatal or serious car accident for any given year, something that could require a different methodological approach. The procedure for MAUP detection could also be improved by developing new routines for testing alternative algorithms for network reshaping and contraction, which were first developed for areal data in the field of geography (see, e.g., Xu, H. Huang, and Dong (2018) and references therein). An additional improvement to the approach could involve the development of spatial or spatio-temporal theoretical point pattern models for car crashes on networks. It is clear that more research is needed to evaluate the full range of possible models for identifying crash ‘hot spots’ on the network. The research presented in this Chapter demonstrates the potential of network-based approaches to work at city scales for flexible and robust estimates of crash rates, down to the road segment level and provides a basis for more further work in the field.

CHAPTER 5

A non-separable first-order spatio-temporal intensity function for events on networks: an application to ambulance interventions

Paper still under development

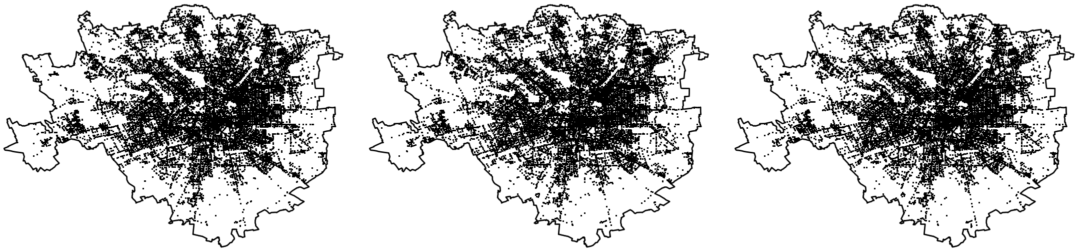
> *Questa è novità, questa è una grossa novità.*
>
> Enzo Spatalino - Voce alla Gente, 2019-09-18

5.1 Emergency Medical Systems data

We analysed the spatio-temporal distribution of all emergency events that occurred in the road network of the municipality of Milan (IT) from 2015-01-01 to 2017-12-31, required an ambulance intervention and were handled by the regional Emergency Medical System (*EMS*), which is called *AREU* (an acronym for *Azienda Regionale Emergenza Urgenza*). In the following paragraphs of this Section, we will describe the algorithms and the procedures that were used to transform the raw information into a data structure suitable for estimating the statistical model detailed in Section 5.2. We will also present an exploratory analysis that summarises the main spatio-temporal characteristics of the data at hand and justify our modelling choices.

We started from a dataset provided by the official regional authorities that included all ambulance dispatches in Lombardia from 2015-01-01 to 2017-12-31. However, we decided to focus only on the city of Milan since it represents one of the most important Italian metropolitan areas, where hundreds of thousands of people pass through every day. Furthermore, the experts of emergency interventions and management of ambulance fleets in AREU reported that the spatio-temporal dynamics in Milan are quite different from the other parts of the region, and require ad-hoc modelling and planning.

Each record in the dataset included the day, the hour, and the GPS coordinates (using Monte Mario Coordinate Reference System) of the corresponding ambulance dispatch, the three building blocks for the spatio-temporal model introduced below. The description of the ambulance interventions is typically enriched using a series of markers that characterise



(a) 2015

(b) 2016

(c) 2017

Figure 5.1: Graphical representation of ambulance interventions that occurred in Milan (IT) from 2015-01-01 to 2017-12-31. We can notice that the spatial distributions are similar among the three years and that the events highlight some the roads in the city. The white areas correspond to not-urban or abandoned areas (e.g. Cimitero Monumentale, Scalo Farini or Ippodromo), large buildings (e.g. Policlinico di Milano or City Life) and parks (such as Parco Sempione or Giardini pubblici Indro Montanelli).

their severity levels (using a code that can be either *green*, *yellow*, or *red*, in increasing order of seriousness), the place (e.g. private house, public office or retirement home) and the reason (e.g. discomfort, illness, car crash, poisoning or tumble). We removed all records with missing spatial or temporal coordinates or anomalies in the other descriptive markers. Moreover, we included only the first ambulance intervention in case of multiple dispatches for the same event, which can occur in case of severe car crashes, heart attacks or other types of life-threatening situations. We did not consider ambulance interventions related to scheduled events, like hospital admissions or discharges. The final sample included 495,950 interventions, 163,488 occurred in 2015, 165,368 in 2016 and 167,094 in 2017.

The spatial representation of the points is depicted in Figure 5.1. First of all, we can notice that the distributions look stable among the three years. The events seem to highlight the skeleton of a road network structure that corresponds to the city ring road and some of the most important arterial thoroughfares. The white areas can be recognised as desert, not-urban places, mainly located in the south or the west. We can also clearly distinguish the shapes of some of the most iconic locations in Milan, such as City Life, Parco Sempione, Scalo Farini or Giardini Indro Montanelli, where the ambulance interventions are usually geo-located at the entrance or the nearest point in the city network. Similar maps are also reported by Bayisa et al. (2020) and Zhou, David S Matteson, et al. (2015). Given the spatial distribution of the emergency interventions that resemble a network structure, we argue that, in this particular case, a network-approach is more appropriate than a planar approach since it takes into account the nature of the data and the particular characteristics of their geo-locations (Baddeley, Nair, et al., 2020; Okabe and Sugihara, 2012).

We explored the temporal dimension of the data, examining the hourly, daily, weekly, and monthly dynamics that govern the total number of emergency calls. We report in Figure 5.2 three time series that represent the daily number of ambulance interventions, divided according to the year of occurrence. The data exhibit a monthly seasonal behaviour, where the global minima are registered around August, in conjunction with the national holidays. Other local peaks and minima could be linked with the most important religious holidays (such as Christmas or Easter), national celebrations (New year's eve) or other sporadic events (such as the heatwave in July 2015 or the ice storms in January 2017). We can notice a common trend behind the temporal dynamics in the three years.

We also considered the temporal dynamics of the emergency calls within a day, and we report in Figure 5.3 three time series that summarise the average number of hourly ambulance interventions divided by the hour of the day and the day of the week. In all cases, we can recognise a distinct shape: after rapidly increasing in the early morning, the time series peaks around 10:00, slowly fall until 20:00 and then sink until the next day. Moreover, the hourly seasonalities are different between weekends and weekdays. In fact, in the latter case, the regional EMS registers, on average, more interventions during the first hours of the day, which is probably linked with the city's nightlife, and lower rates in the morning and the afternoon. Again, we can recognise a similar pattern among the three years. Analogous findings regarding the temporal dimensions of EMS calls are also reported by (Bayisa et al., 2020; David S Matteson et al., 2011; Zhou and David S Matteson, 2016; Zhou, David S Matteson, et al., 2015). The statistical model proposed in Section 5.2 takes into account these seasonalities.

The road network was created starting with data download from Open Street Map (OSM) servers, and, in particular, we used the `openstreetmap.fr`[1] provider, accessed using the `R` package `osmextract`, detailed in Appendix C.

Open Street Map is a project that aims to build an open and editable map of the World (Barrington-Leigh and Millard-Ball, 2017). The basic components of Open Street Map data are called *elements*, and they are divided into *nodes*, which represent points on the earth's surface; *ways*, which are ordered lists of nodes; and *relations*, which are lists of nodes, ways and other relations, where each member has additional information that describes its relationship with the other elements.

A road network is typically represented as the union of a finite set of segments (or, using the OSM jargon, *ways*). More precisely, we downloaded Open Street Map data for the Lombardia region, and then we filtered only the highways that lie inside the polygonal boundary that define the city of Milan using a spatial filter operation. Then, we selected only the *ways* that could be linked with the most important streets of Milan, focusing on the following classes[2] (listed in descending order of importance): *motorways*, *trunks*, *primary*,

[1] See `http://download.openstreetmap.fr/`. Last access: 2020-12-09.

[2] We refer to `https://wiki.openstreetmap.org/wiki/Highways` for a comprehensive description of road network data in Open Street Map and guidelines for its classification system. We also refer to `https://wiki.openstreetmap.org/wiki/IT:Key:highway` for a comparison between the Italian classification system and the classes defined by OSM.

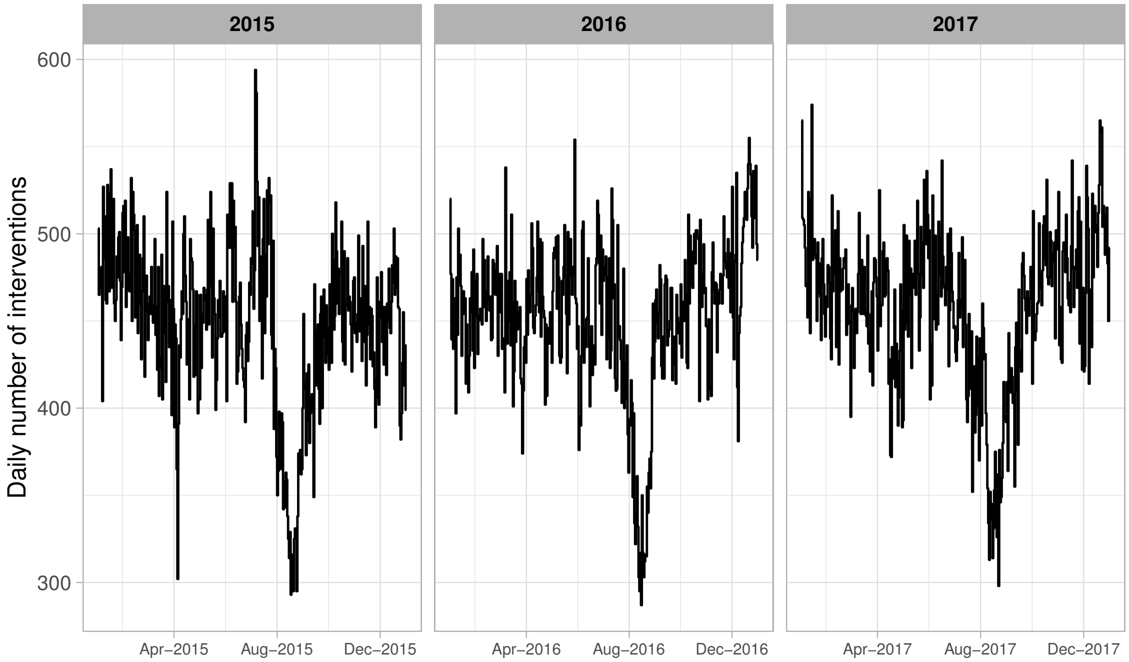


Figure 5.2: Time series that represent the daily number of ambulance interventions that occurred in Milan from 2015-01-01 to 2017-12-31. The data clearly exhibit a seasonal behaviour, and a common distribution among the three years. Some local peaks can be linked to national holidays (such as Easter), heat aves (see July 2015) or ice storms (see January 2017).

secondary, *tertiary*, *unclassified*, and *residential*.
As introduced in Chapter 2, a road network can also be seen as a graph object (Barthélemy, 2011; Karduni, Kermanshah, and Derrible, 2016; S. Marshall et al., 2018; Porta, Crucitti, and Latora, 2006), where each segment is associated with the edges, while the vertices correspond to the intersections, usually located at road junctions, although they may potentially be placed also in between. Its adjacency matrix is typically determined as follows: we say that two edges are *connected* if the corresponding street segments share at least one point (or *node* in OSM jargon) in their boundaries and, by the same reasoning, the vertices are connected if they belong to the same road segment. We took advantage of this dual representation to simplify the road structure, excluding all small clusters of segments, also named *components* in the graph-analysis literature (Kolaczyk and Csárdi, 2014), not-connected to the main street network, probably because of small rounding errors in the coordinates or missing links in OSM data. This pre-processing step creates a fully connected graph (meaning that each vertex can be reached from any other vertex), and it is

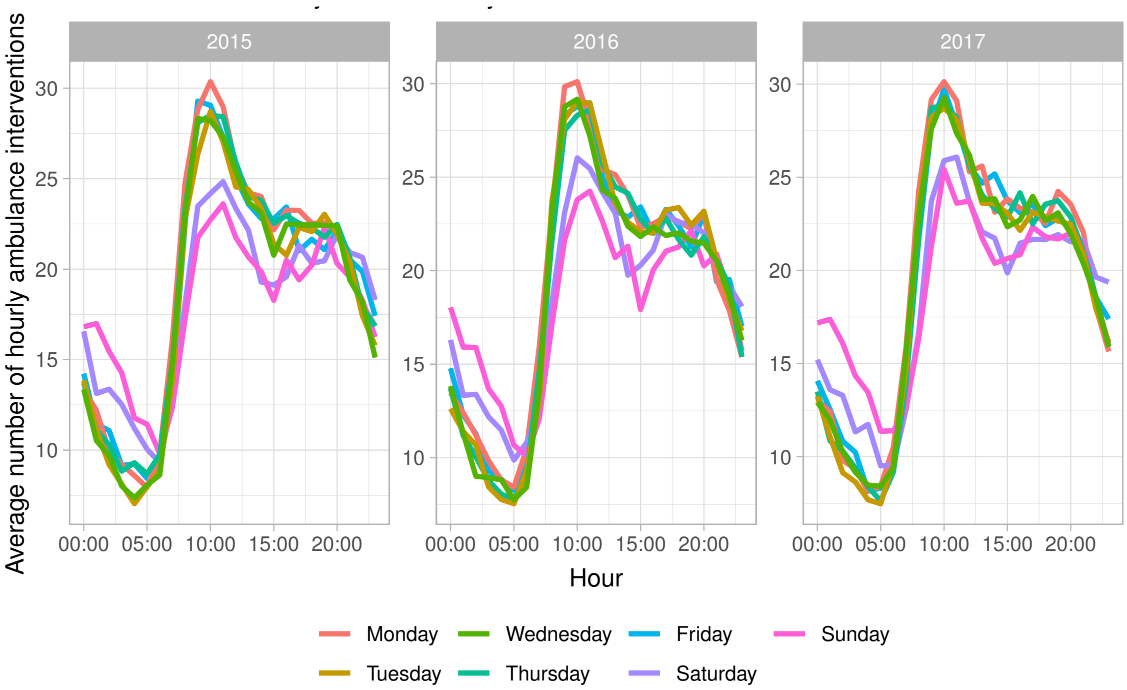


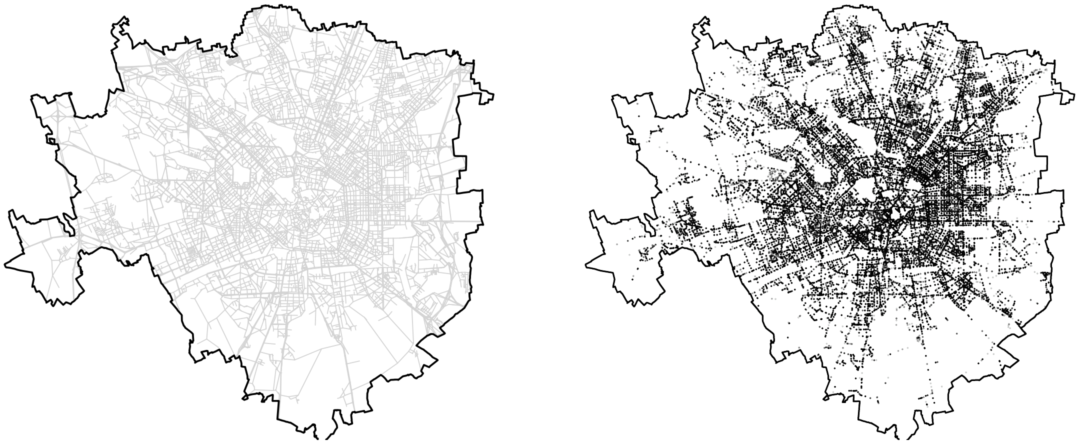
Figure 5.3: Time series that represent the average number of hourly ambulance intervention that occurred in Milan (IT) from 2015 to 2017, divided by the day of the week. Strong seasonal components, related to the hour of the day and the day of the week, can be observed. The three years exhibit similar patterns.

relevant for the kernel estimator that will be proposed and detailed in Section 5.2. We refer to Chapter 2 for more details.

Following the steps detailed before, we created a road network that is long approximately 1850km, including most streets in the municipality of Milan. It is depicted in Figure 5.4a. We can notice that it covers a large portion of the city and that the white areas can be associated with parks, squares and desert or non-urban places.

After creating the road network, we decided to exclude all emergency calls whose GPS locations were found farther than 50 meters from the closest segment of the network[3], since they are probably linked with other streets not included in the network. We dropped approximately 10,000 events. Then, we projected the remaining 489,970 points into the road network, and the result is reported in Figure 5.4b. The events seem to be gathered in a few particular areas like Stazione Centrale, Duomo, Giardini Indro Montanelli, Colonne di San Lorenzo or Porta Genova, all popular and busy areas. The point pattern on a

[3] The distance between a point and a segment was measured using the shortest euclidean perpendicular distance.



(a) *The road network of Milan.*

(b) *The point pattern of ambulance interventions lying on the road network of Milan.*

Figure 5.4: *Left:* Map of the road network of Milan. It includes all major streets in the city. *Right:* Projecting the emergency interventions into the linear network. In both cases, the white areas represent parks, squares, desert or non-urban places and big pedestrian neighbourhoods (e.g. Brera).

linear network represents the starting point for the spatial modelling approach introduced in Section 5.2.

Finally, we explored the spatio-temporal dynamics of the phenomenon, checking the presence of space-time interactions between the hours of the day and the spatial distribution. For this reason, we rounded the occurrence of each emergency event to an integer number such that, for example, all interventions happened between 00 and 01 AM are labelled using the character string **0**. Then, we plotted their spatial distribution divided by the hour of occurrence, and the result is reported in Figure 5.5. We can notice that from 08 AM to 08 PM the points look crowded near the city centre, close to the office areas and the main buildings. On the other hand, the distribution during the night hours look more scattered in the municipality, and it looks like there is a smooth transition between the two scenarios. Similar finding are also reported by Zhou and David S Matteson (2016), Zhou, David S Matteson, et al. (2015), and Zhou and David S. Matteson (2015). The spatial model detailed in Section 5.2 was defined taking into account these interactions.

5.2 Statistical Methods

We consider a continuous one-dimensional spatial region L and a discrete temporal dimension $\mathcal{T} = \{1, 2, \ldots, T\}$ divided into intervals of one hour. The object L represents a linear

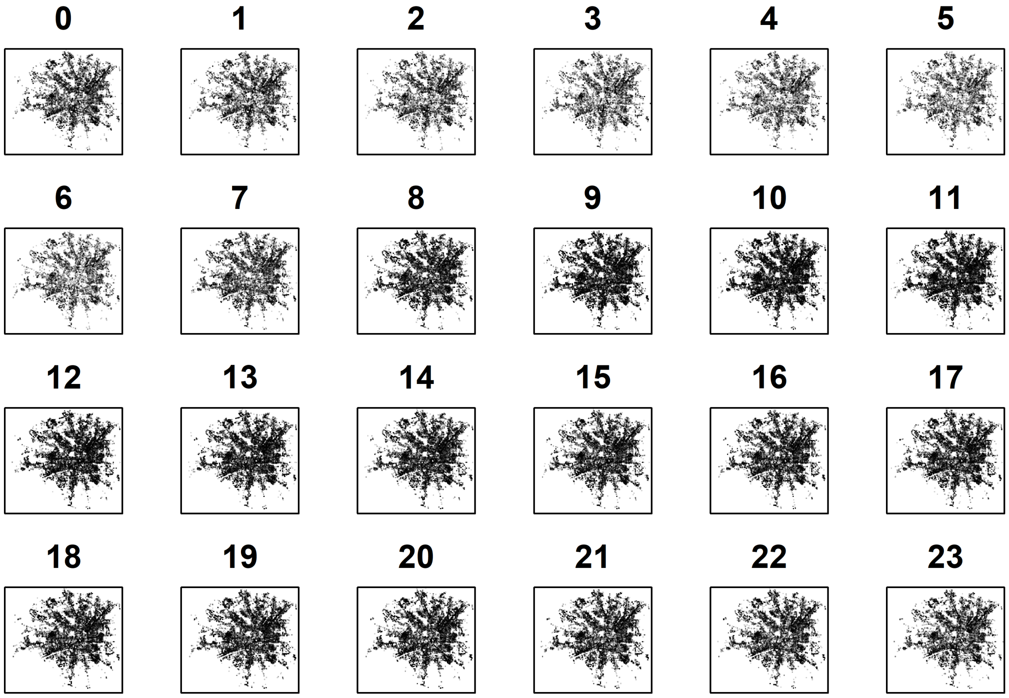


Figure 5.5: Spatial distribution of the ambulance interventions obtained after rounding their time of occurrence to an integer hour.

network, typically defined as the union of a finite set of line segments embedded in a planar region $S \subset \mathbb{R}^2$ (Ang, Baddeley, and Nair, 2012; Okabe and Sugihara, 2012). In this Chapter, L is the street network of Milan, built using the procedures detailed in Section 5.1, while T is equal to 26304, i.e. the hours from 2015-01-01 00:00 to 2018-01-01 00:00.
Let y_t represents the number of ambulance interventions that occurred in the network L at time $t \in \mathcal{T}$, and let $\boldsymbol{s}_{t,i},\ i = 1, \dots, y_t$ denote the location of the ith event. We assume that, independently for each $t \in \mathcal{T}$, the process $\{\boldsymbol{s}_{t,i} : i = 1, \dots, y_t\}$ can be characterized as a *Non Homogeneous Poisson Process* (NHPP) with *intensity function* $\lambda_t(\boldsymbol{s})$ (P. J. Diggle, 2013). The NHPP satisfies the following two properties:

- The number of events occurring in $L' \subseteq L$, which is denoted by $N(L')$, follows a Poisson distribution with parameter $\mu_t(L') = \int_{L'} \lambda_t(\boldsymbol{s})\, \mathrm{d}\boldsymbol{s}$. The object L' represents a finite portion of the network L.
- Let $N(L') = n$, then the n events represent a random sample from a distribution

whose probability density function is proportional to $\lambda_t(\boldsymbol{s})$.

We assume that the intensity function of the process can be decomposed as follows

$$\lambda_t(\boldsymbol{s}) = \mu_t g_t(\boldsymbol{s}) \quad \text{for } \boldsymbol{s} \in S. \tag{5.1}$$

The previous equation, despite being similar to the classical separability assumption for spatio-temporal processes (P. J. Diggle, 2013; Moller and Waagepetersen, 2003), implies that the function $g_t(\boldsymbol{s})$, which, as explained below, represents the spatial dimension of the events, also depends on time t. We decided to adopt this functional form for $\lambda_t(\boldsymbol{s})$ since we noticed space-time interactions in the hourly evolution of ambulance interventions (see the end of Section 5.1). The space-time dependencies will be modelled using a set of weights, which are introduced in Section 5.2.2.

As we have already mentioned, $g_t(\boldsymbol{s})$ represents the spatial dimension at time t, and it must satisfy the two following conditions:

- $g_t(\boldsymbol{s}) > 0 \; \forall t \in \mathcal{T}$ and $\forall \boldsymbol{s} \in L$;
- $\int_L g_t(\boldsymbol{s}) \, \mathrm{d}\boldsymbol{s} = 1 \; \forall t \in \mathcal{T}$.

Hence, μ_t represents the temporal dimension of the process, since it can be viewed as a spatial aggregation of the intensity function at time t. Considering the NHPP hypothesis and the assumptions on $g_t(\boldsymbol{s})$, we can see that $\mu_t = \int_L \lambda_t(\boldsymbol{s}) \, \mathrm{d}\boldsymbol{s}$, and, for all $t \in \mathcal{T}$, $y_t | \lambda_t \sim \text{Poisson}(\mu_t)$. So μ_t represents the *temporal intensity* or *total call volume*. On the other hand, the NHPP hypothesis also implies that $\boldsymbol{s}_{t,i} | \lambda_t, y_t \overset{\text{iid}}{\sim} g_t(\boldsymbol{s})$ for $i = 1, \dots, y_t$. Therefore, we can interpret $g_t(\boldsymbol{s})$ as the spatial intensity of ambulance interventions at time t.

Hereafter we will introduce two statistical models for μ_t and $g_t(\boldsymbol{s})$. The temporal component will be modelled using a dynamic latent factor model based on deterministic temporal covariates, such as the hour of the day or the day of the week. The spatial dimension will be estimated non-parametrically using a network-readaptation of a weighted kernel function. We will also present a procedure for the estimation of the weights.

5.2.1 The temporal model

As mentioned above, we assume that we can model the temporal evolution of the ambulance interventions using a dynamic latent factor model with Poisson distribution. The seasonalities of EMS interventions are taken into account by the inclusion of a pre-determined structure that depends on deterministic temporal covariates. The model's definition and the following notations are based on David S Matteson et al. (2011) and several previous papers (Shen and J. Z. Huang, 2008a,b; Takane and Hunter, 2001; Tsai and Tsay, 2010).

Let y_t represent the number of ambulance dispatches that occurred in the network L at time $t \in \{1, \dots, T\}$. The values of y_t were calculated by rounding and aggregating all EMS

event to the smallest integer hour, such that, for example, all interventions between 00:00 and 01:00 are associated with 00:00.
Moreover, let D be the total number of days under analysis, J be the number of intra-day periods, and $\boldsymbol{Y}$ be the $D \times J$ matrix of EMS counts. In the data at hand, we have $D = 1096$ days and $J = 24$ periods, i.e. the number of hours in a day. Each element of the matrix $\boldsymbol{Y}$, say y_{ij}, $i = 1, \ldots, D$; $j = 1, \ldots, J$, represents the number of ambulance interventions that occurred in the ith day during the jth hour.

The dynamic latent factor model

Considering the hypotheses stated at the beginning of the Section, we say that μ_{ij} represents the expected number of ambulance dispatches during the ith day and jth hour, i.e $y_{ij}|\lambda_{ij} \sim$ Poisson(μ_{ij}). Following David S Matteson et al. (2011), we suppose that the logarithm[4] of μ_{ij} can be approximated using a linear combination of $K \leq J$ latent factors, say $\boldsymbol{f}_1, \ldots, \boldsymbol{f}_K$, which are orthogonal vectors with dimension $J \times 1$. More precisely, we can write

$$\log \boldsymbol{\mu}_i \simeq L_{i1}\boldsymbol{f}_1 + \cdots + L_{iK}\boldsymbol{f}_K, \tag{5.2}$$

where $\boldsymbol{\mu}_i$ represents a vector with dimension $J \times 1$ filled with all EMS counts that occurred during the ith day, while $L_{i1}, \ldots, L_{iK}$ are the loadings of the latent model and also the weights of the linear combination. The K factors are constant among different days and do not depend on the value of i since they represent the intra-day effects in the EMS calls (see Figure 5.3). On the other hand, the loadings are constant within each day but different among different days, and do not depend on the value of j. They represent daily and weekly temporal effects (see Figure 5.2). The value of K must be chosen so that we can successfully approximate the behaviour of EMS counts using the smallest number of factors, and, in practice, it was chosen by testing the predictive performance of the Poisson model using an out-of-sample approach.
Equation (5.2) can also be written in matrix form as follows:

$$\log \boldsymbol{M} = \boldsymbol{L}\boldsymbol{F}', \tag{5.3}$$

where $\boldsymbol{M}$ is a $D \times J$ matrix of expected counts, $\boldsymbol{F} = (\boldsymbol{f}_1, \ldots, \boldsymbol{f}_K)$ is the $J \times K$ matrix obtained by column-binding the latent factors, and $\boldsymbol{L}$ is the $D \times K$ matrix of loadings. At the moment, the model (5.3) is not strictly identifiable since we do not observe neither $\boldsymbol{F}$ nor $\boldsymbol{L}$, so, following David S Matteson et al. (2011), we assume that $\boldsymbol{F}'\boldsymbol{F} = \boldsymbol{I}_K$, where $\boldsymbol{I}_K$ is the $K \times K$ identity matrix.

[4] The logarithm transformation has several benefits. First, the predicted values of μ_{ij} on the original scale (i.e. after the exponential transformation) are forced to be positive. Second, the temporal effects that influence μ_{ij} on a multiplicative scale are transformed into a linear additive scale. Finally, the logarithm function represents the canonical link for the Poisson family (Agresti, 2015).

Factor modelling with covariates via constraints and smoothing splines

The deterministic covariates explored in Section 5.1 were included in the model using a set of constraints on the matrices $\boldsymbol{F}$ and $\boldsymbol{L}$. A daily and weekly structure is imposed on the loadings of the model, that we recall as being constant within each day but vary among different days, and an hourly structure is imposed on the latent factors.
We assume that we can rewrite Equation (5.3) as:

$$\log \boldsymbol{M} = \boldsymbol{L}\boldsymbol{F}' = \boldsymbol{H}\boldsymbol{B}\boldsymbol{F}'$$

where $\boldsymbol{H}$ is a $D \times r$ matrix of constraints related to the matrix $\boldsymbol{L}$, and $\boldsymbol{B}$ denotes an $r \times K$ matrix of unconstrained loadings.
Looking at Figures 5.2 and 5.3, we can notice that the weekly patterns of ambulance interventions evolve smoothly during the years, without any remarkably abrupt change (but for some particular days related to national holidays or sporadic extreme weather conditions), and that the behaviour of EMS events is dramatically different between weekends and weekdays. For these reasons, the temporal constraints on the loadings $\boldsymbol{L}$ are decomposed into two matrices of incidence vectors, named $\boldsymbol{H}^{(1)}$ and $\boldsymbol{H}^{(2)}$, that model the effects related to the day of the week and the week of the year:

$$\log \boldsymbol{M} = \boldsymbol{L}\boldsymbol{F}' = \boldsymbol{H}\boldsymbol{B}\boldsymbol{F}' = (\boldsymbol{H}^{(1)}\boldsymbol{B}^{(1)} + \boldsymbol{H}^{(2)}\boldsymbol{B}^{(2)})\boldsymbol{F}'. \tag{5.4}$$

$\boldsymbol{H}^{(1)}$ is a $D \times 7$ incidence matrix such that, for example, if the first EMS intervention for $t = 1$ happened on Monday, then $\boldsymbol{H}^{(1)}$ can be written as

$$\boldsymbol{H}^{(1)} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \end{bmatrix}.$$

An analogous interpretation holds for matrix $\boldsymbol{H}^{(2)}$, which has dimension $D \times 53$, where 53 is the number of weeks in a year considering that the first and last weeks may have less than seven days. The matrices $\boldsymbol{B}^{(1)}$ and $\boldsymbol{B}^{(2)}$ denote the unconstrained factor loadings with dimensions $7 \times K$ and $53 \times K$, respectively.
Similar reasoning can also be applied to matrix $\boldsymbol{F}$ given the peculiar distribution of EMS hourly interventions displayed in Figure 5.3. In particular, the factors are modelled using an incidence matrix, say $\boldsymbol{H}^{(3)}$, and an unconstrained factors matrix, say $\boldsymbol{B}^{(3)}$, with dimensions $J \times J$ and $J \times K$, respectively:

$$\log \boldsymbol{M} = \boldsymbol{L}\boldsymbol{F}' = \boldsymbol{L}\left(\boldsymbol{H}^{(3)}\boldsymbol{B}^{(3)}\right)' = \boldsymbol{L}\boldsymbol{B}^{(3)'}\boldsymbol{H}^{(3)'} \tag{5.5}$$

The main differences between Equations (5.4) and (5.5) are given by the nature of the constraints. The incidence matrices adopted when modelling the loadings represent a dimension reduction technique since we are imposing $r < D$ constraints (Tsai and Tsay, 2010), while, in the second case, the matrix $\boldsymbol{H}^{(3)}$ is used only to re-parametrise the factors. In both cases, the constraints condition the temporal evolution and represent the covariates for the models described below.

More precisely, given the dynamics represented in Figure 5.3, we assume that the values of μ_{ij} evolve smoothly between the hours of each day, so the factors $\boldsymbol{f}_1, \dots, \boldsymbol{f}_K$ can be estimated as smooth functions that depend on the hourly constraints. The smoothness was included in the estimation process using a spline-based approach via the `R` package `mgcv` (Wood, 2017). If we consider the matrix $\boldsymbol{L}$ as fixed, then Equation (5.5) can be seen as a *varying coefficient model* (Hastie and Tibshirani, 1993) with Poisson response variable, logarithmic link and (hourly) deterministic temporal covariates given by the matrix $\boldsymbol{H}^{(3)}$. The hourly evolution is estimated using a *thin plate regression spline*, i.e. a low-rank isotropic smoother estimated with truncated eigendecomposition and several desirable properties that can be used with large datasets via computationally efficient algorithms (Wood, 2003).

Analogously, if we assume that the values of μ_{ij} evolve smoothly among the weeks of a year, and we consider the matrix $\boldsymbol{F}$ as fixed, then Equation (5.4) can be seen as a *varying coefficient model*, where the unconstrained loadings $\boldsymbol{B}^{(2)}$ are estimated via a cyclic cubic regression spline. On the other hand, the daily constraints $\boldsymbol{H}^{(1)}$ are included just as piecewise constant function.

The values of $\boldsymbol{L}$ and $\boldsymbol{F}$ are initialised using Singular Value Decomposition (SVD) on matrix $\log \boldsymbol{Y}$, and the two modelling steps are repeated until convergence. We refer to David S Matteson et al. (2011) and Shen and J. Z. Huang (2008a) for a more detailed description of the algorithm behind the estimation process.

5.2.2 The spatial model

As we said at the beginning of this Section, the spatial dimension of the EMS interventions, denoted by $g_t(\boldsymbol{s})$, is modelled using a network-version of a Jones-Diggle corrected weighted kernel estimator (Jones, 1993). The network intensity function is defined following the approach detailed in Rakshit, Davies, et al. (2019), while the weight function, which is used to model the temporal seasonalities and the space-time interactions introduced in Section 5.1 and depicted in Figure 5.5, is based on Zhou (2016) and Zhou and David S. Matteson (2015).

More precisely, given a set of observed time periods (previously denoted by $\mathcal{T}$), a future hour u, and a location $\boldsymbol{s}$ on the network, the weighted kernel estimator can be written as

$$\hat{g}_u(\boldsymbol{s}) = \frac{\sum_{t \in \mathcal{T}} \sum_{i=1}^{y_t} w_{\boldsymbol{s}_i}(t, u) K(\boldsymbol{s} - \boldsymbol{s}_{i,t})}{\sum_{t \in \mathcal{T}} \sum_{i=1}^{y_t} w_{\boldsymbol{s}_i}(t, u)}, \tag{5.6}$$

where $K(\cdot)$ indicates a Gaussian network kernel function and $w_{\boldsymbol{s}_i}(t, u)$ represents the weight associated to the $\boldsymbol{s}_{i,t}$ ambulance intervention, which is described in Section 5.2.2.

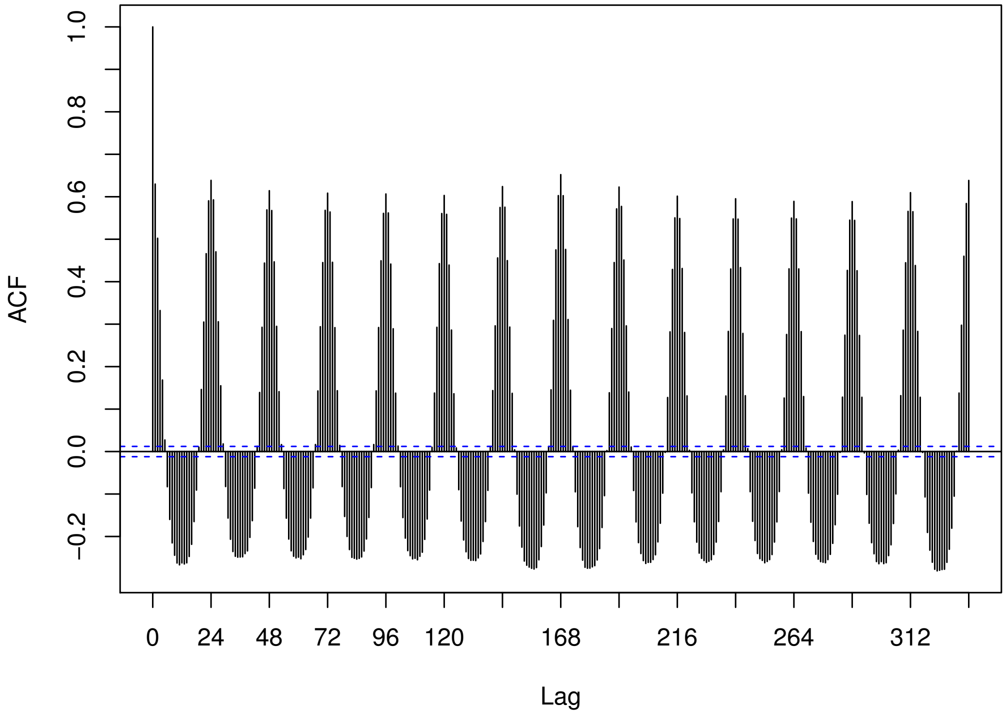


Figure 5.6: Hourly autocorrelation function of EMS interventions. There are some clear hourly, daily, and weekly seasonalities.

The main difference between the proposal in Zhou (2016) and Zhou and David S. Matteson (2015) and the current approach is that we assume $w_{\boldsymbol{s}_i}(t,u)$ depends only on the time periods, i.e. u and t. In contrast, the other authors included a spatial dependence in $w_{\boldsymbol{s}_i}(t,u)$ dividing the planar region under analysis into several areas and defining a different set of weights for each cell. We ignore the spatial dimension since the planar distribution of the street network is not homogeneous, and there is no unambiguous way to divide it. Hence, Equation (5.6) can also be rewritten as

$$\hat{g}_u(\boldsymbol{s}) = \frac{\sum_{t\in\mathcal{T}}\sum_{i=1}^{y_t} w_{\boldsymbol{s}_i}(t,u)K(\boldsymbol{s}-\boldsymbol{s}_{i,t})}{\sum_{t\in\mathcal{T}}\sum_{i=1}^{y_t} w_{\boldsymbol{s}_i}(t,u)} = \frac{\sum_{t\in\mathcal{T}}\sum_{i=1}^{y_t} w(t,u)K(\boldsymbol{s}-\boldsymbol{s}_{i,t})}{\sum_{t\in\mathcal{T}}\sum_{i=1}^{y_t} w(t,u)}, \tag{5.7}$$

since we are assuming that, conditioning on two fixed time periods, all interventions on the road network are linked with the same weights.

Definition of the weight function

As detailed before, the weight function is used to incorporate the space-time interactions into the kernel estimator, creating a non-separability in $\lambda_t(\boldsymbol{s})$, the spatio-temporal intensity.

Considering Figures 5.2 and 5.3 and the EMS counts' hourly AutoCorrelation Function (ACF), which is reported in Figure 5.6, we assume that $w(t,u)$ can be modelled as a function of the time difference between u and t and that it depends on hourly, daily and weekly temporal seasonalities.
The weight function returns the predictive importance of each ambulance intervention given a future time period, and we assume that it can be measured based on the (positive) temporal interactions between two events that are $(u-t)$ time-units apart. Thus, we consider the following functional form for $w(t,u)$, firstly introduced in Zhou and David S. Matteson (2015)

$$w(t,u) = w(u-t) = \rho_1^{u-t} + \rho_2^{u-t} \rho_3^{\sin^2\left(\frac{\pi(u-t)}{24}\right)} \rho_4^{\sin^2\left(\frac{\pi(u-t)}{168}\right)}. \tag{5.8}$$

We can see that it includes a separate coefficient for each seasonal patter: ρ_1 is used to capture the short-term dependence, analogous to AR(1) model, while ρ_3 and ρ_4 measure the daily and weekly seasonalities with periodicity equal to 24 and $24*7 = 168$, respectively. ρ_2 represents a discount factor, which is added to fade-out the daily and seasonal terms that otherwise would never vanish (since they would keep fluctuating between the product of their minimum values and 1). The four parameters are bounded between 0 and 1 to avoid negative weights, exponential growths and difficulties in their interpretations.
The coefficients are estimated implementing an algorithm suggested in Zhou and David S. Matteson (2015). As we mentioned, the objective of the weight function is to approximate the positive temporal correlations between two EMS interventions that are separated by $l = u - t$ time-periods (or lags). Hence, after calculating the empirical hourly ACF of EMS counts up to lag L and taking its positive part, denoted by ACF$^+$, the parameters $\rho_1, \dots, \rho_4$ were estimated by minimising the sum of squared loss between ACF$^+$ and $\rho_0 w(l)$:

$$\underset{\rho_0,\dots,\rho_4}{\operatorname{argmin}} \sum_{l=1}^{L} \left(\text{ACF}^+(l) - \rho_0 w(l)\right)^2 \quad \text{s.t. } 0 \le \rho_j \le 1 \;\forall j = 0,\dots,4 \tag{5.9}$$

The additional coefficient, ρ_0, represents another discount factor without any practical interpretation since it is used only to scale $w(\cdot)$ between 0 and 1, the same as ACF$^+$, while, in its original formulation, the weight function is bounded between 0 and 2.
We choose $L = 672$, which represents four weeks of historical temporal data, while the minimisation problem was solved using the *box-constrained* method implemented in the `R` function `optim()` and defined in Byrd et al. (1995). All parameters were initialised at 0.5.

5.3 Results

In this Section, we present the results obtained after estimating the models described in Section 5.1. All procedures were implemented using the software `R` (R Core Team, 2020) and several external packages. More precisely, as we have already mentioned, the smooth

Table 5.1: Estimates of the Mean Squared Error (MSE) for the temporal model evaluated on the *test* data. The two residuals are defined in Equation (5.10).

Number of factors	MSE with Pearson residuals	MSE with Anscombe residuals
1	1.237250	1.216869
2	1.141232	1.129958
3	1.136537	1.125091
4	1.134588	1.123214
5	1.136711	1.124742
6	1.137685	1.125492

dynamics in the temporal component were estimated using `mgcv` (Wood, 2017), while the network-version of the gaussian weighted kernel defined in Rakshit, Davies, et al. (2019) is implemented in `spatstat` (Baddeley, Rubak, and Turner, 2015). It took approximately 25 minutes to run each iteration of the algorithm for the temporal model using a virtual machine with an Intel Xeon E5-2690 v3 processor, six cores, and 32GB of RAM. On the other side, the functions for estimating the spatial dimension are remarkably efficient and, after approximating the weight function, they usually run in a couple of minutes.

5.3.1 Temporal component

As detailed in Section 5.2.1, the temporal component was estimated using a dynamic latent factor model with deterministic covariates representing hourly, daily and weekly seasonal trends. The temporal effects are included using a set of constraints on the factor and loadings matrices. More precisely, the factors, which should capture the intra-day evolution, are modelled as smooth functions of the hour of the day, while the loadings were constrained according to the day of the week and the week of the year.

The starting point of the algorithm for estimating the matrices $\boldsymbol{F}$ and $\boldsymbol{L}$ is the definition of K, i.e. the number of factors, which, as already mentioned, is chosen using a forecasting perspective. The data at hand represent all EMS interventions from 2015-01-01 to 2018-01-01. Hence, we divided the observations into two groups: the emergency events from 2015-01-01 to 2017-06-30 represent the *training set* of the algorithm, while the predictive performances were evaluated on the second group, i.e. the EMS interventions that occurred from 2017-07-01 to 2017-12-31.

We trained several models with different values of K (from 1 to 6), and we ranked them using the estimates of Mean Squared Error (MSE) criterion on the *test* set, calculated with two types of residuals:

$$\hat{r}_{P,t} = \frac{y_t - \hat{\mu}_t}{\sqrt{\hat{\mu}_t}}; \qquad \hat{r}_{A,t} = \frac{\frac{3}{2}\left(y_t^{2/3} - \hat{\mu}_t^{2/3}\right)}{\hat{\mu}_t^{1/6}}. \tag{5.10}$$

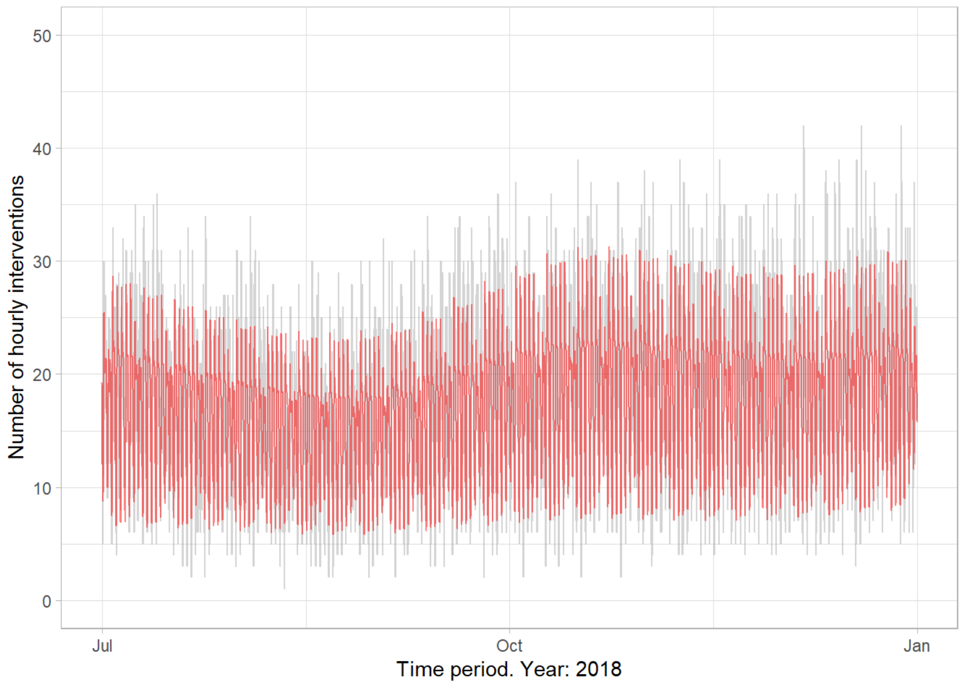
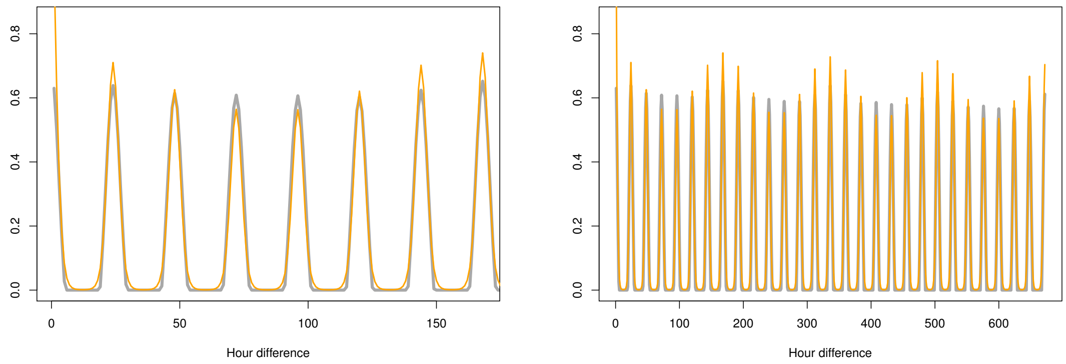


Figure 5.7: Graphical comparison of observed data and factor model's fitted values considering EMS hourly counts from 2017-07-01 to 2017-12-31.

The first one, i.e. $\hat{r}_{P,t}$ represent the Pearson residuals for Poisson GLM or GAM models, while the other ones are the Anscombe residuals (Anscombe et al., 1961). Following David S Matteson et al. (2011), we adopted two different criteria since, as reported in McCullagh and Nelder (1989), the Pearson residuals can be heavily skewed in case of Poisson response variables.

The results are reported in Table 5.1. The optimal value is obtained when the model is run with four factors. It should be pointed out that the observed counts present five missing values from 2015-04-01 at 00:00 to 2015-04-01 04:00:00 which, for each value of K, were imputed using a dynamic latent factor model with the same structure that was trained using the data until 2015-03-31 at 23:00.

Finally, after rerunning the model with $K = 4$ factors on the complete dataset, we checked its predictive accuracy using a graphical approach. More precisely, we calculated the fitted values from 2017-07-01 to 2017-12-31, and we compared them to the observed counts. The result is displayed in Figure 5.7, which shows a good agreement between the two time-series.



(a) One week of lagged counts

(b) Four weeks of lagged counts

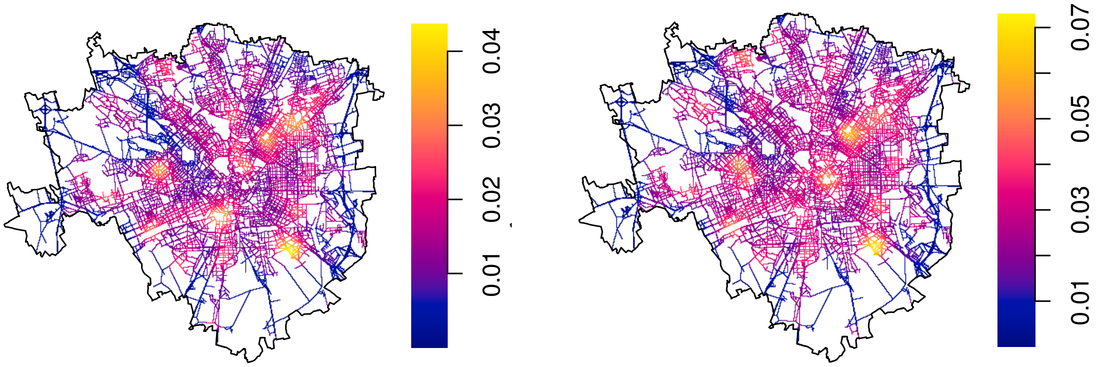
Figure 5.8: The observed positive part of the ACF (grey) and the estimated weight function (orange) considering lagged counts for one (a) and four (b) weeks.

5.3.2 Estimating the weight function

As introduced in Section 5.2.2, the spatial component was estimated combining a network Gaussian smoothing kernel with a weight function that measures the predictive importance of each ambulance intervention. The weights are used to mimic the (positive) interactions between two EMS interventions separated by l temporal lags, replicating the hourly, daily, and weekly seasonalities present in the ACF, which is displayed in Figure 5.6.

As reported in Equation (5.8), the weight function depends on four parameters, that represent the three seasonal components and a discount factor. They were estimated solving the minimisation problem introduced in Equation (5.9). We found that the optimal value of ρ_1 is equal to 0.448, which points out a mildly strong short-term correlation; ρ_3 was found approximately equal to 0.002, while ρ_4 is much higher, being equal to 0.744. The second seasonal parameter, i.e. $\hat{\rho}_3$, indicates that the daily component oscillates between 0.002 and 1. Given the periodic behaviour of sin function, the maximum value of $\hat{\rho}_3^{\sin^2\left(\frac{\pi l}{24}\right)}$, i.e. the daily effect on the weight function, is obtained when the lag l is approximately a multiple of 24, and the minimum is reached when the time difference is close to 12 or its multiples. The value of $\hat{\rho}_4$ points out that the weekly effect is smoother and varies between 0.744 and 1. Similarly, the maximum value of the weekly effect is registered when the lag l is close to 168 or its multiples. The optimal values of ρ_0 and ρ_2, i.e. the two discount factors, were found equal to 0.75 and 0.99, respectively, which says that the daily and weakly seasonalities fade-out slowly.

We display in Figure 5.8 a graphical comparison between the observed positive part of the hourly ACF and the estimated weight function. Figure 5.8a shows one week of lagged counts, while Figure 5.8b shows the complete set of lags up to four weeks. In both cases, the weight function successfully mimics the ACF. The only minor flaws are related to the



(a) 2018-01-01 03:00

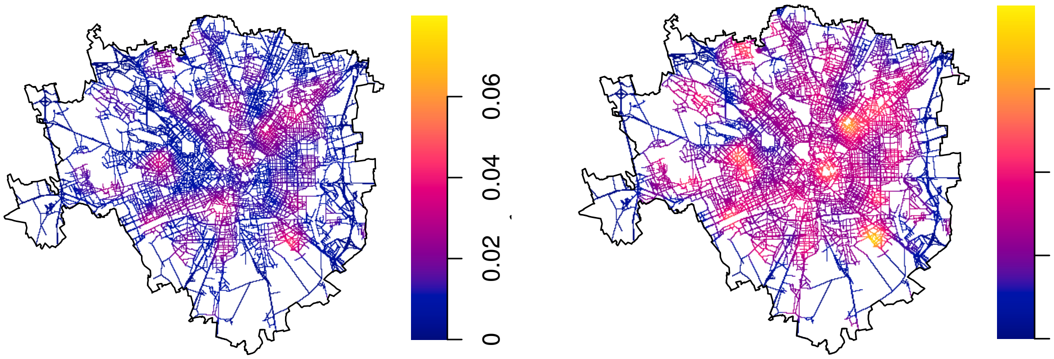
(b) 2018-01-01 15:00

Figure 5.9: Estimates of the spatial intensity function $\hat{g}_u(\boldsymbol{s})$ considering two future time periods: 2018-01-03 at 03:00 (left) and 2018-01-03 at 15:00 (right).

smoothed versions of daily and weekly seasonalities that look respectively slightly weaker and stronger than their counterparts.

5.3.3 Spatial component

After estimating the weight function, we applied Equation (5.7) to obtain the predicted spatial intensity $\hat{g}_u(\boldsymbol{s})$ for a future time period u. In particular, considering that the data at hand report the EMS interventions from 2015-01-01 at 00:00 to 2018-01-01 at 00:00, we decided to forecast $\hat{g}_u(\boldsymbol{s})$ considering two future time periods: 2018-01-03 at 03:00 and 2018-01-03 at 15:00. The results are reported in Figure 5.9.
The first map shows that the EMS interventions are scattered in several parts of the municipality, and highlights some roads of the network that could be linked with nightlife areas (such as Porta Genova or Colonne di San Lorenzo). The second map draws attention to the zones close to Duomo and other significant working places, while there is no emphasis on the nightlife areas. In both cases, the main train station (Stazione Centrale), a popular square (Piazzale Loreto) and several retirement houses (such as Pio Albergo Trivulzio) stand out in the spatial intensity function's estimates.
Maps in Figure 5.9 are drawn using different scales, which implies that they cannot be directly compared. Hence, to emphasise the spatio-temporal dynamics of the EMS dispatches, we report in Figure 5.10 the same maps created using a common scale. Although they clearly point out the same areas as before, they also highlight the temporal patterns in the average hourly number of ambulance dispatches considering that most EMS interventions occur during the morning on the first part of the afternoon (see Figure 5.3).



(a) 2018-01-01 03:00

(b) 2018-01-01 15:00

Figure 5.10: Estimates of the spatial intensity function $\hat{g}_u(\boldsymbol{s})$ considering two future time periods: 2018-01-03 at 03:00 (left) and 2018-01-03 at 15:00 (right). The two maps are analogous to Figure 5.9, but they are reported using the same scale.

5.4 Conclusions and future works

This paper investigated the spatio-temporal distribution of ambulance interventions that occurred from 2015-01-01 to 2017-21-31 in the municipality of Milan, one of the most hectic, vibrant, and congested cities in Europe. We assumed that the emergency events are the realisation of a point pattern on a linear network, representing the most important streets of the city. More precisely, the spatial support was downloaded from Open Street Map, and several pre-processing steps were applied to manipulate the data and create a fully-connected graph. The street network and the EMS data are visualised in Figure 5.4. Preliminary analysis, summarised in Section 5.1, revealed that the temporal evolution presents several types of seasonalities, due to hourly, daily and weekly patterns. The spatial distribution of the interventions showed that the locations tend to concentrate along the street network and seem to be clustered near several popular and busy areas (like Stazione Centrale, Porta Genova or Corvetto). Moreover, we noticed the presence of space-time interactions in the hourly distribution of the events: in the morning and in the afternoon the ambulance dispatches look concentrated in the city centre, while, they are more spread all around the city in the night.

We divided the intervention times into discrete intervals of one hour, and we assumed that, independently for each time span, the EMS interventions follow a non-homogeneous Poisson process with a non-separable intensity function, defined in Equation (5.1). The temporal component was approximated via a dynamic latent factor model with deterministic covariates representing the seasonal trends mentioned above. The hourly and weekly effects were modelled using a spline approach, to grasp the smooth dynamics noticed in Figures 5.2

and 5.3. The spatial dimension was estimated using a non-parametric Gaussian network kernel function that was combined with a set of weights that capture the space-time interactions. The weight function, denoted by $w(t, u)$, measures the (positive) temporal correlation between two ambulance dispatches separated by $l = u - t$ temporal lags. This function was defined using four parameters that represent the temporal dynamics registered in EMS counts. They were estimated solving the minimisation problem in Equation (5.9).

The temporal model was trained using different values of K, i.e. the number of factors, and we found that the best predictive performances occur when $K = 4$. The fitted values seem to approximate the EMS counts reasonably well, as reported in Figure 5.7. We exemplified the spatial approach estimating the weight function and the spatial intensity considering two future time periods: one during the first hours of 2018-01-03 and another in the first part of the afternoon of the same day. Figures 5.8 and 5.9 display the estimated intensities, showing that the spatial and spatio-temporal patterns are captured effectively.

The ideas presented in this paper could be extended in several ways. First, we might take into account the multivariate nature of the EMS data, introduced in Section 5.1, estimating the relative risk of EMS interventions, i.e. the ratio of the rates of occurrences for different severity levels. This approach was recently explored in McSwiggan, Baddeley, and Nair (2020) for the multivariate analysis of car crashes on a linear network. Second, we plan to focus on the second-order characteristics of a spatio-temporal point process, developing methods and packages for estimating the K function in case of non-separable intensities (Ripley, 1977). This type of problems was recently studied by Rakshit, Baddeley, and Nair (2019), developing several algorithms and code for efficient estimation of the K function on a linear network that ignores the temporal component, and M. M. Moradi and Mateu (2020), where the authors proposed theoretical models and computational implementations to estimate the network K function in case of a cyclic separable spatio-temporal intensity. Finally, classical point pattern models such as Log-Gaussian Cox processes or self-exciting processes could be extended to a network support. These developments also represent the starting point for the inclusion of relevant spatial covariates measured on the road network, such as urban traffic (given by motorcars and public transports), environmental measurements, socio-demographic indices, or the road segment's type, that can be directly obtained from Open Street Map.

Chapter 6

Conclusions and Future Developments

Mamma mia la monnezza c'ho fatto.

Renè Ferretti su *Caprera* - Boris (1x03)

The present manuscript focuses on the analysis of structures and models for spatial data lying on or alongside road networks. Starting from computational and practical problems related to transportation, road safety, and emergency interventions, we presented various statistical techniques based on lattice and point-pattern approaches.

The first Chapter briefly introduced the basic concepts and data sources related to the analysis of spatial data lying on street networks. We mentioned some of the peculiar computational and methodological problems that make road network analysis challenging and, at the same time, exciting.

Chapter 2 reviewed software implementations for representing street network data, focusing on two `R` packages named `stplanr` and `dodgr`. Their basic structure and relative pros and cons were discussed using several increasingly complex examples taken from Open Street Map. The particular properties of road network data were highlighted, showcasing code for estimating shortest paths and centrality measures. Moreover, we would like to point out that the ideas summarised in Chapter 2 represented one of the starting points for developing two new `R` packages. The first one, named `osmextract`, is extensively detailed in Appendix C, while the second one, named `sfnetworks`, represents an alternative approach for spatial networks analysis (Van der Meer et al., 2021). It links the spatial and graph dimensions using `sf` and `tidygraph` objects, extending the `stplanr` approach in several ways and fixing some of its inherent problems. In the next weeks, we plan to submit both packages to CRAN and keep working on their development.

Chapter 3 proposed a Dynamic Zero-Inflated Poisson model to compare car crashes determinants and monitor accidents exposure considering an extremely detailed urban road network. We illustrated several techniques to summarise and combine spatially misaligned open-source data and defined a road safety and a road risk index that provide local information at the segment level. The proposed model is quite simple from a statistical perspective, but it represents an original attempt to develop a localised road safety index considering a street network that covers a large area. We plan to extend the social dimension of the paper, developing a causal approach for identifying the so-called black spots and

their determinants.
Chapter 4 explored several increasingly complex multivariate Bayesian hierarchical models to analyse the spatial distribution of severe and slight car crashes that occurred in the road network of Leeds from 2011 to 2018. The best model included a multivariate CAR spatial random effect with a different autoregression coefficient for each severity level. We tested its robustness against different hyperpriors and adjacency matrices, and the results are detailed in Appendix B. We proposed a novel procedure to contract a road network and test the severity of MAUP (Modifiable Areal Unit Problem) for spatial models developed using a network-lattice support. In the next months, we plan to improve the methodology for estimating traffic exposure at the network level, combining data from official sources with graph metrics.
Finally, Chapter 5 introduced some preliminary results related to the development of an inhomogeneous spatio-temporal Poisson process to analyse the distribution of ambulance interventions in Milan. A non-separable first-order intensity function was considered to capture the space-time interactions noticed during the preliminary analysis. The algorithm for estimating the spatial component via a weighted kernel function was exemplified considering two future time points, and it seems that the main spatial and temporal trends were successfully modelled. At the time of writing, we are developing diagnostics for testing the predictive accuracy. Moreover, in the next months we plan to extend the existing ideas in two directions:

1. develop a network extensions of classical point process models, such as Log-Gaussian Cox processes or Self-exciting processes;
2. add relevant spatio-temporal covariates to the EMS interventions model, such as population density, urban traffic, or commuting flows.

Maybe this is a little pretentious, but I sincerely hope that this manuscript may represent a useful starting point towards the development of innovative software and statistically principled methods for geo-referenced data on spatial networks.

Formal Acknowledgements

Map data copyrighted OpenStreetMap contributors and available from
https://www.openstreetmap.org.

Appendix A

A note on Integrated Nested Laplace Approximation (INLA)

A.1 Introduction

The *Integrated Nested Laplace Approximation* (INLA) is a deterministic statistical methodology to perform approximate Bayesian inference using analytical approximations and numerical integrations (Håvard Rue, Martino, and Chopin, 2009). It represents a fast and accurate alternative to the traditional MCMC methods (Robert and Casella, 2013) for a particular class of models named *Latent Gaussian Models* (LGM), which are introduced below.

The MCMC approach is a versatile tool to handle complex and general Bayesian models. However, even if the inference is usually handled by powerful software such as `WinBugs` (David Spiegelhalter et al., 2003), `JAGS` (Plummer, 2012) or `stan` (Carpenter et al., 2017), the MCMC step requires a large amount of CPU and it is very time consuming. These two problems are extremely relevant for large[1] spatial and temporal models, especially for parameter's tuning and comparing alternative specifications.

Hence, considering the large amounts of spatial data available worldwide, the INLA methodology and the corresponding `R` package (also named `INLA`) are getting more and more popular, which lead to a widespread adoption in several research fields such as measurement error models (Muff et al., 2015), network meta-analysis (Sauter and Held, 2016), outbreak detection (Salmon et al., 2015), disease mapping (Moraga, 2019), and, more generally, spatial and spatio-temporal models (Lindgren, Håvard Rue, et al., 2015) analogous to the methods presented in Chapter 4.

In the following Sections, we briefly review the basic ideas and the key components that define the INLA approach. Excellent introductions and comprehensive books with many practical examples are provided by (Blangiardo and Cameletti, 2015; Cameletti, 2017; Gómez-Rubio, 2020; Moraga, 2019; Håvard Rue, Riebler, et al., 2017), amongst others. We refer to these works for a more detailed presentation.

[1] The term *large* refers not only to the dimensionality of the dataset, but also to the complexity and flexibility of the model.

A.2 Latent Gaussian Models

As mentioned above, the INLA methodology is restricted to a particular class of models called Latent Gaussian Models (LGM), which includes a wide variety of statistical approaches like GLM, GLMM (Generalised Linear Mixed Models), GAM, GAMM (Generalised Additive Mixed Models), Survival models, Geostatistical model, and logGaussian Cox Processes (Agresti, 2015; P. J. Diggle, 2013; P. J. Diggle, Tawn, and Moyeed, 1998; Fong, Håvard Rue, and Wakefield, 2010; Miller Jr, 2011; Wood, 2017).
Indicating by $\boldsymbol{x}$ the structure of a Gaussian random field, usually specified as

$$\boldsymbol{x}|\boldsymbol{\theta}_2 \sim N\left(\boldsymbol{\mu}(\boldsymbol{\theta_2}), \boldsymbol{Q}^{-1}(\boldsymbol{\theta}_2)\right),$$

and by $\boldsymbol{y} = (y_1, \ldots, y_n)$ a set of observed data conditionally independent given $\boldsymbol{x}$ and $\boldsymbol{\theta}_1$, the Latent Gaussian Models assume that

$$\boldsymbol{y}|\boldsymbol{x}, \boldsymbol{\theta}_1 \sim p(\boldsymbol{y}|\boldsymbol{x}, \boldsymbol{\theta}_1) = \prod_i p(y_i|x_i, \boldsymbol{\theta}_1).$$

The Gaussian random field $\boldsymbol{x}$ describes the model's dependence structure, $\boldsymbol{\theta}_1$ and $\boldsymbol{\theta}_2$ are sets of hyperparameters, $\boldsymbol{\mu}(\boldsymbol{\theta}_2)$ denotes the mean vector, and $\boldsymbol{Q}^{-1}(\boldsymbol{\theta}_2)$ the precision matrix Furthermore, it is usually assumed that $\boldsymbol{x}$ represents a Gaussian *Markov* Random Field (GMRF), which means that the elements of $\boldsymbol{x}$ are *conditionally independent* given the remaining ones, i.e. $x_i \perp\!\!\!\perp x_j|\boldsymbol{x}_{-ij}$ (Havard Rue and Held, 2005). In the previous equation, $\boldsymbol{x}_{-ij}$ denotes the random field without elements i and j.
The conditional independence property usually implies that the precision matrix $\boldsymbol{Q}^{-1}$ is very sparse. Hence, the GMRF hypothesis has important consequences and relevant computational benefits since it implies INLA can take advantage of efficient algorithms for analytical approximations involving sparse matrices.
A LGM can also be specified using a three-level hierarchical structure, analogous to the approach adopted in Chapter 4 and detailed in Equations (4.1) and (4.2):

$$\begin{aligned}
&\boldsymbol{y}|\boldsymbol{x}, \boldsymbol{\theta}_1 \sim \prod_{i \in \mathcal{I}} p(y_i|\boldsymbol{x}, \boldsymbol{\theta}_1) && \text{(the likelihood)} \\
&\boldsymbol{x}|\boldsymbol{\theta}_2 \sim N\left(\boldsymbol{\mu}(\boldsymbol{\theta_2}), \boldsymbol{Q}^{-1}(\boldsymbol{\theta}_2)\right) && \text{(the GMRF assumption)} \\
&\boldsymbol{\theta} = (\boldsymbol{\theta}_1, \boldsymbol{\theta}_2) \sim p(\boldsymbol{\theta}) && \text{(the hyperpriors)}
\end{aligned} \tag{A.1}$$

The hyperparameters $\boldsymbol{\theta}$ control the latent Gaussian field and the distribution of $(y_1, \ldots, y_n)$. The INLA framework usually assumes that the dimension of $\boldsymbol{\theta}$ is small, typically from 2 to 5 and not exceeding 20. For example, following the terminology introduced in Chapter 4, the *baseline models* included four hyperparameters (i.e. the variances of the two unstructured errors and the two spatial components), while the most general model considered three extra hyperparameters. On the other hand, the GMRF can be very large (e.g. 10^8 components, most not observed).

Additive Models

If we assume that the observations $\boldsymbol{y}$ depend on a parameter $\boldsymbol{\varphi}$, which can be linked to a set of covariates and random effects through an appropriate link function, then we can re-establish a relationship between classical GAMM models and LGM. More precisely, if we denote the linear predictor as $g(\boldsymbol{\varphi}) = \boldsymbol{\eta}$, then the classical additive model setup can be written as:

$$g(\varphi_i) = \eta_i = \beta_0 + \sum_j \beta_j z_{ij} + \sum_k f_k.$$

The parameter β_0 is the overall intercept, $\boldsymbol{z}$ are fixed covariates with linear effects, while $\{f_k\}$ indicates a set of functions defined using covariates or indices representing, for example, smoothing splines, unstructured errors or temporal and spatial effects (such as the CAR and MCAR distributions presented in Section 4.3). If the model components are assumed to be a priori independent and the coefficients of fixed effects have a Gaussian prior, then the joint distribution of

$$\boldsymbol{x} = (\boldsymbol{\eta}, \boldsymbol{\beta}, \boldsymbol{f}_1, \boldsymbol{f}_2, \dots)$$

is Gaussian, yielding the GMRF structure of hierarchical LGM introduced in Equation (A.1). As mentioned before, the critical assumption for computational efficiency is that the dimension of $\boldsymbol{\theta}$ remains small even when the latent fields is large.

A.3 The Integrated Nested Laplace Approximation

As the name suggests, one of the key ingredients of INLA methodology is the Laplace approximation (LA) (Barndorff-Nielsen, 1989). The main idea behind LA is to approximate the integral of a distribution function $f(x)$ using a Taylor's series expansion of its logarithm centred around its mode, denoted as $x^* = \operatorname{argmax}\log(f(x))$:

$$\begin{aligned}\log(f(x)) &\simeq \log f(x^*) + (x - x^*)\frac{\partial \log f(x)}{\partial x}\bigg|_{x=x^*} + \frac{(x-x^*)^2}{2}\frac{\partial^2 \log f(x)}{\partial x^2}\bigg|_{x=x^*} \\ &= \log f(x^*) + \frac{x - x^*}{2} f''(x^*)\end{aligned}$$

Then, the integral of interest is approximated matching the mean and the curvature of a Gaussian distribution:

$$\int f(x)\ \mathrm{d}x = \int \exp\left(\log(f(x))\right)\ \mathrm{d}x \simeq \int \exp\left[\log f(x^*) + \frac{1}{2}(x - x^*)^2 f''(x^*)\right]\ \mathrm{d}x$$

The previous equation implies that, under LA, the distribution of $f(x)$ is approximated as Gaussian with mean $f(x^*)$ and variance $-\frac{1}{f''(x^*)}$.

The objective of Bayesian Inference is to compute the posterior distribution of the latent field, $\boldsymbol{x}$, and the hyperparameters, $\boldsymbol{\theta}$, which, starting from the LGM hierarchical structure defined in Equation (A.1), can be expressed as

$$p(\boldsymbol{x}, \boldsymbol{\theta}|\boldsymbol{y}) \propto p(\boldsymbol{\theta})p(\boldsymbol{x}|\boldsymbol{\theta}) \prod_i p(y_i|x_i, \boldsymbol{\theta})$$

Nevertheless, considering the assumptions stated before on the behaviour and the dimensionality of $\boldsymbol{\theta}$ and $\boldsymbol{x}$, INLA does not attempt to estimate the posterior distribution written before, but focuses on the following marginals:

$$p(\theta_j|\boldsymbol{y}) = \int \pi(\boldsymbol{\theta}|\boldsymbol{y}) \; \mathrm{d}\boldsymbol{\theta}_{-j}, \qquad j = 1, \ldots, \dim(\boldsymbol{\theta}); \tag{A.2}$$

$$p(x_i|\boldsymbol{y}) = \int p(x_i, \boldsymbol{\theta}|\boldsymbol{y})p(\boldsymbol{\theta}|\boldsymbol{y}) \; \mathrm{d}\boldsymbol{\theta}, \qquad i = 1, \ldots, \dim(\boldsymbol{x}). \tag{A.3}$$

Approximating the Posterior Marginals for the Hyperparameters

The first step consists in creating an approximation for the joint posterior distribution of the hyperparameters using the following derivation:

$$\begin{aligned}
p(\boldsymbol{\theta}|\boldsymbol{y}) &= \frac{p(\boldsymbol{x}, \boldsymbol{\theta}|\boldsymbol{y})}{p(\boldsymbol{x}|\boldsymbol{\theta}, \boldsymbol{y})} && \text{Def. of Cond. Prob.} \\
&= \frac{p(\boldsymbol{y}|\boldsymbol{x}, \boldsymbol{\theta})p(\boldsymbol{x}, \boldsymbol{\theta})}{p(\boldsymbol{y})} \frac{1}{p(\boldsymbol{x}|\boldsymbol{\theta}, \boldsymbol{y})} && \text{Bayes Thorem} \\
&= \frac{p(\boldsymbol{y}|\boldsymbol{x}, \boldsymbol{\theta})p(\boldsymbol{x}|\boldsymbol{\theta})p(\boldsymbol{\theta})}{p(\boldsymbol{y})} \frac{1}{p(\boldsymbol{x}|\boldsymbol{\theta}, \boldsymbol{y})} && \text{Def. of Cond. Prob} \qquad \text{(A.4)} \\
&\propto \frac{p(\boldsymbol{y}|\boldsymbol{x}, \boldsymbol{\theta})p(\boldsymbol{x}|\boldsymbol{\theta})p(\boldsymbol{\theta})}{p(\boldsymbol{x}|\boldsymbol{\theta}, \boldsymbol{y})} && \text{Since we are conditioning on } \boldsymbol{y} \\
&\simeq \left. \frac{p(\boldsymbol{y}|\boldsymbol{x}, \boldsymbol{\theta})p(\boldsymbol{x}|\boldsymbol{\theta})p(\boldsymbol{\theta})}{\tilde{p}(\boldsymbol{x}|\boldsymbol{\theta}, \boldsymbol{y})} \right|_{\boldsymbol{x}=\boldsymbol{x}^*(\boldsymbol{\theta})} := \tilde{p}(\boldsymbol{\theta}|\boldsymbol{y}) && \text{Laplace approximation}
\end{aligned}$$

More precisely, in Equation (A.4), $\tilde{p}(\boldsymbol{x}|\boldsymbol{\theta}, \boldsymbol{y})$ denotes the Laplace approximation of $p(\boldsymbol{x}|\boldsymbol{\theta}, \boldsymbol{y})$ and $\boldsymbol{x}^*(\boldsymbol{\theta})$ represents its mode for a given $\boldsymbol{\theta}$. Then, because $\boldsymbol{\theta}$ is of low dimension, the desired marginals, i.e. $p(\theta_j|\boldsymbol{y})$, can be directly derived from the approximated joint posterior distribution using a low number of evaluation points (Martins et al., 2013). Usually, INLA adopts variance-stabilizing corrections and Fisher-transform of correlations to obtain better-behaved posterior densities and improve the Laplace approximation in the last step of Equation (A.4). Sauter and Held (2016) developed an additional correction term to further improve the Laplace approximation.

Approximating the Posterior Marginals for the Latent Field

The second step is related to the approximation of the posterior marginal distributions for the GMRF, defined in Equation (A.3). The integral is solved numerically using a finite weighted sum:

$$\tilde{p}(x_i|\boldsymbol{y}) = \sum_{\boldsymbol{\theta}^*} \tilde{p}(x_i|\boldsymbol{\theta}^*, \boldsymbol{y})\tilde{p}(\boldsymbol{\theta}^*|\boldsymbol{y})\Delta(\boldsymbol{\theta}^*), \tag{A.5}$$

where $\tilde{p}(x_i|\boldsymbol{\theta}^*, \boldsymbol{y})$ represents an estimate of $p(x_i|\boldsymbol{y}, \boldsymbol{\theta})$, $\tilde{p}(\boldsymbol{\theta}^*|\boldsymbol{y})$ are the approximate posterior marginals obtained from Equation (A.4), $\boldsymbol{\theta}^*$ denotes a set of integration points, and $\Delta(\boldsymbol{\theta}^*)$ the corresponding weights.

INLA implements several strategies for defining the integrations points and the corresponding weights, but the most relevant are named *grid strategy* and *composite central design strategy*. The former scheme is usually more precise than the latter, but it has a computational cost that grows exponentially in the dimension of $\boldsymbol{\theta}$. Hence, it is recommended only when $\dim \boldsymbol{\theta} \leq 4$. The *composite central design strategy* involves a spherical approximation of the integral such that the integration points are located on the level set for the joint posterior of $\boldsymbol{\theta}$ (Box and Wilson, 1951).

The default behaviour for approximating $p(x_i|\boldsymbol{y}, \boldsymbol{\theta})$ is the so-called *Simplified Laplace Approximation*, which is based on a Taylor expansion of Laplace approximation, with a linear and a cubic correction term:

$$\log p(x_i|\boldsymbol{\theta}, \boldsymbol{y}) \simeq -\frac{1}{2}x_i^2 + b_i(\boldsymbol{\theta})x_i + \frac{1}{6}c_i(\boldsymbol{\theta})x_i^3.$$

This approach is the default one, since it represents a reasonable trade-off between accuracy and computational speed. INLA implements two other integration strategies. The first one, named *Gaussian*, directly approximates $p(x_i|\boldsymbol{y}, \boldsymbol{\theta})$ using a Gaussian distribution and a Cholseky decomposition for the precision matrix. Although it is extremely fast to compute, this approximation is usually not very accurate. The second one is called *Laplace* and it directly adopts the Laplace approximation, following a similar derivation as in Equation (A.4). The second scheme works really well, but it is usually intractable for large models since it involves heavy computations which must be repeated several times (up to $\dim(\boldsymbol{x})$, which can be very large).

A.4 Conclusion

Now we can summarise the reasons why the methodology was named *Integrated Nested Laplace Approximation*:

1. The *Laplace Approximation* is the basis for deriving the posterior marginal distributions;
2. The posterior marginals of $\boldsymbol{x}$, the latent field, involve a Laplace approximation for $p(\boldsymbol{\theta}|\boldsymbol{y})$. Hence, the two computing steps are *nested*.

3. The analytical approximations are obtained by numerical *Integration*, as reported, for example, in Equation (A.5).

To conclude, it should also be noticed that INLA can provide estimates of additional and useful quantities such as the posterior linear predictors and the predictive densities (also explored in Section 4 and Appendix B), the Deviance Information Criterion (David J Spiegelhalter et al., 2002), and the Watanabe-Akaike infromation criterion (Watanabe, 2010).

Appendix B

Supplementary materials related to the paper presented in Chapter 4

B.1 Exploring the temporal dimension

Following the procedures detailed in Section 4.2, we estimated, for each year, the number of car crashes that occurred in each segment of the city network. We display the results in Figure B.1, where the red lines represent the segments that registered at least one car crash for a given year. We notice that the spatial distribution looks more or less the same for all years. Moreover, we calculated that approximately 40% of road segments reported zero counts for 2011-2018, while another 40% of road segments registered two or more crashes, which imply that the spatial distribution is constant among different years. For these reasons we preferred to ignore the temporal dimension in Chapter 4.

B.2 Probability Integral Transform values

The *Probability Integral Transform* (PIT) values (or *posterior predictive p-values*) represent a common criterion for criticism of a Bayesian Hierarchical model. They measure, for each elementary unit, the probability that a new observation is lower or equal than the observed value of the corresponding response variable (Gelman, Meng, and H. Stern, 1996; H. S. Stern and N. Cressie, 2000):

$$\text{PIT}_i = \mathbb{P}(Y_i \leq y_i^{\text{obs}} | \boldsymbol{y}).$$

The term $\boldsymbol{y}$ represents the vector of observed counts. If the model fits the observations well, then the histogram of PIT values should resemble a uniform distribution between 0 and 1. We refer to Boulieri et al. (2017) for an application of PIT criterion to road safety data.

PIT values may be adjusted in case of discrete counts using a continuity correction (E. Marshall and Spiegelhalter, 2003):

$$\text{PIT}_i = \mathbb{P}(Y_i < y_i^{\text{obs}} | \boldsymbol{y}) + \frac{1}{2}\mathbb{P}(Y_i = y_i^{\text{obs}} | \boldsymbol{y}).$$

INLA implements a leave-one-out cross-validated version of PIT values (Gómez-Rubio,

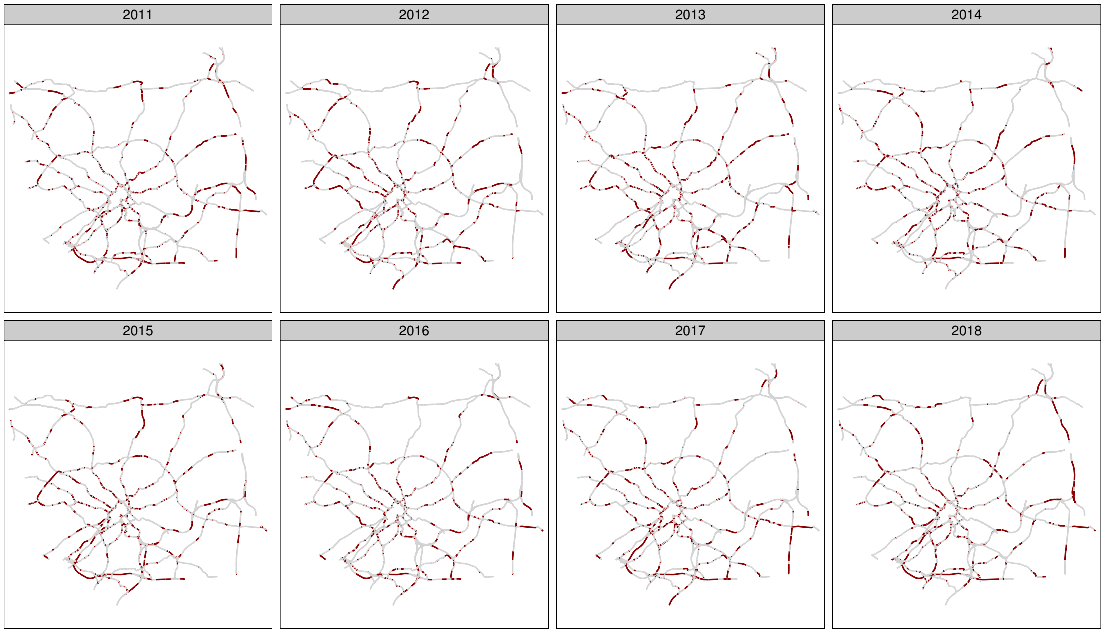


Figure B.1: Spatio-temporal representation of car crashes counts for Leeds city network. The red lines represent those segments that registered at least one car accident for a given year.

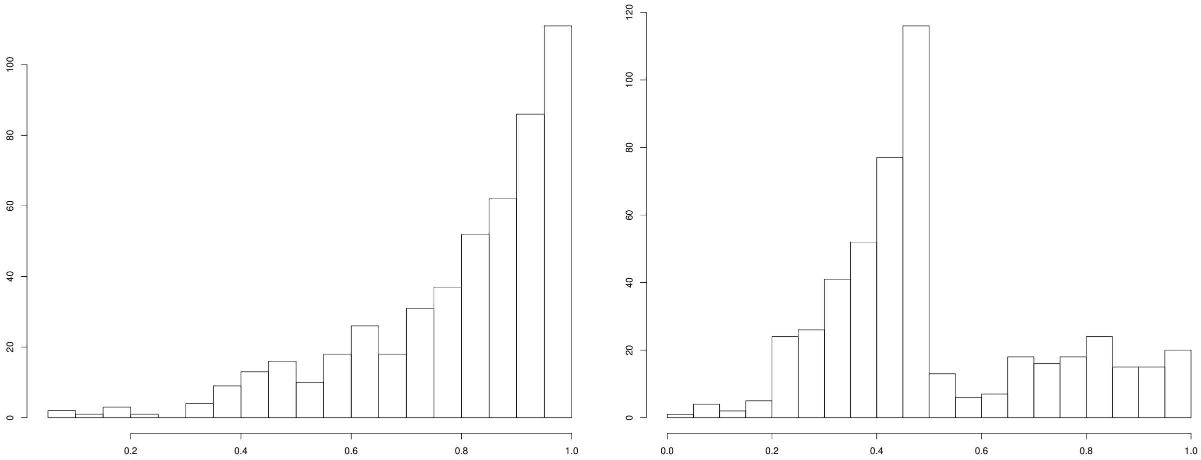
2020; Held, Schrödle, and Håvard Rue, 2010):

$$\mathrm{PIT}_i = \mathbb{P}(Y_i \leq y_i^{\mathrm{obs}} | \boldsymbol{y}_{-i}),$$

where the vector $\boldsymbol{y}_{-i}$ denotes the observed counts minus the ith observation. These quantities can also be adjusted using a continuity correction.

Unfortunately, PIT values do not seem to adapt well to a Poisson regression with sparse counts. A small simulation study was implemented to exemplify this behaviour using the `R` software and the `INLA` package (R Core Team, 2020; Håvard Rue, Riebler, et al., 2017). The `R` code is reported in Listing B.1. The simulation design is as follows. First, we sampled $n = 500$ Poisson random variables with mean $E_i \exp(\lambda_i)$, where $E_i \sim \mathrm{Uniform}(0, 1)$ and $\lambda_i \sim N(-0.33, 1)$ (see `R` code from line five to ten). Approximately 63% of simulated counts are equal to 0, which is close to the proportion of segments that registered no slight or severe car crash in our dataset.

Then, we estimated a Poisson regression model specifying the true random mechanism behind $\boldsymbol{y}$ (see lines thirteen to twenty of the R code) and calculated the leave-one-out cross-validated PIT values. The histogram of PIT values obtained without the continuity correction is reported in Figure B.2a, whereas the histogram of the PIT values with the



(a) Histogram of PIT values.

(b) Histogram of PIT values after continuity correction.

Figure B.2: Histograms of PIT values obtained running a Bayesian hierarchical model with Poisson responses. Check the code behind the simulation in Listing B.1.

continuity correction is reported in Figure B.2b. Both graphs are clearly far from uniform. We argue that this is related to the sparseness of simulated counts, since we found that the problem gets mitigated for less sparse Poisson random variables. We do not report these further results in detail.

Listing B.1: The R code used to perform the simulation study and to estimate the PIT values, whose distributions are reported in Figures B.2a and B.2b.

```
# packages
library(INLA)

# fake data
set.seed(07102020)
n <- 500
x <- rnorm(n)
lambda <- -0.33 + x
E <- runif(n)
y <- rpois(n, E * exp(lambda))

# run INLA model
inla(
formula = y ~ x,
family = "poisson",
data = list(y = y, x = x, E = E),
offset = log(E),
control.compute = list(cpo = TRUE),
```

```r
control.inla = list(strategy = "laplace", int.strategy = "grid")
)
```

B.3 The R code for estimating model (G)

We report in Listing B.2 the R code used to estimate model (G).

Listing B.2: The R code that defines model (G).

```r
model_G <- function(
  cmd = c(
    "graph", "Q", "mu", "initial", "log.norm.const", "log.prior", "quit"
  ),
  theta = NULL
) {
  interpret.theta <- function() {
    #> Function for changing from internal scale to external scale

    #> First k parameters are the autocorrelation parameters
    alpha <- vapply(
      theta[1:k],
      function(x) alpha.min + (alpha.max - alpha.min) * stats::plogis(x),
      numeric(1)
    )

    #> The next k parameters are the marginal precisions
    mprec <- vapply(
      theta[(k + 1):(2 * k)],
      exp,
      numeric(1)
    )

    #> the last (k * (k - 1)) / 2 are the correlation parameters
    #> ordered by columns.
    corre <- vapply(
      theta[(2 * k + 1):(k * (k + 3) / 2)],
      function(x) 2 * stats::plogis(x) - 1,
      numeric(1)
    )

    param <- c(alpha, mprec, corre)

    #> length non-diagonal elements
    n <- (k - 1) * k / 2

    #> intial matrix with 1s at the diagonal
    M <- diag(1, k)

    #> Adding correlation parameters (lower.tri) and (upper.tri)
    M[lower.tri(M)] <- param[(2 * k + 1):(k * (k + 3) / 2)]
    M[upper.tri(M)] <- t(M)[upper.tri(M)]

    #> Preparing the st. dev matrix
    st.dev <- sqrt(1 / param[(k + 1):(2 * k)])

    #> Matrix of st. dev.
    st.dev.mat <- matrix(st.dev, ncol = 1) %*%
      matrix(st.dev, nrow = 1)

    #> Final inverse matrix
    M <- M * st.dev.mat

    #> Inverting matrix
    PREC <- solve(M)

    return(list(alpha = alpha, param = param, VACOV = M, PREC = PREC))
  }

  #> Graph of precision matrix; i.e., a 0/1 representation
  #> of precision matrix
  graph <- function() {
    #> Build the blockdiagonal matrix
    Rs <- vector("list", length = k)
    D_W <- Matrix::Diagonal(n = nrow(W), x = Matrix::rowSums(W))

    for (j in seq_len(k)) {
      #> I choose alpha = 0.5 for no particular reason
      R <- t(chol(D_W - 0.5 * W))
      Rs[[j]] <- R
    }
    Bdiag_R <- Matrix::bdiag(Rs)

    #> Build the central part of the matrix product
    central_block <- Matrix::kronecker(
      matrix(1, nrow = k, ncol = k),
      Matrix::Diagonal(nrow(W), 1)
    )

    G <- Bdiag_R %*% central_block %*% t(Bdiag_R)
    G
  }

  Q <- function() {
    #> Parameters in model scale
    param <- interpret.theta()

    #> Build the blockdiagonal matrix
    Rs <- vector("list", length = k)
    D_W <- Matrix::Diagonal(n = nrow(W), x = Matrix::rowSums(W))
```

```r
    for (j in seq_len(k)) {
      R <- t(chol(D_W - param$alpha[j] * W))
      Rs[[j]] <- R
    }
    Bdiag_R <- Matrix::bdiag(Rs)

    #> Build the central part of the matrix product
    central_block <- Matrix::kronecker(
      param$PREC,
      Matrix::Diagonal(nrow(W), 1)
    )

    Q <- Bdiag_R %*% central_block %*% t(Bdiag_R)
    Q
  }

  #> Mean of model
  mu <- function() {
    return(numeric(0))
  }

  log.norm.const <- function() {
    #> return the log(normalising constant) for the model
    val <- numeric(0)
    return(val)
  }

  log.prior <- function() {
    #> return the log-prior for the hyperparameters.
    #> Uniform prior in (alpha.min, alpha.max) on model scale
    param <- interpret.theta()

    #> log-Prior for the autocorrelation parameter
    val <- 0
    for (j in 1:k) {
      val <- val - theta[j] - 2 * log(1 + exp(-theta[j]))
    }

    #> Whishart prior for joint matrix of hyperparameters
    val <- val + log(
      MCMCpack::dwish(
        W = param$PREC,
        v = k,
        S = diag(rep(lambda_wish, k))
      )
    )
    #> This is for precisions
    val <- val + sum(theta[(k + 1):(2 * k)])
    #> This is for correlation terms
    val <- val + sum(
      log(2) + theta[(2 * k + 1):(k * (k + 3) / 2)] -
      2 * log(1 + exp(theta[(2 * k + 1):(k * (k + 3) / 2)]))
    )

    return(val)
  }

  initial <- function() {
    #> return initial values
    #> The Initial values form a diagonal matrix
    return(c(rep(0, k), rep(log(1), k), rep(0, (k * (k - 1) / 2))))
  }

  quit <- function() {
    return(invisible())
  }

  val <- do.call(match.arg(cmd), args = list())
  return(val)
}
```

B.4 First strategy for criticism of model (G)

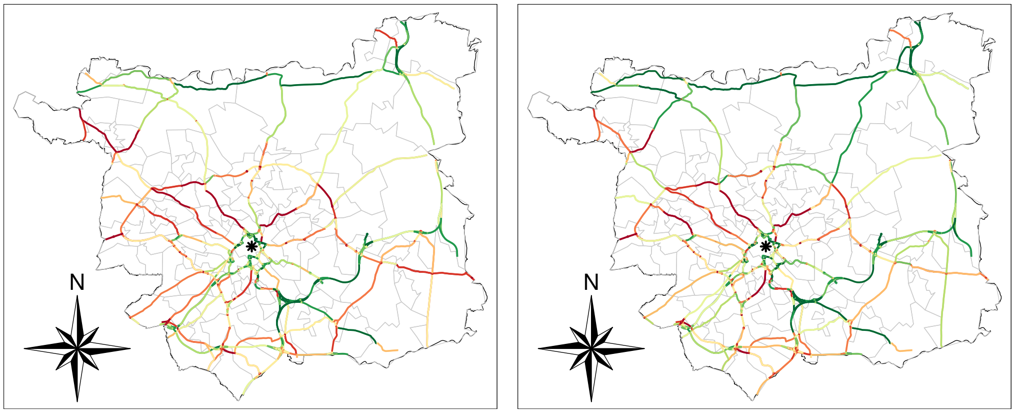
We report in Algorithm 1 the pseudo-code that describes the procedure used to estimate the balanced accuracy distribution. We run the algorithm using $N = 5000$ iterations and the quantiles associated to the following set of probabilities: $\boldsymbol{p} = (0.5, 0.55, 0.6, 0.65, 0.7, 0.75, 0.8, 0.825, 0.85, 0.875, 0.9, 0.925, 0.95, 0.975, 0.99)$. We also run the same procedure using the posterior mean instead of a particular quantile (i.e. the red curve in Figure 4.6).

B.5 Alternative specification for the prior distribution

We report in Tables B.1 and B.2 a summary of the estimates of the fixed and random effects for two alternative specifications of model (G), differing in the hyperprior assigned to Ω, the precision matrix of the PMCAR random effect. More precisely, the two models were defined using, respectively, a Wishart hyperprior with rank equal to 2 and scale matrix equal to $\text{diag}(0.5, 0.5)$ (first model) and $\text{diag}(2, 2)$ (second model). We will refer to these models using a code that summarises the corresponding hyperprior distribution.

B.6 Alternative specification for the neighbourhood matrix

We report in Tables B.3 and B.4 a summary of the estimates of the fixed and random effects for two alternative specifications of model (G), differing in the definition of the adjacency



(a) Severe car crashes

(b) Slight car crashes

Figure B.3: Two maps representing the posterior means for severe and slight car crashes rates, estimated using model (G) on the contracted network. The colours go from red (higher quantiles) to green (lower quantiles). The black star represents Leeds City Center.

matrix $\boldsymbol{W}$. More precisely, the two models were defined using, respectively, a second and a third order adjacency matrix for $\boldsymbol{W}$. We will refer to these models as $\boldsymbol{W}_2$ and $\boldsymbol{W}_3$. Lastly, we report in Tables B.5 and B.6 a summary of the estimates of fixed and random effects for four alternative specifications of model (G), that were defined with a spatial neighbourhood adjacency matrix using, respectively, a threshold of 50, 100, 250, and 500 meters.

B.7 Car crashes rates for the contracted network

We display in Figure B.3 two maps that represent the posterior means of car crashes rates obtained running model (G) on the contracted network. The two maps look similar to their not-contracted counterparts (see Figure 4.5). In both cases, the areas that are located north and north-west to the city center are associated with higher levels of car crashes rates. Moreover, we can graphically check the effects of removing redundant vertices since, after merging contiguous road segments, we obtained a smoother estimate of car crashes rates.

B.8 DIC and WAIC for alternative specifications

We report in Table B.7 the estimates of DIC and WAIC for model (G) and all alternative specifications. We can notice how all models whose adjacency matrix ignored the network structure performed significantly worse than model (G).

Algorithm 1: Pseudo-code illustrating the steps used to estimate the balanced accuracy. The same procedure was applied independently for the two severity levels.

```
Input: $N$ (number of iterations) and $\boldsymbol{p}$ (vector of probabilities)
/* For example $\boldsymbol{p} = (0.5, 0.75, 0.95)$                                  */
Output: $S$
/* $S$ is the matrix of estimates of balanced accuracy.                             */
Data: Posterior distribution of fitted values calculated using model (G)
Result: Estimates of balanced accuracy distribution
/* The algorithm is divided into two blocks: first we compute quantiles
   associated to the chosen probabilities, then we estimate balanced
   accuracy.                                                                        */
begin
    $\boldsymbol{M} \leftarrow$ Matrix(0, length($\boldsymbol{p}$), $n$) ;       /* Store quantiles */
    for $i \in \boldsymbol{p}$ do
        for $j = 1$ to $n$ do
            $\boldsymbol{M}[i, j] \leftarrow$ quantile of order $i$ for road segment $j$;
        end
    end
end
begin
    /* Store estimates of balanced accuracy                                         */
    $\boldsymbol{S} \leftarrow$ Matrix(0, length($\boldsymbol{p}$), $N$);
    for $i \in \boldsymbol{p}$ do
        for $j = 1$ to $N$ do
            predCounts $\leftarrow n$ poisson random deviates with mean $\boldsymbol{M}[i, ]$;
            binCounts $\leftarrow$ discretize predCounts using a threshold of 1;
            $\boldsymbol{S}[i, j] \leftarrow$ balancedAccuracy(binCounts, obsCounts);
            /* obsCounts are the observed counts                                    */
        end
    end
end
```

	Scale matrix: diag$(0.5, 0.5)$		Scale matrix: diag$(2, 2)$	
	Severe Crashes	Slight Crashes	Severe Crashes	Slight Crashes
β_0	-14.490	-12.832	-14.461	-12.831
	(0.123)	(0.113)	(0.130)	(0.124)
Betweenness	-0.063	-0.069	-0.063	-0.066
	(0.052)	(0.034)	(0.052)	(0.034)
Motorways	-0.775	-0.169	-0.769	-0.162
	(0.177)	(0.121)	(0.173)	(0.119)
Primary Roads	0.469	0.558	0.470	0.558
	(0.132)	(0.103)	(0.130)	(0.102)

Table B.1: Means (standard deviations) for the posterior distributions of the fixed effects.

ID		$\sigma^2_{\theta_1}$	$\sigma^2_{\theta_2}$	ρ_θ	ρ_1	ρ_2	$\sigma^2_{\phi_1}$	$\sigma^2_{\phi_2}$	ρ_ϕ
diag$(0.5, 0.5)$	mean:	0.425	0.608	0.401	0.994	0.996	0.404	0.346	0.814
	sd:	(0.083)	(0.049)	(0.013)	(0.002)	(0.001)	(0.067)	(0.043)	(0.034)
diag$(2, 2)$	mean:	0.459	0.652	0.403	0.996	0.997	0.302	0.271	0.862
	sd:	(0.087)	(0.049)	(0.013)	(0.002)	(0.000)	(0.063)	(0.039)	(0.029)

Table B.2: Means (standard deviations) for the posterior distributions of the hyperparameters.

	Second order adjacency: $\boldsymbol{W}_2$		Third order adjacency: $\boldsymbol{W}_3$	
	Severe Crashes	Slight Crashes	Severe Crashes	Slight Crashes
β_0	-14.503	-12.900	-14.502	-12.904
	(0.167)	(0.179)	(0.163)	(0.166)
Betweenness	-0.022	-0.029	-0.042	-0.039
	(0.050)	(0.032)	(0.050)	(0.032)
Motorways	-0.757	-0.153	-0.742	-0.144
	(0.161)	(0.103)	(0.160)	(0.103)
Primary Roads	0.475	0.585	0.466	0.570
	(0.123)	(0.090)	(0.122)	(0.091)

Table B.3: Means (standard deviations) for the posterior distributions of the fixed effects.

ID		$\sigma^2_{\theta_1}$	$\sigma^2_{\theta_2}$	ρ_θ	ρ_1	ρ_2	$\sigma^2_{\phi_1}$	$\sigma^2_{\phi_2}$	ρ_ϕ
$\boldsymbol{W}_2$	mean:	0.611	0.859	0.417	0.999	0.999	0.373	0.334	0.834
	sd:	(0.094)	(0.051)	(0.007)	(0.000)	(0.000)	(0.259)	(0.254)	(0.754)
$\boldsymbol{W}_3$	mean:	0.513	0.754	0.405	0.998	0.998	1.448	1.401	0.893
	sd:	(0.083)	(0.041)	(0.007)	(0.001)	(0.001)	(0.184)	(0.139)	(0.029)

Table B.4: Means (standard deviations) for the posterior distributions of the hyperparameters.

	Spatial adjacency: 50m		Spatial adjacency: 100m	
	Severe Crashes	Slight Crashes	Severe Crashes	Slight Crashes
β_0	−14.485 (0.133)	−12.841 (0.129)	−14.446 (0.142)	−12.797 (0.144)
Betweenness	−0.054 (0.051)	−0.060 (0.033)	−0.030 (0.050)	−0.041 (0.032)
Motorways	−0.792 (0.174)	−0.176 (0.117)	−0.858 (0.172)	−0.224 (0.113)
Primary Roads	0.464 (0.132)	0.568 (0.101)	0.428 (0.131)	0.546 (0.099)
	Spatial adjacency: 250m		Spatial adjacency: 500m	
	Severe Crashes	Slight Crashes	Severe Crashes	Slight Crashes
β_0	−14.424 (0.162)	−12.794 (0.153)	−14.397 (0.154)	−12.777 (0.135)
Betweenness	0.026 (0.050)	0.008 (0.031)	0.056 (0.047)	0.047 (0.030)
Motorways	−0.968 (0.166)	−0.303 (0.104)	−1.052 (0.161)	−0.402 (0.099)
Primary Roads	0.413 (0.122)	0.544 (0.088)	0.384 (0.110)	0.476 (0.081)

Table B.5: Means (standard deviations) for the posterior distributions of the fixed effects.

ID		$\sigma^2_{\theta_1}$	$\sigma^2_{\theta_2}$	ρ_θ	ρ_1	ρ_2	$\sigma^2_{\phi_1}$	$\sigma^2_{\phi_2}$	ρ_ϕ
50m	mean:	0.501	0.690	0.408	0.997	0.998	0.308	0.272	0.836
	sd:	(0.088)	(0.046)	(0.010)	(0.001)	(0.001)	(0.054)	(0.033)	(0.032)
100m	mean:	0.543	0.749	0.411	0.998	0.999	0.353	0.313	0.847
	sd:	(0.095)	(0.049)	(0.009)	(0.001)	(0.001)	(0.071)	(0.044)	(0.033)
250m	mean:	0.647	0.873	0.416	0.999	0.999	0.638	0.623	0.870
	sd:	(0.087)	(0.047)	(0.006)	(0.000)	(0.000)	(0.098)	(0.073)	(0.031)
500m	mean:	0.695	0.893	0.418	0.998	0.998	2.726	3.139	0.927
	sd:	(0.100)	(0.055)	(0.007)	(0.001)	(0.001)	(0.485)	(0.389)	(0.020)

Table B.6: Means (standard deviations) for the posterior distributions of the hyperparameters.

ID	DIC	WAIC
(G)	14 113.35	14 093.90
$\text{diag}(0.5, 0.5)$	14 110.41	14 087.88
$\text{diag}(2, 2)$	14 117.31	14 100.84
$\boldsymbol{W}_2$	14 198.69	14 153.61
$\boldsymbol{W}_3$	14 172.20	14 139.50
50m	14 125.23	14 100.95
100m	14 139.76	14 112.47
250m	14 181.43	14 137.60
500m	14 241.58	14 177.59

Table B.7: DIC and WAIC for model (G) and all its alternative specifications.

Appendix C

osmextract: An R Package to Download, Convert and Read Open Street Map Data Extracts

There is no I in team, but there is a U in bug.

Somewhere online

The following appendix is based on one of the package's vignettes. The R *code can be browsed at the following page:* https://github.com/ITSLeeds/osmextract.

C.1 Introduction

Open Street Map (OSM) is an online database that provides open-licence geospatial data mapping, among the other things, roads, rivers, buildings, coastal lines, political and administrative boundaries worldwide (OpenStreetMap contributors, 2017). It is used by several popular services like *Facebook*, *Flickr*, *Foursquare*, *Moovit*, *Niantic*, *Snapchat*, and *Tableu*, and it is the only mapping database to which all major internet companies continue to contribute. By this definition, at the moment it is the most important geo-database (Anderson, Sarkar, and Palen, 2019; Barrington-Leigh and Millard-Ball, 2017). It also provides geocoding and reverse geocoding routines via the Nominatim API (Open Street Map, 2017). Open Street Map uses a peculiar data structure, which has relevant consequences on every software related to OSM. In fact, the basic components of Open Street Map data are called *elements*, and they are divided into:

nodes: representing the building blocks of *ways* and other points on the earth's surface without a physical size (such as traffic lights, road signs or crossings);

ways: ordered lists of *nodes*, typically linked with streets, rivers, shops and buildings;

relations: lists of nodes, ways and other relations, where each member has additional information that describes its relationship with the other elements.

The characteristics of each element may be described using a *tag*, which is just a pair of a *key* and a *value*. Several examples of tags, keys and values are provided in Section C.3.3 when introducing the R code for filtering OSM extracts.

In this appendix, we will present osmextract, an R package to download, convert and read-in bulk OSM data hosted by external providers such as Geofabrik GmbH and bbbike. We aim to make it easier for people to access OSM extracts for reproducible research and answer a common question: how to get the data into a statistical environment, in an appropriate format, as part of a computationally efficient and reproducible workflow? Other R packages answer parts of this question. osmdata, for example, provides an interface to the Overpass API, which is ideal for downloading small OSM datasets (Padgham et al., 2017). However, the API is rate limited, making it hard to download large datasets. As a case study, the following code can be used to try downloading all cycleways[1] in England:

```
library("osmdata")
cycleways_england = opq("England") %>%
  add_osm_feature(key = "highway", value = "cycleway") %>%
  osmdata_sf()
Error in check_for_error(doc) : General overpass server error; returned:
Runtime error: Query timed out in query at line 4 after 26 seconds.
```

As we can see, the function stops with an error message after approximately 30 seconds, saying that the query timed out.

On the other hand, osmextract adopts a different approach linking OSM extracts created and formatted by external providers instead of querying the Open Street Map database. The same request can be made with osmextract as follows, which reads-in almost 100,000 lines in less than 10 seconds, after downloading and converting the data. The download-and-conversion operations of England's OSM extract, which are extensively explained in Section C.3, take approximately eight minutes using a laptop with i7-7500U processor and 8GB of RAM. The result is depicted in Figure C.1.

```
cycleways_england = oe_get(
  place = "England",
  provider = "geofabrik",
  quiet = FALSE,
  query = "SELECT * FROM 'lines' WHERE highway = 'cycleway'"
)
```

osmextract is designed to complement osmdata, which has advantages over our package for small datasets: osmdata is likely to be quicker for datasets less than a few tens of MB in size, provides up-to-date data and has an intuitive interface. osmdata can provide data in a range of formats, such as sf ((OGC) Open Geospatial Consortium Inc, 2011), sp (R. S. Bivand, E. Pebesma, and Gómez-Rubio, 2013) or sc (Sumner and Padgham, 2020), while osmextract only returns sf objects. osmextract's niche is that it provides a fast way to download large OSM datasets in the highly compressed pbf format and read them in via the fast C library GDAL (GDAL/OGR contributors, 2020) and the popular R package for working with geographic data sf (E. J. Pebesma, 2018). There are several other projects that

[1] The exact definition of *cycleway* is quite tricky since OSM uses different formats to indicate the cycleways. We are going to present just a textbook example.

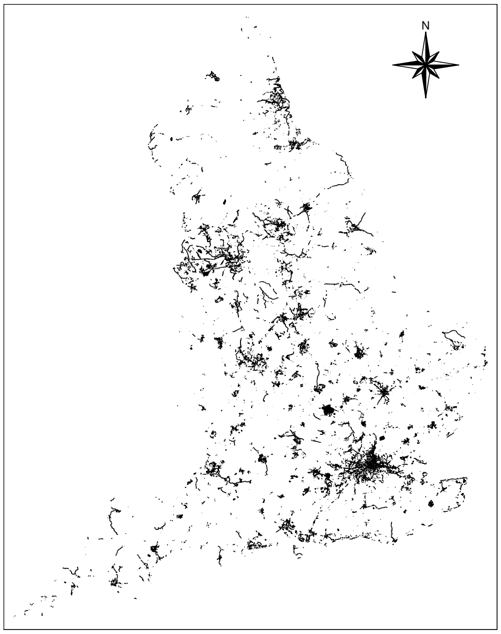


Figure C.1: Map of the cycleways in England. The data were downloaded using `osmextract` and plotted via the `R` package `tmap` (Tennekes, 2018).

focus on OSM data such as `pyrosm` (Tenkanen, 2020), `pydriosm` (Fu, 2020), `osmium` (Map, 2020a), `osmosis` (Map, 2020b), and `OpenStreetMapX.jl` (Szufel, 2020), mainly developed using different software like `C++`, `python` (Van Rossum and Drake, 2009) or `Julia` (Zappa Nardelli et al., 2018). In the near future, we will compare our package with the alternative implementations.

C.1.1 Install and load the package

At the time of writing, the package is not on `CRAN`, but the development version can be installed from Github as follows:

```r
install.packages("remotes"); library("remotes")
install_github("ITSLeeds/osmextract")
```

Loading the package generates an important message about the license associated with OSM data:

```r
library("osmextract")
#> Data (c) OpenStreetMap contributors, ODbL 1.0.
#> https://www.openstreetmap.org/copyright.
```

There are important legal considerations that should be taken into account before using OSM data, especially for people working in a for-profit capacity. Anyone using OSM data

is bound by law to adhere to the ODBL licence, and we refer to the package's introductory vignette for more details.
As we said before, `osmextract` returns data using `sf` format, so we also load the `R` package with the same name that define all `st_*` functions used in the next Section:

```r
library("sf")
```

The rest of the appendix is organised as follows. In Section C.2 we present a brief overview of the OSM providers currently supported by `osmextract`, explaining their pros and cons. Section C.3 introduces the five most important functions in `osmextract`, which define the building blocks of the package and are used to match, download, convert and read-in an OSM extract. Finally, Section C.4 concludes the appendix and presents some future works.

C.2 Open Street Map Providers

At the time of writing, the package is linked with three OSM extracts providers: *Geofabrik*, *bbbike*, and *openstreetmap.fr*. *Geofabrik* is a company that creates map products and offers free downloads of OSM extracts that are updated daily. These extracts are based on a hierarchical division of the world into different regions. The first level covers the six continents (plus the Russian Federation); the second level contains most countries and several special areas (like the Alps, Britain and Ireland, US MidWest, US Northeast, US Pacific, US South and US West); the third and last level represents some local regions (mainly in Europe, Russia, Canada and South America). *openstreetmap.fr* is a web-service that provides OSM data for several zones worldwide, which are updated minutely. *Geofabrik* and *openstreetmap.fr* are based on similar partitions of the world. The latter is more detailed in some countries (mainly France, Italy, China, India, Russia, and Brazil), but it does not store extracts related to several areas in Africa and South America, while the former covers the whole world. Finally, *bbbike* is different from the other providers since it saves OSM data for more than 200 cities worldwide. We refer to the package's vignette for a more detailed comparison of the provider's partitions.
`osmextract` summarises the OSM data stored by each provider using an `sf` dataframe object that records, among the other things, the URLs and the polygonal boundaries of each area. These dataframe objects are used by `oe_match()` (detailed in Section C.3.1) to match a place with one of the extracts stored by the providers.
The function `oe_providers()` can be used to print a short summary of all providers supported by `osmextract`:

```r
oe_providers()
#>   available_providers          dataframe_name number_of_zones number_of_fields
#> [1]           geofabrik          geofabrik_zones             430               14
#> [2]              bbbike             bbbike_zones             235               10
#> [3]    openstreetmap_fr   openstreetmap_fr_zones             835                6
```

The previous output can be interpreted as follows:

available_providers: summarizes the name of the supported providers;
database_name: stores the name of the corresponding sf dataframe objects;
number_of_zones: the number of zones (or rows);
number_of_fields: the number of fields (or columns).

The most important fields of each dataframe object are:

id: A unique identifier, containing letters, numbers, and the characters - and /.
name: The, usually English, long-form name of the area.
pbf: Link to the latest .pbf file for this region.
geometry: A polygonal boundary around the region, stored in sfg[2] format.

We refer to the help pages of the dataframe objects (e.g. ?geofabrik_zones) for more details.

C.3 The main functions

This Section describes the essential characteristics of the five main functions that compose the package. We refer to the introductory vignette and the help pages for more details. These functions are used to

1. oe_match(): Match an input place with one of the files stored by OSM providers;
2. oe_download(): Download the chosen file;
3. oe_download(): Convert between .pbf and .gpkg formats;
4. oe_read(): Read .pbf and .gpkg files;
5. oe_get(): Match, download, translate, and import data, all in one step.

They are presented following the same order in which they are typically used. It should be noticed that we adopted a common oe_* prefix so that a user can take advantage of auto-completion features implemented by the IDE (e.g. Rstudio).

C.3.1 oe_match: Match OSM extracts

The function oe_match() takes in input a string through the parameter place, and it returns a named list of length two with the URL and the size (in bytes) of a .pbf file representing a geographical zone stored by one of the supported providers. The .pbf format is a highly optimised binary format used by OSM providers to store and share OSM extracts. For example:

```
oe_match(place = "Italy", quiet = TRUE)
#> $url
#> [1] "https://download.geofabrik.de/europe/italy-latest.osm.pbf"
#> $file_size
#> [1] 1544340778
```

[2] The sfg class is defined in the package sf (E. J. Pebesma, 2018).

The geographical zone is chosen by calculating the Approximate String Distance between the input `place` and one of the fields in the provider's dataset. Then, the function selects the closest match. By default, it uses the `name` field and *Geofabrik* provider, but a user can select a different column via the argument `match_by`. We refer to the help page of the chosen provider for a detailed description of all available fields. A useful and interesting alternative field is represented by the (unique and unambiguous) iso3166-1 alpha2 codes:

```
oe_match(place = "US", match_by = "iso3166_1_alpha2")
#> $url
#> [1] "https://download.geofabrik.de/north-america/us-latest.osm.pbf"
#> $file_size
#> [1] 6982945396
```

The parameter `max_string_dist` (which defaults to 1) represents the maximum tolerable distance between the input `place` and the closest match in `match_by` column before the function prints a warning message or stops with an error. This value can always be increased to help the matching operations, but that can lead to false matches:

```
oe_match("London", max_string_dist = 3)
#> The input place was matched with: Jordan
#> $url
#> [1] "https://download.geofabrik.de/asia/jordan-latest.osm.pbf"
#> $file_size
#> [1] 27400228
```

If the approximate string distance between the closest match and the input `place` is greater than `max_string_dist`, then `oe_match()` will also check the other supported providers. For example:

```
oe_match("leeds")
#> No exact match found for place = leeds and provider = geofabrik.
#> Best match is Laos.
#> Checking the other providers.
#> An exact string match was found using provider = bbbike.
#> $url
#> [1] "https://download.bbbike.org/osm/bbbike/Leeds/Leeds.osm.pbf"
#> $file_size
#> [1] 19376705
```

Finally, if there is no tolerable match with any of the supported providers and `match_by` argument is equal to `"name"`, then `oe_match()` will use the Nominatim API to geolocate the input `place` and perform a spatial matching operation, introduced below.

```
oe_match("Milan")
#> No exact match found for place = Milan and provider = geofabrik.
#> Best match is Iran.
#> Checking the other providers.
#> No exact match found in any OSM provider data. Searching for the location online.
#> $url
#> [1] "https://download.geofabrik.de/europe/italy/nord-ovest-latest.osm.pbf"
#> $file_size
#> [1] 416306623
```

`oe_match()` returns a warning message if there are multiple zones equidistant (according to approximate string distance) from the input `place`. In that case, it selects the first match:

```
oe_match("Belin")
Warning: The input place was matched with multiple
geographical zones: Benin - Berlin.
#> Selecting the first match.
#> The input place was matched with: Benin
```

Matching zones with geographic inputs

The input `place` can also be specified using an `sfc_POINT` object[3] with arbitrary Coordinate Reference System (CRS), as documented in the following example. The function will return a named list of length two with the URL and the size of a `.pbf` file representing a zone that geographically intersects the `sfc_POINT` (or an error, if the input point does not cross any area). If the input `place` intersects multiple geographically nested zones, the function returns the area with the highest `level`. The meaning of the `level` fields depends on the chosen provider, so we refer to the help pages for more details. We could roughly say that the higher is the level, the lower is the geographical area's administrative unit. If there are multiple matches within the same `level`, then `oe_match()` will return the area whose centroid is closest to the input `place`.

```
milan_duomo = st_sfc(st_point(c(1514924, 5034552)), crs = 3003)
oe_match(milan_duomo)
#> The input place was matched with multiple geographical areas.
#> Selecting the areas with the highest "level".
#> $url
#> [1] "https://download.geofabrik.de/europe/italy/nord-ovest-latest.osm.pbf"
#> $file_size
#> [1] 416306623
```

The input `place` can also be specified using a numeric vector of coordinates. In that case, the CRS is supposed to be EPSG:4326. For example:

```
oe_match(c(9.1916, 45.4650)) #> Duomo di Milano using EPSG: 4326
```

The output is the same as before.
Most of the following examples are based on a small and simple OSM extract that can be retrieved as follows:

```
its_details = oe_match("ITS Leeds", provider = "test")
```

[3] The `sfc_POINT` objects are defined in the `R` package `sf`.

C.3.2 oe_download: Download OSM extracts

The oe_download() function is used to download .pbf files representing OSM extracts. It takes in input a URL, through the parameter file_url, and it downloads the requested data in a directory (specified by the parameter download_directory):

```
oe_download(
  file_url = its_details$url,
  file_size = its_details$file_size,
  provider = "test",
  download_directory = tempdir()
)
```

The argument provider can be omitted if the input file_url is associated with one of the supported providers.
The default value for download_directory is tempdir(), which is a function that returns a path to a temporary directory, erased every time R is restarted). A user can set a persisting directory as the default value by adding the character string OSMEXT_DOWNLOAD_DIRECTORY = /path/for/osm/data to the .Renviron file, e.g. with:

```
edit_r_environ()
#> Add a line containing: OSMEXT_DOWNLOAD_DIRECTORY=/path/to/save/files
```

The function edit_r_environ() is defined in the R package usethis (Wickham et al., 2019). The default download_directory can always be checked using oe_download_directory(). We strongly advise all users to set a persistent directory since downloading and converting (see Section C.3.3) .pbf files are expensive operations, that are skipped by all oe_*() functions if they detect that the input file was already downloaded and/or converted. More precisely, oe_download() runs several checks before actually downloading a new file, to avoid overloading the OSM providers. The first step is the definition of the path associated with the input file_url. The path is created by pasting together the download_directory, the name of the chosen provider (specified by provider argument or inferred from the input file_url), and the base-name of the URL. For example, if file_url points to the OSM extract of Italy, i.e. https://download.geofabrik.de/europe/italy-latest.osm.pbf and download_directory is equal to /tmp/, then the path of the new file is built as follows: /tmp/geofabrik_italy-latest.osm.pbf. In the second step, the function checks if the new path already exists and, in that case, it returns it (without downloading anything[4]). Finally, it downloads a new file, and it returns its path.

C.3.3 oe_vectortranslate: Convert to .gpkg format

The oe_vectortranslate() function translates a .pbf file into .gpkg format. GeoPackage (.gpkg) is an *open, stardards-based, platform-independent, portable, self-descripting, compact format for transferring geospatial information* (Open Geospatial Consortium (OGC),

[4] The parameter force_download can override this behaviour.

2020). It takes in input a string representing the path to an existing .pbf file and it returns the path to the newly generated .gpkg file. The .gpkg file is created in the same directory as the input .pbf file and with the same name. The conversion is performed using ogr2ogr (GDAL/OGR contributors, 2020) through vectortranslate utility in gdal_utils(). We decided to adopt this approach following the suggestions of the maintainers of GDAL. Moreover, GeoPackage files have database capabilities like random access and querying that are extremely important for OSM data (see Section C.3.5).
The simplest example works as follows:

```
its_pbf = oe_download(its_details$url, provider = "test")
its_gpkg = oe_vectortranslate(its_pbf)
```

The vectortranslate operation can be customised in several ways modifying the parameters layer, extra_tags, osmconf_ini, and vectortranslate_options.

layer argument

The .pbf files processed by GDAL are usually categorized into 5 layers, named points, lines, multilinestrings, multipolygons and other_relations following the structure used by the OSM database (see Section C.1). The oe_vectortranslate() function can covert only one layer at a time, specified through the parameter layer. The default value is "lines", since that's the most used layer according to our experience. Several layers with different names can be stored in the same .gpkg file.
The .pbf files always contain all five layers:

```
st_layers(its_pbf, do_count = TRUE)
#> Driver: OSM
#> Available layers:
#>         layer_name       geometry_type features fields
#> 1           points               Point      186     10
#> 2            lines         Line String      189      9
#> 3 multilinestrings   Multi Line String       10      4
#> 4    multipolygons       Multi Polygon      104     25
#> 5  other_relations Geometry Collection        3      4
```

while, by default, oe_vectortranslate() convert only the lines layer:

```
st_layers(its_gpkg, do_count = TRUE)
#> Driver: GPKG
#> Available layers:
#>         layer_name       geometry_type features fields
#> 1            lines         Line String      189      9
```

Another layer can be added as follows:

```
its_gpkg = oe_vectortranslate(its_pbf, layer = "points")
st_layers(its_gpkg, do_count = TRUE)
#> Driver: GPKG
#> Available layers:
```

```
#>          layer_name       geometry_type features fields
#> 1            points               Point      186     10
#> 2             lines         Line String      189      9
```

These considerations are important since `oe_read()` (see Section C.3.4) can only read one layer at a time.

`osmconf_ini` and `extra_tags` argument

The arguments `osmconf_ini` and `extra_tags` are used to modify how GDAL reads and processes a `.pbf` file. More precisely, several operations that GDAL performs on a `.pbf` file are governed by a `CONFIG` file, that can be checked at the following link: https://github.com/OSGeo/gdal/blob/master/gdal/data/osmconf.ini. The `osmextract` package stores a local copy which is used as the standard `CONFIG` file.
As we said in the Introduction, the basic components of OSM data are called *elements*, and are divided into *nodes*, *ways* and *relations*. Thus, for example, the code at line 7 of that CONFIG file is used to determine which *ways* are assumed to be `POLYGONS` (following the definition in Simple Feature standards ((OGC) Open Geospatial Consortium Inc, 2011)) if they are closed.
The parameter `osmconf_ini` can be used to specify the path to a different `CONFIG` file, in case a user needs more control over GDAL operations. See Section C.3.5 for an example. If `osmconf_ini` is equal to `NULL` (the default), then `oe_vectortranslate()` uses the standard `CONFIG` file.
Another example can be built as follows. OSM data are usually described using several *tags*, i.e. pairs of *keys* and *values*. The code at lines 33, 53, 85, 103, and 121 of the default `CONFIG` file is used to determine, for each layer, which tags should be explicitly reported as columns (while all other tags are stored in the `other_tags` field). The parameter `extra_tags`, which defaults to `NULL`, manipulates which (extra) tags should be explicitly reported in the `.gpkg` file. A complete list of OSM tags and Map features is reported in the OSM wiki: https://wiki.openstreetmap.org/wiki/Map_Features. It should be noted that the argument `extra_tags` is ignored if `osmconf_ini` is not `NULL` (since we can not know how a non-standard `CONFIG` file was generated).
The `oe_get_keys()` function can be used to check all `keys` that are stored in the `other_tags` field for a given `.gpkg` file. For example,

```
oe_get_keys(its_gpkg, layer = "lines")
#> [1] "bicycle" "foot" "maxspeed" "access" "lanes" "oneway" "lit"
#> [2] ... more keys
```

Then, the `.gpkg` file can be recreated adding new tags:

```
its_gpkg = oe_vectortranslate(its_pbf, extra_tags = c("bicycle", "foot"))
```

We present more complex (and realistic) examples in Section C.3.5.

vectortranslate_options argument

The parameter vectortranslate_options is used to control the arguments that are passed to ogr2ogr via gdal_utils() when converting between .pbf and .gpkg formats. ogr2ogr can perform various operations during the translation process, such as spatial filters or SQL queries. These operations are determined by the argument vectortranslate_options. If NULL (default value), then vectortranslate_options is set equal to:

```
c(
  "-f", "GPKG",
  "-overwrite",
  "-oo", paste0("CONFIG_FILE=", osmconf_ini),
  "-lco", "GEOMETRY_NAME=geometry",
  layer
)
```

The options "-f" and "GPKG" says that the output format is GPKG. This is mandatory when the version of GDAL is smaller than 2.3. "-overwrite" is used to delete an existing layer and recreate it empty. The string "-oo", paste0("CONFIG_FILE=", osmconf_ini) modifies the *open options* of the .pbf file and set the path of the CONFIG file. We refer to the help page of GDAL OSM driver for more details on its open options. The options "-lco", "GEOMETRY_NAME=geometry" say that the name of the geometry column in the .GPKG file should be geometry (default value is geom). The string -lco is an acronym for *Layer Creation Options*, and we refer to the help page of GDAL GPKG driver for more details. Finally, the layer argument specifies which layer should be converted. The arguments that are passed to vectortranslate_options can also be used to perform queries during the vectortranslate process, as shown in Section C.3.5.

Other notes

By default, vectortranslate operations are skipped if oe_vectortranslate() detects a file having the same path as the input file, .gpkg extension and a layer with the same name as the parameter layer with all extra_tags. In that case, the function will return the path of the .gpkg file[5]. If osmconf_ini or vectortranslate_options parameters are not NULL, the vectortranslate operations are never skipped.

C.3.4 oe_read: Read-in OSM data

The oe_read() function is a wrapper around oe_download(), oe_vectortranslate(), and sf::st_read(). It is used for reading-in a .pbf or .gpkg file that is specified using its path or URL. For example, the following code can be used for reading-in the its-gpkg file:

```
oe_read(its_gpkg)
#> Reading layer ‘lines’ from data source "..." using driver ‘GPKG’
```

[5] This behaviour can be overwritten setting force_vectortranslate = TRUE.

```
#> Simple feature collection with 189 features and 11 fields
#> geometry type:  LINESTRING
#> dimension:      XY
#> bbox:           xmin: -1.5624 ymin: 53.8047 xmax: -1.5481 ymax: 53.8110
#> geographic CRS: WGS 84
```

If the input `file_path` points to a (typically small) `.pbf` file, the vectortranslate operations can be skipped using the parameter `skip_vectortranslate`. The input object can also be specified using a URL:

```
my_url = paste0(
 "https://github.com/ITSLeeds/osmextract/",
 "raw/master/inst/its-example.osm.pbf"
)
oe_read(my_url, provider = "test", skip_vectortranslate = TRUE)
#> Reading layer ‘lines’ from data source "..." using driver ‘OSM’
#> Simple feature collection with 189 features and 9 fields
#> geometry type:  LINESTRING
#> dimension:      XY
#> bbox:           xmin: -1.5624 ymin: 53.8047 xmax: -1.5481 ymax: 53.8110
#> geographic CRS: WGS 84
```

The `provider` argument must always be specified in case of non-supported providers.

C.3.5 `oe_get`: Do it all in one step

To simplify the steps outlined above, while enabling modularity if needs be, we packaged them all into a single function that works as follows:

```
its_lines = oe_get("ITS Leeds")
```

The output is depicted in Figure C.2. `oe_get()` is a wrapper around `oe_match()` and `oe_read()`, and it summarises the algorithm that we use for importing OSM extracts:

1. match the input `place` with the URL of a `.pbf` zone through `oe_match()`;
2. if necessary, download the file using `oe_download()`;
3. convert it into `.gpkg` format using `oe_vectortranslate()`. As explained in Section C.3.3, the conversion could be skipped in some cases;
4. read-in one layer of the `.gpkg` file using `st_read()`.

The arguments `osmconf_ini`, `vectortranslate_options`, `query` and `wkt_filter` (the last two are defined in `st_read()`) can be used to further optimize the process of getting OSM extracts into `R`.

`osmconf_ini` argument

The following example shows how to create an ad-hoc `CONFIG` file. First, we load a local copy of the standard `osmconf.ini`, taken from `https://github.com/OSGeo/gdal/blob/master/gdal/data/osmconf.ini`:

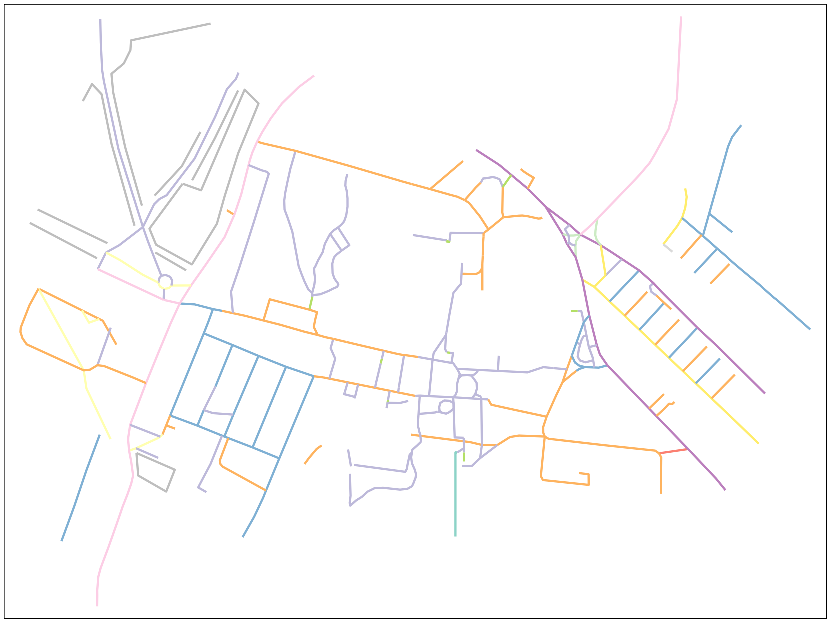


Figure C.2: A smalll road network associated with the `its_lines` object. It is located in proximity of the Institute of Transport Studies, Leeds (UK). The road segments are coloured according to value of the `highway` key.

```
custom_osmconf_ini = readLines(
  con = system.file("osmconf.ini", package = "osmextract")
)
```

Then, we modify the code at lines 18 and 21 asking GDAL to report all nodes and ways, even without any significant tag:

```
custom_osmconf_ini[[18]] = "report_all_nodes=yes"
custom_osmconf_ini[[21]] = "report_all_ways=yes"
```

We change also lines 45 and 53, removing the `osm_id` field and altering the default columns:

```
custom_osmconf_ini[[45]] = "osm_id=no"
custom_osmconf_ini[[53]] = "attributes=highway,lanes".
```

A local copy of the new `CONFIG` file can be used during `ogr2ogr` conversion:

```
temp_ini = tempfile(fileext = ".ini")
writeLines(custom_osmconf_ini, temp_ini)
oe_get("ITS Leeds", provider = "test", osmconf_ini = temp_ini)
#> Simple feature collection with 191 features and 4 fields
#> Further output ...
```

We can see that the output has 2 extra features (since we set "report_all_nodes=yes" and "report_all_ways=yes") and only 4 columns, i.e. the new default attributes, highway and lanes, and two other fields, z_order and other_tags, that were created by GDAL.

vectortranslate_options argument

As we said above, the parameter vectortranslate_options is used to modify the options that are passed to ogr2ogr. This is extremely important because, if we tune it, then we can analyse small parts of an enormous .pbf files without fully reading it in memory.
The first example shows how to use the argument -t_srs (acronym for *Target Spatial Reference System*) to modify the CRS of a .pbf object while performing vectortranslate operations:

```
my_vectortranslate = c(
  "-f", "GPKG", #> output file format
  "-overwrite", #> overwrite an existing layer
  "-lco", "GEOMETRY_NAME=geometry", #> layer creation options,
  "-t_srs", "EPSG:27700", #> change the CRS
  "lines" #> layer
)
oe_get("ITS Leeds", vectortranslate_options = my_vectortranslate)
#> Extra output ...
#> bbox:  xmin: 428911.1 ymin: 434356.9 xmax: 429858.1 ymax: 435067
#> projected CRS:  OSGB 1936 / British National Grid
```

The default CRS of all OSM extracts obtained by Geofabrik and several other providers is EPSG:4326, i.e. latitude and longitude coordinates expressed via WGS84 ellipsoid, while the code EPSG:27700 indicates the British National Grid. Hence, the parameter -t_srs can be used to transform geographical data into projected coordinates, which may be essential for some statistical software like spatstat (Baddeley, Rubak, and Turner, 2015). The same operation can also be performed in R with the sf package, but the conversion can be slow for large spatial objects.
The next example demonstrates how to use -select and -where options to run a query during the vectortranslate process. The starting point is analogous to the previous example:

```
my_vectortranslate = c(
  "-f", "GPKG",
  "-overwrite",
  "-lco", "GEOMETRY_NAME=geometry",
  "-select", "osm_id, highway",
  "-where", "highway IN ('primary', 'secondary', 'tertiary')",
  "lines"
)
```

The options "-select" and "-where" specify an SQL-like query. The first option is used to select one or more columns from one layer of the .pbf file, while the second option filters only those features where the value associated to the "highway" key is equal to "primary", "secondary" or "tertiary". In fact, GDAL and ogr2ogr runs the OGR SQL dialect, and we refer to its online manual for more details: https://gdal.org/user/ogr_sql_dialect.html. These arguments are fundamental for users that need to subset a small portion of a bigger .pbf file. For example, the following code extracts all primary, secondary and tertiary roads from the .pbf file of Portugal (240MB) stored by Geofabrik. After downloading the data, it runs in approximately 40 seconds using a laptop with i7-7500U processor and 8GB of RAM.

```
my_vectortranslate = c(
  "-f", "GPKG",
  "-overwrite",
  "-lco", "GEOMETRY_NAME=geometry",
  "-select", "osm_id, highway",
  "-where", "highway IN (’primary’, ’secondary’, ’tertiary’)",
  "lines"
)

system.time({
  portugal = oe_get("Portugal", vectortranslate_options = my_vectortranslate)
})
#>  user  system elapsed
#> 26.91   12.64   38.78
```

The equivalent R code read-in several useless features and takes almost six times as much. The difference is even more pronounced for larger extracts.

```
system.time({
  portugal = oe_get("Portugal", force_vectortranslate = TRUE)
  portugal = portugal %>%
    select(osm_id, highway) %>%
    filter(highway %in% c("primary", "secondary", "tertiary"))
})
#>   user  system elapsed
#> 172.52   32.61  217.68
```

The functions select and filter used in the R code above are defined in the package dplyr (Wickham et al., 2019) and are analogous to the corresponding SQL keys.
The argument vectortranslate_options can also be tuned to perform spatial filter operations during the vectortranslate process. The option -spat, illustrated in the following example, can be used to filter only those features that intersect a given rectangular bounding box, specified as c(xmin, ymin, xmax, ymax). For example:

```
my_vectortranslate = c(
"-f", "GPKG",
"-overwrite",
"-spat", c(-1.559184, 53.807739, -1.557375, 53.808094),
"lines"
)
its_small = oe_get(
```

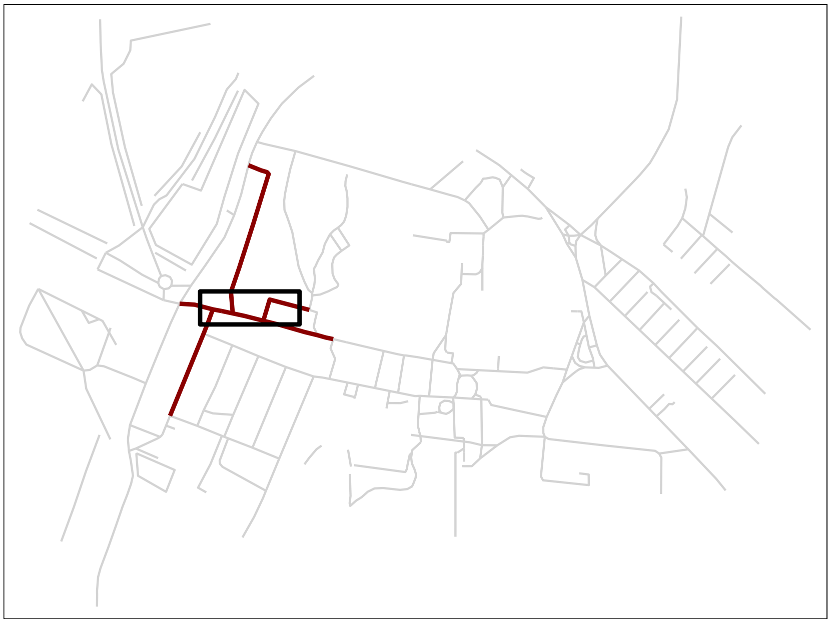


Figure C.3: Spatial filter based on a rectangular region applied to the data during vectortranslate processes. The red segments intersect the polygon.

```
  place = "ITS Leeds",
  vectortranslate_options = my_vectortranslate
)
```

The output is represented in Figure C.3, where the bounding box was highlighted in black, the intersecting streets in red and all the other roads in grey.
Finally, the options `-clipsrc` and `-clipdst` can be used to perform more complex operations such as spatial filtering with generic polygons or clipping. The former option crops the features considering a polygon specified in the original CRS, while the latter manipulates the data after the projection. In both cases, the polygon must be specified using the Well Known Text format ((OGC) Open Geospatial Consortium Inc, 2011). The following example shows how to download from Geofabrik servers the `.pbf` extract associated with the West Yorkshire region and apply a spatial filter while performing vectortranslate operations. We select and clip only the road segments that intersect a 5 kilometers circular buffer centred in Chapeltown, one of the neighbourhoods of Leeds.

```
chapeltown <- st_sfc(st_point(c(430964.5, 435700.3)), crs = 27700) %>%
  st_buffer(5000) #> Create a 5km circular buffer

my_vectortranslate = c(
  "-f", "GPKG",
  "-overwrite",
  "-select", "highway",
  "-where", "highway IN ('motorway', 'trunk',
  'primary', 'secondary', 'tertiary', 'unclassified')",
  "-t_srs", "EPSG:27700",
  "-clipdst", st_as_text(chapeltown), #> specify the spatial filter
  "-nlt", "PROMOTE_TO_MULTI", #> promote the geometry type
  "lines"
)
system.time({
  leeds_small = oe_get(
  "West Yorkshire",
  vectortranslate_options = my_vectortranslate
)
})
#> user  system elapsed
#> 8.33    1.01    9.21
```

The options `"-t_srs"`, `"-select"` and `"-where"` have the same interpretation as before. The option `"-clipdst"` says that we want to clip the OSM extract after the reprojection to `EPSG:27700`. The function `st_as_text()` converts an `sfg` polygon into Well Known Text format, which is mandatory for the spatial filter. Hence, `st_as_text(chapeltown)` specifies the polygon used for clipping. The last step may return invalid `LINESTRING` geometries. For this reason, the `-nlt` and `PROMOTE_TO_MULTI` options are used to override the default geometry type and *promote* the `LINESTRING(s)` into `MULTILINESTRING(s)`. The result is reported in Figure C.4, where we highlight the bounding circle and the road segments within using a dark-red colour, while all the other road segments in Leeds are coloured in grey. The operations take approximately 9 seconds, while the equivalent R code, reported below, takes more than four times as much. The time difference is more and more relevant for larger OSM data.

```
system.time({
  west_yorkshire <- oe_get(
    place = "West Yorkshire",
    force_vectortranslate = TRUE
  ) %>%
    st_transform(27700)

  west_yorkshire_small <- west_yorkshire %>%
    filter(highway %in% c(
      "motorway", "trunk", "primary", "secondary", "tertiary", "unclassified"
    ))

  chapeltown_roads <- st_crop(west_yorkshire, chapeltown)
})
#>   user  system elapsed
#>  31.78    3.09   38.22
```

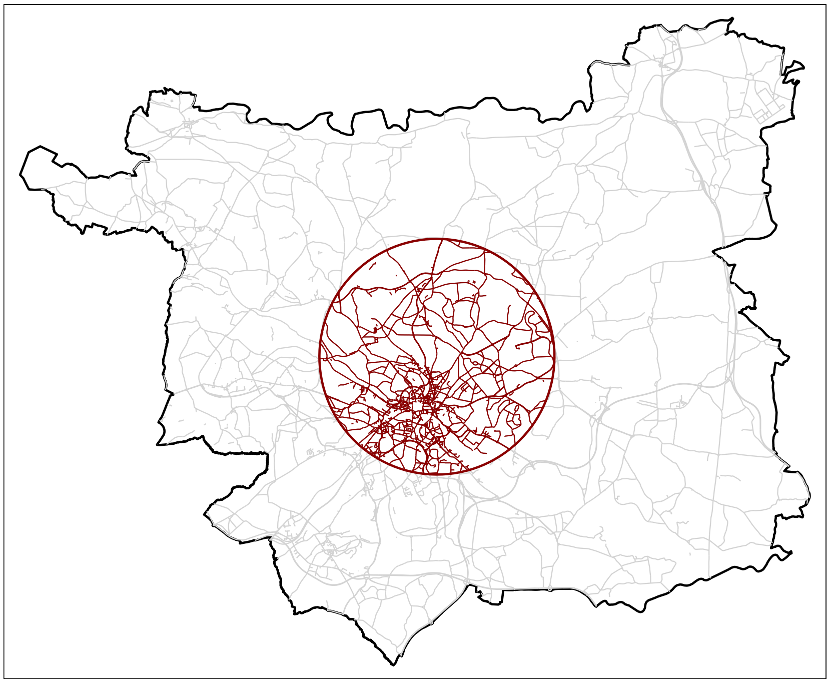


Figure C.4: Spatial filter based on a circular polygonal region centred in the Chapeltown neighbourhood of Leeds.

`query` and `wkt_filter` arguments

The last two options that we introduce are `query` and `wkt_filter`. They are defined in the `R` package `sf` and represent a useful compromise between the `GDAL` and the `R` approaches explained above, especially when a user needs to apply different queries to the same (typically small or medium-size) OSM extract. In fact, the two parameters create regular queries and spatial filters, respectively, that are applied immediately before reading-in the `.gpkg` file. The following code, for example, mimics the operations illustrated above, reading-in the road segments that intersect the circular buffer defined around Chapeltown neighbourhood:

```
system.time({
  oe_get(
    place = "West Yorkshire",
    force_vectortranslate = TRUE,
    query =
      "SELECT *
      FROM 'lines'
      WHERE highway IN
      ('motorway', 'trunk', 'primary', 'secondary',
      'tertiary', 'unclassified')",
    wkt_filter = st_as_text(st_transform(chapeltown, 4326)),
  )
})
#>  user  system elapsed
#> 15.86    2.87   20.64
```

This approach has its pros and cons. First of all, we can see that it's slightly slower than the GDAL routines, mainly because several unnecessary features are being converted to the `.gpkg` format. Hence, it may become unfeasible for larger `.pbf` files, probably starting from 500MB. We will test more cases and add more benchmarks in the near future. On the other side, the syntax is cleaner, the approach is more intuitive and, most importantly, it does not require a new (time-consuming) `ogr2ogr` conversion every time a user defines a new query. For these reasons, this is the suggested approach for querying a medium-size OSM extract.

C.4 Conclusions and next steps

In this appendix we reviewed the basic components of Open Street Map data, we introduced some of its external providers, such as *Geofabrik* or *bbbike*, and we detailed the main functionalities included in the `R` package `osmextract`. The most important routine is probably `oe_get()`, which is a wrapper around the other main functions, and it used to match, download, convert and read-in an OSM extract. It has several arguments that are used to modify every aspect of the process, such as the matching operations or the vectortranslate conversion.

We created several examples to showcase the main functionalities, comparing the new routines with traditional approaches implemented only in `R`. We found that the proposed methods, which integrate `R` and `GDAL`, outperform the classical ways, querying, filtering and reading-in a medium-sized `.pbf` file several times faster. This effect is even more pronounced for larger files.

At the time of writing, the package is under review by the ROpenSci foundation and we are working on the suggested changes. More precisely, we are developing a new `level` argument that should make the spatial matching operations more intuitive. We have already implemented a link between `oe_match()` and `Nominatim` servers to search for a location online if there's no match in the providers' data. After completing the review process, we will submit the package to CRAN.

Finally, the next version of the package will also include a more intuitive approach for defining spatial and regular queries through the `vectortranslate_options` argument, analogous to `query` and `wkt_filter` parameters.

Bibliography

(OGC) Open Geospatial Consortium Inc (2011). *OpenGIS Implementation Specification for Geographic information - Simple feature access - Part 1: Common architecture*. Tech. rep. OGC 06-103r4. (OGC) Open Geospatial Consortium Inc. URL: https://www.ogc.org/standards/sfa.

Abdel-Aty, Mohamed et al. (2013). "Geographical unit based analysis in the context of transportation safety planning". In: *Transportation Research Part A: Policy and Practice* 49, pp. 62–75.

Agresti, Alan (2015). *Foundations of linear and generalized linear models*. John Wiley & Sons.

Aguero-Valverde, Jonathan and Paul P Jovanis (2006). "Spatial analysis of fatal and injury crashes in Pennsylvania". In: *Accident Analysis & Prevention* 38.3, pp. 618–625.

– (2008). "Analysis of Road Crash Frequency with Spatial Models". In: *Transportation Research Record* 2061.1, pp. 55–63.

Alarifi, Saif A et al. (2018). "Exploring the effect of different neighboring structures on spatial hierarchical joint crash frequency models". In: *Transportation Research Record* 2672.38, pp. 210–222.

Anderson, Jennings, Dipto Sarkar, and Leysia Palen (2019). "Corporate editors in the evolving landscape of OpenStreetMap". In: *ISPRS International Journal of Geo-Information* 8.5, p. 232.

Ang, Qi Wei, Adrian Baddeley, and Gopalan Nair (2012). "Geometrically corrected second order analysis of events on a linear network, with applications to ecology and criminology". In: *Scandinavian Journal of Statistics* 39.4, pp. 591–617.

Anscombe, Francis John et al. (1961). "Examination of residuals". In: *Proceedings of the Fourth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Contributions to the Theory of Statistics*. The Regents of the University of California.

Appelhans, Tim (2020). *leafgl: High-Performance 'WebGl' Rendering for Package 'leaflet'*. R package version 0.1.1. URL: https://CRAN.R-project.org/package=leafgl.

Baddeley, Adrian, Gopalan Nair, et al. (2020). "Analysing point patterns on networks—A review". In: *Spatial Statistics*, p. 100435.

Baddeley, Adrian, Ege Rubak, and Rolf Turner (2015). *Spatial point patterns: methodology and applications with R*. CRC press.

Banerjee, Prithish et al. (2018). "A note on the Adaptive LASSO for Zero-Inflated Poisson regression". In: *Journal of Probability and Statistics* 2018.

Bao, Qiong et al. (2012). "Improved hierarchical fuzzy TOPSIS for road safety performance evaluation". In: *Knowledge-based systems* 32, pp. 84–90.

Barndorff-Nielsen, Ole Eiler (1989). *Asymptotic techniques; for use in statistics*. Tech. rep.

Barrington-Leigh, Christopher and Adam Millard-Ball (2017). "The world's user-generated road map is more than 80% complete". In: *PloS one* 12.8, e0180698.

Barthélemy, Marc (2011). "Spatial networks". In: *Physics Reports* 499.1-3, pp. 1–101.

Barua, Sudip, Karim El-Basyouny, and Md Tazul Islam (2014). "A full Bayesian multivariate count data model of collision severity with spatial correlation". In: *Analytic Methods in Accident Research* 3, pp. 28–43.

El-Basyouny, Karim and Tarek Sayed (2009). "Urban arterial accident prediction models with spatial effects". In: *Transportation Research Record* 2102.1, pp. 27–33.

Bayisa, Fekadu L et al. (2020). "Large-scale modelling and forecasting of ambulance calls in northern Sweden using spatio-temporal log-Gaussian Cox processes". In: *Spatial Statistics* 39, p. 100471.

Besag, Julian (1974). "Spatial interaction and the statistical analysis of lattice systems". In: *Journal of the Royal Statistical Society: Series B (Methodological)* 36.2, pp. 192–225.

Besag, Julian and Charles Kooperberg (1995). "On conditional and intrinsic autoregressions". In: *Biometrika* 82.4, pp. 733–746.

Bivand, Roger (2019). *rgrass7: Interface Between GRASS 7 Geographical Information System and R*. R package version 0.2-1. URL: https://CRAN.R-project.org/package=rgrass7.

Bivand, Roger S., Edzer Pebesma, and Virgilio Gómez-Rubio (June 2013). *Applied Spatial Data Analysis with R*. en. 2nd ed. Springer Science & Business Media.

Blangiardo, Marta and Michela Cameletti (2015). *Spatial and spatio-temporal Bayesian models with R-INLA*. John Wiley & Sons.

Blondel, Vincent D et al. (2008). "Fast unfolding of communities in large networks". In: *Journal of statistical mechanics: theory and experiment* 2008.10, P10008.

Boeing, Geoff (2017). "OSMnx: New methods for acquiring, constructing, analyzing, and visualizing complex street networks". In: *Computers, Environment and Urban Systems* 65, pp. 126–139.

Borruso, Giuseppe (2003). "Network density and the delimitation of urban areas". In: *Transactions in GIS* 7.2, pp. 177–191.

Borruso, Giuseppe (2005). "Network density estimation: analysis of point patterns over a network". In: *International Conference on Computational Science and Its Applications*. Springer, pp. 126–132.

– (2008). "Network density estimation: a GIS approach for analysing point patterns in a network space". In: *Transactions in GIS* 12.3, pp. 377–402.

Botella-Rocamora, P, A Lopez-Quilez, and MA Martinez-Beneito (2013). "Spatial moving average risk smoothing". In: *Statistics in Medicine* 32.15, pp. 2595–2612.

Boulieri, Areti et al. (2017). "A space–time multivariate Bayesian model to analyse road traffic accidents by severity". In: *Journal of the Royal Statistical Society: Series A (Statistics in Society)* 180.1, pp. 119–139.

Box, George EP and Kenneth B Wilson (1951). "On the experimental attainment of optimum conditions". In: *Journal of the royal statistical society: Series b (Methodological)* 13.1, pp. 1–38.

Braunholtz, David and Duncan Elliott (2019). *Estimating and adjusting for changes in the method of severity reporting for road accidents and casualty data*. Available at: https://assets.publishing.service.gov.uk/government/uploads/system/uploads/attachment_data/file/820588/severity-reporting-methodology-final-report.odt. Department for Transport.

Briz-Redón, Álvaro, Francisco Martínez-Ruiz, and Francisco Montes (2019a). "Estimating the occurrence of traffic accidents near school locations: a case study from Valencia (Spain) including several approaches". In: *Accident Analysis & Prevention* 132, p. 105237.

– (2019b). "Investigation of the consequences of the modifiable areal unit problem in macroscopic traffic safety analysis: a case study accounting for scale and zoning". In: *Accident Analysis & Prevention* 132, p. 105276.

Brodersen, Kay Henning et al. (2010). "The balanced accuracy and its posterior distribution". In: *2010 20th International Conference on Pattern Recognition*. IEEE, pp. 3121–3124.

Byrd, Richard H et al. (1995). “A limited memory algorithm for bound constrained optimization”. In: *SIAM Journal on scientific computing* 16.5, pp. 1190–1208.

Cameletti, Michela (2017). *An Introduction to Bayesian Computing with INLA*. URL: https://drive.google.com/file/d/1eW_0P6I_zWu3y_J8666o1MQghPcbCuf5.

Cameron, A Colin and Pravin K Trivedi (2013). *Regression analysis of count data*. Vol. 53. Cambridge university press.

Cardillo, Alessio et al. (2006). “Structural properties of planar graphs of urban street patterns”. In: *Physical Review E* 73.6, p. 066107.

Carlin, Bradley P, Sudipto Banerjee, et al. (2003). “Hierarchical multivariate CAR models for spatio-temporally correlated survival data”. In: *Bayesian Statistics* 7.7, pp. 45–63.

Carpenter, Bob et al. (2017). “Stan: A probabilistic programming language”. In: *Journal of statistical software* 76.1.

Chang, Winston et al. (2020). *shiny: Web Application Framework for R*. R package version 1.5.0. URL: https://CRAN.R-project.org/package=shiny.

Chorley, Richard J and Petercoed Haggett (1967). *Models in geography*. Methuen and Co.

Cooley, David (2020). *googleway: Accesses Google Maps APIs to Retrieve Data and Plot Maps*. R package version 2.7.3. URL: https://CRAN.R-project.org/package=googleway.

Cooper, Crispin HV and Alain JF Chiaradia (2020). “sDNA: 3-d spatial network analysis for GIS, CAD, Command Line & Python”. In: *SoftwareX* 12, p. 100525.

Cressie, Noel AC (1993). *Statistics for Spatial Data*. John Wiley & Sons, Ltd.

Cronie, Ottmar, Mehdi Moradi, and Jorge Mateu (2020). “Inhomogeneous higher-order summary statistics for point processes on linear networks”. In: *Statistics and Computing*.

Crucitti, Paolo, Vito Latora, and Sergio Porta (2006). “Centrality measures in spatial networks of urban streets”. In: *Physical Review E* 73.3, p. 036125.

Csardi, Gabor, Tamas Nepusz, et al. (2006). “The igraph software package for complex network research”. In: *InterJournal, complex systems* 1695.5, pp. 1–9.

Cunningham, Rebecca M, Maureen A Walton, and Patrick M Carter (2018). “The major causes of death in children and adolescents in the United States”. In: *New England Journal of Medicine* 379.25, pp. 2468–2475.

De Jong, Piet, Gillian Z Heller, et al. (2008). “Generalized linear models for insurance data”. In: *Cambridge Books*.

Department for Transport (2020). *Reported road casualties in Great Britain: 2019 annual report*. Available at: https://assets.publishing.service.gov.uk/government/uploads/system/uploads/attachment_data/file/922717/reported-road-casualties-annual-report-2019.pdf.

Diggle, Peter (1985). “A kernel method for smoothing point process data”. In: *Journal of the Royal Statistical Society: Series C (Applied Statistics)* 34.2, pp. 138–147.

Diggle, Peter J (2013). *Statistical analysis of spatial and spatio-temporal point patterns*. CRC press.

Diggle, Peter J, Jonathan A Tawn, and Rana A Moyeed (1998). “Model-based geostatistics”. In: *Journal of the Royal Statistical Society: Series C (Applied Statistics)* 47.3, pp. 299–350.

Dubin, Robin A (1988). “Estimation of regression coefficients in the presence of spatially autocorrelated error terms”. In: *The Review of Economics and Statistics*, pp. 466–474.

Dugas, Charles et al. (2003). “Statistical learning algorithms applied to automobile insurance ratemaking”. In: *CAS Forum*. Vol. 1. 1. Citeseer, pp. 179–214.

Dunnington, Dewey (2020a). *ggspatial: Spatial Data Framework for ggplot2*. R package version 1.1.4. URL: https://CRAN.R-project.org/package=ggspatial.

– (2020b). *qgisprocess: Use 'QGIS' Processing Algorithms.* URL: https://github.com/paleolimbot/qgisprocess.

Egilmez, Gokhan and Deborah McAvoy (2013). "Benchmarking road safety of US states: A DEA-based Malmquist productivity index approach". In: *Accident Analysis & Prevention* 53, pp. 55–64.

Euler, Leonhard (1741). "Solutio problematis ad geometriam situs pertinentis". In: *Comment. Acad. Sci. Petropolitanae 8*, pp. 128–140.

Fong, Youyi, Håvard Rue, and Jon Wakefield (2010). "Bayesian inference for generalized linear mixed models". In: *Biostatistics* 11.3, pp. 397–412.

Fortunato, Santo (2010). "Community detection in graphs". In: *Physics reports* 486.3-5, pp. 75–174.

Fotheringham, A Stewart, Chris Brunsdon, and Martin Charlton (2003). *Geographically weighted regression: the analysis of spatially varying relationships.* John Wiley & Sons.

Freni-Sterrantino, Anna, Massimo Ventrucci, and Håvard Rue (2018). "A note on intrinsic conditional autoregressive models for disconnected graphs". In: *Spatial and spatio-temporal epidemiology* 26, pp. 25–34.

Fu, Qian (2020). *PyDriosm: an open-source tool for downloading, reading and PostgreSQL-based I/O of OpenStreetMap data.* URL: https://pydriosm.readthedocs.io/en/latest/.

GDAL/OGR contributors (2020). *GDAL/OGR Geospatial Data Abstraction software Library.* Open Source Geospatial Foundation. URL: https://gdal.org.

Gelfand, Alan E and Penelope Vounatsou (2003). "Proper multivariate conditional autoregressive models for spatial data analysis". In: *Biostatistics* 4.1, pp. 11–15.

Gelman, Andrew, Jessica Hwang, and Aki Vehtari (2014). "Understanding predictive information criteria for Bayesian models". In: *Statistics and Computing* 24.6, pp. 997–1016.

Gelman, Andrew, Xiao-Li Meng, and Hal Stern (1996). "Posterior predictive assessment of model fitness via realized discrepancies". In: *Statistica Sinica*, pp. 733–760.

Gitelman, Victoria, Etti Doveh, and Shalom Hakkert (2010). "Designing a composite indicator for road safety". In: *Safety science* 48.9, pp. 1212–1224.

Gómez-Rubio, Virgilio (2020). *Bayesian Inference with INLA.* CRC Press.

GRASS Development Team (2017). *Geographic Resources Analysis Support System (GRASS GIS) Software, Version 7.2.* Open Source Geospatial Foundation. URL: http://grass.osgeo.org.

Guo, Lijia (2003). "Applying data mining techniques in property/casualty insurance". In: *in CAS 2003 Winter Forum, Data Management, Quality, and Technology Call Papers and Ratemaking Discussion Papers, CAS.* Citeseer.

Hagberg, Aric A., Daniel A. Schult, and Pieter J. Swart (2008). "Exploring Network Structure, Dynamics, and Function using NetworkX". In: *Proceedings of the 7th Python in Science Conference.* Ed. by Gaël Varoquaux, Travis Vaught, and Jarrod Millman. Pasadena, CA USA, pp. 11–15.

Haggett, Peter and Richard J Chorley (1969). *Network analysis in geography.* Vol. 1. Hodder Education.

Harris, Paul, Chris Brunsdon, and Martin Charlton (2011). "Geographically weighted principal components analysis". In: *International Journal of Geographical Information Science* 25.10, pp. 1717–1736.

Hastie, Trevor and Robert Tibshirani (1993). "Varying-coefficient models". In: *Journal of the Royal Statistical Society: Series B (Methodological)* 55.4, pp. 757–779.

Held, Leonhard, Birgit Schrödle, and Håvard Rue (2010). "Posterior and cross-validatory predictive checks: a comparison of MCMC and INLA". In: *Statistical modelling and regression structures.* Springer, pp. 91–110.

Hermans, Elke, Filip Van den Bossche, and Geert Wets (2008). "Combining road safety information in a performance index". In: *Accident Analysis & Prevention* 40.4, pp. 1337–1344.

– (2009). "Uncertainty assessment of the road safety index". In: *Reliability Engineering & System Safety* 94.7, pp. 1220–1228.

Hodges, James S, Bradley P Carlin, and Qiao Fan (2003). "On the precision of the conditionally autoregressive prior in spatial models". In: *Biometrics* 59.2, pp. 317–322.

Huang, Helai et al. (2016). "Macro and micro models for zonal crash prediction with application in hot zones identification". In: *Journal of Transport Geography* 54, pp. 248–256.

Jiang, Bin (2007). "A topological pattern of urban street networks: universality and peculiarity". In: *Physica A: Statistical Mechanics and its Applications* 384.2, pp. 647–655.

Jolliffe, Ian (2002). *Principal Component Analysis*. Ed. by Springer. 2nd ed. Springer Series in Statistics.

Jones, M Chris (1993). "Simple boundary correction for kernel density estimation". In: *Statistics and computing* 3.3, pp. 135–146.

Karduni, Alireza, Amirhassan Kermanshah, and Sybil Derrible (2016). "A protocol to convert spatial polyline data to network formats and applications to world urban road networks". In: *Scientific Data* 3.1, pp. 1–7.

Kirk, David S, Nicolo Cavalli, and Noli Brazil (2020). "The implications of ridehailing for risky driving and road accident injuries and fatalities". In: *Social Science & Medicine* 250, p. 112793.

Kolaczyk, Eric D and Gábor Csárdi (2014). *Statistical analysis of network data with R*. Vol. 65. Springer.

Lambert, Diane (1992). "Zero-inflated Poisson regression, with an application to defects in manufacturing". In: *Technometrics* 34.1, pp. 1–14.

Lämmer, Stefan, Björn Gehlsen, and Dirk Helbing (2006). "Scaling laws in the spatial structure of urban road networks". In: *Physica A: Statistical Mechanics and its Applications* 363.1, pp. 89–95.

Lindgren, Finn, Håvard Rue, et al. (2015). "Bayesian spatial modelling with R-INLA". In: *Journal of Statistical Software* 63.19, pp. 1–25.

Lord, Dominique and Fred Mannering (2010). "The statistical analysis of crash-frequency data: a review and assessment of methodological alternatives". In: *Transportation research part A: policy and practice* 44.5, pp. 291–305.

Lord, Dominique, Simon Washington, and John N Ivan (2007). "Further notes on the application of zero-inflated models in highway safety". In: *Accident Analysis & Prevention* 39.1, pp. 53–57.

Lovelace, Robin and Richard Ellison (2018). "stplanr: A package for transport planning". In: *The R Journal* 10.2, pp. 7–23.

Lovelace, Robin, Anna Goodman, et al. (2017). "The Propensity to Cycle Tool: An open source online system for sustainable transport planning". In: *Journal of transport and land use* 10.1, pp. 505–528.

Lovelace, Robin, Malcolm Morgan, et al. (2019). "stats19: A package for working with open road crash data". In: *Journal of Open Source Software* 4.33, p. 1181.

Lovelace, Robin, Jakub Nowosad, and Jannes Muenchow (2019). *Geocomputation with R*. CRC Press.

Lu, Binbin et al. (2018). "Shp2graph: Tools to Convert a Spatial Network into an Igraph Graph in R". In: *ISPRS International Journal of Geo-Information* 7.8, p. 293. URL: https://doi.org/10.3390/ijgi7080293.

Lu, Yongmei and Xuwei Chen (2007). "On the false alarm of planar K-function when analyzing urban crime distributed along streets". In: *Social Science Research* 36.2, pp. 611–632.

Ma, Xiaoxiang, Suren Chen, and Feng Chen (2017). "Multivariate space-time modeling of crash frequencies by injury severity levels". In: *Analytic Methods in Accident Research* 15, pp. 29–40.

MacKay, Murray (Jan. 1972). "Traffic Accidents—a Modern Epidemic". In: *International Journal of Environmental Studies* 3.1-4, pp. 223–227. ISSN: 0020-7233. DOI: 10.1080/00207237208709519.

Mannering, Fred (2018). "Temporal instability and the analysis of highway accident data". In: *Analytic methods in accident research* 17, pp. 1–13.

Mannering, Fred L and Chandra R Bhat (2014). "Analytic methods in accident research: Methodological frontier and future directions". In: *Analytic methods in accident research* 1, pp. 1–22.

Map, Open Street (2020a). *Osmium Library: A fast and flexible C++ library for working with OpenStreetMap data*. URL: https://osmcode.org/libosmium/.

– (2020b). *Osmosis: A command line Java application for processing OSM data*. URL: https://wiki.openstreetmap.org/wiki/Osmosis.

Mardia, KV (1988). "Multi-dimensional multivariate Gaussian Markov random fields with application to image processing". In: *Journal of Multivariate Analysis* 24.2, pp. 265–284.

Marshall, EC and DJ Spiegelhalter (2003). "Approximate cross-validatory predictive checks in disease mapping models". In: *Statistics in Medicine* 22.10, pp. 1649–1660.

Marshall, Stephen et al. (2018). "Street network studies: from networks to models and their representations". In: *Networks and Spatial Economics* 18.3, pp. 735–749.

Martínez-Beneito, Miguel A and Paloma Botella-Rocamora (2019). *Disease Mapping: From Foundations to Multidimensional Modeling*. CRC Press.

Martins, Thiago G et al. (2013). "Bayesian computing with INLA: new features". In: *Computational Statistics & Data Analysis* 67, pp. 68–83.

Matteson, David S et al. (2011). "Forecasting emergency medical service call arrival rates". In: *The Annals of Applied Statistics* 5.2B, pp. 1379–1406.

McCullagh, P. and J.A. Nelder (1989). *Generalized Linear Models, Second Edition*. Chapman and Hall/CRC Monographs on Statistics and Applied Probability Series. Chapman & Hall. ISBN: 9780412317606.

McSwiggan, Greg, Adrian Baddeley, and Gopalan Nair (2017). "Kernel density estimation on a linear network". In: *Scandinavian Journal of Statistics* 44.2, pp. 324–345.

– (2020). "Estimation of relative risk for events on a linear network". In: *Statistics and Computing* 30.2, pp. 469–484.

Miaou, Shaw-Pin (1994). "The relationship between truck accidents and geometric design of road sections: Poisson versus negative binomial regressions". In: *Accident Analysis & Prevention* 26.4, pp. 471–482.

Miaou, Shaw-Pin and Harry Lum (1993). "Modeling vehicle accidents and highway geometric design relationships". In: *Accident Analysis & Prevention* 25.6, pp. 689–709.

Miaou, Shaw-Pin and Joon Jin Song (2005). "Bayesian ranking of sites for engineering safety improvements: decision parameter, treatability concept, statistical criterion, and spatial dependence". In: *Accident Analysis & Prevention* 37.4, pp. 699–720.

Miaou, Shaw-Pin, Joon Jin Song, and Bani K Mallick (2003). "Roadway traffic crash mapping: a space-time modeling approach". In: *Journal of transportation and Statistics* 6, pp. 33–58.

Miller Jr, Rupert G (2011). *Survival analysis*. Vol. 66. John Wiley & Sons.

Moller, Jesper and Rasmus Plenge Waagepetersen (2003). *Statistical inference and simulation for spatial point processes*. CRC Press.

Moradi, M Mehdi and Jorge Mateu (2020). “First-and second-order characteristics of spatio-temporal point processes on linear networks”. In: *Journal of Computational and Graphical Statistics* 29.3, pp. 432–443.

Moradi, M Mehdi, Francisco J Rodríguez-Cortés, and Jorge Mateu (2018). “On kernel-based intensity estimation of spatial point patterns on linear networks”. In: *Journal of Computational and Graphical Statistics* 27.2, pp. 302–311.

Moraga, Paula (2019). *Geospatial health data: Modeling and visualization with R-INLA and shiny.* CRC Press.

Morris, Richard G and Marc Barthelemy (2014). “Spatial effects: Transport on interdependent networks”. In: *Networks of networks: the last frontier of complexity.* Springer, pp. 145–161.

Muff, Stefanie et al. (2015). “Bayesian analysis of measurement error models using integrated nested Laplace approximations”. In: *Journal of the Royal Statistical Society: Series C: Applied Statistics*, pp. 231–252.

Nantulya, Vinand M and Michael R Reich (May 2002). “The Neglected Epidemic: Road Traffic Injuries in Developing Countries”. In: *BMJ : British Medical Journal* 324.7346, pp. 1139–1141. ISSN: 0959-8138.

Noland, Robert B and Mohammed A Quddus (2004). “A spatially disaggregate analysis of road casualties in England”. In: *Accident Analysis & Prevention* 36.6, pp. 973–984.

Okabe, Atsuyuki, Toshiaki Satoh, and Kokichi Sugihara (2009). “A kernel density estimation method for networks, its computational method and a GIS-based tool”. In: *International Journal of Geographical Information Science* 23.1, pp. 7–32.

Okabe, Atsuyuki and Kokichi Sugihara (2012). *Spatial analysis along networks: statistical and computational methods.* John Wiley & Sons.

Open Geospatial Consortium (OGC) (2020). *GeoPackage.* URL: http://www.geopackage.org/.

Open Street Map (2017). *Nominatim: Open-source geocoding with OpenStreetMap data.* URL: https://nominatim.org/.

– (2020). *Open Street Map data - Accuracy.* Accessed: 2020-12-05. URL: https://wiki.openstreetmap.org/wiki/Accuracy#Topology.

Openshaw, Stan (1981). “The modifiable areal unit problem”. In: *Quantitative Geography: A British View*, pp. 60–69.

OpenStreetMap contributors (2017). *Planet dump retrieved from https://planet.osm.org.* URL: https://www.openstreetmap.org.

Ordnance Survey (2020). *Ordnance Survey.* Accessed: 2020-05-21. URL: https://www.ordnancesurvey.co.uk/.

PACTS (2020). *Roads Policing and Its Contribution to Road Safety.* Tech. rep. Parliamentary Advisory Council for Transport Safety.

Padgham, Mark (Feb. 2019). “dodgr: An R package for network flow aggregation”. In: *Transport Findings.* DOI: 10.32866/6945.

Padgham, Mark et al. (2017). “osmdata”. In: *Journal of Open Source Software* 2.14.

Palmi-Perales, Francisco, Virgilio Gomez-Rubio, and Miguel A. Martinez-Beneito (2019). *Bayesian Multivariate Spatial Models for Lattice Data with INLA.* arXiv: 1909.10804 [stat.CO].

Pebesma, Edzer J (2018). “Simple features for R: Standardized support for spatial vector data.” In: *The R Journal* 10.1, p. 439.

Pedersen, Thomas Lin (2020). *tidygraph: A Tidy API for Graph Manipulation.* R package version 1.2.0. URL: https://CRAN.R-project.org/package=tidygraph.

Plummer, Martyn (2012). "JAGS Version 3.3. 0 user manual". In: *International Agency for Research on Cancer, Lyon, France.*

Porta, Sergio, Paolo Crucitti, and Vito Latora (2006). "The network analysis of urban streets: a dual approach". In: *Physica A: Statistical Mechanics and its Applications* 369.2, pp. 853–866.

QGIS Development Team (2020) (2020). *QGIS Geographic Information System. Open Source Geospatial Foundation Project.* URL: http://qgis.osgeo.org/.

R Core Team (2020). *R: A Language and Environment for Statistical Computing.* R Foundation for Statistical Computing. Vienna, Austria. URL: https://www.R-project.org/.

Rakshit, Suman, Adrian Baddeley, and Gopalan Nair (2019). "Efficient Code for Second Order Analysis of Events on a Linear Network". In: *Journal of Statistical Software* 90.1, pp. 1–37.

Rakshit, Suman, Tilman Davies, et al. (2019). "Fast Kernel Smoothing of Point Patterns on a Large Network using Two-dimensional Convolution". In: *International Statistical Review* 87.3, pp. 531–556.

Ripley, Brian D (1977). "Modelling spatial patterns". In: *Journal of the Royal Statistical Society: Series B (Methodological)* 39.2, pp. 172–192.

Robert, Christian and George Casella (2013). *Monte Carlo statistical methods.* Springer Science & Business Media.

Rosić, Miroslav et al. (2017). "Method for selection of optimal road safety composite index with examples from DEA and TOPSIS method". In: *Accident Analysis & Prevention* 98, pp. 277–286.

Rosolino, Vaiana et al. (2014). "Road safety performance assessment: a new road network Risk Index for info mobility". In: *Procedia-social and behavioral sciences* 111, pp. 624–633.

Rue, Havard and Leonhard Held (2005). *Gaussian Markov random fields: theory and applications.* CRC press.

Rue, Håvard, Sara Martino, and Nicolas Chopin (2009). "Approximate Bayesian inference for latent Gaussian models by using integrated nested Laplace approximations". In: *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 71.2, pp. 319–392.

Rue, Håvard, Andrea Riebler, et al. (2017). "Bayesian computing with INLA: a review". In: *Annual Review of Statistics and Its Application* 4, pp. 395–421.

Salmon, Maëlle et al. (2015). "Bayesian outbreak detection in the presence of reporting delays". In: *Biometrical Journal* 57.6, pp. 1051–1067.

Sauter, Rafael and Leonhard Held (2016). "Quasi-complete separation in random effects of binary response mixed models". In: *Journal of Statistical Computation and Simulation* 86.14, pp. 2781–2796.

Savolainen, Peter T et al. (2011). "The statistical analysis of highway crash-injury severities: a review and assessment of methodological alternatives". In: *Accident Analysis & Prevention* 43.5, pp. 1666–1676.

Shah, Syyed Adnan Raheel et al. (2018). "Road safety risk assessment: an analysis of transport policy and management for low-, middle-, and high-income Asian countries". In: *Sustainability* 10.2, p. 389.

Shankar, Venkataraman, Fred Mannering, and Woodrow Barfield (1995). "Effect of roadway geometrics and environmental factors on rural freeway accident frequencies". In: *Accident Analysis & Prevention* 27.3, pp. 371–389.

Shen, Haipeng and Jianhua Z. Huang (June 2008a). "Forecasting time series of inhomogeneous Poisson processes with application to call center workforce management". In: *Ann. Appl. Stat.* 2.2, pp. 601–623. DOI: 10.1214/08-AOAS164. URL: https://doi.org/10.1214/08-AOAS164.

– (2008b). “Interday Forecasting and Intraday Updating of Call Center Arrivals”. In: *Manufacturing & Service Operations Management* 10.3, pp. 391–410. DOI: 10.1287/msom.1070.0179.

Snow, John (1855). *On the Mode of Communication of Cholera*. John Churchill.

Spiegelhalter, David et al. (2003). *WinBUGS user manual*.

Spiegelhalter, David J et al. (2002). “Bayesian measures of model complexity and fit”. In: *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 64.4, pp. 583–639.

Staudt, Christian L., Aleksejs Sazonovs, and Henning Meyerhenke (Dec. 2016). “NetworKit: A tool suite for large-scale complex network analysis”. en. In: *Network Science* 4.4, pp. 508–530. ISSN: 2050-1242, 2050-1250. DOI: 10.1017/nws.2016.20. (Visited on 04/30/2020).

Stern, Hal S and Noel Cressie (2000). “Posterior predictive model checks for disease mapping models”. In: *Statistics in Medicine* 19.17-18, pp. 2377–2397.

Sugihara, Kokichi, Toshiaki Satoh, and Atsuyuki Okabe (2010). “Simple and unbiased kernel function for network analysis”. In: *2010 10th International Symposium on Communications and Information Technologies*. IEEE, pp. 827–832.

Sumner, Michael D. and Mark Padgham (2020). *silicate: Common Forms for Complex Hierarchical and Relational Data Structures*. R package version 0.7.0. URL: https://CRAN.R-project.org/package=silicate.

Szufel, Przemysław (2020). *OpenStreetMapX.jl: OpenStreetMap (*.osm) support for Julia 1.0 and up*. URL: https://github.com/pszufe/OpenStreetMapX.jl.

Takane, Yoshio and Michael A Hunter (2001). “Constrained principal component analysis: a comprehensive theory”. In: *Applicable Algebra in Engineering, Communication and Computing* 12.5, pp. 391–419.

Tenkanen, Henrikki (2020). *Pyrosm: OpenStreetMap PBF data parser for Python*. URL: https://pyrosm.readthedocs.io/en/latest/.

Tennekes, Martijn (2018). “tmap: Thematic Maps in R”. In: *Journal of Statistical Software* 84.6, pp. 1–39.

Thomas, Isabelle (1996). “Spatial data aggregation: exploratory analysis of road accidents”. In: *Accident Analysis & Prevention* 28.2, pp. 251–264.

Tsai, Henghsiu and Ruey S Tsay (2010). “Constrained factor models”. In: *Journal of the American Statistical Association* 105.492, pp. 1593–1605.

UK Data Service Census Support (2014). *Census Support: Flow Data*. URL: http://wicid.ukdataservice.ac.uk/.

Ukkusuri, Satish et al. (2012). “The role of built environment on pedestrian crash frequency”. In: *Safety Science* 50.4, pp. 1141–1151.

Van der Meer, Lucas et al. (2021). *sfnetworks: Tidy Geospatial Networks in R*. URL: https://luukvdmeer.github.io/sfnetworks/.

Van Rossum, Guido and Fred L. Drake (2009). *Python 3 Reference Manual*. Scotts Valley, CA: CreateSpace. ISBN: 1441412697.

Wall, Melanie M (2004). “A close look at the spatial structure implied by the CAR and SAR models”. In: *Journal of Statistical Planning and Inference* 121.2, pp. 311–324.

Wang, Chao, Mohammed A Quddus, and Stephen G Ison (2009). “Impact of traffic congestion on road accidents: a spatial analysis of the M25 motorway in England”. In: *Accident Analysis & Prevention* 41.4, pp. 798–808.

– (2011). “Predicting accident frequency at their severity levels and its application in site ranking using a two-stage mixed multivariate model”. In: *Accident Analysis & Prevention* 43.6, pp. 1979–1990.

Wang, Xuesong et al. (2016). “Macro-level safety analysis of pedestrian crashes in Shanghai, China”. In: *Accident Analysis & Prevention* 96, pp. 12–21.

Wang, Zhu, Shuangge Ma, and Ching-Yun Wang (2015). “Variable selection for zero-inflated and overdispersed data with application to health care demand in Germany”. In: *Biometrical Journal* 57.5, pp. 867–884.

Watanabe, Sumio (2010). “Asymptotic Equivalence of Bayes Cross Validation and Widely Applicable Information Criterion in Singular Learning Theory”. In: *Journal of Machine Learning Research* 11, pp. 3571–3594.

Wickham, Hadley et al. (2019). “Welcome to the tidyverse”. In: *Journal of Open Source Software* 4.43, p. 1686. DOI: 10.21105/joss.01686.

Wood, Simon N (2003). “Thin plate regression splines”. In: *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 65.1, pp. 95–114.

– (2017). *Generalized additive models: an introduction with R*. CRC press.

World Health Organization (2018). *Global Status Report On Road Safety 2018*. en. World Health Organization. ISBN: 978-92-4-156568-4.

Xie, Zhixiao and Jun Yan (2008). “Kernel density estimation of traffic accidents in a network space”. In: *Computers, environment and urban systems* 32.5, pp. 396–406.

Xu, Pengpeng, Helai Huang, and Ni Dong (2018). “The modifiable areal unit problem in traffic safety: basic issue, potential solutions and future research”. In: *Journal of Traffic and Transportation Engineering (English edition)* 5.1, pp. 73–82.

Yamada, Ikuho and Jean-Claude Thill (2004). “Comparison of planar and network K-functions in traffic accident analysis”. In: *Journal of Transport Geography* 12.2, pp. 149–158.

Zappa, Diego et al. (2019). “Text Mining In Insurance: From Unstructured Data To Meaning”. In.

Zappa Nardelli, Francesco et al. (Oct. 2018). “Julia Subtyping: A Rational Reconstruction”. In: *Proc. ACM Program. Lang.* 2.OOPSLA, 113:1–113:27. ISSN: 2475-1421. DOI: 10.1145/3276483. URL: https://doi.acm.org/10.1145/3276483.

Zhai, Xiaoqi et al. (2019). “The influence of zonal configurations on macro-level crash modeling”. In: *Transportmetrica A: transport science* 15.2, pp. 417–434.

Zhou, Zhengyi (2016). “Predicting ambulance demand: Challenges and methods”. In: *arXiv preprint arXiv:1606.05363*.

Zhou, Zhengyi and David S Matteson (2016). “Temporal and Spatiotemporal Models for Ambulance Demand”. In: *Healthcare Analytics: From Data to Knowledge to Healthcare Improvement*, p. 389.

Zhou, Zhengyi, David S Matteson, et al. (2015). “A spatio-temporal point process model for ambulance demand”. In: *Journal of the American Statistical Association* 110.509, pp. 6–15.

Zhou, Zhengyi and David S. Matteson (2015). “Predicting Ambulance Demand: A Spatio-Temporal Kernel Approach”. In: *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. KDD ’15. Sydney, NSW, Australia: Association for Computing Machinery, pp. 2297–2303. ISBN: 9781450336642.

Ziakopoulos, Apostolos and George Yannis (2020). “A review of spatial approaches in road safety”. In: *Accident Analysis & Prevention* 135, p. 105323.