POLITECNICO DI TORINO

Master of Science
Computer Engineering

Master's Thesis

Demand model generation for shared mobility: a KDE approach

Supervisor
Prof. Luca Vassio

Candidate
Maurizio Pinna

Co-Supervisors
Prof. Marco Mellia
Prof. Danilo Giordano
MSc. Alessandro Ciociola

April 2022

Summary

In recent times important changes on how people move around the cities have been happening. The expression "smart mobility" can well describe it. This definition includes: technology, infrastructures (park slots, charging stations, traffic signs, vehicles), mobility solutions and people. The aim is to lower the traffic and pollution, to create smart traffic flows with no interruption, and also boost the economies of scale in order to give everyone greater access to mobility. As a matter of fact, energy consumption is by far the greatest source of CO2 emissions, with 76% worldwide. This includes, among others, road transportation with 12.5%. In this optic, European Union has set the objective of reducing by 30% the emissions compared to 1990, by 2030. Furthermore, it has been proposed starting from 2035, an effective ban for new fossil-fuel cars, a clear signal to the car makers in order to accelerate their innovation on electric vehicles. In this context several business models inspired by sharing economy and Information Communication Technology grew up.

Vehicle sharing is the rental of vehicles by the hour or by the minute as opposed to traditional day or week-long rentals. Members of the system have access to a fleet of vehicles that they can rent on an as-needed basis. The fee charged is based on the length of the rental in hours or minutes. With the so called free floating vehicle sharing, where vehicles do not have home parking spaces but are instead can be parked anywhere within a city's operating area. The ever-increasing installation of IoT object leads to a consequent generation of BigData. IoT and Big Data are strictly interconnected, originating a continuous cycle: data creation from IoT, data collection and analysis, with big data analytics pipelines, new configuration of the manufacturing and maintenance processes with the information extracted from data.

In this thesis, we created a demand model that is capable to describe properly the free floating services users habits. The aim is to use this to extend an existing data-driven simulator named ODySSEUS, developed by Smart Data research group from which this work is supervised. The model

is based on Poisson Processes for time domain, and a four dimension Kernel Density Estimate (KDE) for space. We use different datasets from the most important car sharing company in Europe, cleaning and applying different pre-processing steps on them, in order to be able to extract useful information. We concentrate only in space domain and in particular in the optimization of the bandwidth parameter of KDE through machine learning cross validation approaches. We first investigate on the difference between discretized input and output and continuous one. We show the importance of optimizing properly the Fixed Bandwidth KDE, moreover we introduced Variable Bandwidth KDE approach. Results shows the difference between discretized input and output and continuous one, with different advantages with the continuous approach. Also, in terms of log-likelihood metric, the importance of optimizing the hyper-parameters bandwidth with machine learning cross validation approach. Yet, we illustrate how variable KDE can lead to some advantages in some cases.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Climate change, the biggest human challenge

Global climate change has become an absolute priority topic especially in the last few years, and for sure it is going the be one of the biggest challenge that society has ever faced. There is no doubt on climate emergency in the scientific community, and about 90% - 100% of publishing papers scientist agrees that human activity is the cause of the recent global warming. [27] What is really impacting on overheating are greenhouse gases (GHG), carbon dioxide and methane for most. According to the last Intergovernmental Panel on Climate Change, in 2019 atmospheric CO2 concentrations were higher that at any time in at least 2 million years. While since 1750, CO2 concentration raised about 47%. Global surface has increased faster since 1970 than in any other 50-year period at least over the last 2000 years.[39]

Energy consumption is by far the greatest source of the emissions, with 76% worldwide. This includes, among others, road transportation with 12.5% [30]

For those reasons governments all over the world are cooperating to find a solution. Paris Agreement was adopted by 196 countries at COP21 with the aim to limit global warming below 2, preferably to 1.5 degrees Celsius, compared to pre-industrial levels. To achieve this long-term temperature goal, countries aim to reach global peaking of GHG emissions as soon as possible to achieve a climate neutral world by mid century. European Union has set the objective of reducing by 30% the emissions compared to 1990, by 2030.

Furthermore, it has been proposed an effective ban for new fossil-fuel cars from 2035, a clear signal to the car makers in order to accelerate their innovation on electric vehicles. In this context, there is also an increasing pressure from the civil society. The awareness of the problem is spread more then ever, while international climate movement are more and more participated, with people asking the governments to act in the shortest time possible in order to preserve the world for the future generations. [33] With this in mind, green economy was born. This term describes an economy in which economic growth and environmental responsibility work together in a mutually reinforcing fashion while supporting progress on social development.[17]

1.2 How transportation is changing

Transportation sector is responsible for a remarkable part of the CO2 emissions. In the last years electric vehicles (EV) have earned a lot popularity. Several reasons have contributed to this trend. Primarily, technology is constantly advancing, so research are focused on developing vehicles significantly more efficient then the Internal Combustion Engine (ICE) ones. Then wider adoption of EV is a possible solution to lower the emission. Lastly, from the consumer point of view, the price of electricity is much lower than the price of fossil fuels. [43] [31]

1.2.1 Shared mobility and micro mobility

In addition to the revolution of the vehicles' engines and the way they are powered, a mutation how people decide to move is on going. The expression "smart mobility" can well describe it. This definition includes: technology, infrastructures (park slots, charging stations, traffic signs, vehicles), mobility solutions (e.g. new models) and people. Smart mobility does not only mean alternative transport options, but a wider and more complex phenomena based on the following principles:

- **Flexibility:** different mode of transport allow people to choose which one suites best in a given context
- **Efficiency:** The user can reach destination with the minimum effort and in the shortest time period

- **Integration:** The complete route can be planned without being forced to consider which transport options are used
- **Clean technologies:** Migration from internal combustion engine vehicles to zero emissions ones
- **Accessibility:** Everyone must be able to have access to different forms of smart mobility
- **Social benefits:** Smart mobility has to contribute to an increasing life quality

The final goal of smart mobility in the cities is to lower the traffic and pollution, to create smart traffic flows with no interruption, and also boost the economies of scale in order to give everyone greater access to mobility.

In this context, several business model inspired by sharing economy and Information and Communication Technologies (ICT) are playing a key role. Car sharing is the rental of vehicles by the hour or by the minute as opposed to traditional day- or week-long rentals. Members of the system have access to a fleet of vehicles that they can rent on an as-needed basis. The fee charged is based on the length of the rental in hours or minutes. New operators such as Car2go or Enjoy in Europe, created variation in the original business model, with the so called Free Floating Car Sharing (FFCS), where vehicles do not have home parking spaces but are instead can be parked anywhere within a city's operating area. Because the vehicles do not need to be returned to their starting point to complete a rental, this service is also known as one-way car sharing. Vehicles could be picked up or dropped off anywhere within the operating area, and while is possible to drive them outside it, the rental period can only be ended when the vehicle returns to the operating area. [36] This is possible thanks to a mobile app, where through GPS system, users can locate vehicles, book one of them, and open it. Figure 1.1 shows a car sharing mobile application through which is possible to use the services.

The majority of this services are still based on ICE vehicles, but some players are starting to adopt EV vehicle in their fleet. Technological development, especially in batteries manufacturing, has also increased the emergence of shared micro-mobility services including dock-less e-scooters, dock-less and docked bikes and e-bikes. The variety and availability of such services in major cities worldwide have grown rapidly, allowing an increasing number of users to choose between several modes and companies. The

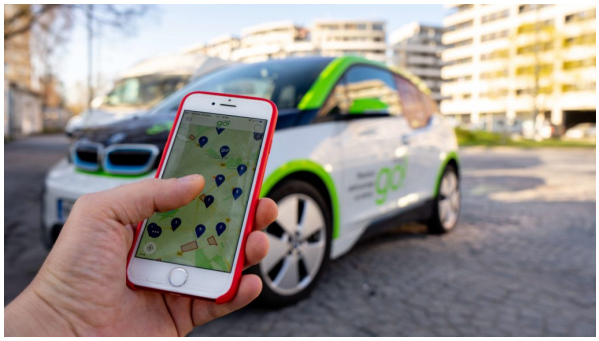

Figure 1.1. Car sharing mobile application.

working principle is basically the same of the FFCS one. Every provider has its own mobile application, by which the user can use the system. The main difference from EV is that when e-scooters and e-bikes run out of charge, have to be picked up in order to be charged. This is what the majority of the companies choose to do, but other solutions - such as a contribution from the user that brings the light vehicle at home - has been proposed as in [24]

Nowadays, data is one the most important components in all fields of research which is not surprising as the amount of data generated is constantly growing. The services mentioned above have contributed in large part to the creation of this data. The higher granularity of data the better information systems can be developed for improving the allocation of resources. Fundamental questions that need to be answered are how travellers adopt and use each mode, how usage varies between different modes, and how they impact urban mobility and its sustainability overall.

1.3 The development of IoT sensor, big data and data analysis

Talking about data analysis is impossible not to mention the constant growing presence of different sensors placed in several objects, that are able to collect raw data and share them on the internet. Those objects can be found different daily life scenarios, involving individuals and business companies:

household appliances that compose the so called Smart Home; manufacturing machines in Industry 4.0; vehicles and smart roads, all together integrated to shape the smart cities. The before mentioned can all be resumed with the definition *Internet of Things (IoT)*: the acronym is generally used to indicate every group of physical device, able to receive and share data on a wireless connection, with a limited human interaction on this process.

The ever-increasing installation of IoT object leads consequent generation of *BigData*, this term refers to those collection of data that are so huge, variegated, speedy transmitted that is very hard or almost impossible to analyze with ordinary techniques. Said that with the proper paradigms is possible to filter, order, analyze and extract value from them

So in this context *IoT* and *Big Data* are strictly interconnected, originating a continuous cycle:

- Data creation from IoT
- Data collection and analysis, with big data analytics pipelines
- New configuration of the manufacturing and maintenance processes with the information extracted from data.

The particular scenario under our analysis perfectly fits the dissertation because, beside the free floating application itself used by the customers, every vehicle is connected to the Internet, and collects and share important information such state of charge, position, average speed, and so on.

1.4 Research question

In this thesis we are going to implement a data-driven mobility demand model to characterize over the time and space the users' habits of different sharing services. The model is based on Poisson processes for the time domain and Kernel Density Estimate for the space. In a nutshell, Kernel Density Estimation is a non-parametric way to estimate the probability function of a random variable. The objective of density estimation is starting from a limited sample of data, and build assumptions of the underlying density function everywhere, also in the points of the domain where there is no data observation. In one dimension the concept is strictly similar to histograms. The difference is that with KDE, the contribution of every data point is smoothed out from a single point, to the region nearby. Summing up

all the components, gives the resulting estimation of the data and its probability density function. The explanatory figure 1.2 taken from Wikipedia, can help to understand the logic behind it. However, more details are going to be illustrated in the next chapters.

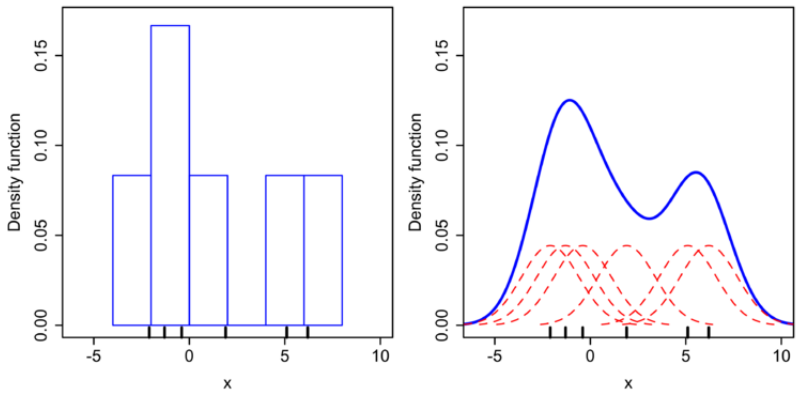

Figure 1.2. Histogram on the left and kernel Density Estimate with Gaussian Kernels on the right Source:[1]

The main focus is on the bandwidth parameter of the KDE, in order to understand:

- The difference between continue input and output KDE and discretized input and output one.
- How the bandwidth varies over different time slots
- How can data-driven approaches be used for the selection of the optimal bandwidth
- How a variable bandwidth KDE can overcome some limitations of the traditional fixed bandwidth KDE.

The final goal of this work is also to integrate the model in ODySSEUS, an Origin-Destination Simulator of Shared E-mobility in Urban Scenarios. This software was firstly implemented in [23] [25] [20] [26], and it was already extended in other works such as [53] [29].

1.5 Thesis organization

The thesis is organized as follows:

In chapter 2 are illustrated the Mathematical background which contains all the mathematical tools used and analyzed in this specific use domain; the topological background useful to understand, how different preprocessing step can help extract different information from the datasets; the technological background to know how this thesis has been developed and where is going to operate inside ODySSEUS.

In chapter 3 we concentrate on finding the optimal KDE bandwidth trough cross validation methods. We analyze also the difference between different coordinate systems, and why UTM one is preferred among the others. Finally we make an anlysys on how the bandwidth changes depending on the cardinality of the dataset and how samples are distributed.

Chapter 4 is dedicated to the variable or adaptive bandwidth kernel estimation. Here is discussed the model that we choose to investigate on it, how a variable bandwidth affects the final pdf, and what can be the advantages of this type of estimators.

Chapter 2

Background and previous work

2.1 Mathematical background

In these paragraph are contained definitions and discussion of the mathematical tools used for the work. This includes the revision of huge amount of literature, in particular in paragraph 2.1.2 where we mention different previous work, that are strictly mathematical. This because in section 2.4 we concentrate principally on those works that use in practice these tools both in the same research domain and others.

2.1.1 Multivariate Kernel Density Estimate

Kernel Density Estimate is a non parametric technique for the empirical estimation of density functions. [2]The multivariate form is defined as:

$$\widehat{f}_{\mathbf{H}}(\mathbf{x}) = \frac{1}{n}\sum_{i=1}^{n} K_{\mathbf{H}}(\mathbf{x} - \mathbf{x_i}) \tag{2.1}$$

where:

- $\mathbf{x} = (x_1, x_2, ..., x_d)^T, \mathbf{x_i} = (x_{i1}, x_{i2}, ..., x_{id})^T, i = 1,2, ..., n$ are d-vectors;
- **H** is the bandwidth (or smoothing) d×d matrix which is symmetric and positive definite;
- K is the Kernel function which is a symmetric multivariate density;

The choice of the kernel function does not effect the final distribution so much. As that, generally a Gaussian Kernel is preferred, because it is computationally efficient, yet assumes a uniform relationship between the variables. In this case $K_{\mathbf{H}}(\mathbf{x})$ in equation 2.1 is equal to:

$$K_{\mathbf{H}}(\mathbf{x}) = (2\pi)^{-\frac{d}{2}} |\mathbf{H}|^{-\frac{1}{2}} e^{-\frac{1}{2}\mathbf{x^T H^{-1} x}}$$

as a consequence, the equation 2.1 can be rewritten as:

$$\widehat{f}_{\mathbf{H}}(\mathbf{x}) = \frac{1}{n}\sum_{i=1}^{n} (2\pi)^{-\frac{d}{2}} |\mathbf{H}|^{-\frac{1}{2}} e^{-\frac{1}{2}\mathbf{(x-x_i)^T H^{-1}(x-x_i)}} \tag{2.2}$$

The choice of the bandwidth instead is the key parameter for the Probability Density Function characterization. Three possibles modalities are possible for doing this[22]:

- For a problem with d-variables, a full symmetrical bandwidth matrix

  $$\mathbf{H} = \begin{bmatrix} h_{11}^2 & \cdots & h_{1d}^2 \\ \vdots & \ldots & \vdots \\ h_{d1}^2 & \cdots & h_{dd}^2 \end{bmatrix}$$

  that is a positive-definite matrix with d(d+1)/2 parameters., in which $h_{ik}^2 = h_{ki}^2$

- A diagonal matrix with only d parameters,
  $\mathbf{H} = diag(h_1^2, h_2^2, ..., h_d^2) = \mathbf{h} \otimes \mathbf{h}$, where $\mathbf{h} = (h_1, h_2, ..., h_d)^T$

- A diagonal matrix with one parameter, $\mathbf{H} = h^2\mathbf{I}$, where $\mathbf{I}$ is the identity matrix.

In figure 2.1 is possible to see the three alternatives in a bi-variate case displayed. On the left the diagonal matrix with one parameter. At the center the diagonal matrix with d parameters and on the right the fully symmetrical bandwidth matrix with d(d+1)/2 parameters.

2.1.2 Variable Bandwidth KDE

In some cases a fixed bandwidth kernel density estimate cannot describe properly a pdf, especially when it is characterized by multi-modality. With

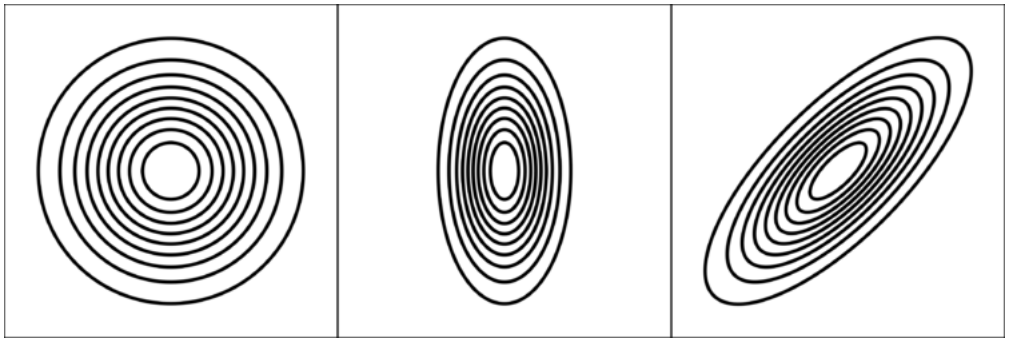

Figure 2.1. Comparison of the three bandwidths selection modalities in a bi-variate case. Source: [2]

this in mind, adaptive or variable bandwidth KDE (i.e., VKDE) was introduced, aiming to perform better than fixed bandwidth KDE [41] [46]. Basically a VKDE allows enlarging the bandwidth in zones of sparse data and restrict it, in zones with a lot of samples. There are different ways of define a VKDE, each one with some advantages and disadvantages both between each others, yet with respect to the fixed bandwidth KDE.

It is possible to distinguish VKDE in two main groups. Those which falls in the definition of Balloon Estimators, and Simple Point Estimators. The difference among them is how the different bandwidths are assigned in the space. [46]

Balloon Estimators

In models falling within this type a different but fixed bandwidth is selected for each estimation point $\mathbf{x}$. The estimate of f at $\mathbf{x}$ is then an average of identically scaled kernel at each data point. The equation that can characterize this behaviour is:

$$\widehat{f} = \frac{1}{nh\left(\mathbf{x}\right)^{d}} \sum_{i=1}^{n} K\left(\frac{\mathbf{x} - \mathbf{x_i}}{h\left(\mathbf{x}\right)}\right) \tag{2.3}$$

and in the multivariate form will be $\mathbf{H} = h(\mathbf{x})\mathbf{I}_d$.

This type of estimators have been introduced for the first time by Loftsgaarden and Quesenberry [38]in the form of k-th nearest neighbour estimator:

$$\widehat{f}\left(\mathbf{x}\right) = \frac{k}{nV_d h_k\left(\mathbf{x}\right)^{d}}$$

where $h_k(\mathbf{x})$ is the Euclidian distance from $\mathbf{x}$ to the kth nearest sample point and V_d is volume of unit sphere in $\Re^d$.

One of the most important disadvantages of balloon estimators is that when considered as global estimate, usually fail to integrate to one[45] [52] [50]. Because of this and for other reasons such as lack of packages that implements this estimators, we are not going deeper into this.

Sample Point Estimators

This class of estimation uses a different bandwidth for each data point and it can be described by:

$$\widehat{f}(\mathbf{x}) = \frac{1}{n}\sum_{i=1}^{n}\frac{1}{h(\mathbf{x}_i)^d}K\left(\frac{\mathbf{x}-\mathbf{x}_i}{h(\mathbf{x}_i)}\right) \tag{2.4}$$

The estimate of f at every $\mathbf{x}$ is then an average of differently scaled kernels centered at each data point.

With the bandwidth matrix in the multivariate in the form $\mathbf{H}(\mathbf{x_i}) = h(\mathbf{x}_i)\mathbf{I}_d$.

It was presented by Breiman, Meisel and Purcell in [21]. Asymptotically is equally to choose

$$h_i \propto f(\mathbf{x}_i)^{-\frac{1}{d}}$$

Abramson in [19] and [18] proposed a square root law, so that

$$h_i \propto f(\mathbf{x}_i)^{-\frac{1}{2}}$$

for any dimension.

Silverman in [49] [51] [50] provided an implementation of a sample point estimator that is composed by steps.

Primarily a fixed KDE $\widetilde{f}(\mathbf{x})$ is calculated, ensuring $\widetilde{f}(\mathbf{x}_i) > 0$ for every i. Next, the local bandwidth multiplicative factors λ_i are defined as:

$$\lambda_i = \left(\frac{\widetilde{f}(\mathbf{x}_i)}{g}\right)^{-\alpha}$$

where $\log g$ defined as

$$\frac{1}{n}\sum_i \log \widetilde{f}(\mathbf{x}_i)$$

is the geometric mean of the $\widetilde{f}(\mathbf{x}_i)$.

α is a modulation parameter so that $0 < \alpha < 1$. This determines how much the local bandwidth is enlarged in low sample density areas and how much is restricted in high sample density ones. $\alpha = 0$ is equivalent to the fixed KDE, while $\alpha = 1$ corresponds to the nearest neighbour approach.

Then, the **adaptive KDE** is defined as

$$\widehat{f}(\mathbf{x}_i) = \frac{1}{n}\sum_{i=1}^{n} \frac{1}{(h\lambda_i)^d} K\left(\frac{\mathbf{x}-\mathbf{x}_i}{h\lambda_i}\right) \tag{2.5}$$

This estimator has been further investigated for the scope of this work, in order to understand if it can lead to considerable advantages.

Other solution has been proposed in literature such as [56] where the local bandwidth b_i can be viewed as the average merging distance required in forming the nested sequence of clusters; or [34] where each bandwidth b_i is determined with the Integrate Squared Error criterion combined with an L_2 regularization term.

2.1.3 Maximum Likelihood Estimation (MLE)

From Wikipedia: "Maximum likelihood estimation (MLE) is a method of estimating the parameters of an assumed probability distribution, given some observed data. This is achieved by maximizing a likelihood function so that, under the assumed statistical model, the observed data is most probable. The point in the parameter space that maximizes the likelihood function is called the maximum likelihood estimate. The logic of maximum likelihood is both intuitive and flexible, and as such the method has become a dominant means of statistical inference.

From a statistical standpoint, a given set of observations is a random sample from an unknown population. The goal of maximum likelihood estimation is to make inferences about the population that is most likely to have generated the sample, specifically the joint probability distribution of the random variables $\{y_1, y_2, ...\}$, not necessarily independent and identically distributed. Associated with each probability distribution is a unique vector $\theta = [\theta_1, \theta_2, ..., \theta_k]^T$ of parameters that index the probability distribution. Evaluating the joint density at the observed data sample $\{y_1, y_2, ...\}$ gives a real-valued function, $L_n(\theta) = L_n(\theta; y) = f_n(y; \theta)$ which is called the likelihood function. The goal of maximum likelihood estimation is to find

the values of the model parameters that maximize the likelihood function over the parameter space, that is $\widehat{\theta} = \arg\max \widehat{L}_n(\theta; y)$"[3]

2.2 Topographical background

2.2.1 Geographical coordinate system and WGS84

The geographic coordinate system (GCS) is a spherical or ellipsoidal coordinate system for measuring and communicating positions directly on the Earth as latitude and longitude. [4] The World Geodetic System (WGS) is a standard for use in cartography, geodesy, and satellite navigation including GPS. The latest revision is WGS 84 (also known as WGS 1984, EPSG:4326), established and maintained by the United States National Geospatial-Intelligence Agency since 1984, and last revised in 2014[5]. Decimal degrees (DD) express latitude and longitude geographic coordinates as decimal fractions of a degree. DD are used in many geographic information systems (GIS), web mapping applications such as OpenStreetMap, and GPS devices.

2.2.2 Haversine Distance

The Haversine formula determines the great-circle distance between two points on a sphere given their longitudes and latitudes. Important in navigation, it is a special case of a more general formula in spherical trigonometry, the law of Haversines, that relates the sides and angles of spherical triangles.

$$d = 2{\cdot}r{\cdot}\arcsin\sqrt{hav(\varphi_2 - \varphi_1) + (1 - hav(\varphi_1 - \varphi_2) - hav(\varphi_1 + \varphi_2)) \cdot hav(\lambda_2 - \lambda_1)} \tag{2.6}$$

where:
λ_1, λ_2 are the longitude of point 1 and longitude of point 2
φ_1, φ_2 are the latitude of point 1 and latitude of point 2.

2.2.3 Mercator Projection and Web Mercator Coordinate System

The Mercator projection is a cylindrical map projection. It became the standard map projection for navigation because it is unique in representing

north as up and south as down everywhere while preserving local directions and shapes. The map is thereby conformal. As a side effect, the Mercator projection inflates the size of objects away from the equator. This inflation is very small near the equator but accelerates with increasing latitude to become infinite at the poles. As a result, landmasses such as Greenland and Antarctica appear far larger than they actually are relative to landmasses near the equator, such as Central Africa. [6] Many major online street mapping services (Bing Maps, Google Maps, Mapbox, MapQuest, OpenStreetMap, Yahoo! Maps, and others) use a variant of the Mercator projection for their map images called Web Mercator or Google Web Mercator. [7] As a projection, longitude and latitude are expressed in meters.

2.2.4 UTM coordinate system

The Universal Transverse Mercator (UTM) is a map projection system for assigning coordinates to locations on the surface of the Earth. It differs from global latitude/longitude in that it divides earth into 60 zones and projects each to the plane as a basis for its coordinates. Specifying a location means specifying the zone and the x, y coordinate in that plane. [8] The projection

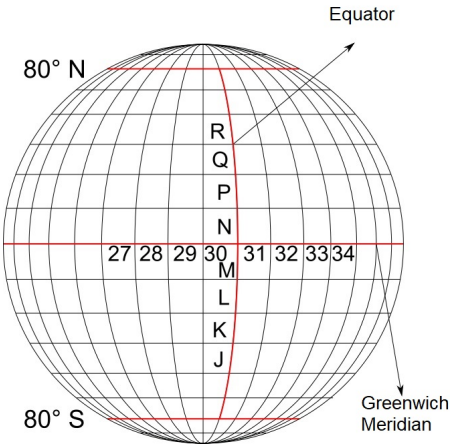

Figure 2.2. UTM coordinates zones grid

from spheroid to a UTM zone is some parameterization of the transverse Mercator projection. The parameters vary by nation or region or mapping system. Most zones in UTM span 6 degrees of longitude, and each has a designated central meridian. Web Mercator and UTM coordinate systems

share some properties, yet have some important differncies:

- Web Mercator (EPSG:3857) is Direct Mercator, it does not deform distances along the equator. UTM is a group of Transverse, they do not deform distances along each central meridian (not exactly since they are secant).

- Web Mercator is spherical (but based on ellipsoidal datum, so not exactly conformal). UTM are ellipsoidal projections, and conformal.

- Web Mercator is tangential to the equator. UTM are secant in the vicinity of each central meridian, as it is possible to understand in figure 2.3

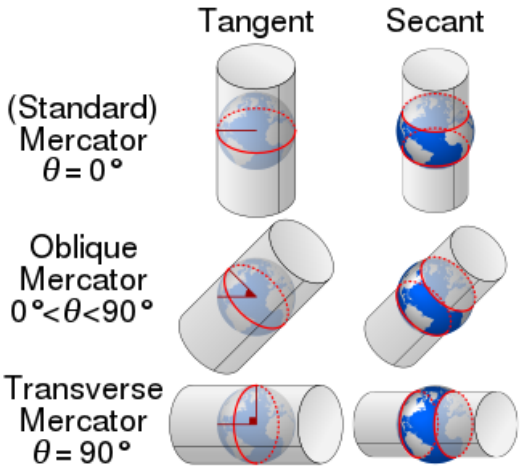

Figure 2.3. Comparison of tangent and secant forms of normal, oblique and transverse Mercator projections with standard parallels in red.

- Web Mercator is one for the globe, so distances deforms far away the equator. UTM is divided in zones, so when the distance is being deformed far away each central meridian, you enter another zone, with another central meridian.

- In Web Mercator parallels and meridians are horizontal and vertical lines. In UTM zones, not necessarilly: central meridian is a vertical line and the equator is an horizontal line, but all meridians and parallels are families of orthogonal curves.

2.3 ICT background

2.3.1 Python

Python is one of the most popular programming languages. It supports various paradigms, such the object oriented programming one. It integrates well with other software components making it a general purpose language that can be used to build a full end to end pipeline.

The language can be used for different area such as Web Development, Data Science and Machine Learning, Simulation, and everyone of this are part of the research project which this thesis is part of. For this reason Python has been preferred over other languages such as R. Another selling point is there are several libraries that can be imported. Strictly related to this work, the most important that have been used are:

- **Numpy:** adding support for large, multi-dimensional arrays and matrices, along with a large collection of high-level mathematical functions to operate on these arrays. The core functionality of NumPy is its "ndarray", for n-dimensional array, data structure. These arrays are homogeneously typed.[9]
- **Pandas:** a software library for data manipulation and analysis. In particular, it offers data structures and operations for manipulating numerical tables and time series. [40] [10]
- **Geopandas:** an open source project to make working with geospatial data in python easier. GeoPandas extends the datatypes used by pandas to allow spatial operations on geometric types. Geometric operations are performed by shapely. [35] [11]
- **Matplotlib:** a comprehensive library for creating static, animated, and interactive visualizations. [32] [12]
- **Scikit-Learn:** a free software machine learning library. It features various classification, regression and clustering algorithms. It is designed to interoperate with the Python numerical and scientific libraries NumPy and SciPy. [42][13]

  The two most important features used are:

  – KernelDensity: Kernel Density Estimation algorithm which uses the Ball Tree or KD Tree for efficient queries.

  - GridSearchCV: Grid Search generates a different combination of all the specified hyperparameters and their values and calculates the performance for each combination and selects the best value for the hyperparameters. This makes the processing time-consuming and expensive based on the number of hyperparameters involved. In particular, we used this for the bandwidth, global bandwidth and alpha parameter of the different Kernel Density Estmations.

- **Contextily:** a package to retrieve tile maps from the internet. It can add those tiles as basemap to matplotlib figures or write tile maps to disk into geospatial raster files.[14]

- **AwKDE:** small library that is able to generate multivariate Variable Bandwidth Kernel Density Estimation. [15] It uses the pybind11 package which makes creating C++ bindings super convenient. Only the evaluation is written in a small C++ snippet to speed it up, the rest is a pure python implementation. It is based [54] [48] [50] on which we are going to to take a closer look in the next chapter.

2.3.2 ODySSEUS

As mentioned in the introduction chapter ODySSEUS is an Origin-Destination Simulator of Shared E-mobility in Urban Scenarios developed by Smart Data Research group by which this work is supervised. Citing the official website [16]: it is composed by four modules each one coming with its own API, command line interface and GUI:

- **City Data Manager** (previously UMAP [23])
  - Upload or collect raw input data from different sources.
  - Provide utilities to normalise data into a common format (e.g. column naming).
  - Provide a unified interface to access and analyse normalised data.

- **Demand Modelling**
  - Create different demand models using among the others Poisson-KDE that is the object of this thesis.
  - Evaluate the goodness of a demand model under different viewpoints and compare different demand models.

  - Create demand-side simulation scenarios

- **Supply Modelling**
  - Create fleets choosing between different modes and vehicles.
  - Create and configure refueling/charging infrastructures and energy mix for electricity production.

- **Simulator**
  - Create and run simulation scenarios based on supply and demand configurations.
  - Support for trip-level simulations and timeframe-level simulations .
  - Collect several performance metrics (satisfied demand, fleet handling cost, equivalent CO2 emissions, gross profit, ...)
  - Detailed interface to analyse simulation results

As introduced in section 1.4 this work is almost totally focused on City Data Manager and Demand Modelling module. In particular regarding the first one, different preprocessing steps are applied on raw data, while in Demand Modelling we operate redesigning the KDE model, introducing different optimizations as generally illustrated in Thesis Organization 1.5. Being the modules strictly interconnected, the innovation introduced in the first two modules affects also the Simulator one, in particular when it operates in Event Generation mode, in the way the trips are generated.

2.4 Related Works

While in section 2.1.2 are described several studies that analyze different types of KDE from a strictly mathematical point of view; in this section instead, we are going to examine some works where KDE is implemented in a practical way. As mentioned in 1.4 this thesis started from [23] [25] [20] [26] where a 4 dimensional KDE was implemented (in combination with Poisson processes in time domain) with a discretization of the input data, so that each city is divided in a grid composed by square bins (with sides equals to 500m for cars and 300m for scooters), each data point identified by longitude and latitude is associated with a single bin, identified with a zone-id. Gaussian Kernel was selected and a bandwidth equal to 1, which correspond to a

bin. The dimensions that characterize the KDE are the origin and destination point, each one composed by two coordinates on the city grid: Origin-x, Origin-y, Destination-x, Destination-y. As a consequence also the output of the KDE is discretized, which means that, when a sample operation is performed, four coordinates that indicates respectively the number of row and columns for the origin and the destination cell are extracted.

Concerning about other works that involves KDE in diffent domains of the urban mobility one, in [28] the authors, for the estimation of environmental contours of sea states, propose the use of bivariate KDE with adaptive bandwidth fro generating the join probability distribution of significant wave height and energy period. The estimator is of the type defined in 2.5

In [45] fixed univariate KDE and a Simple Point Estimator for Variable KDE is used to model the tag distribution of High-Troughput Sequencing data, a central technology in genome-wide studies of protein DNA interactions.

In [44], the authors compare KDE and V-KDE to model the distributions of the dispersion of pulsars and fast radio bursts to measure the Milky Way halo using data-driven techniques.

In [37], the authors use KDE to monitor the spatial risk of disease (cancer) in public health. The authors estimate a relative risk function comparing fixed and adaptive KDE, showing that they could detect risk areas with case data from a population-based cancer registry.

In [47], the authors estimate the soft bit error rate for an arbitrary communication system by estimating the conditional pdf of soft receiver outputs. To do this, they use a computational technique with Gaussian KDE and a particular Maximum Likelihood criterion to select the optimal smoothing parameter.

Other works use KDE in the context of mobility, but do not compare it with V-KDE. For example, in [55] the authors combined mobility data and geotagged social media data to explore the semantics behind people's movements. Again, the authors choose the bandwidth matrix $H = hI$ with a single parameter. Differently from our work, the smoothing parameter selection is chosen based on previous studies rather than cross-validation.

Another example is [57], where the authors propose a fixed KDE to estimate the dynamic geographical distribution of cell phone users. Then, they combined it with neural networks to predict future spatial and temporal distributions.

2.5 Datasets

The dataset used in this thesis were collected from car2go with the platform UMAP [23] in different cities worldwide, with the majority in Europe. Each raw dataset is normalized, so that the columns described in table 2.1 are coherent within different cities.

Among the preprocessing operation, different filters for the trips are applied: trips with origin or destination point that are located outside of the given city area are removed, as well as trips with an average speed lower then 120 km/h, a duration time smaller than one minute or greater then 60 minutes.

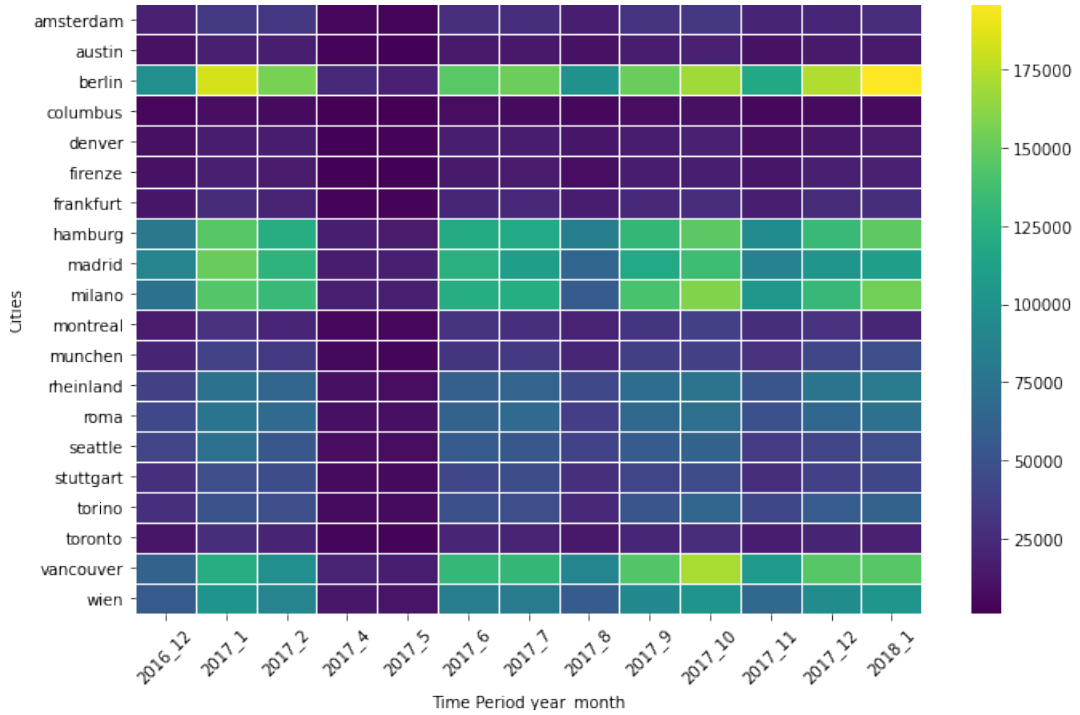

Figure 2.4. Heatmap of the dataset available collected with UMAP [23]

In figure 2.4 are represented the car sharing dataset of car2go. For limiting the duplication, only two of them (Amsterdam, Torino) are reported in this work, even if the results have been implemented in the software ODySSEUS for every city.

Attribute	Description	Example
Plate	Unique vehicle number	6-ZFG-77
Start time	Time stamp at which the trips started,in the respective time zone	2017-01-01 00:03:56+01:00
End time	Time stamp at which the trips ended,in the respective time zone	2017-01-01 00:45:30+01:00
Start longitude	Starting point longitude expressed in decimal degrees	4.90196
Start latitude	Starting point longitude expressed in decimal degrees	52.34124
End longitude	Ending point longitude expressed in decimal degrees	4.85089
End latitude	Ending point longitude expressed in decimal degrees	52.37312
Euclidean Distance	Euclidean distance from starting to ending point expressed in meters	4959.308
Duration	Total time the vehicle has been used for the given trip expressed in seconds	2088.035
Driving distance	Driving distance multiplied for a factor that considers the topography of each city	5107.30
Average speed	Average speed calculated dividing the driving distance by the duration, expressed in m/s	2.44598
Average speed	Average speed conversion from m/s to km/h	8.8055

Table 2.1. Datasets description. Other attributes, obtained directly from the time ones, has not been reported in the table for the sake of brevity. For example: year, month, daytype(weekday or weekend) and hour.

Chapter 3

Fixed KDE bandwidth optimization

In this chapter we are going to analyze how different bandwidths can affect the Kernel Density Function.

We maintained the approach used in [25] [20] [26] in which the temporal slot is divided in weekdays (from Monday to Friday) and weekend (Saturday and Sunday) and 24 slot for each daytype, one for every hour of the day. As a result, 48 time slots are considered, which means 48 KDEs.

Among three different option available to define the bandwidth parameter shown in section 2.1.1, we operate with a diagonal bandwidth matrix, composed by one parameter h, multiplied by the identity matrix. This for different reasons: firstly the most supported libraries available in Python (Scikit-Learn) in which is possible to implement KDE, operate in this way. Using Python is better from an engineering and integration point of view, since the entire ODySSEUS software is built in python too. Operating with a Scikit-Learn module like KernelDenisty one, leads to some important advantages. Since the class inherits from Scikit-Learn BaseEstimator(), is possible in combination with GridSearchCV, to customize or set the different the hyper-parameters and parameters in the most suitable way, case by case. We made a comparison between the cross-validation approach used by Scikit-Learn Kernel Density in combination with GridSearhcCV, and the one adopted by the library statsmodels. In the first one the bandwidth matrix H is composed by only one parameter h, while in last one, H is a diagonal matrix of d parameters. Still, in statsmodels it is not possible to select the number of folds used for the cross validation, neither the

bandwidths values set from which the optimal one will be selected. We compared the results and most importantly the execution time of the two different methods. For doing this we take a subset of the entire dataset (month of January, Weekday 09 time slot) with 1821 samples points and two-features. Scikit-Learn and GridSearch took 7.5 seconds to find the optimal bandwidth, while statsmodels KDEMultivariate took 1.34 minutes. It is then possible to understand that, when the dataset cardinality scale-up, the computation time tends to explode, as claimed by authors in [22]. The optimal banwidth values are comparable: 130.38m for Scikit-Learn KDE and [99.163, 124.112] for statsmodels. We don't certainly know if the fact statsmodels takes so long, is due to the fact that it computes d parameters for the bandwidths instead of one, or if it is caused by a less optimized implementation with respect to Scikit-Learn. Also authors in [22] stated that the computation time explodes when a bandwidth matrix composed of d parameters is used. Anyway for the reason just explained, we prefer using the Scikit-Learn approach, with the bandwidth matrix composed by one parameter. Furthermore we will see that when UTM coordinate system is used the simplification in the geographic case is acceptable since each dimension has the same unit of measurement (meters).

Set that we are going to operate with Scikit-Learn (at least for this entire chapter) different Kernel Density Estimates are fitted with a range of bandwidths. GridSearchCV from Scikit-Learn as mentioned in section 2.3.1 is used to find the optimal bandwidth, performing a n-fold cross validation. The trips which compose the sample space are divided into n-subsets a KDE is built using the samples from n-1 subsets, and the goodness is evaluated on the remaining subset by calculating the log-likelihood $\sum \log \widehat{p}\,(x_i)$. The procedure is replicated n-times, using a different subset as test set at every iteration. Then, an average score is calculated for each bandwidth, and the one with the higher score is selected as best.

In our case a **10-fold cross validation** is performed. This because it is a good trade-off between the cardinality of the trips of each time slot, the execution time and the variation of the log-likelihood.

3.1 Bi-dimensional KDE

The final goal is to implement a four dimensional KDE. This because the origin and the destination points of the trips are not uncorrelated. For

example if a user starts a trip from a neighbourhood in the city center, is high improbable that the trip will end in the same neighbourhood. Said that, several analysis ad the consequent results are proposed first in two-dimensions, considering only the starting point of the trips. This is done for readability and interpretability reasons, as it is possible to produce 2-D representation of the density functions. Moreover, a bi-dimensional demand model is perfectly suitable in those cases which the starting point and the final destination are not correlated, yet when the aim is to make an estimate from a fixed starting o destination point. With that being said, the same methodologies used in two dimension are applied in four dimensions.

3.1.1 KDE with discretized input and output

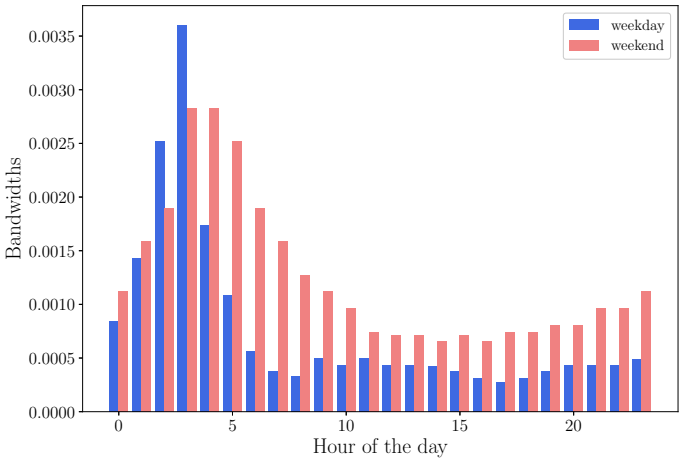

Figure 3.1. Amsterdam optimal bandwidth values for each time slot KDE. BW values expressed as fraction of square bin.

In section 2.4 we explained that a KDE with discretized input has been used in previous work. In this preliminary phase we explore the possibility of cross validation, using a range of bandwidths that are a fraction or a multiple of the single bin.

The results of this cross validation experiment with discretized input and output, are shown in figures 3.1, 3.2. In next section can be noticed that the general trend is comparable with the continuous versions. In general

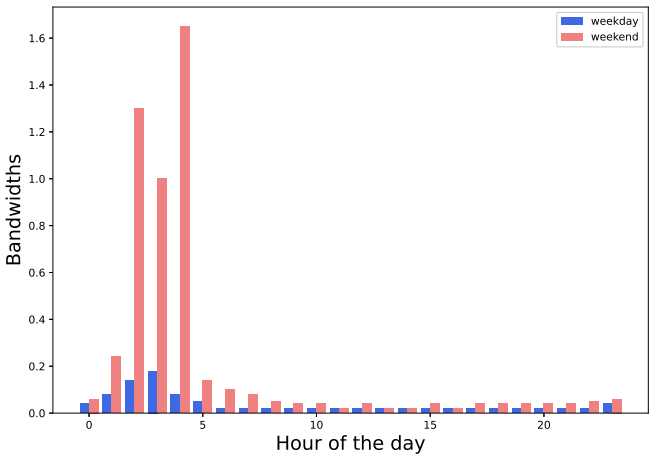

Figure 3.2. Turin optimal bandwidth values for each time slot KDE. BW values expressed as fraction of square bin.

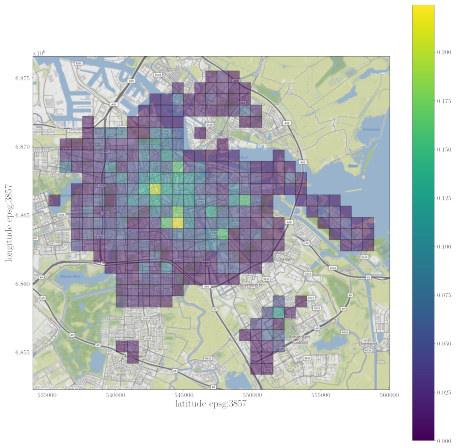

Figure 3.3. Amsterdam optimal bandwidth (0.12) KDE with discretized input and output. Time slot WD23.

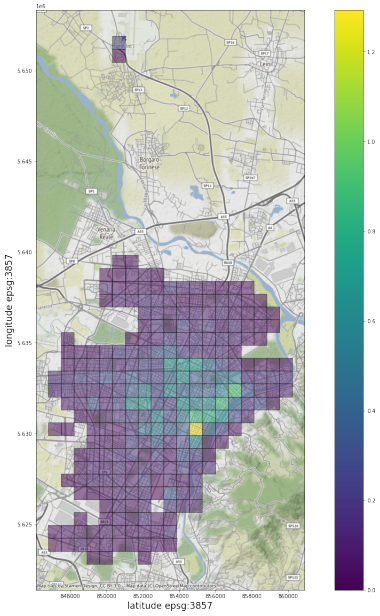

Figure 3.4. Turin optimal bandwidth (0.04) KDE with discretized input and output. Time slot WD23.

we have larger bandwidth values in those time slots less populated of trips, and smaller ones vice-versa. Even if the inputs and outputs are discretized, it is well observable (fig. 3.1, 3.4) for both cities, even we considered on purpose different time slots, that most of the trips are concentrated near the city centre. We will compare next, how useful can be this discretization combined with those type of qualitative plots.

3.1.2 Continuous input data, WGS84 coordinate system

Applying a discretization on the input data, increments a lot the probability that different sample points fall in the same bin. This can lead to a deterioration of the precision, yet the selection of the bandwidth more difficult and less precise, for the characteristics of the log-likelihood evaluation function.

Said that, another important advantage to operate on a continue domain is that is possible to apply a discretization after the sample on KDE has been made. With the precious possibility to change the bins side's dimension without having to tune the model again, therefore avoiding defining how the discretization is computed a priori (i.e. before the model is tuned), as in the case of discretized input and output.

For this reason, as first step it has been decided not to discretize the input data and firstly to keep it with the WGS84 coordinate system, as it has been collected.

Preliminary, we use all the trips for 2017. We divide the time space in 48 slot yet, so every slot is characterized by its own KDE.

Figures 3.5, 3.6 shows the bookings' trend grouped for each of the 48 slot in the entire year of 2017. Graphs of each single months are not reported, but they shows for every city, basically the same trend. It is possible to notice most of the trips are concentrated in weekdays in work commuting hours. During weekends instead, early in the morning the car sharing services are less used, while from mid day until late at night, more trips are completed.

Euclidean Distance

For computing the distance between two samples in GridSearchCV function, it is possible to use different metrics. In this sub-section we illustrate the cross validation with the Euclidean Distance metric.

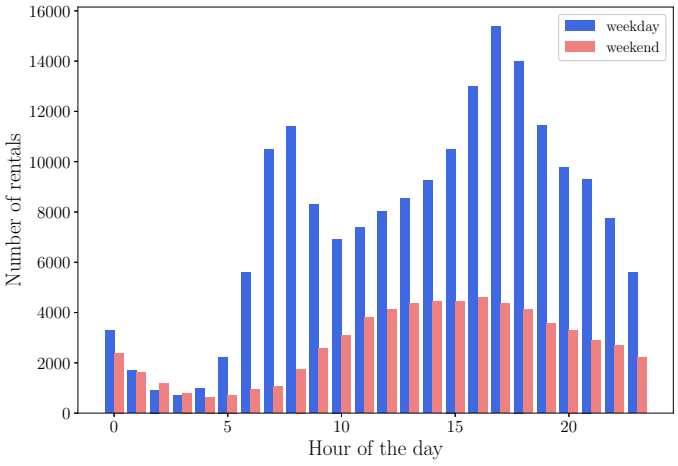

Figure 3.5. Booking counts for each of the 48 time slot for the city of Amsterdam. Year 2017.

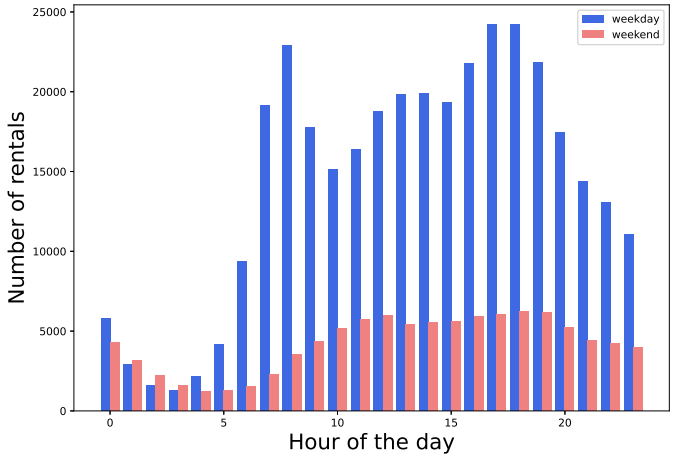

Figure 3.6. Booking counts for each of the 48 time slot for the city of Turin. Year 2017.

In figures 3.7, 3.16 respectively for Amsterdam and Turin are reported the optimal bandwidth values calculated for each KDE of the relative time slot, in the entire 2017 year. As is reasonable to expect, it's possible to see that there is a sort of inverse proportionality, between the size of the

bandwidth and the number of trips in each slot. The more trips are present in a time slot, the smaller will be the bandwidth and vice-versa.

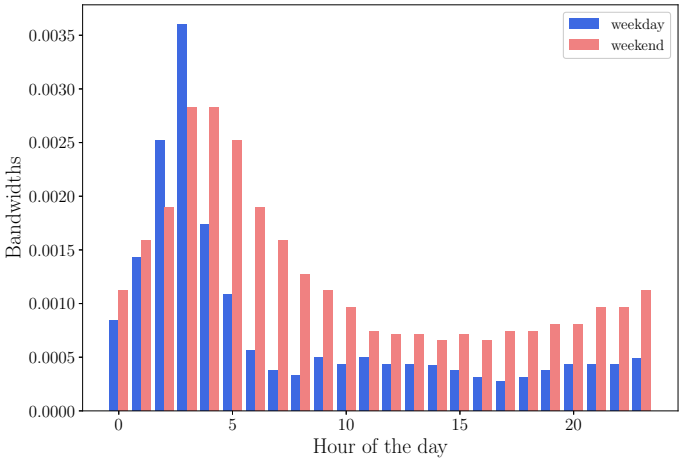

Figure 3.7. Amsterdam optimal bandwidth values for each time slot KDE. Values expressed in decimal degrees.

In figures 3.10, 3.11 is possible to see. for the city of Amsterdam, the representation of the KDE probability density function, fitted with the optimal bandwidth, on the sample data of time slot Weekend 23, of 2017.

The log-likelihood plots 3.8 3.9, clearly shows their peaks, where are situated the optimal bandwidth values. For time slots with fewer trips, in particular early in the morning at weekend, the shape of the curve is less explanatory, simply because of graphic scale and boundaries settings. Anyway it has been preferred to group all time slots together.

In the 3D representation is possible to notice how most of the high values of the density is concentrated in the city center, while this characteristic, is partially obfuscated in 2D-plot which involves a sample only in the centroid of each bin. This clearly introduces an important imprecision on this type of qualitative plots. This particular combination of continuous KDE sampled only in centroids of each bin for making a 2-D plot, returns a qualitative plot, that appears to be less useful even of those with discretized input and output.

This behaviour can be observed even better in figures 3.12 3.13 where, it has been used on purpose a bandwidth of 0.0001 DD that is much smaller

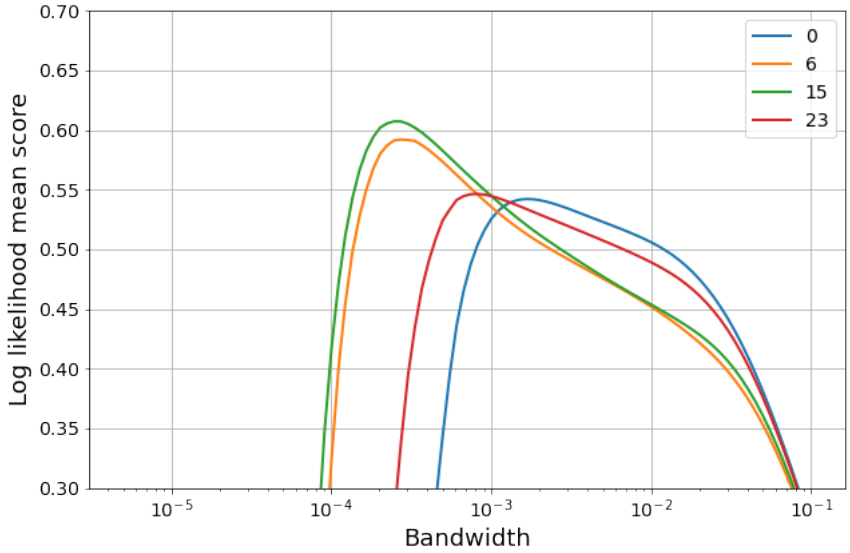

Figure 3.8. Mean of the log-likelihood score for the different folds. Cross validation with Euclidean distance metric, for the weekdays time slots. City of Amsterdam.

than the optimal one (0.0012 DD). This leads to a distribution that is similar to several deltas located in correspondence of some of the sample points. While in 3-D representation is possible to understand completely this habit, from figure 3.12 instead, it seems that the entire pdf is totally flat, which is not correct. Apart from representation optimization, this is called under smoothing in Kernel Density Estimation.

The opposite of under-smoothing is over-smoothing, which happens when the bandwidth is too large, related to the distribution of the data samples. This can be observed in figures 3.14, 3.15. This because we used for fitting the WE23 KDE a bandwidth of 0.1 DD that is approximately 11 kilometers, which is about more than the half of the side of the city area of Amsterdam under examination.

The same remarks are valid also for the city of Turin, illustrated in figures 3.19, 3.20, for which we have not reported the figures regarding under-smooth and over-smooth, that produce effects strictly comparable.

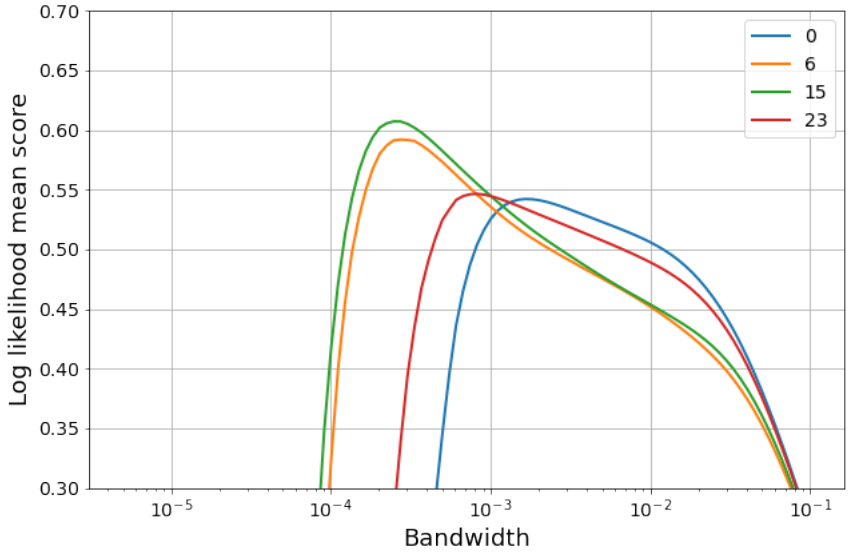

Figure 3.9. Mean of the log-likelihood score for the different folds. Cross validation with Euclidean distance metric, for the weekend time slots. City of Amsterdam.

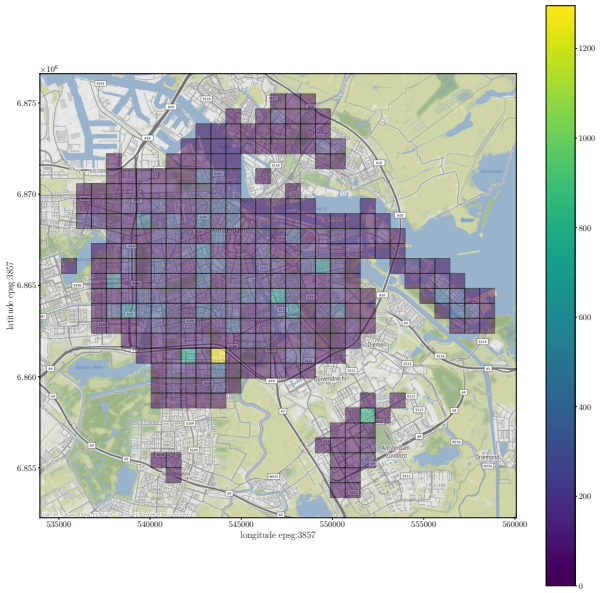

Figure 3.10. Amsterdam optimal bandwidth (0.00112 DD, WGS84) KDE sample in centroid of each bin, Time slot WD23

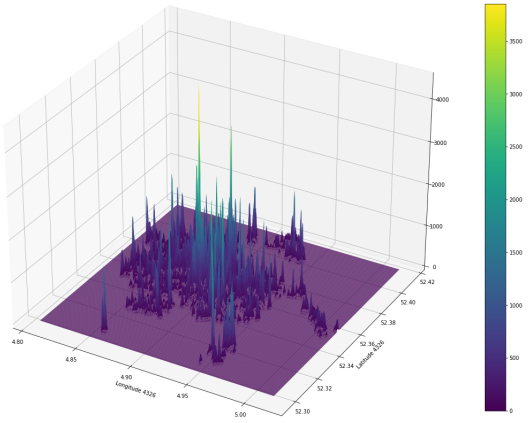

Figure 3.11. Amsterdam 3D surface plot optimal bandwidth (0.000491 DD) KDE sample. Euclidean Distance cross validation. Time slot WD23

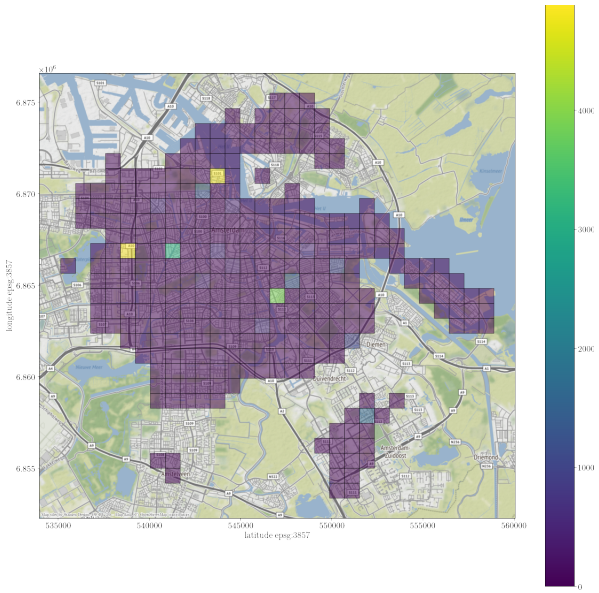

Figure 3.12. Amsterdam extra small bandwidth (0.0001 DD) KDE sample in centroid of each bin. Time slot WE23

Haversine Distance

In two dimensions, another distance metric that can be used in GridSearchCV is the Haversine distance, defined in section 2.6. This takes into account that

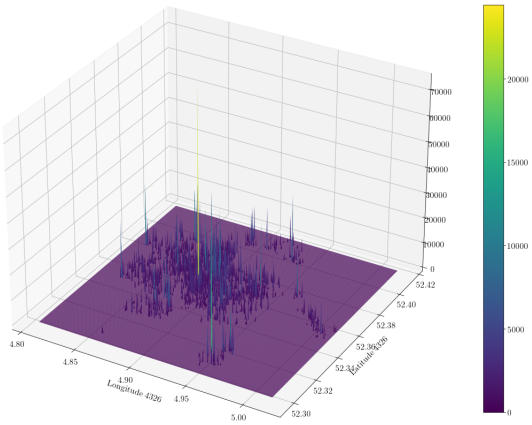

Figure 3.13. Amsterdam 3D surface plot extra small bandwidth (0.0001 DD) KDE sample. Time slot WD23

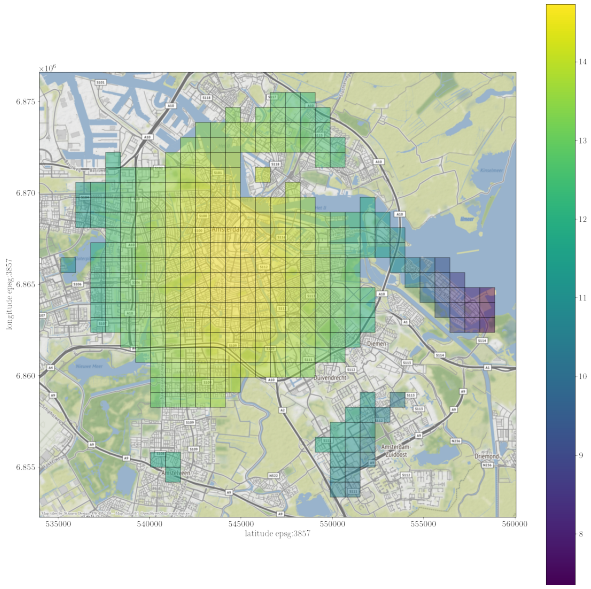

Figure 3.14. Amsterdam extra large bandwidth (0.1 DD) KDE sample in centroid of each bin. Time slot WD23

distances in longitude and in latitude are not the same, which means that for approximately in the city of Amsterdam 0.001 DD corresponds to 67 meters in longitude and 111 meters in latitude. In other words a difference of two longitudes expressed in decimal degrees, corresponds to a different length expressed in meters, depending on the latitude at which the difference is

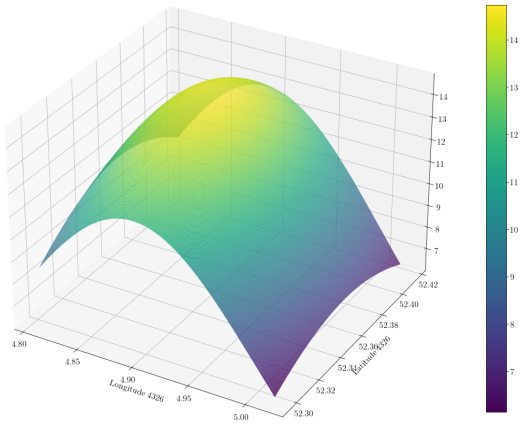

Figure 3.15. Amsterdam 3D surface plot extra large bandwidth (0.1 DD) KDE sample. Time slot WD23

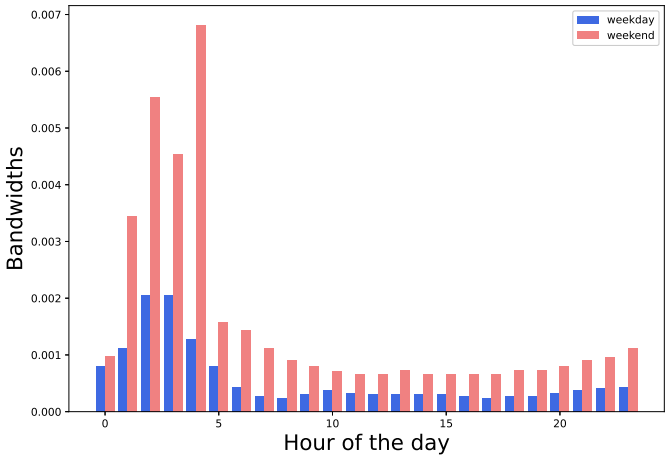

Figure 3.16. Turin optimal bandwidth values for each time slot KDE. Values expressed in decimal degrees.

calculated.

In conclusion, a degree of longitude is widest at the equator with a distance of 111.321 kilometers. The distance gradually shrinks to zero as they meet at the poles. At 40 degrees north or south, the distance between a degree of longitude is 85 kilometers.

In general bandwidths obtained with Haversine distance metric in cross

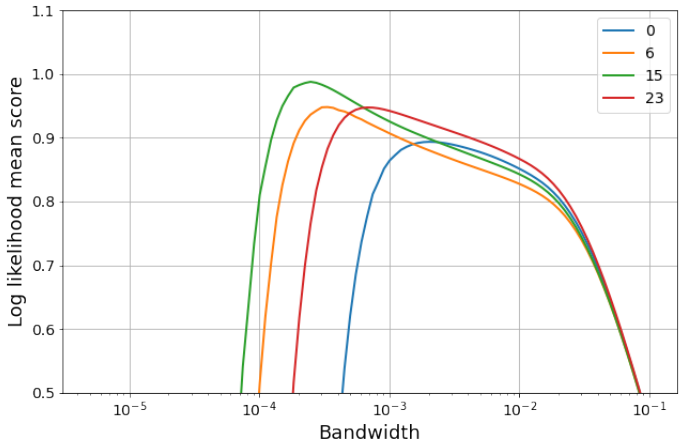

Figure 3.17. Mean of the log-likelihood score for the different folds. Cross validation with Euclidean distance metric, for the weekdays time slots. City of Turin.

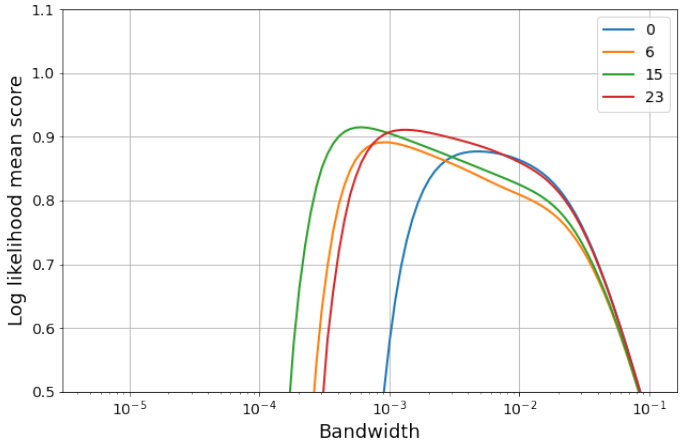

Figure 3.18. Mean of the log-likelihood score for the different folds. Cross validation with Euclidean distance metric, for the weekend time slots. City of Turin.

validation are slightly smaller (fig. 3.21, 3.22). This affects also the resulting KDE distribution (fig. 3.23, 3.24), which compared to the ones obtained with Euclidean distance metric, appears with the same general shape, but with more spikes and a a bit more scattered.

The log-likelihood plots have not been reported, but shows the same

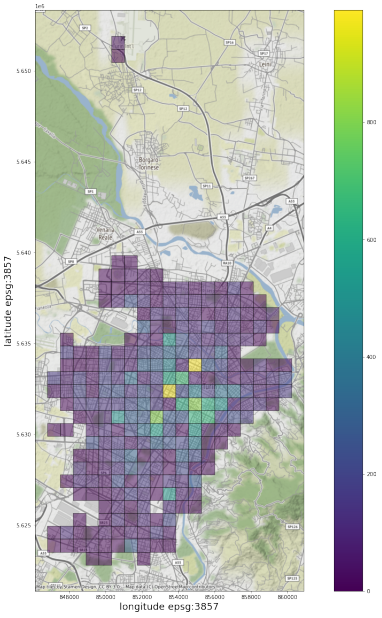

Figure 3.19. Torino optimal bandwidthple. Euclidean Distance cross validation. (0.0005 DD) KDE sample in centroids.Time slot WD23
Time slot WD23

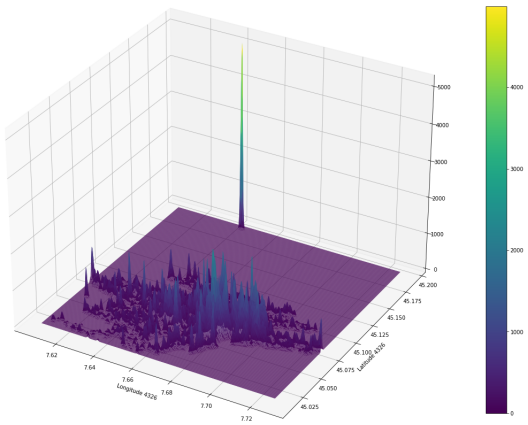

Figure 3.20. Torino 3D surface plot optimal bandwidth (0.0005 DD) KDE sample. Euclidean Distance cross validation. Time slot WD23

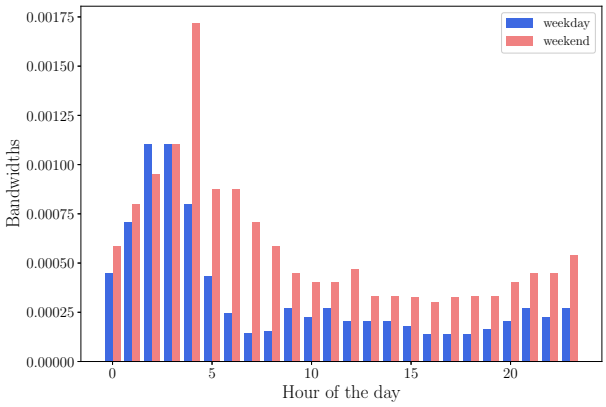

Figure 3.21. Amsterdam optimal bandwidth values for each time slot KDE. Cross Validation with Haversine distance. Values expressed in decimal degrees.

exactly trend, with the only difference on the maximum log-likelihood values, and the corresponding bandwidth, on which those values falls.

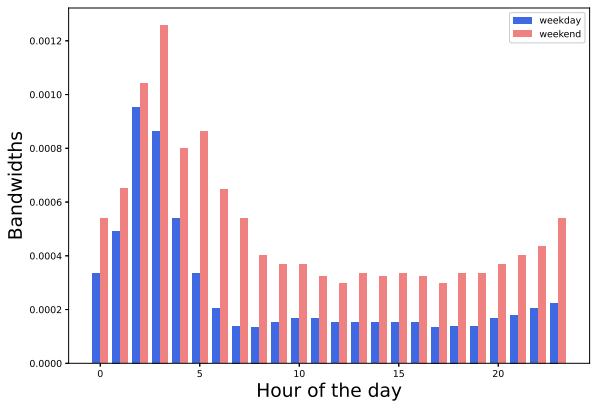

Figure 3.22. Turin optimal bandwidth values for each time slot KDE. Cross Validation with Haversine distance. Values expressed in decimal degrees.

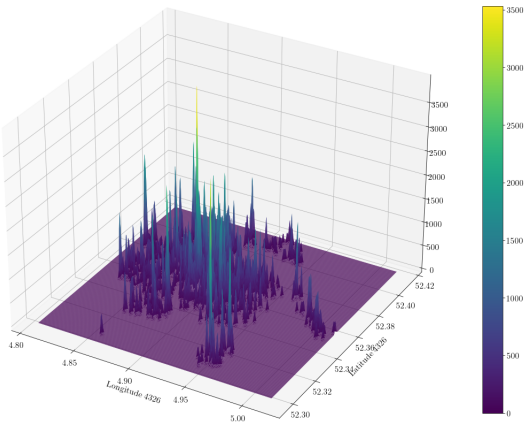

Figure 3.23. Amsterdam 3D surface plot optimal bandwidth (0.00053 DD) KDE sample. Haversine Distance cross validation. Time slot WD23

3.1.3 Continuous input data, UTM coordinate system

As it has been said before, the WSG84 coordinate system in which data have been collected, have some disadvantages for our purpose. The system is without any doubt more precise with respect to other ones, but for a few reason it has been decided to adopt the UTM coordinate system, which has been analyzed and compared in section 2.2.3, to continue the work.

Thanks to the UTM system is possible to:

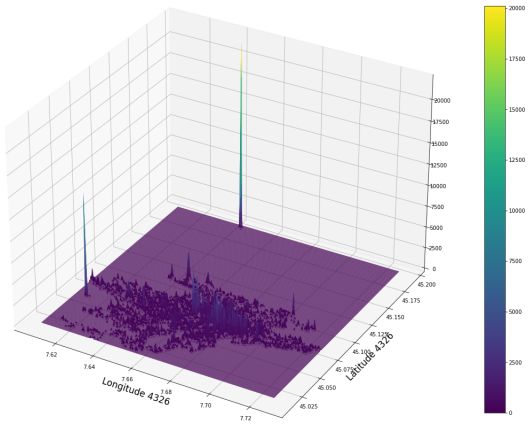

Figure 3.24. Torino 3D surface plot optimal bandwidth (0.00023 DD) KDE sample. Haversine Distance cross validation. Time slot WD23

- Have better qualitative 2-D plot. In fact, while with WGS84 system, if we desired to have a plot that displayed both the density and map, to have intelligible information, only a sample in each bin centroid was done, and as we have seen in the related figures, produced a poor representation precision. With UTM system instead, we can produce a sort of continue contour plot with the map above it. For being precise the definition of continue is not totally true, because it is an interpolation of a 500x500 point grid.

- The discrepancy described in section 3.1.2 about moving for a given amount of Decimal Degrees in Longitude rather then in Latitude is solved, because UTM use meters. So it is not necessary anymore to use Haversine distance for cross validation, which still could not be used in four dimensions KDE, and produces a non symmetrical bandwidth.

Cross validation algorithm, performed still with trips from all 2017, shows the same trend emerged with discretized input and output and WGS84 coordinate system (fig.3.25, 3.26). Looking at the city of Amsterdam, for time slot WD23 the optimal bandwidth found is 67meters with UTM coordinate system, 0.000796 DD with WGS84. Converting it to meters in one of the two dimension we have that is more or less 75 meters, therefore results appear to be coherent.

3D representation of the Kernel Density Estimate pdf are comparable to the ones with WGS84 coordinates. As mentioned above, with UTM system

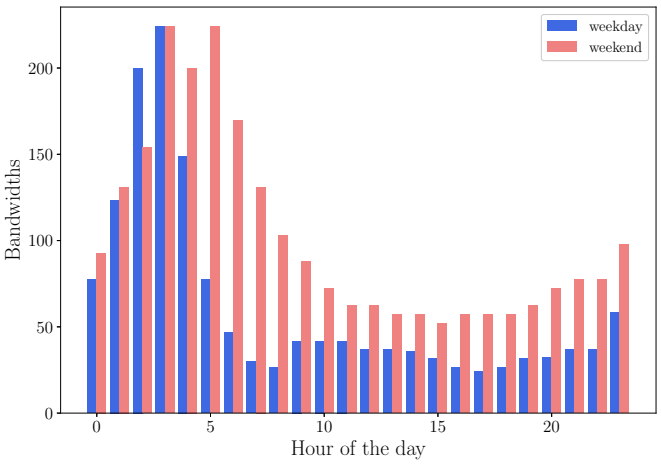

Figure 3.25. Amsterdam optimal bandwidth values for each time slot KDE fitted with UTM coordinates. Values expressed in meters.

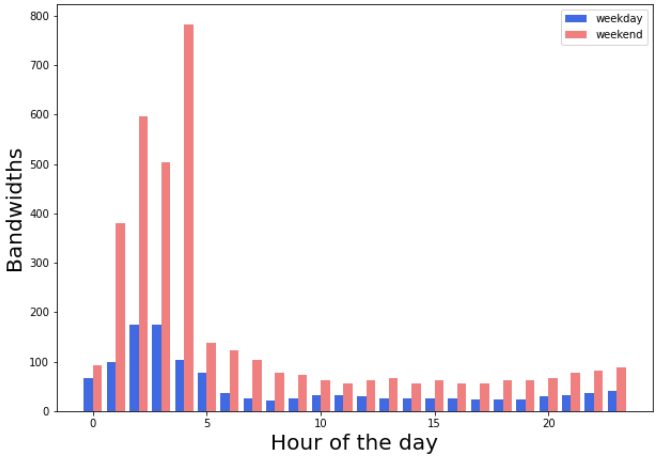

Figure 3.26. Torino optimal bandwidth values for each time slot KDE fitted with UTM coordinates. Values expressed in meters.

is possible to produce continue contour plot of the density, over the city map. These are shown in figures 3.29 3.30. Its clear that most of the request from both cities comes from the city centre. The Amsterdam case is particularly interesting because canals obviously affect the way people use the service.

For better readability we propose the plots also in log-scale. It is possible

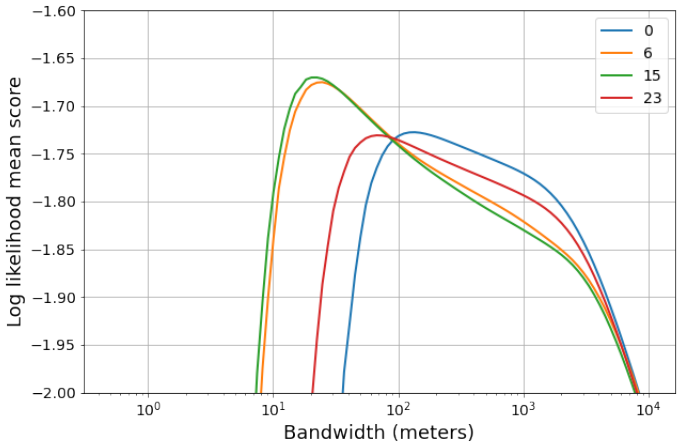

Figure 3.27. Mean of the log-likelihood score for the different folds. UTM coordinate system. Weekdays time slots. City of Amsterdam.

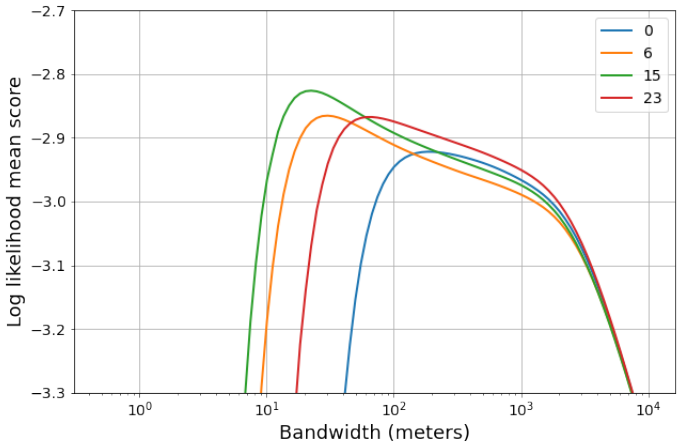

Figure 3.28. Mean of the log-likelihood score for the different folds. UTM coordinate system. Weekdays time slots. City of Turin.

to observe it in figures 3.31 3.32

Until now, we used for the optimal bandwidth research the entire dataset of 2017 respectively to each city. Anyway with ODySSEUS software is possible to simulate different scenarios, that may include only a subset of the dataset we considered so far. For example it can be useful to focus only on winter months to study some specific users' habits from the trips, or a given subset can better satisfy some requirements compared to another. A similar

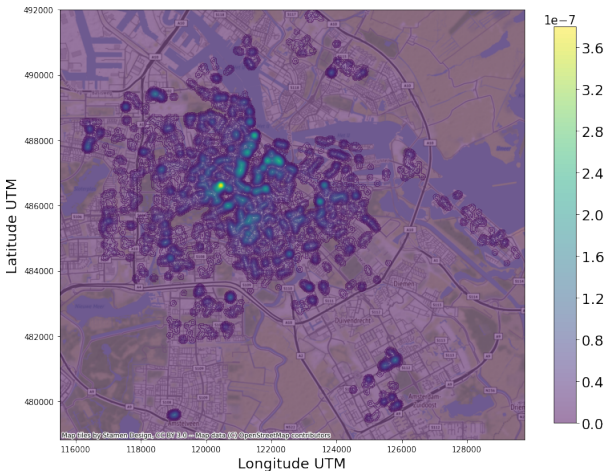

Figure 3.29. Amsterdam contour plot UTM coordinates. 12 Month dataset. Optimal BW = 67 meters. WD23 time slot.

situation is described in this previous work of M.Cocca Et Al. [26] where only two months from September to November 2017 were selected.

For this reason, some runs of the same optimal bandwidth research algorithm were performed with different possible subsets of the entire dataset. In particular, we started the iteration from only one month (January 2017) and added each iteration a month, until the 12th (December 2017).

Results can be summarized in figures 3.33, 3.34, 3.35, 3.36 Once again, is possible to notice that the smaller the cardinality of the subset for each time slot, the the greater will be the bandwidth and vice-versa. This trend is maintained in general trough all the different month subset, with some exceptions. This can be due to the different trips' starting point distribution in space. In fact, the more the points are distant each others, the greater will be the optimal bandwidth value that will be selected with the log likelihood approach by the cross validation algorithm.

3.2 Four dimensional KDE

As previously introduced in section 3.1, a 2D-KDE is not sufficient to characterize a mobility demand. This is because rental origins and destinations are correlated. Therefore, it is important to implement a four-dimensional KDE to correctly estimate an origin-destination (OD) matrix. We report the same experiment as before, where we find the optimal h for each hourly

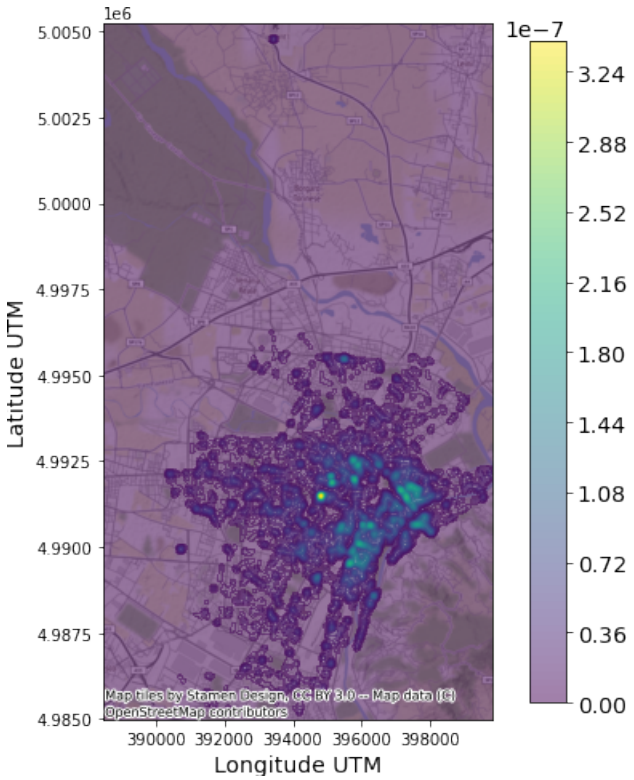

Figure 3.30. Torino contour plot UTM coordinates. 12 Month dataset. Optimal BW = 63 meters. WD23 time slot.

time slot, but here modelling the 4D matrix.

In figures 3.37, 3.38 we show the log-likelihood score for 4 hourly time slots in 4D, the x-axis is logarithmic to illustrate the difference in log-likelihood. As it is possible to observe how the optimal bandwidth values are much larger than in 2D, with optimal values between 200 m and 400 m. Values below 100 m result in very poor performance. Moreover, if we can notice that the likelihood scores are lower with respect to the 2D case. This reflects the strong dependencies between the origin and destination of the rentals that makes the data inherently sparser.

Focusing now on the optimal bandwidth trend of each subset in four dimensions. From figures 3.39, 3.40, 3.41, 3.41 we can observe that with a dataset of 1 month, the optimal bandwidths between 2D and 4D are comparable. By increasing the amount of data, the bandwidth remains larger for 4D in all cases. This is because the patterns in 2D are quite repetitive, while in 4D there is more variability, and KDE should keep the bandwidth

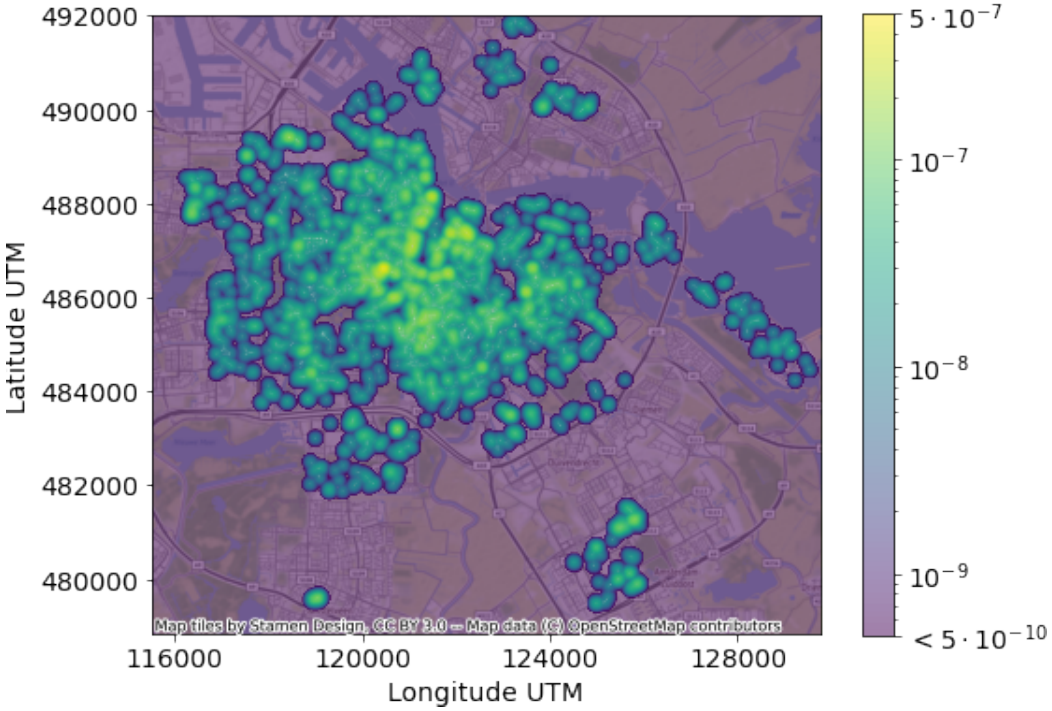

Figure 3.31. Amsterdam contour plot UTM coordinates. 12 Month dataset. Optimal BW = 67 meters. WD23 time slot. Logarithmic scale

larger to generalize the samples and avoid under-smoothing. Also, in 4D there is a big difference in the optimal bandwidth between 6 and 12 months because the model is still learning the 4D patterns.

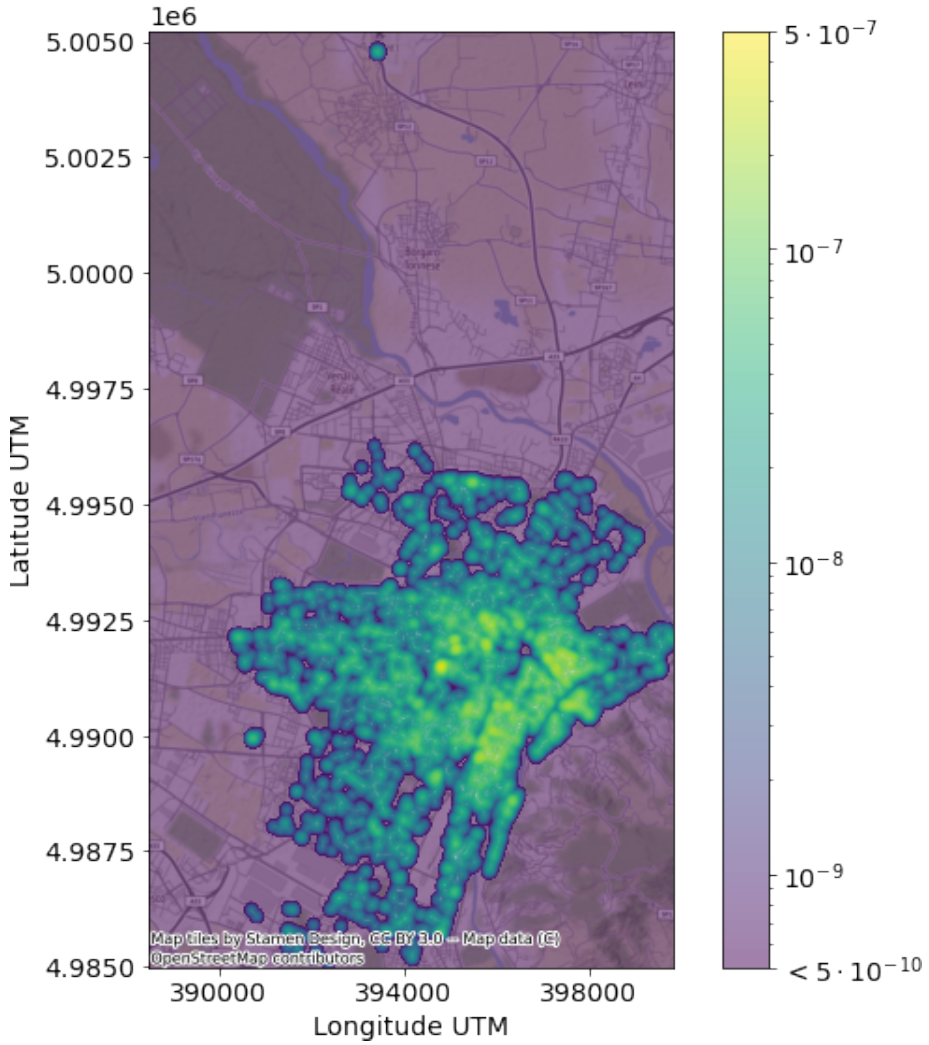

Figure 3.32. Torino contour plot UTM coordinates. 12 Month dataset. Optimal BW = 63 meters. WD23 time slot.

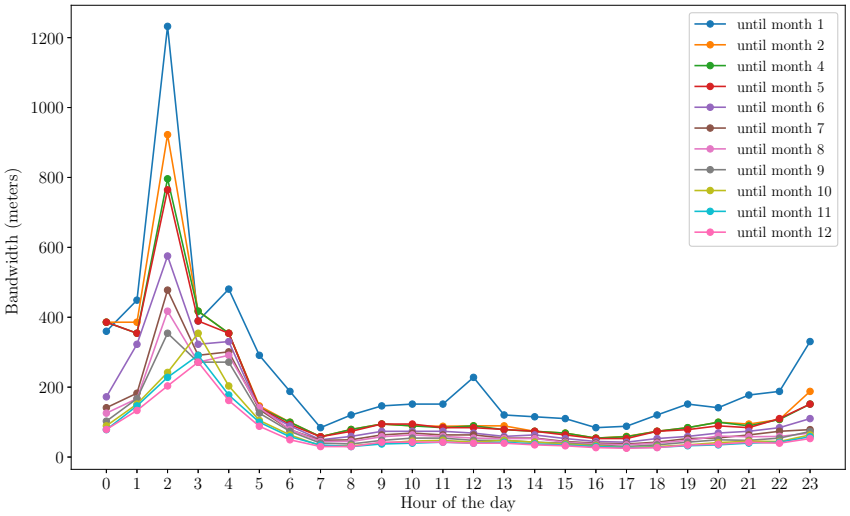

Figure 3.33. Amsterdam UTM Weekdays optimal bandwidth trends. Every line corresponds to a subset.

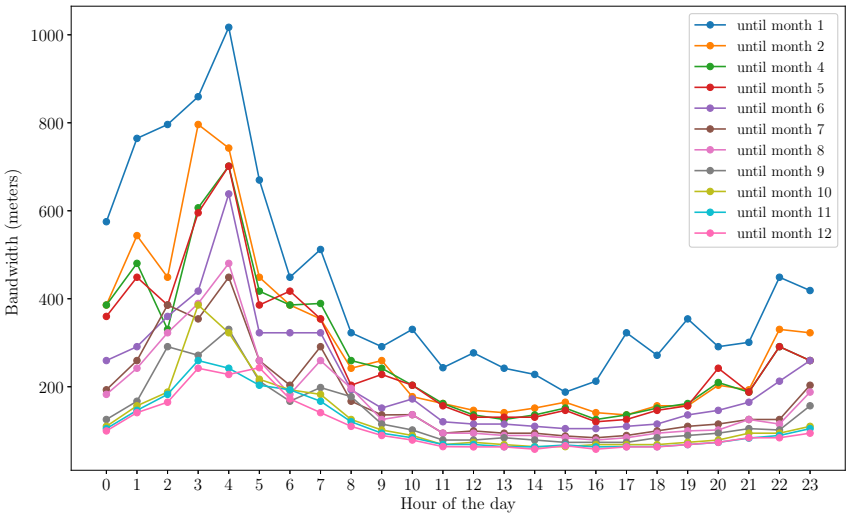

Figure 3.34. Amsterdam UTM Weekends optimal bandwidth trends. Every line corresponds to a subset.

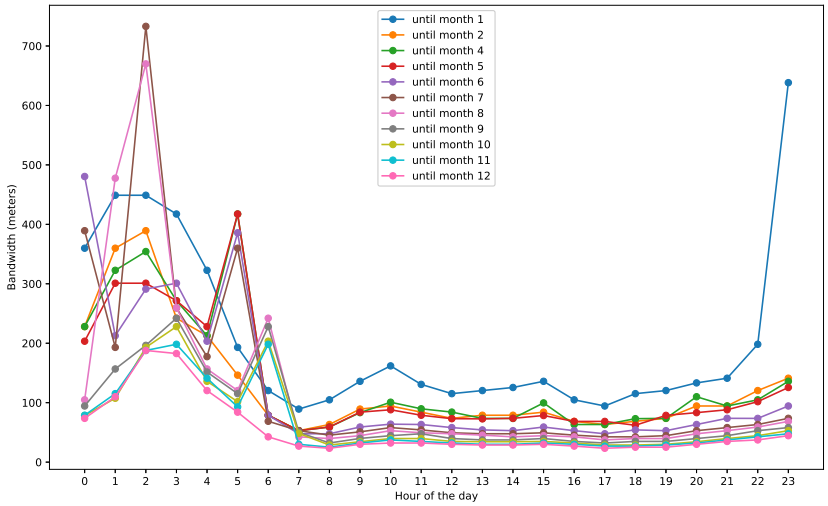

Figure 3.35. Torino UTM Weekdays optimal bandwidth trends. Every line corresponds to a subset.

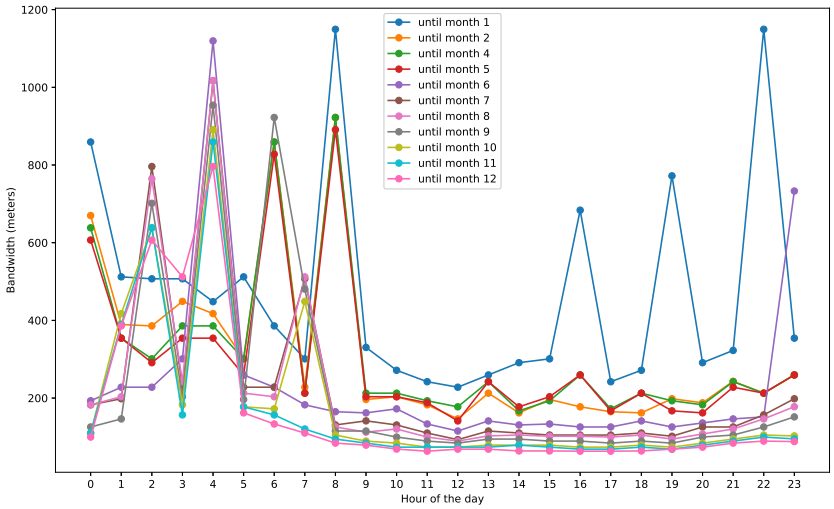

Figure 3.36. Torino UTM Weekends optimal bandwidth trends. Every line corresponds to a subset.

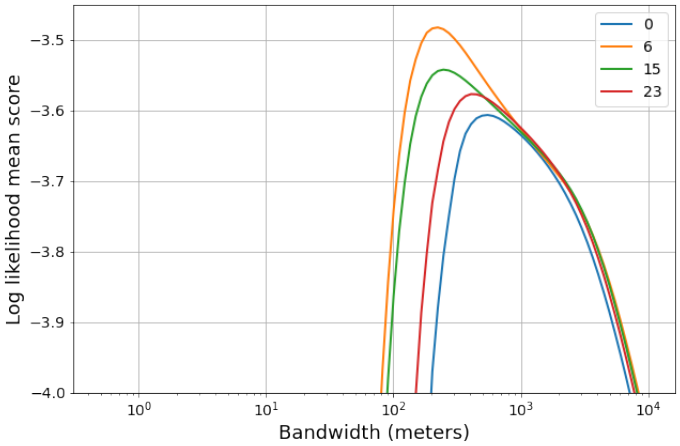

Figure 3.37. Mean of the log-likelihood score for the 4D-KDE different folds. UTM coordinate system. Weekdays time slots. City of Amsterdam.

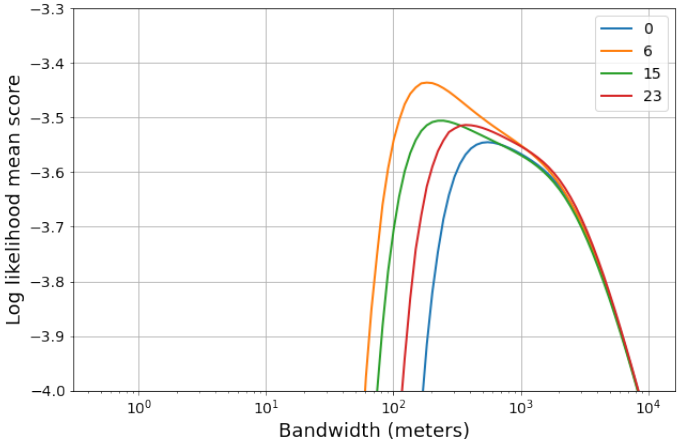

Figure 3.38. Mean of the log-likelihood score for the 4D-KDE different folds. UTM coordinate system. Weekdays time slots. City of Turin.

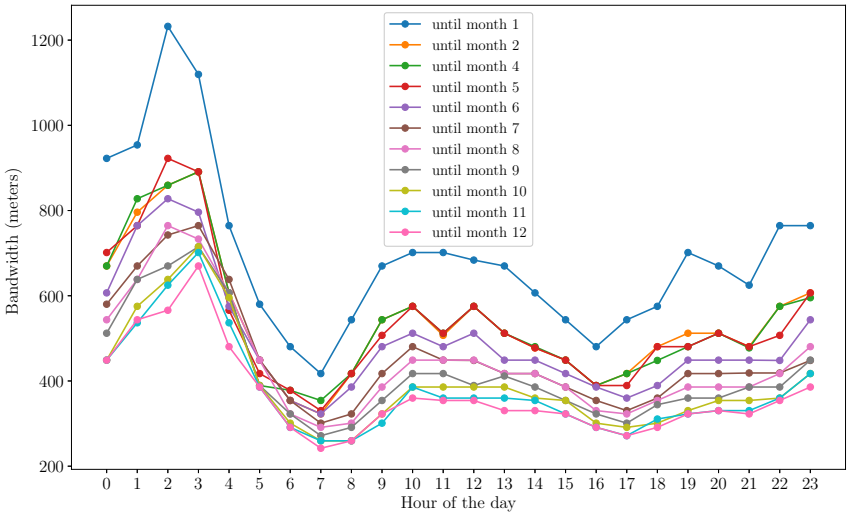

Figure 3.39. Amsterdam UTM Weekdays 4D-KDE optimal bandwidth trends. Every line corresponds to a subset.

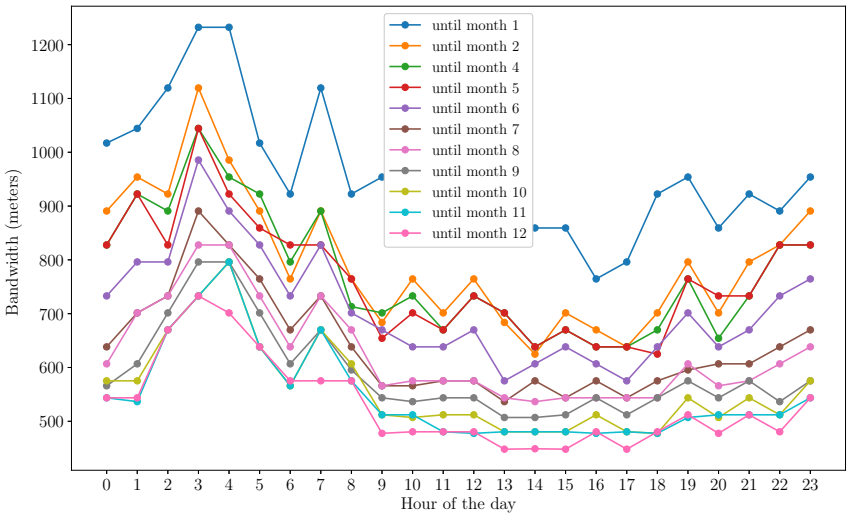

Figure 3.40. Amsterdam UTM Weekdays 4D-KDE optimal bandwidth trends. Every line corresponds to a subset.

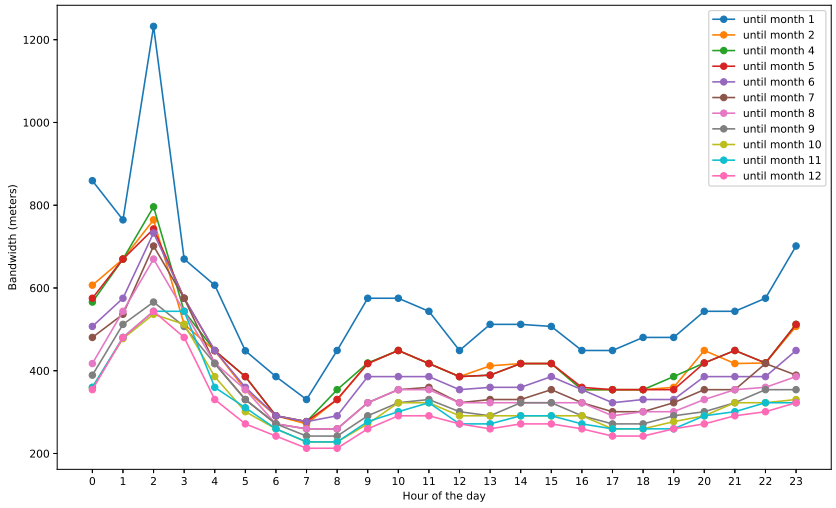

Figure 3.41. Torino UTM Weekdays 4D-KDE optimal bandwidth trends. Every line corresponds to a subset.

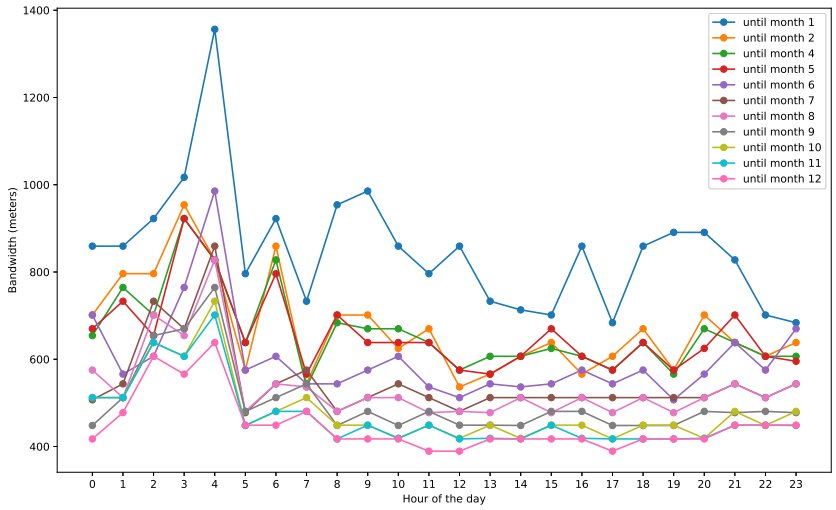

Figure 3.42. Torino UTM Weekdays 4D-KDE optimal bandwidth trends. Every line corresponds to a subset.

Chapter 4

Variable Bandwidth KDE optimization

In section 2.1.2 we introduced variable bandwidth KDE models, and we made an important distinction between two main types of estimators: Ballon and Simple Point estimators. As previously explained, the first one suffers of an important deficiency: it usually fails to integrate to one. For this reason, but also because in the open source community is available a python library that implements a Simple Point estimator, we have chosen to adopt this last one. The library just mentioned has been briefly introduced in paragraph 2.3.1, while the mathematical background on which is built, is described in 2.1.2.

Now we are going at first to explore how this library work with our data, and how it models the probability density functions in different scenarios, and then, how it is possible to optimize the two principal hyper-parameters that characterize the pdf. For doing this we continue to use data in UTM coordinates, therefore every contourplot in this chapter has been done with this coordinate system. Moreover as already done in the previous chapter, an important part of the dissertation is firstly presented in two dimensions, to ensure a better readability and awareness of the problem.

Looking at AwKDE documentation, the core function is GaussianKDE(). Two essential hyper-parameters have to be provided to this: ***glob_bw*** defines the global bandwidth used to compute the first estimation, that is then cached to compute the adaptive KDE. ***alpha*** instead defines the modulation parameter as already explained. Together they characterize the equation 2.5.

With *alpha* set to None, GaussianKDE behaves as a Fixed Bandwidth

KDE. For this reason, but also because there is almost no support nor community that operates with this library, we first verify the reliability of the library, ensuring that different pdf, obtained at first with Scikit-Learn KernelDensity() function, and then with AwKDE GaussianKDE() one in fixed bandwidth mode, actually matches or are comparable.

The comparison is not immediately straightaway because GaussianKDE(), when the fit method is invoked to fit the data, it receives as input the matrix of samples composed by n samples and m features (in 2 dimensions m=2 while in 4 dimensions m=4), and applies a transformation of the samples, returning a sample matrix with zero mean and identity co-variance matrix. This transformation is applied also when we want to evaluate KDE value in a particular given point, or in a set of different given points. While instead, for extract randomly a sample point from the KDE, which is in other words sampling the fitted pdf, a back transformation to the original space is performed.

According to this, the bandwidth too has to be scaled. So we proceed to divide the optimal bandwidth obtained with cross validation approach with Scikit-Learn (*OpBwSL*), for the average of the standard deviation of the samples in each dimension. This approximately gives the optimal bandwidth for each time slot, for standardized samples. Then, to select the optimal AwKDE bandwidth, a cross validation process is executed, using a range of 50 equally spaced points in the neighbourhood of the just mentioned *OpBwSL*, with bounds

$$[OpBwSL/2\cdot, OpBwSL \cdot 1{,}5]$$

Another possible solution could be to use an array of logarithmically spaced points from the equivalent of 0.5 meters to the equivalent of 10000 meters (i.e. from 0.0002 to 4.151) and then refine the operation using 50 spaced points around the neighbourhood of the first found optimal bandwidth. Anyhow, the result obtained fitting the KDE with the optimal bandwidth with the two different libraries are almost the same, as it is possible to observe comparing figure 4.1 and figure 3.29.

In [19] [18] Abramson highlighted that alpha equal to 0.5 can perform well in simulation studies both in one and two dimensions, other like Terrell and Scott [52] showed that the adaptive square-root law gives up to 50 % of gain up in terms of Mean Integrated Squared Error compared to the fixed bandwidth approach.

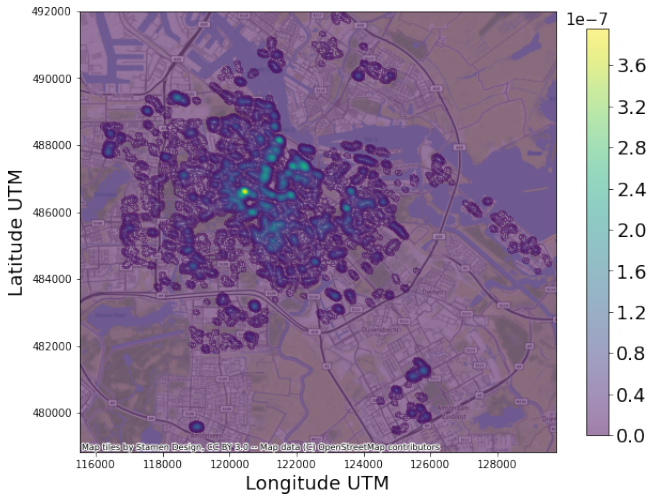

Figure 4.1. Amsterdam AwKDE contour plot. 12 Month dataset. WD23 time slot. Optimal glob_bw = 0.0333, alpha=None

MISE can be very handful to compare performances with synthetic data, especially when there is a continuous probability density function available as reference. The problem we are facing in this case study is quite different, because we are trying to build a continue probability density function from all the collected data point, there can exist an original pdf to use as a comparison. For this reason we did not use this metric.

We then start with the choice of *alpha*=0.5 to observe the benefits if any, of using a V-KDE. As before we have done for fixed-KDE, we optimize the global bandwidth. In order to better understand how AwKDE works, we take into consideration the same time slot used in the previous example: weekday at 11 p.m. We also show the first result in 2 dimension, in order to have also a qualitative comparison with the previous results. First we understand how local bandwidth enlarges in low sample density areas and restricts in high density ones.

In figures 4.2 4.3 we see in percentage how much the bandwidth is enlarged or restricted. A red color means the local values of the bandwidth has increased. A blue color instead reflects that the Variable-KDE has restricted the local bandwidth. For what regards the city of Amsterdam, looking at the results, the Variable-KDE automatically restricts the bandwidth up to 60% in those areas where there is a large number of samples, e.g., in the city centre, a high-traffic area where the majority of the rentals occurs. In suburban neighborhoods, where less rentals occur, the bandwidth increases

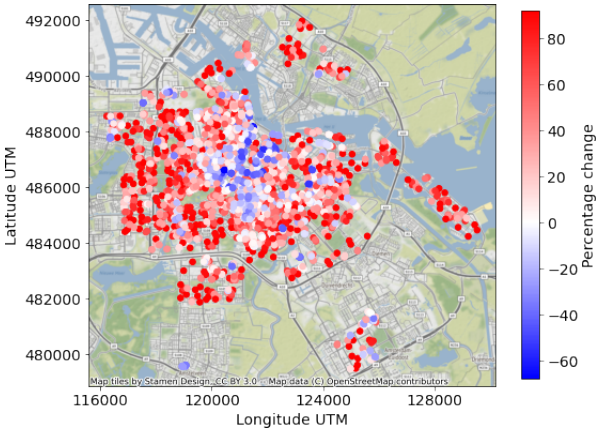

Figure 4.2. Amsterdam AwKDE scatter plot local bandwidth percentage incrementdecrement with respect to glob_bw. glob_bw=0.11574, alpha=0.5

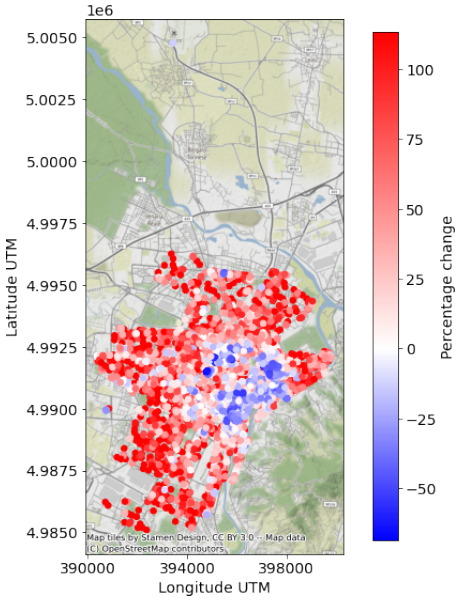

Figure 4.3. Torino AwKDE scatter plot local bandwidth percentage incrementdecrement with respect to glob_bw. glob_bw=0.02385 (equal to 76.18496 meters), alpha=0.5

up to 80% to automatically adapt to the lack of samples in the data. Same considerations can be made for Turin.

We can observe the impact in the estimated density of trips in the obtained Variable KDE reported in Figures4.44.5. Qualitatively, observe the

Figure 4.4. Amsterdam AwKDE contour plot. Alpha=0.5 Optimized glob_bw = 0.02321 (equal to 55 meters)

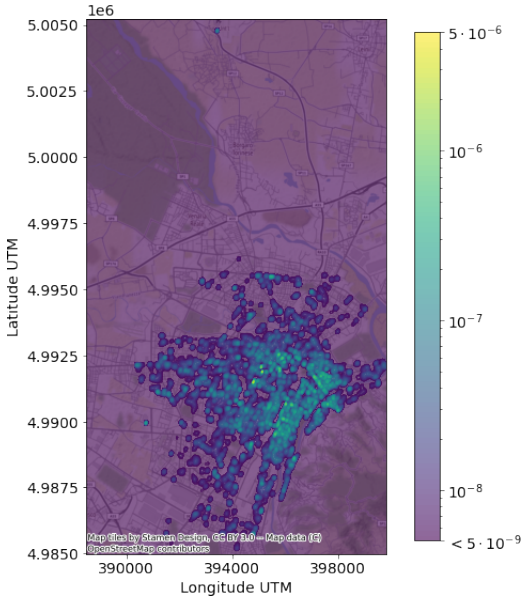

Figure 4.5. Torino AwKDE contour plot. Alpha=0.5 Optimized glob_bw = 0.0238 (equal to 5 meters)

more concentrated and higher presence of peaks (yellow dots) in city center. Instead, in areas with low density, the Gaussian distributions are less peaked (violet areas) and more smoothed to generalize the fewer samples at disposal. If we compare the log-likelihood scores obtained with standard KDE, we observe a slight improvement with V-KDE, with the log-likelihood

score that improves from -1.73 to -1.71 in the 2D case at 11 pm for the city of Amsterdam. It also possible to compare the figures mmentioned above with the contour plots obtained with fixed Bandwidth KDE in 3.31 for Amsterdam and 3.32.

4.1 AwKDE hyper-parameters tuning with Cross-Validation approach

Until now, we have seen how adaptive bandwidth behaves. For doing this, we adopted Silverman's law, adopting *alpha=0.5* and finding the optimal global bandwidth, with a cross validation approach, relying on log-likelihood metric.

Going further, we want to investigate how GaussianKDE() behaves if we try to optimize both hyper-parameters *glob_bw* and *alpha* simultaneously, tuning them together with GridSearchCV, which generates a grid with every possible combination of hyper-parameters values. This cross-validation approach is made possible even in AwKDE, because GaussianKDE() implements Scikit-Learn BaseEstimator class that is the one that is used by different other well-known libraries such as scikit-learn itself for the cross-validation.

For generating the the hyper-parameter grid we provided to GridSearchCV function, the following list of hyper-parameters:

- as *alphas* an array of equally spaced values from 0.1 to 0.9, plus the value "None", which means using KDE in non-adaptive mode . *alpha*=1 has not been inserted in the possible values because, as discoloured in the previous chapter, it is equivalent to a Balloon estimator approach type, which suffers of the previously exposed defects.

- As *glob_bw* instead we used a logarithmically spaced array of points from the equivalent of 0.5 meters to 10000 meters for showing as the log-likelihood values drop if the global bandwidth is too small or too large. As alternative is possible indeed to select the global bandwidth in the neighborhood of the of the previously founded optimal one.

Results in shows that for different time slots, the *glob_bw* in general is almost always halved and for which regards *alpha* values, in almost in every time-slot and every subset the optimization process chooses values in

the range [0.6, 0.9]. Again as an example of this, we show in figures 4.6 4.7 the variable KDE density obtained. Both again are in WD23 timeslot and for both *alpha=0.9* has been selected. For the city of Amsterdam the log-likelihood improves again to -1.69. However, comparing with $\alpha = 0.5$ in Figures 4.4 4.5, it is now hard to observe any significant benefits. In fact, we notice that, even though the fitting improved, we suffer from a possible over-fitting problem: in the $\alpha = 0.9$ case, peaks in the most central zones are even more evident, almost reflecting the presence of the charging stations that are present in for example in Amsterdam. Here, cars are returned with very high frequency, since those parking slots are reserved for car sharing customers. This causes a sizeable amount of the samples in the original trace to be concentrated almost exactly in the same location. While this is a property of the dataset, it does not reflect the actual mobility demand of users, but rather the artificial opportunity to park a car in such reserved parking positions. In these cases, one might not want to learn these patterns that are very specific and artificially introduced. Notice that the same problem would occur if the data collection is affected by a discretization of the origin/destination coordinates, which would cause several points in the data to appear in the exact same position. Therefore, we warn about the limitations of optimizing α in these practical cases, and we recommend following Silverman's law and choosing $\alpha = 0.5$ for simplicity. Similar considerations apply in 4D for OD matrices V-KDE.

4.2 Addition of random Gaussian noise to the samples

In order to investigate better what we exposed in the previous section, in particular the fact that tuning both *glob_bw* and *alpha* hyperparameters leds to *alpha* almost ever close to the maximun (i.e. 0.9), we decided to operate adding a random Gaussian noise to the samples. This to observe if such high values of *alpha* are still selected and if peaks in the most central zones are so evident as before, suggesting a possible overfitting.

In particular, for generating the noise we used a normal distribution with *mean* $\mu = 0$ and *standard deviation* equal to

$$\sigma = \frac{max - min}{\tau}$$

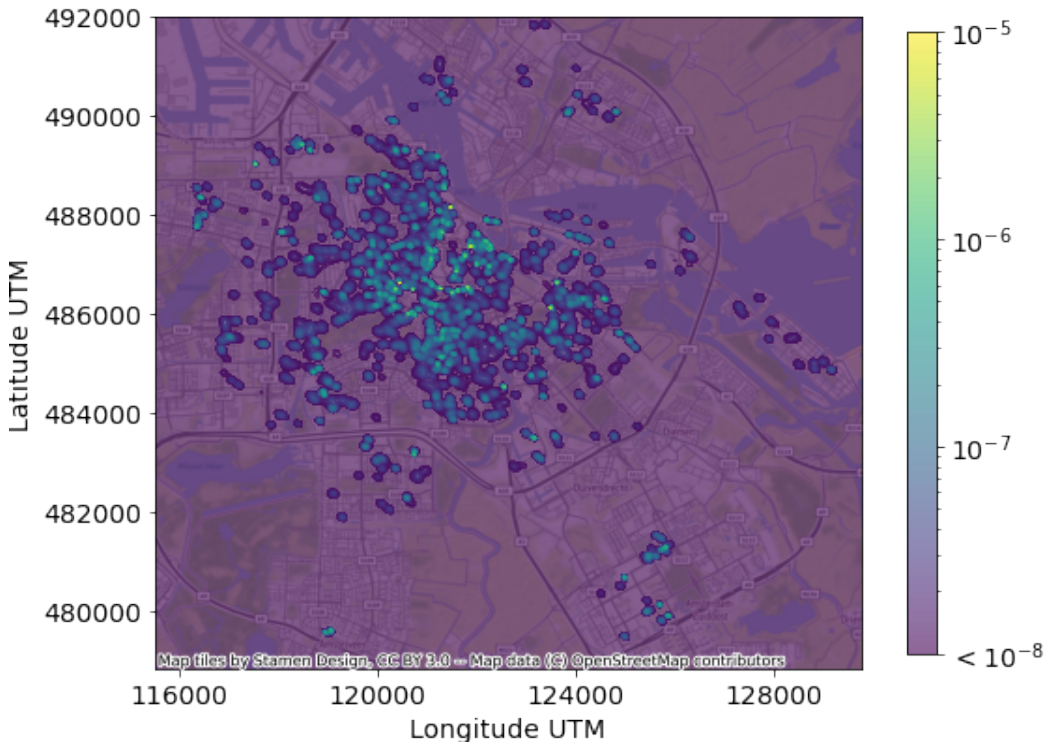

Figure 4.6. Amsterdam AwKDE contour plot. Optimized Alpha=0.9 and glob_bw = 0.0186 (equal to 44 meters). Timeslot WD23

where *max* and *min* are in order the maximum longitude or latitude value, and the minimum longitude or latitude value, while τ is a dividing factor. We tried with different values, and even a $\tau = 200$ is enough to avoid the cross-validation over-fitting problem just described. This means that with this given τ, assuming the case of Turin, the longitude range is about 10 kilometers, so about 95% of the samples are moved less then 100 meters, and about 68% of this less then 50 meters. In longitude instead, because the city airport (which is out of town) is served too, the range is approximately 20km, and therefore, about about 95% of the samples are moved less then 200 meters, and about 68% of this less then 100 meters. Therefore this does not effect the representation of how users exploit the service and their habits (100 meters corresponds barely to few buildings in a street) but allows to overcome some log-likelihood metric limits. For visualization purposes we can select a subset of 6 month in a time-slot that have enough samples to understand how they are going to be moved, but not so many to make things

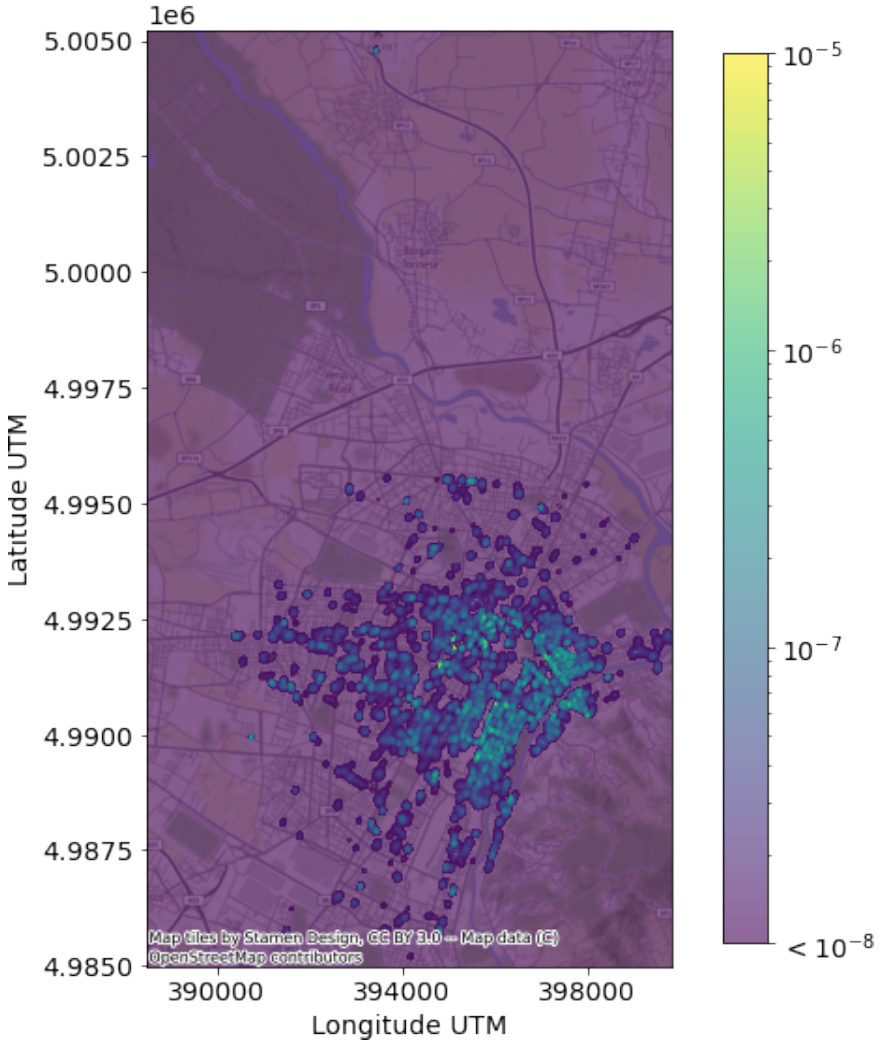

Figure 4.7. Torino AwKDE contour plot. Optimized Alpha=0.9 and glob_bw = 0.02006 (equal to 42 meters). Timeslot WD23

unreadable in a scatter plot. Therefore in figure 4.8 is possible to observe how the noised samples are distributed with respect to the original ones. Caselle Airport is not included because there no trips from there at 2 a.m. in the weekdays.

What we observe in general is that, starting from the situation without noise just described above, where the *glob_bw* is almost always halved and *alpha* is selected to be always high, introducing the Gaussian noise, for different values of tau, the smaller is τ, ie the bigger is the standard deviation

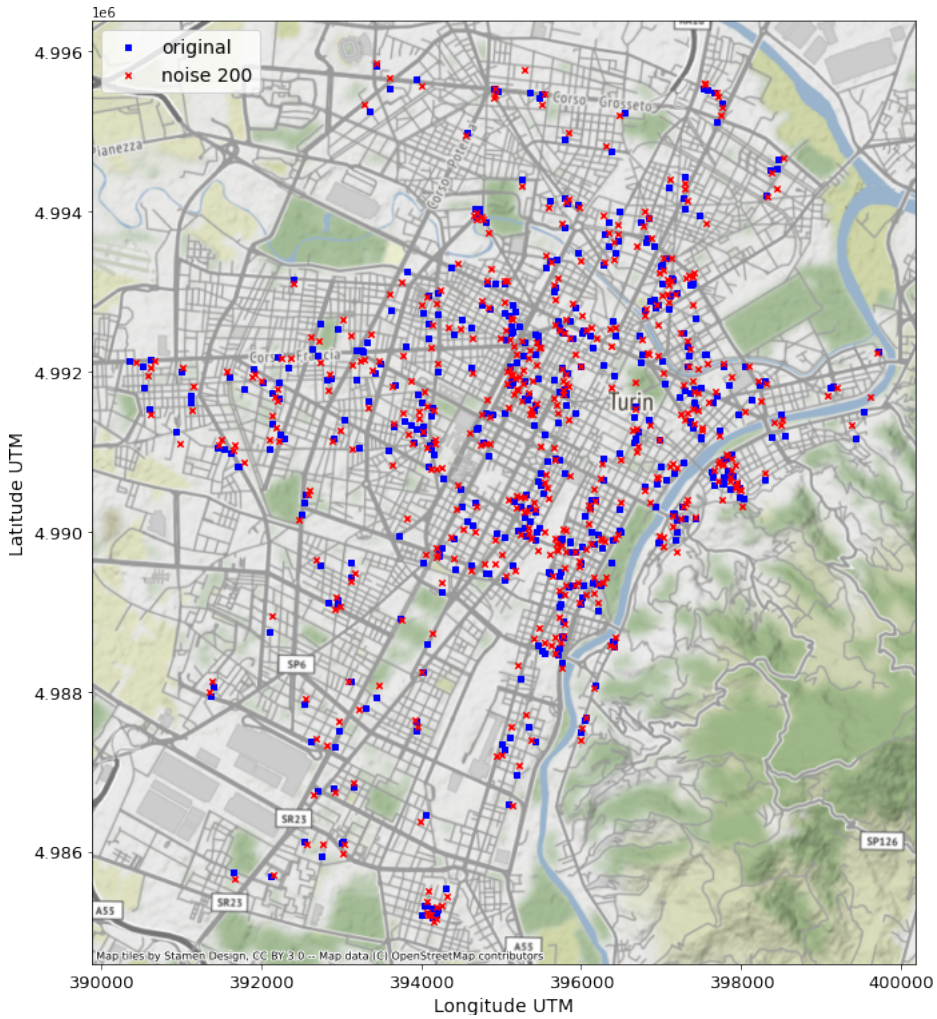

Figure 4.8. Turin scatter plot comparing original samples and samples with Gaussian noise $\tau = 200$. 6 month dataset, WD02 time slot.

of the introduced noise, the bigger is the optimal *glob_bw* selected, and the smaller is the *alpha*. With a τ=200, as mentioned above, values of *alpha* around 0.5 are preferred, which is the value that we found in the Abramson Law.

Regarding instead the *glob_bw*, the smaller is τ, the bigger is the global bandwidth. With a τ=200 as in the example is selected a *glob_bw* near equal to the optimal one obtained without *alpha* in the hyper-parameter Grid-Search. But the most important thing we can observe from figures 4.8 **??** is that regardless of the *alpha* value, for very large or very small values

of the global bandwidth, there is a significant drop of the log-likelihood. On the other hand, especially from figure **??** we see that is possible to adopt a *glob_bw* value that is just bigger than the optimal one, without any significant appearing worsening in performances. What is important is to prevent to adopt values too low, that can cause an asymptotically drop of the log-likelihood.

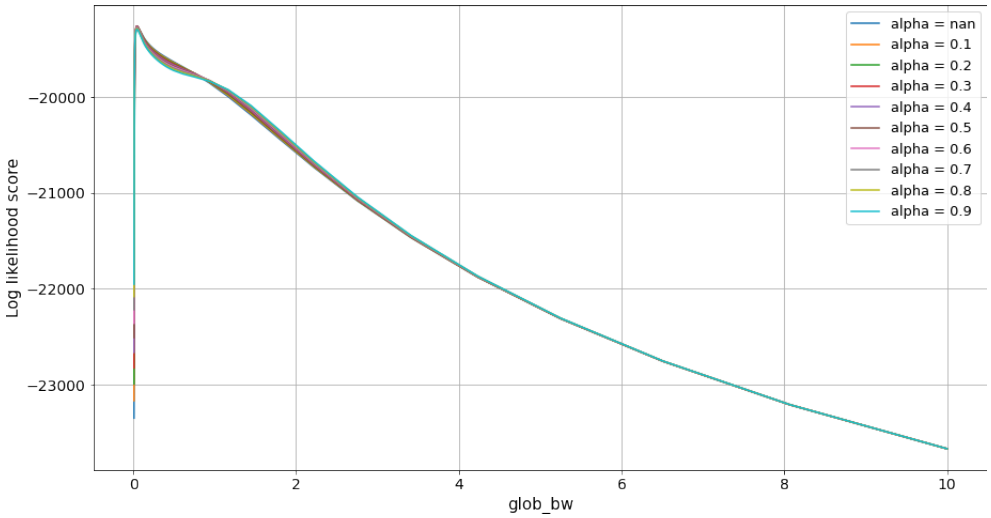

Figure 4.9. Log-likelihood plot related to the *glob_bw* for each value of *alpha*. City of Torino, 6 month dataset, WD07 timeslot. Gaussian noise $\tau = 200$

4.2.1 Four dimensions Variable KDE

The same observations made in the previous section about two dimensional Variable KDE, can be repeated for the four dimension case. Comparing the same sub-sample of the dataset (6 months starting from January to June) and the same time slot Weekday at 7 a.m., with a $\tau = 200$ an *alpha=0.5* is selected. This is not ever the case, in fact changing time-slot for example the *alpha=0.5* selected as best in the entire grid, oscillates around 0.3, 0.4, 0.5. This encourage as selecting as *alpha* 0.5 in according to Abramson's law. Furthermore also the observation made comparing fixed bandwidth 2-D KDE and 4-D one are valid, in fact we can compare the *glob_bw* in 4-D VKDE (0.14294) and the one in 2-D VKDE (0.049336) and see that increased, as figured out in the Fixed bandwidth scenario. This because in

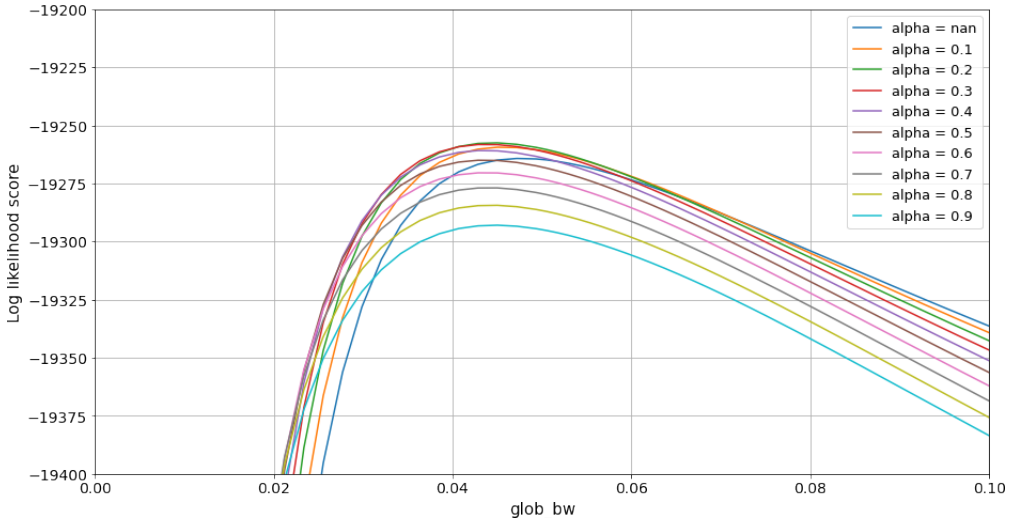

Figure 4.10. Log-likelihood plot related to the *glob_bw* for each value of *alpha*. City of Torino, 6 month dataset, WD07 timeslot. Gaussian noise $\tau = 200$

four dimension the model has less confidence in modeling some habits in the same way.
Finally in figures 4.11 4.12 we can observe basically the same trend showed in two dimensions, with a peak of the log likelihood that indicates the best way to fit the data. Also here is important to prevent global bandwidth values too small, in order to avoid a significant drop of the log-likelihood.

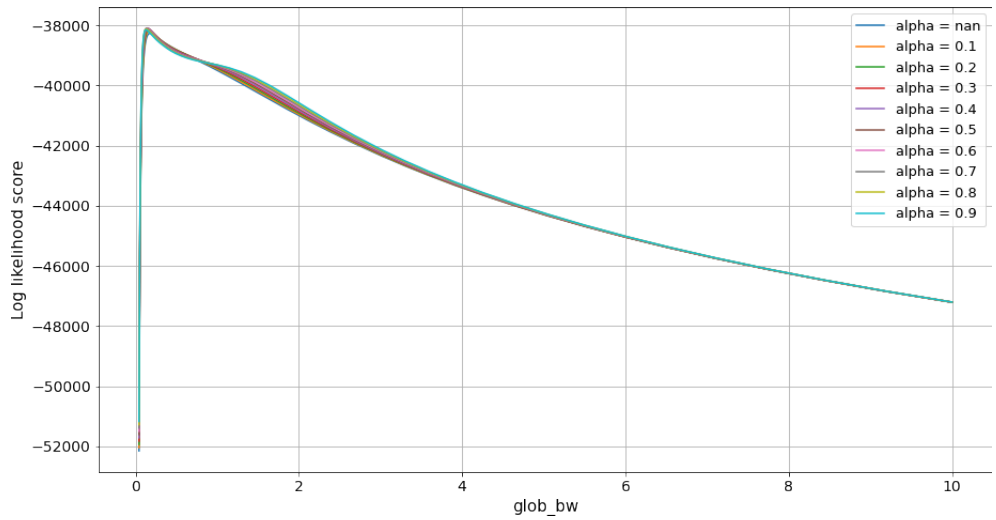

Figure 4.11. Log-likelihood 4D-KDE plot related to the *glob_bw* for each value of *alpha*. City of Torino, 6 month dataset, WD07 timeslot. Gaussian noise $\tau = 200$

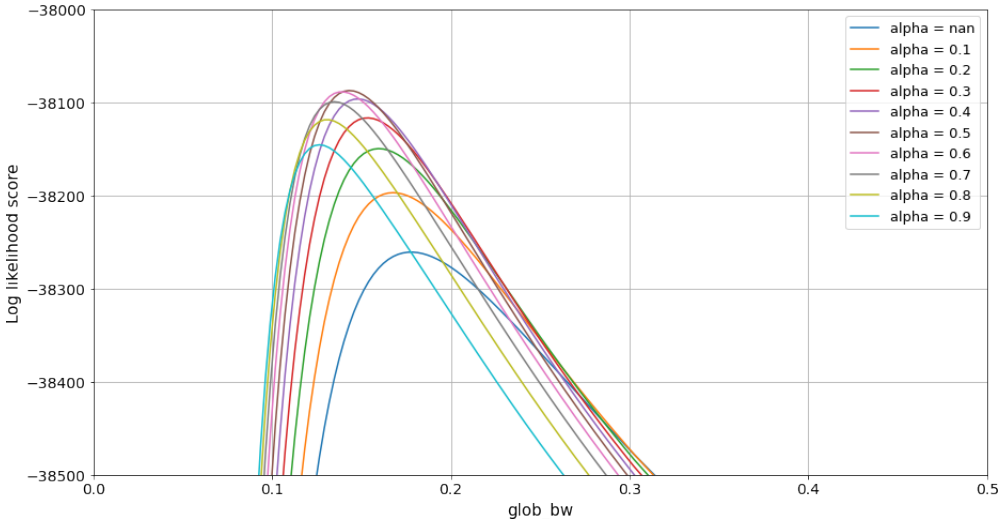

Figure 4.12. Log-likelihood 4D-KDE plot related to the *glob_bw* for each value of *alpha*. City of Torino, 6 month dataset, WD07 timeslot. Gaussian noise $\tau = 200$

Chapter 5

Conclusions

In this thesis we built a demand model that is capable do describe properly the internal combustion car sharing services' user habits. The tool can be integrated with any mobility simulators as ODySSEUS, an Origin Destination Simulator of Electric Shared Mobility in Urban Scenarios written in Python. The importance of having a model that can well fit and generalize the user habits, that interoperate with a data-driven simulator is given by the fact that is possible to study different what if scenarios, and see how a full electric car sharing system can work in different cases. Free floating electric sharing systems represent a valid sustainable alternative to help people moving around the city. Thanks to them is possible to lower the number of the cars around the city, the occupation of the parking slot. Another great value is that since the fleets are composed by full electric vehicles, they contribute to lower the CO2 emissions, and the city air pollution in general.

For the demand model we concentrate in totally in space domain. In particular we use the Kernel Density Estimate, that allows to start from a limited sample of data, and make assumptions of the underlying density function everywhere, also in the points of the domain where there is no data observation. We underlined the importance of preprocessing steps in the Knowledge Discovery Data-mining process, in particular filtering the trips in order to keep only the valuable ones and applying the proper topological transformation to them. We figured out the difference of using discretized input and output data, with continuous Gaussian Kernels with respect to continuous inputs and outputs. In particular the second approach leads to some advantages as several increment of precision and the possibility to

comfortably discretize the sample after they have been extracted from KDE, hence avoiding to fitting a different model based on discretization.

We show how traditional KDE models fail to generalize the information obtained from mobility datasets, if not properly tuned. This mostly comes from the heterogeneity of the data, which changes over time (e.g., day and night) and space (e.g., dense or suburban areas). These issues call for fine grained parameter tuning, that we solve by automatically finding the optimal parameters for classical and variable KDE approaches. About VKDE we exploited the possibility of tuning with Cross Validation approach both *alpha* and *glob_bw* parameter together, and see how the log-likelihood changes according to different combination of them. In conclusion we verified that selecting an *alpha = 0.5* according to Silverman's Law is overall a good choice. Finally comparing the same time slots in 2-D and 4-D KDE we illustrated that the general trend are similar for both, but as it is more difficult to find similar patterns in 4-D, bandwidths values are higher. This approach can be applied to many different problems involving geo-spatial estimation of density, such as telecommunication cell data. A beneficial property is the invariance of the model to scale: indeed, the optimization is agnostic on the road network, and on whether traces come are related to urban, regional, or inter-continental densities. This work paves the way for better integration between now available big datasets of mobility traces and modelling and simulation tools for mobility and transportation.

Bibliography

[1] https://en.wikipedia.org/wiki/Kernel_density_estimation.

[2] https://en.wikipedia.org/wiki/Multivariate_kernel_density_estimation.

[3] https://en.wikipedia.org/wiki/Maximum_likelihood_estimation.

[4] https://en.wikipedia.org/wiki/Geographic_coordinate_system.

[5] https://en.wikipedia.org/wiki/World_Geodetic_System.

[6] https://en.wikipedia.org/wiki/Mercator_projection.

[7] https://en.wikipedia.org/wiki/Web_Mercator_projection.

[8] https://en.wikipedia.org/wiki/Universal_Transverse_Mercator_coordinate_system#cite_note-2.

[9] https://numpy.org/.

[10] https://pandas.pydata.org/.

[11] https://geopandas.org/.

[12] https://matplotlib.org/.

[13] https://scikit-learn.org/.

[14] https://contextily.readthedocs.io/.

[15] https://github.com/mennthor/awkde.

[16] https://smartdata.polito.it/odysseus-an-origin-destination-simulator-of-shared-e-mobility-in-urban-scenarios.

[17] Icc ten conditions for a transition towards a green economy, 2012.

[18] Ian S Abramson. Arbitrariness of the pilot estimator in adaptive kernel methods. *Journal of Multivariate analysis*, 12(4):562–567, 1982.

[19] Ian S Abramson. On bandwidth variation in kernel estimates-a square root law. *The annals of Statistics*, pages 1217–1223, 1982.

[20] Michelangelo Barulli, Alessandro Ciociola, Michele Cocca, Luca Vassio,

Danilo Giordano, and Marco Mellia. On scalability of electric car sharing in smart cities. In *2020 IEEE International Smart Cities Conference (ISC2)*, pages 1–8. IEEE, 2020.

[21] Leo Breiman, William Meisel, and Edward Purcell. Variable kernel estimates of multivariate densities. *Technometrics*, 19(2):135–144, 1977.

[22] Q Chen, RJ Wynne, P Goulding, and D Sandoz. The application of principal component analysis and kernel density estimation to enhance process monitoring. *Control Engineering Practice*, 8(5):531–543, 2000.

[23] Alessandro Ciociola, Michele Cocca, Danilo Giordano, Marco Mellia, Andrea Morichetta, Andrian Putina, and Flavia Salutari. Umap: Urban mobility analysis platform to harvest car sharing data. In *2017 IEEE SmartWorld, Ubiquitous Intelligence Computing, Advanced Trusted Computed, Scalable Computing Communications, Cloud Big Data Computing, Internet of People and Smart City Innovation (SmartWorld/SCALCOM/UIC/ATC/CBDCom/IOP/SCI)*, pages 1–8, 2017.

[24] Alessandro Ciociola, Michele Cocca, Danilo Giordano, Luca Vassio, and Marco Mellia. E-scooter sharing: Leveraging open data for system design. In *2020 IEEE/ACM 24th International Symposium on Distributed Simulation and Real Time Applications (DS-RT)*, pages 1–8, 2020.

[25] Alessandro Ciociola, Dena Markudova, Luca Vassio, Danilo Giordano, Marco Mellia, and Michela Meo. Impact of charging infrastructure and policies on electric car sharing systems. In *2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC)*, pages 1–6. IEEE, 2020.

[26] Michele Cocca, Danilo Giordano, Marco Mellia, and Luca Vassio. Free floating electric car sharing: A data driven approach for system design. *IEEE Transactions on Intelligent Transportation Systems*, 20(12):4691–4703, 2019.

[27] John Cook, Naomi Oreskes, Peter T Doran, William R L Anderegg, Bart Verheggen, Ed W Maibach, J Stuart Carlton, Stephan Lewandowsky, Andrew G Skuce, Sarah A Green, Dana Nuccitelli, Peter Jacobs, Mark Richardson, Bärbel Winkler, Rob Painting, and Ken Rice. Consensus on consensus: a synthesis of consensus estimates on human-caused global warming. *Environmental Research Letters*, 11(4):048002, apr 2016.

[28] Aubrey Eckert-Gallup and Nevin Martin. Kernel density estimation (kde) with adaptive bandwidth selection for environmental contours of extreme sea states. In *OCEANS 2016 MTS/IEEE Monterey*, pages 1–5.

IEEE, 2016.

[29] Edoardo Fassio, Alessandro Ciociola, Danilo Giordano, Michel Noussan, Luca Vassio, and Marco Mellia. Environmental and economic comparison of icev and ev in car sharing. In *2021 IEEE International Intelligent Transportation Systems Conference (ITSC)*, pages 1621–1626, 2021.

[30] Mengpin Ge, Johannes Friedrich, and Leandro Vigna. 4 charts explain greenhouse gas emissions by countries and sectors, Feb 2020.

[31] Mikhail Granovskii, Ibrahim Dincer, and Marc A. Rosen. Economic and environmental comparison of conventional, hybrid, electric and hydrogen fuel cell vehicles. *Journal of Power Sources*, 159(2):1186–1193, 2006.

[32] J. D. Hunter. Matplotlib: A 2d graphics environment. *Computing in Science & Engineering*, 9(3):90–95, 2007.

[33] Stephan Hügel and Anna R. Davies. Public participation, engagement, and climate change adaptation: A review of the research literature. *WIREs Climate Change*, 11(4):e645, 2020.

[34] Yi Jin, Yulin He, and Defa Huang. An improved variable kernel density estimator based on l2 regularization. *Mathematics*, 9(16):2004, 2021.

[35] Kelsey Jordahl, Joris Van den Bossche, Martin Fleischmann, Jacob Wasserman, James McBride, Jeffrey Gerard, Jeff Tratner, Matthew Perry, Adrian Garcia Badaracco, Carson Farmer, Geir Arne Hjelle, Alan D. Snow, Micah Cochran, Sean Gillies, Lucas Culbertson, Matt Bartos, Nick Eubank, maxalbert, Aleksey Bilogur, Sergio Rey, Christopher Ren, Dani Arribas-Bel, Leah Wasser, Levi John Wolf, Martin Journois, Joshua Wilson, Adam Greenhall, Chris Holdgraf, Filipe, and François Leblanc. geopandas/geopandas: v0.8.1, July 2020.

[36] Katherine Kortum, Robert Schönduwe, Benjamin Stolte, and Benno Bock. Free-floating carsharing: City-specific growth rates and success factors. *Transportation Research Procedia*, 19:328–340, 2016. `https://doi.org/10.1016/j.trpro.2016.12.092`.

[37] Dorothea Lemke, Volkmar Mattauch, Oliver Heidinger, Edzer Pebesma, and Hans-Werner Hense. Comparing adaptive and fixed bandwidth-based kernel density estimates in spatial cancer epidemiology. *International Journal of Health Geographics*, 14(1):1–10, 2015.

[38] Don O Loftsgaarden and Charles P Quesenberry. A nonparametric estimate of a multivariate density function. *The Annals of Mathematical Statistics*, 36(3):1049–1051, 1965.

[39] MassonDelmotte, P. Zhai, S.L. Connors A. Pirani, C.Péan, S. Berger,

N. Caud, Y. Chen, L. Goldfarb, M.I. Gomis, M.Huang, K. Leitzell, E. Lonnoy, J.B.R. Matthews, T.K. Maycock, T. Waterfield, O. Yelekçi, and B. Zhou R. Yu. The physical science basis. contribution of working group i to the sixth assessment report of the intergovernmental panel on climate change, 2021.

[40] Wes McKinney et al. Data structures for statistical computing in python. In *Proceedings of the 9th Python in Science Conference*, volume 445, pages 51–56. Austin, TX, 2010.

[41] Michael C Minnotte. *A test of mode existence with applications to multimodality*. PhD thesis, Rice University, 1993.

[42] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. Scikit-learn: Machine learning in python. *Journal of machine learning research*, 12(Oct):2825–2830, 2011.

[43] Dario Pevec, Jurica Babic, and Vedran Podobnik. Electric vehicles: A data science perspective review. *Electronics*, 8(10), 2019.

[44] E Platts, J Xavier Prochaska, and Casey J Law. A data-driven technique using millisecond transients to measure the milky way halo. *The Astrophysical Journal Letters*, 895(2):L49, 2020.

[45] Parameswaran Ramachandran and Theodore J Perkins. Adaptive bandwidth kernel density estimation for next-generation sequencing data. In *BMC proceedings*, volume 7, pages 1–10. BioMed Central, 2013.

[46] Stephan R Sain. *Adaptive kernel density estimation*. PhD thesis, Rice University, 1994.

[47] Samir Saoudi, Jia Dong, and Tarik Ait-Idir. Joint maximum likelihood and expectation maximization methods for unsupervised iterative soft bit error rate estimation. In *2012 IEEE Vehicular Technology Conference (VTC Fall)*, pages 1–5. IEEE, 2012.

[48] David W Scott. *Multivariate density estimation: theory, practice, and visualization*. John Wiley & Sons, 2015.

[49] Bernard W Silverman. Using kernel density estimates to investigate multimodality. *Journal of the Royal Statistical Society: Series B (Methodological)*, 43(1):97–99, 1981.

[50] Bernard W Silverman. *Density estimation for statistics and data analysis*. Routledge, 2018.

[51] Bernhard W Silverman. Algorithm as 176: Kernel density estimation

using the fast fourier transform. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, 31(1):93–99, 1982.

[52] George R Terrell and David W Scott. Variable kernel density estimation. *The Annals of Statistics*, pages 1236–1265, 1992.

[53] Leonardo Tolomei, Stefano Fiorini, Alessandro Ciociola, Luca Vassio, Danilo Giordano, and Marco Mellia. Benefits of relocation on e-scooter sharing-a data-informed approach. In *2021 IEEE International Intelligent Transportation Systems Conference (ITSC)*, pages 3170–3175. IEEE, 2021.

[54] Bin Wang and Xiaofeng Wang. Bandwidth selection for weighted kernel density estimation. 09 2007.

[55] Fei Wu, Hongjian Wang, Zhenhui Li, Wang-Chien Lee, and Zhuojie Huang. Semmobi: A semantic annotation system for mobility data. In *Proceedings of the 24th International Conference on World Wide Web*, pages 255–258, 2015.

[56] Tiee-Jian Wu, Ching-Fu Chen, and Huang-Yu Chen. A variable bandwidth selector in multivariate kernel density estimation. *Statistics & probability letters*, 77(4):462–467, 2007.

[57] Guangyuan Zhang, Xiaoping Rui, Stefan Poslad, Xianfeng Song, Yonglei Fan, and Zixiang Ma. Large-scale, fine-grained, spatial, and temporal analysis, and prediction of mobile phone users' distributions based upon a convolution long short-term model. *Sensors*, 19(9):2156, 2019.