Expert Systems With Applications 44 (2016) 275–288

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa

Time-evolving O-D matrix estimation using high-speed GPS data streams

Luís Moreira-Matias[a,*], João Gama[b,e], Michel Ferreira[f,d], João Mendes-Moreira[b,c], Luis Damas[g]

[a] NEC Laboratories Europe, Heidelberg 69115, Germany
[b] LIAAD-INESC TEC, Porto 4200-465, Portugal
[c] Dep. de Eng. Informática, Fac. de Engenharia, U. Porto, Porto 4200–465, Portugal
[d] Instituto de Telecomunicações, Porto 4200-465, Portugal
[e] Faculdade de Economia, U.Porto, 4200-465 Porto, Portugal
[f] Dept de Ciência dos Computadores, Fac. de Ciências, U.Porto, 4200-465 Porto, Portugal
[g] Geolink 4050-275 Porto, Portugal

ARTICLE INFO

Keywords:
Online machine learning
Data streams
Mass-based clustering
Incremental discretization
Urban mobility analysis
Origin-destination matrix

ABSTRACT

Portable digital devices equipped with GPS antennas are ubiquitous sources of continuous information for location-based Expert and Intelligent Systems. The availability of these traces on the human mobility patterns is growing explosively. To mine this data is a fascinating challenge which can produce a big impact on both travelers and transit agencies.

This paper proposes a novel incremental framework to maintain statistics on the urban mobility dynamics over a time-evolving origin-destination (O-D) matrix. The main motivation behind such task is to be able to learn from the location-based samples which are continuously being produced, independently on their source, dimensionality or (high) communicational rate. By doing so, the authors aimed to obtain a generalist framework capable of summarizing relevant context-aware information which is able to follow, as close as possible, the stochastic dynamics on the human mobility behavior. Its potential impact ranges Expert Systems for decision support across multiple industries, from demand estimation for public transportation planning till travel time prediction for intelligent routing systems, among others.

The proposed methodology settles on three steps: (i) Half-Space trees are used to divide the city area into dense subregions of equal mass. The uncovered regions form an O-D matrix which can be updated by transforming the trees'leaves into conditional nodes (and vice-versa). The (ii) Partioning Incremental Algorithm is then employed to discretize the target variable's historical values on each matrix cell. Finally, a (iii) dimensional hierarchy is defined to discretize the domains of the independent variables depending on the cell's samples.

A Taxi Network running on a mid-sized city in Portugal was selected as a case study. The Travel Time Estimation (TTE) problem was regarded as a real-world application. Experiments using one million data samples were conducted to validate the methodology. The results obtained highlight the straightforward contribution of this method: it is capable of resisting to the drift while still approximating context-aware solutions through a multidimensional discretization of the feature space. It is a step ahead in estimating the real-time mobility dynamics, regardless of its application field.

1. Introduction

Today, there is a vast number of widespread portable *gadgets* such as smartphones, laptops and GPS navigational devices which are capable of tracking and communicate their position *continuously*. These devices represent an opportunity to trace their owners' activities. These activities are closely related to the underlying patterns of the daily human behavior. Researchers worldwide aim to transform these patterns into useful information about urban mobility dynamics. These patterns are valuable assets for various research fields, from disease containment to traffic management (Castro, Zhang, Li, & Pan, 2013).

These devices work as ubiquitous sensors of human mobility flows. They do it so by producing *continuous* records of GPS data, leaving a *trace* of their spatiotemporal activity. Mining this new type

[*] Corresponding author. Tel.: +4962214342261.
E-mail addresses: luis.matias@fe.up.pt, luis.matias@neclab.eu (L. Moreira-Matias), jgama@fep.up.pt (J. Gama), michel@dcc.fc.up.pt (M. Ferreira), jmoreira@fe.up.pt (J. Mendes-Moreira), luis@geolink.pt (L. Damas).

http://dx.doi.org/10.1016/j.eswa.2015.08.048
0957-4174/© 2015 Elsevier Ltd. All rights reserved.

of data is an huge challenge. Moreover, the availability of the data broadcasted by such mobility tracing devices is largely increasing. Consequently, it is reasonable to conclude that, in a near future, we will be able to *freely* collect in real-time mobility traces from multiple and **heterogeneous** sources such as buses, taxis, trams, private cars, individual smartphones, loop counters or even video cameras. Some cities are already opening their data repositories to the world, allowing the real-time collection of mobility based data (e.g. Bristol City Council, 2015). Moreover, the advances on communicational devices – which exhibit a growing trend on increasing their bandwidth while still reducing the cost per communicated packet (e.g. 4G Huang, Qian, Gerber, Mao, Sen, and Spatscheck, 2012) –, point a clear trend to support an expectation that such samples can be transmitted on a very **high rate**. Such characteristics imply that any Expert and Intelligent system built upon such data will have to deal with an evergrowing training set of heterogeneous samples in a fast mutation.

Generically, we can point three of the most common questions on Urban Mobility mining as follows:

1. Where are we traveling from/to? (i.e. spatial analysis Liu, Andris, Biderman, & Ratti, 2009);
2. How long it takes to go from point A to point B? (i.e. temporal analysis Mendes-Moreira, Jorge, de Sousa, & Soares, 2012; Moreira-Matias, Gama, Mendes-Moreira, & de Sousa, 2014);
3. When are we traveling? Or, how many of us are going from point A to point B in the time instant/interval t? (i.e. demand analysis Moreira-Matias, Gama, Ferreira, Mendes-Moreira, & Damas, 2013b).

Commonly, the question (1) is formulated as a Clustering problem (Liu et al., 2009), while the remaining ones (2,3) are formulated as Regression (Mendes-Moreira et al., 2012) ones. Typical approaches to such unsupervised/supervised learning problems (such as the (1) Expectation-Maximization algorithm (Fraley & Raftery, 2002) or (2,3) Support Vector Regression Cortes and Vapnik, 1995) are known by formulating them (partially or fully) as **local optimization problems**. Most of the well-known solvers for such problems require multiple scans over all the samples within a given training set. This type of learners is also known as offline learning methods. Despite their innumerous successful applications, these formulations are not adequate to model the dynamic nature of urban mobility problems, which usually include multiple Concept Drifts during a single day (i.e. samples that have distinct characteristics from the ones existing in the training set, which provoke a shift on the characteristics of the Statistical Population that we are aiming to generalize).

Three simple and yet impactful examples on such issues can be a (i) car accident/breakdown, a (ii) fast weather change or (iii) a big convention going on a (usually not that busy) conference center. The (i) would impact the short-term expected/predicted (2) link travel time on the affected road, as well on the ones which are more directly connected to those - which should be foreseen by any straightforward intelligent routing algorithm (e.g. Ge, Xiong, Tuzhilin, Xiao, Gruteser, & Pazzani, 2010). The (ii) may impact both the transportation demand (e.g. (2,3) bursty peaks in taxis due to heavy rains Kamga, Yazici, and Singhal, 2013) and the travel time within the entire urban area (i.e. a global reduction of the (2) average link speed), which may impact a swift change of the control policies on a given public transportation network of interest (e.g. real-time bus bunching mitigation Moreira-Matias et al., 2014). On the other hand, a type-(iii) event will seasonly change the typical (1,3) demand patterns within a given city area - which may be of interest of any smart recommendation system of profitable areas for taxi operations (Ge et al., 2010).

The abovementioned examples illustrate the need of update the learning model throughout the day on an constant manner to adequately handle the most recent information within. However, to carry out such learning task using the aforementioned problem formulations for offline learning require an unaffordable amount of time and/or resources. Recently, online learning methods had become popular by their abilities on learning from high-speed data streams. They usually do it so by relaxing some of the typical constrains of the offline learning methods using approximations to conventional learners (e.g. Hoeffding bounds Domingos and Hulten, 2000) or keeping just sufficient statistics of the input samples (e.g. Online Forest Trees in Gama, Medas, and Rocha, 2004).

The **Origin-Destination** (O-D) matrix is a state-of-the-art technique to analyze urban mobility in general (Lee, Shin, & Park, 2008; Phithakkitnukoon, Veloso, Bento, Biderman, & Ratti, 2010; Yue, Zhuang, Li, & Mao, 2009) which can cover most of the typical questions (e.g. 1,2,3) on this research topic. It consists of dividing an urban area into two finite sets of k_o, k_d non-overlapping subregions which entirely cover the initial one. Then, each cell of a $(j_o \times j_d)$: $j_o \leq k_o \wedge j_d \leq k_d$ matrix is used to generate relevant information on the city dynamics, including traffic flow analysis and transportation supply/demand prediction, among others. This information is often inferred using a broad range of algorithms and statistical models over the GPS data streams produced by each network's vehicle. Commonly, an O-D matrix comprises a time-dimension in its cells – which contain some sort of summarization of the data samples within. Consequently, it is a **discretization** method for both time and space. Despite the continuous characteristics of the GPS streams, most works on O-D matrices based on taxi GPS traces employ batch learning methods (Lee et al., 2008; Liu et al., 2009; Phithakkitnukoon et al., 2010; Qi, Li, Li, Pan, Wang, & Zhang, 2011; Yue et al., 2009; Zhang, Li, Zhou, Chen, Sun, & Li, 2011) or Bayesian statistics (Hazelton, 2008; Li, 2005; Parry & Hazelton, 2012; Perrakis, Karlis, Cools, & Janssens, 2015) using, at most, two different data sources.

This application paper proposes **incremental** discretization techniques to maintain accurate statistics of interest over a **time-evolving** O-D matrix. These statistics can be used as a bedrock for real-time analysis on human mobility dynamics, or as a valuable training input for machine learning algorithms. Our goal is to provide a sustainable learning framework, from a computational point of view, which can deal with a continuous stream of GPS traces broadcasted by multiple and heterogeneous sources. Moreover, we also intend to admit samples defined in multiple dimensional spaces by using the rich additional information that different sources may include on each sample (e.g. driver's gender and/or age, passenger load). To the best of our knowledge, this approach meets no parallel in the existing literature on data driven Expert and Intelligent systems on urban dynamics.

This methodology starts by employing **spatial discretization** using a mass-based clustering technique (K.Ting & Wells, 2010) to divide an urban area into a set of subregions. Then, the resulting clusters are incrementally updated over time using Half-Space (HS) trees (Bentley, 1975). Thirdly, a **temporal discretization** is performed by creating hierarchized dimensional spans (Chen, Chen, Lin, & Ramakrishnan, 2005) whose size depends on the amount of information available in each matrix cell. Finally, the Partition Incremental Discretization (PiD) algorithm (Gama & Pinto, 2006) is proposed to address the *incremental* maintenance of histograms on one (or more) continuous time-dependent variables of interest about the historical origin-destination data.

A large taxi fleet running in the city of Porto, Portugal, was selected as a case study. A time stamped dataset containing the spatial O-D coordinates of one million trips was used to conduct the experiments. In this study, the travel time was selected as the target variable. Travel Time Estimation (TTE) was performed as an application case of this framework.

Contributions. The results demonstrate that the proposed incremental discretization framework is a straightforward contribution in **four** distinct aspects: (i) to monitor the evolution of urban dynamics in real-time by maintaining a flexible sample-by-sample discretization method over the O-D continuous spatial space; (ii) to obtain statistics with distinct levels of detail about the flows between each

O-D pair, thereby reducing the variance in each O-D pairs' cell; (iii) to build induction models capable of characterizing the expected behavior of a given random variable about the city mobility dynamics using the aforementioned statistics as input; (iv) it discards the need of converging for any local/global *minima* by performing a series of approximations. These approximations are driven by the nature of the most recent samples. By doing so, the inference methods built upon this learning framework are able to simultaneously return reliable results while keeping intact its ability of dealing with one or multiple concept drifts.

The remainder of the paper is structured as follows: the Section 2 defines the problem. The related work is briefly revised in Section 3. The fourth Section describes the two-layer framework employed to incrementally estimate the O-D matrix. Section 5 starts by describing a histogram-based technique to discretize the target variable; then, a discussion is provided on how the histograms can *follow* the evolution of the O-D matrix. A multidimensional discretization model is also proposed to handle discretization in multiple dimensions. The fifth Section briefly describes the Case Study addressed in this work along with some details about the data employed in the experiments. The Section 7 starts by presenting an application case for the methodology (i.e. TTE), along with the experimental setup and its results. The Section 8 discusses the results obtained, as well as the application of this framework in real-world problems. Finally, conclusions are drawn, as well as future research directions on this topic.

2. Problem statement

O-D matrices are a widespread analysis technique employed in many research fields. This work addresses both the generation and maintenance of O-D matrices by mining (A) a *high-speed* continuous flow of origin/destination spatial points (discarding the path followed between the points). This task can be divided into two distinct stages. Firstly, (B) the urban area is divided into two finite sets of non-overlapping k_o, k_d subregions. Then, (C) the origin and destination (i.e. $j_o, j_d : j_o \leq k_o \wedge j_d \leq k_d$) subregions of those initial decomposition are selected as Regions of Interest (ROI) to form the final O-D matrix. A ROI corresponds to an O-D *hotspot* in a city. Then, it is necessary to decide how to store all the data inside each matrix cells given its multiple dimensions. These problems are formulated along this Section. The symbols and notations used in this paper are provided in Table 1.

Table 1
Notation and symbols employed along this section.

$\mathbb{D}$	Urban area to decompose
$v(lat, lon)$	An O-D location represented by a pair of coordinates
Ψ	Set of initial subregions / stage1 city decomposition
ψ	Membership function to get the location's cluster in Ψ
k	Number of subregions after the stage1 city decomposition
Γ	Parameter set to refine each subregion by density (stage2)
Ω	Set of final subregions / stage2 city decomposition
ω	Membership function to get the location's cluster in Ω
j	Number of subregions after the stage2 city decomposition
M	Resulting O-D matrix
S	Initial *finite* dataset of O-D locations
s_i	Data points inside the cluster Psi_i or Ω_i
$hDim_i$	Dimension chose to split a subregion Psi_i or Ω_i (i.e. lat./lon.)
θ_i	Break point to split a subregion in the $hDim_i$
C	Regions of Psi_i that must be refined (i.e. *candidates*)
c	Number of regions in C
κ	Maximum number of points in memory about one cluster
s_i	Set of data points inside the cluster Psi_i or Ω_i kept in memory
n	Total number of data points/locations in memory
N	Total number of data points/locations processed
α	Max. threshold for the mass ratio contained in a single O-D region
rt	Min. threshold for excessive mass ratio to refine a O-D region
ξ	Min. threshold for mass ratio contained in a O-D region
ϕ	Min. threshold for mass density in a O-D region
p	Split/merging test periodicity on the `layer-on`
ρ_i	Be the mass density of a region Ψ_i
a_i	Area occupied by a region Ψ_i
sm_i	Set of data points in region Ψ_i
su_i	Number of data points contained in a region Ψ_i after its last update
ϑ	Highest mass value contained inside one subregion
θ_{Ψ_i}	Split point to divide a region Ψ_i in two with equal masses

2.1. Learning from high speed data streams

Typically, **data streams** comprise a (a) *neverending* flow of data samples. Moreover, the (b) data distribution may not be *stationary*. These characteristics disable the use of many state-of-the-art machine learning algorithms. **High-speed** data streams assume that it is not possible to *scan* all the past samples before predicting the target value of the following sample (Gama, 2010). Let $X = \{x_1, x_2, \ldots, x_n\}$ be a dataset produced by a high-speed data stream until time instant t. Let *learner*() be a batch learning algorithm of interest where *model*(X, t) is the predictive model inferred by it at instant t. Finally, let λ_X be the expected sample arrival rate. The worst-case time complexity of the *learner*() is, at best, a single-scan complexity (i.e. $O(n)$). (c) High-speed data streams assume the validity of the following equation

$$T(n) = c \times n : \lim_{n \to \infty} \frac{\lambda_X}{T(n)} = 0 \tag{1}$$

where $T(n)$ is the time required by the learner algorithm to perform an individual scan for every past n samples, and c is the constant time required to process each sample. In fact, the average number of samples that may be used by any learning algorithm applicable to X is given by $\tau = \frac{c}{\lambda_X}$. In these conditions, a learner is allowed to *inspect* just a *small* number of past samples to update its model before the following sample arrives. In extreme scenarios, the learner may be forced to process just one instance at a time (i.e. $\tau = 1$). This is called an *incremental* learning method. This paper follows two assumptions: (1) a GPS data source is an (a) *infinite* stream of (b) *time-evolving* data; (2) its (c) *high* arrival rate implies that processing is made just one instance at a time.

2.2. City decomposition

A city region is a continuous two-dimensional area (i.e. a subset of $\mathbb{R}^2$), which is difficult to work with. Consequently, it is common practice to *decompose* the city into k disjoint areas to perform any data analysis of interest (Castro et al., 2013). Let $v_a(lat_a, lon_a)$ be a pair of geographic coordinates representing a *location*. Let $\mathbb{D} \subseteq \mathbb{R}^2$ be an urban area of interest defined by two rectangular vertices with the coordinates (v_1, v_2): $lat_1 > lat_2 \wedge lon_1 < lon_2$. Implicitly, it is possible to infer the following

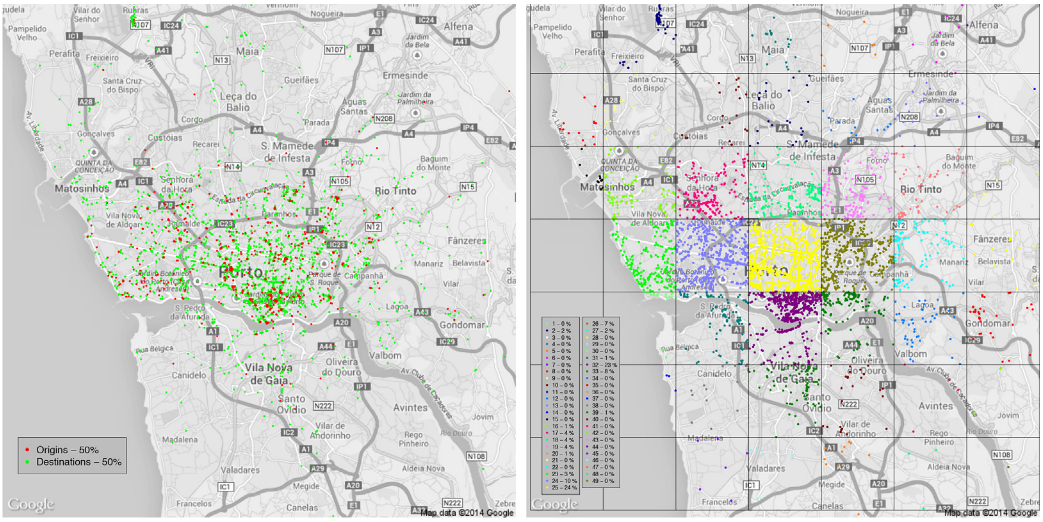
$$\mathbb{D} = [lon_1, lon_2] \times [lat_2, lat_1] \tag{2}$$

The **city decomposition** is a pair (Ψ, ψ), where Ψ is a finite set of regions and $\psi : \mathbb{D} \to \Psi$ is a membership function mapping any location $v_a \in \mathbb{D}$ to a region given by $\psi(v_a) \in \Psi$. This work uses the definitions in the Eq. 3 presented below. An example of this process is illustrated in Fig. 1.

$$\bigcup_{i=1}^{k} \Psi_i = \mathbb{D} \wedge \Psi_i \cap \Psi_l = \emptyset, \ \forall i, l \in \{1, \ldots, k\} : i \neq l \tag{3}$$

2.3. ROI selection

The ROI selection is commonly made by employing a threshold-based 0–1 function ω over some user-defined continuous criteria γ_i, such as the O-D location number or *density* within an input region Ψ_i. Formally, it is possible to define ω: $\Psi \to \Omega$ as a membership function $\omega(\Gamma)$, which can be used to iteratively form the ROI set Ω



(a) 5000 O-D pairs' coordinates. (b) Grid-Based City Decomposition.

Fig. 1. A naive example on city decomposition.

from the original subregion set Ψ. It does so based on the criteria set $\Gamma = \bigcup_{i=1}^{k} \gamma_i$. Consequently, $k \equiv |\Psi| \wedge j \equiv |\Theta| : j \le k \wedge \Theta \subseteq \Psi$.

In various works, only one spatial dimension is considered as they decompose the city according to the destinations or the origins, and not based on the relationship between these locations (e.g., the passenger demand (Lee et al., 2008) or the service offer quantity analysis Phithakkitnukoon et al., 2010). An O-D matrix M comprises the relationships between two ROI sets (i.e. origin and destination). It can be formed using two distinct approaches: (*i*) a unique pair of functions (ψ, ω) to generate both the O-D ROI sets (Ω_o, Ω_d) or (*ii*) two distinct pairs of functions $\{(\psi_o, \omega_o), (\psi_d, \omega_d)\}$ that produce two separate decompositions on the discretization of the origin/destination continuous spaces. For very large datasets, it is expected that $\Omega_o \simeq \Omega_d$ as they contain the city's ROI. However, it is very common to observe *seasonal* changes throughout *time* (i.e.: similarly to human behavior). Therefore, it is common to employ a type-*i* approach where $\Omega \equiv \Omega_o \equiv \Omega_d$. Consequently, M is represented as a quadratic matrix with size $j_o \times j_d : j_o = j_d$. The temporal discretization is then performed on the matrix cells (as suggested by previous works on related topics Lee et al., 2008; Phithakkitnukoon et al., 2010; Yue et al., 2009). The present work follows a type-*i* approach which also benefits from those assumptions.

2.4. Mobility modeling

After performing such spatial discretization, it is necessary to decide how to structure the data inside each sample x_i which is within inside each ROI considered. The problem is that those samples of trip-based locations may contain additional rich information (e.g. driver's age, gender, weather-based, etc.) rather than only a timestamp. Consequently, they are represented as multi-dimensional features rather than unidimensional ones. The challenge lies on make everything cope together on a learning system that is able to learn from any type of location-based samples, independently on the dimensions where they are defined. Moreover, we aim on letting this system evolve over time to let it express the most current state of the network at each moment - including forgetting mechanisms for deprecated data.

In the next Section, we perform a small overview on the related work on this topic, while the subsequent sections describe the methodology proposed to address the abovementioned problem.

3. Literature review

The estimation of time-dependent O-D Matrices is a thrilling problem in many research areas. Each area may face the problem using different approaches, assumptions and ends. From the abovementioned problem formulation, we can state two distinct subproblems: (1) the identification of the ROIs to form the O-D pairs and (2) the mobility modeling through analyzing one or multiple variables of interest inside each cell. The related work on those subproblems are briefly analyzed below.

3.1. ROI identification

The ROI identification is a problem of high relevance, especially on the taxi industry due to its need of an ubiquitous demand analysis. However, such analysis is also relevant for other transportation-related topics such as car sharing (for the deployment of the pick-up/drop-off stations) or even on route planning on mass transit agencies (Ceder, 2002). This analysis is firstly performed by decomposing the city into one or two-dimensional O-D zones/matrices, respectively (as previously stated in Section 2.2). This problem can be seen as a non-overlapping spatial clustering process. In (Phithakkitnukoon et al., 2010), a grid-based decomposition was used to predict the number of vacant taxis in a subregion. A Naive Bayes classifier was applied to historical GPS data of Lisbon, Portugal. (Liu et al., 2009) analyzed the Taxi Driver's Mobility Intelligence by categorizing the drivers based on their profitability. It did it so by employing a three dimensional clustering technique (*space* × *space* × *time*). In (Zhang et al., 2011), a low granularity grid was employed to infer anomalous trajectories. By detecting abnormal sequences of origin-destination cells, the authors expect to avoid frauds.

(Lee et al., 2008) created a recommendation model based framework to describe the spatiotemporal structure of the passenger demand on Jeju Island, South Korea. A spatial k-Means was employed to form time-dependent clusters. Hierarchical clustering is employed in (Qi et al., 2011; Yue et al., 2009) to mine time-dependent attractive areas concerning the passenger-finding problem. As many clustering based algorithms, these works employ the Euclidean distance to form its partitions. Consequently, these approaches are useless when using highly dimensional data. They also discard some of the most important characteristics of the GPS stream by employing batch learners and low volumes of data on their test beds. An mass-based incremental clustering framework like our own is capable of modeling seasonal demand patterns on a spatial dimension (e.g. a large conference taking place on a congress place or a big soccer match). Such characteristic is key to highlight its contribution facing the previous work on this topic.

For these reasons, the authors believe that the present approach has no parallel in the existing literature on ROI discovery for the spatial definition of O-D matrices.

3.2. Mobility modeling

Data driven approaches to estimate reliable O-D matrices for mobility modeling have been extensively studied in the literature (Barceló, Montero, Marqués, & Carmona, 2010; Bera & Rao, 2011; Li, 2005; Lu, Zhou, & Zhang, 2013; Park, Murphey, McGee, Kristinsson, Kuang, & Phillips, 2014; Perrakis et al., 2015; Toledo & Kolechkina, 2013). Commonly, this type of technique is used to characterize one specific variable of interest over multiple pairs of (known) O-D pairs. The most common types are traffic flow counts (Perrakis et al., 2015), speed profiling (Park et al., 2014) and travel time prediction (Barceló et al., 2010).

Traditionally, the most usual formulation for the O-D estimation problem aims to do it so using *incomplete* data. Such issue can lead to a non-deterministic solution, where multiple plausible non-unique O-D matrices which may diverge considerably from each other (as suggested by Lu et al., 2013). The main problem within is, given a fixed list of O-D pairs previously selected, estimate a reliable O-D of flow counts given a stream of incomplete mobility data (e.g.: traffic link counts from Automatic Vehicle Identifiers which have a natural low penetration rate as it only models data acquired from vehicles that used a toll-based road to go from O to D). Then, the problem is modeled using parametric estimation techniques such as Maximum Likelihood (Spiess, 1987), Generalized Least Squares (Toledo & Kolechkina, 2013) or, most commonly, Bayesian Inference (Hazelton, 2008; Li, 2005; Parry & Hazelton, 2012; Perrakis et al., 2015). The main drawback of the first approach regarding the latter two is that it requires the user to make an assumption on the functional form of the underlying probability distribution (typically, a Multivariate Normal Distribution or a Poisson one Bera and Rao, 2011). On the other hand, the parametrization of the model in the last two techniques is done by adding one additional source of static data (e.g. mobility surveys Kuwahara & Sullivan, 1987) which is obtained *apriori*. The advantage of this type of techniques is to leverage on a previous *belief* of the O-D matrix values in order to update them with a small percentage of the current stream evidences, thus tackling the eventual low penetration rates of the data acquisition system used[1].

This paper focuses on analyzing the O-D urban dynamics. It differs from the abovementioned approaches because it focuses on human behavioral patterns rather than on modeling traffic patterns *per se*. Even so, there is a clear overlap between the two topics that is worthy to be analyzed on three fundamental points: (i) the type of assumptions made within, (ii) the type of data considered and (iii) the amount of data available today *versus* the quantity of information that it may, or not, contain.

The (i) typical abovementioned approaches to O-D modeling require one (or multiple) assumptions in order to provide accurate forecasts of the target variable. Obviously, this is a clear limitation of such approaches as any of those assumptions may not reflect adequately the evolution of the network status (e.g. a probability density function describing the travel time between two locations which main connecting link's flow is currently affected by a car accident). Recently, some approaches have tried to model the noise introduced by such stochastic events using state-based models such as Kalman Filters (Barceló et al., 2010). However, their approach is still parametric as they assume that noise follows a Gaussian distribution (i.e. *white* noise). Per opposition, the **methodology proposed in this paper is completely non-parametric** and still incremental. Consequently, it is able to accommodate any type of functional forms on the probability distributions that describe the target variable behavior for each O-D pair.

Even if the most well known approaches for O-D take, at most, two data sources as input, there are works utilizing multiple different types (iii) such as loop counters (Djukic, van Lint, & Hoogendoorn, 2012), toll-based Automatic Vehicle Identifiers (Zhou & Mahmassani, 2006), smartphones via bluetooth connections (Barceló et al., 2010) or simple handovers (Iqbal, Choudhury, Wang, & González, 2014), probe car data from private users (Giannotti, Nanni, Pedreschi, Pinelli, Renso, Rinzivillo, & Trasarti, 2011) and from public transportation networks such as buses (Munizaga & Palma, 2012), trains (van der Hurk, Kroon, Maróti, & Vervest, 2015) and taxis (Zheng, Liu, Yuan, & Xie, 2011), or even multimodal smartcards (Ji, Mishalani, & McCord, 2015). However, just a few works consider more dimensions than the spatial and temporal one to construct the O-D matrix. A rare exception is the work presented by (Verbas, Mahmassani, & Zhang, 2011), where the vehicle classes are considered to estimate flow counts. Per opposition, this methodology is able to **accommodate rich types of data associated with the O-D locations**, which may be dispersed by multiple and distinct dimensions. This rich data is key to create scenario-oriented learning, which aim to model the expected network behavior under specific conditions - as it is properly described along Section 5.3.

Such multiple mobility data sources are opening new possibilities on the O-D matrix estimation problem. The main breakthrough relies on the data availability, which is today considerably (and increasingly) higher than it used to be some time ago due to the recent advances on communicational frameworks (e.g. 4G Huang et al., 2012). Moreover, the large increasing of the urban areas worldwide is pushing them to find solutions to maintain sustainable mobility levels within. One of the solutions is to create **open repositories of mobility data**, which are accessible by anyone. Portland (U.S. Department of Transportation, 2015), Dublin (Dublinked, 2015), Porto (Moreira-Matias, Azevedo, Mendes-Moreira, Ferreira, & Gama, 2015) and Ottawa (CKAN, 2015) are some of the examples which published location-based data from their public transportation networks (buses, taxis and light tram), along with loop counters. (Bristol City Council, 2015) went to another level by providing an unprecedented real-time access to multiple city mobility indicators (such as flow-counts, congestion levels or car accidents) to anyone interested on accessing it so. Such trend is clearly pointing that the problem will shift from dealing with a low penetration rate to handle an *excessive* flow of data, given by mixing all those heterogeneous sources of mobility data. Consequently, the problem on O-D matrix estimation is now on how to learn from such multiple sources. Our work results on a flexible discretization framework which summarizes this data on different levels of detail, depending on the amount of data available on a given multidimensional chain. This characteristic is key to successfully deal with the high speed data streams generated by those multiple sources. The authors want to claim it so even considering that the experiments conducted to validate the methodology hereby proposed used (rich) probe car data (i.e. taxis) standalone. This ability comes from the way that the framework is designed. The mechanisms that validate such claim are presented in detail along Section 5.

The two works that have more similarities to our own are presented by (Zheng et al., 2011) and (Giannotti et al., 2011), respectively. The work of (Zheng et al., 2011) also models urban mobility using O-D matrix matrices using taxi trajectories. However, **they constrain the matrix spatial boundaries to be major roads** - which can be faced as a limitation of such approach. (Giannotti et al., 2011)

[1] The reader can consult the Section 2.3 in (Bera & Rao, 2011) to know more about this type of parametric methods;

propose a new query-oriented logical language to store information typical trajectories collected from an unbounded stream of locations collected from probe car data. Despite the straightforward characteristics of this framework in terms of scalability, it does not model other context-aware data dimensions (such as the vehicle class or the users' charateristics). The different discretization levels of this framework - which extend the classical spatiotemporal concept - can be seen as an opportunity to maintain multiple statistics of interest of the O-D patterns instead of just one. By these reasons, the multidimensional tree-based discretization of the trips' attributes is straightforward on the real-time O-D matrix estimation problem, regardless of the research goal and scope. This methodology is presented along the next two Sections.

4. Online O-D matrix estimation

One of the major problems of decomposing an area into a set of subregions Ψ is guaranteeing that each subregion contains sufficient data points to characterize it. An example of this problem is the popular grid-based decomposition (see Fig. 1b), where the city is decomposed on equal-sized regions based on a user-defined width/height (Castro et al., 2013). Its popularity resides on its simplicity. However, it is naive as it is independent from the data spatial distribution. It results in regions containing an excess/deficit of data samples. The goal with this work is to decompose a city area into equal-sized subregions regarding the *number of points* within (i.e. **mass**).

Let $S = \{v_1, v_2, \ldots, v_n\} : S \subseteq \mathbb{D}$ be a set of n O-D locations of interest and `spcls` be a *data driven* spatial discretization function defined as follows

$$\texttt{spcls}(\mathbb{D}, S, \Gamma) = \{\Psi, \psi, \Omega, \omega\} : \gamma_i = \gamma_l, \forall i, l \in \{1, \ldots, k\} \tag{4}$$

Finally, let $s_i \subseteq S$ be the set of data points contained in a subregion Ψ_i, where $|s_i|$ is the region mass. The high-level goal is to build an online unsupervised learning method `spcls` that minimizes the value of mass standard deviation ($\sigma_{|s_i|}$).

The incremental estimation of an O-D Matrix without any prior knowledge is a difficult task. A two-layer discretization algorithm is proposed to overcome this problem. In the `layer-off`, (A) a batch learning algorithm starts by performing hierarchical mass-based clustering (K.Ting & Wells, 2010) to find the best k subregions that meet this last high-level goal. Then, a density-based function is defined as ω. Finally, the O-D matrix is built based on the resulting Ω. The second layer (`layer-on`) (B) goes from the output of the previous layer to incrementally update a *sufficient* amount of statistics about the regions in Ω. This methodology is thoroughly described along this section.

4.1. `layer-off`: Batch O-D matrix estimation

Let Ψ_i be a *rectangular* subregion defined by two vertices $v_{i,1}, v_{i,2}$ whose coordinates are defined as follows:

$$\begin{aligned} lat_{i,1} &= \max(lat_i), lon_{i,1} = \min(lon_i), \\ lat_{i,2} &= \min(lat_i), lon_{i,2} = \max(lon_i) : lat_i, lon_i \in \Psi_i \end{aligned} \tag{5}$$

This algorithm starts by initializing $\Psi = \Psi_1, k = 1 : \Psi_1 = \mathbb{D}$. Then, it iteratively runs a cycle composed of five steps: firstly, it selects the ith subregion as $\arg\max_{i \in \{1,\ldots,k\}} |\Psi_i|$. Secondly, the length of the vertical/horizontal ith subregion is computed using the *Haversine* distance between two geographic coordinates (Robusto, 1957). Then, one of the latitude/longitude is selected as the largest/shortest dimension $hDim_i$, $lDim_i$ based on that length. The third step consists on finding the **binary** split point θ_Ψ which divides the region space Ψ into two regions with *equal* masses. Fourthly, it creates a new $k + 1_{th}$ subregion defined by $\{\Psi_{k+1}, s_{k+1}\}$, where $\Psi_{k+1} \subset \Psi_i$ is defined by the area's breakpoint θ_Ψ of the $hDim_i$ dimension. s_{k+1} is defined as follows:

$$s_{k+1} = \left\{ v_o | \, v_o[hDim_i] \geq \theta_\Psi, \forall o \in \{1, \ldots, |s_i|\} \right\} \tag{6}$$

Finally, the algorithm updates the k number of partitions as $k' = k + 1$, as well as the sets Ψ_i, s_i as Ψ'_i, s'_i defined in the following equations.

$$\Psi'_i = \{v_o | \, v_o \in \Psi_i \wedge v_o \notin \Psi_{k'}, \forall o\} \tag{7}$$

$$s'_i = \{v_q | v_q \in s_i \wedge v_q \notin s_{k'}, \forall q\} \tag{8}$$

This cycle only stops when $\vartheta \leq \alpha$, where α is a user-defined parameter (commonly a small ratio of n) and $\vartheta = \max_{i \in \{1,\ldots,k\}} |s_i|$. It defines the desired *granularity* level.

The suggested mass-based partitioning method follows closely the method proposed by (K.Ting & Wells, 2010). This application case is a two-dimensional case as $\mathbb{D} \subseteq \mathbb{R}^2$. Its implementation is also made through a **Half-Space tree** where the concept of *space* is given by each region's **mass**. The split point θ is computed as the *median* value of $hDim_i$ on s_i. Consequently, ψ will be a decision tree where the *leaves* will contain a cluster and the *nodes* will contain a split-point condition regarding one of the two spatial dimensions.

After the initial decomposition, an ROI selection is performed. Let ρ_i be the mass density of a region Ψ_i given by $\rho_i = |s_i|/a_i, \forall i$, where a_i is an area occupied by the region Φ_i. Let ϕ, rt be a user-defined minimum density-based threshold and a mass-based threshold ratio, respectively. Let ξ denotes a minimum mass-based threshold ratio where $\xi \ll \alpha$. The membership function ω: $\Psi \to \Omega_1$ can be defined as follows:

$$\omega_1(\rho_i, \phi) = \begin{cases} 1 \; if \; \rho_i \geq \phi \vee |s_i| \geq \frac{\alpha \times n}{1+rt} \\ 0 \; if \; \rho_i < \phi \wedge |s_i| < \frac{\alpha \times n}{1+rt} \end{cases} : 0 < rt \ll 1 \tag{9}$$

The remaining regions form a set of c region *candidates* $C = \{\Psi_i | \Psi_i \in \Psi \wedge \psi \notin \Omega_1\}$ which may need to be *refined*. The goal now is to find subregions in each region C_i which have, at least, $1 - rt$ percentage of the total data points $\in C_i$, i.e. $|s_i|$. For that, the method runs a four-step cycle: firstly, it selects the ith subregion candidate as $\arg\min_{i \in \{1,\ldots,c\}} \rho_i$. Secondly, it discards the candidate C_i if $|s_i| < \xi \times n$. Such test aims to *filter* regions without a relevant quantity of O-D flows within. Thirdly, it finds a split point θ to divide C_i into $\{C_{c+1}, C_{c+2}\}$ as $|s_{c+1}|/|s_{c+2}| \simeq rt$, using an approach similar to the one employed in stage 1. Finally, C and Ψ are updated as follows $C' = C \setminus C_i$ and $\Psi' = \Psi \setminus \{C_i\} \cup \{C_{c+1}\} \cup \{C_{c+2}\}$. The ROI set Ω is updated as $\Omega' = \Omega \cup \{C_{c+2}\}$ if $\omega(\rho_{c+2}, \phi) = 1$. Otherwise, C_{c+2} returns to the candidate set as $C'' = C' \cup \{C_{c+2}\}$. This cycle runs continuous until $C \equiv \emptyset$.

4.2. `layer-on`: Incremental O-D matrix Estimation

Let $S_t = \{v_1, v_2, \ldots\}$ be an **infinite** set of locations where N is the number of samples achieved at time instant t defined as $|S_t| = N : \lim_{t\to\infty} N = \infty$. Let $sm = \{sm_1, \ldots, sm_k\}$ be the set containing the data points sm_i within a subregion Ψ_i and n be the number of points stored in *memory* at instant t. After performing the *first* run of `layer-off`, $n = N$ and $s \equiv sm$. However, this relationship cannot be maintained as the *memory* has a bounded domain, while N has an unbounded domain. Therefore, n is constrained as $\lim_{t\to\infty} n \ll N$.

To define the domain boundaries of n, it is necessary to describe the *minimum* amount of information required to characterize the spatial data distribution in Ψ. This information can be used to reconstruct Ψ, Ω at all times by using the points in $sm : \sum_{i=1}^{k} |sm_i| = n$. To do so, the `layer-on` starts by setting the maximum number of points $\kappa = \arg\max_{i \in \{1..k\}} |sm_i| \in sm$ as the one obtained at the time instant immediately after the *first* run of `layer-off`. Consequently, the domain of n meets its constraint as $\lim_{t\to\infty} n = \kappa \times k \ll N$.

Let $\bar{lat}_i, \bar{lon}_i$ be the average latitude/longitude of the $|s_i|$ O-D points in region Ψ_i at time instant t. This algorithm iteratively processes each new sample $v_N \in S_t$ in a three-step loop: firstly, it determines $R_i = \psi(v_N) : R_i \in \Psi$ as the O-D subregion to which v_N belongs. Secondly, it updates the number of points $|s_i|$, as well as $\bar{lat}_i, \bar{lon}_i$, based on v_N. sm_i is also updated as $sm'_i = sm_i \cup \{v_N\}$. However, if $|sm'_i| > \kappa$, a *forgetting* mechanism is launched. The algorithm deletes the most outdated data point (sm'_{i_1}) from the memory as $sm''_i = sm'_i \setminus \{sm'_{i_1}\}$. Its goal is to maintain the n inside a bounded domain. Finally, the algorithm determines which of the current partitions in Ψ meets the **merge/split** criteria. This operation is periodically performed, where p represents its period . p can be defined in time as (i.e. $p \geq \lambda_{St}$, where λ_{St} stands for the expected arrival rate of new locations $v_N \in S_t$) or in space (i.e. each p samples) as $p \geq 1$. The value of p sets how *reactive* our model will be.

4.2.1. Merging partitions

By merging, the algorithm aims at *recovering* the regions from the ROI set Ω where the number of O-D points increases more than expected. Let su represent the region mass after its last update (i.e. merge/split). su is initialized as $su_i = |sm_i| : i \in \{1, \ldots, k\}$ right after the *last* run of the `layer-off`. The `merge` operator is launched in every region in $C = \{C_i | C_i \notin \Omega \wedge C_i \in \Psi \wedge |s_i| > 2 \times su_i\}$. The merge operation starts by finding the deepest conditional node *node* of ψ which divides C_i from another region Ψ_{old} (it is an operation with a worst-case time complexity of $O(k)$). Secondly, the operation transforms the *node* into a leaf node with the cluster of the newest region Ψ_{new} defined as $\Psi_{new} = \Psi_{old} \cup C_i$. Ψ, k and s are updated accordingly as $\Psi' = \Psi \cup \{\Psi_{new}\} \setminus \{\Psi_{old}\} \setminus \{C_i\}$, $k' = k + 1$ and $s' = s \cup \{s_{new}\} \setminus \{s_{old}\} \setminus \{s_i\}$. Ω and su are also updated as $\Psi_{new} \in \Omega$: $|s_{new}| \geq \xi \times N$ and $su_{new} = |s_{new}|$.

4.2.2. Splitting partitions

The splits follow a similar approach as the one proposed in Stage 1 of the `layer-off`. The `split` operator is triggered in every region in $C = \{C_i | C_i \in C : C_i \in \Omega \wedge |s_i| > \alpha \times N\}$. The main difference resides in defining the split point θ. In this layer, it is not possible to conduct a single-scan operation on multiple data points $\in sm_i$ to calculate the median. Instead, $\bar{lat}_i, \bar{lon}_i$ are used, - which are easily maintained following an *incremental* logic.

4.3. Two-layer framework

Similarly to many incremental learning algorithms (Gama & Pinto, 2006), the $\texttt{spcls}(\mathbb{D}, S, \Gamma)$ maintains two distinct *layers*: the `layer-off`, which determines the best possible ROI set Ω by employing unsupervised batch learning methods over the entire dataset available, and `layer-on`, which approximates Ω by updating itself to each new data point. This flexibility comprises an error which grows as the `split` operator is invoked in the `layer-on`. To mitigate this effect, the framework can launch the `layer-off` on-demand. The foundation for this ability is s. It is a set of data points that keeps the most recent data points of each existing region Ψ_i. s is maintained using a *sliding window* whose size is determined by the constant κ, which is obviously correlated to the parameters α and n.

Therefore, the `spcls` can be classified as an unsupervised learning method which is also incremental. Its parameter set Γ is defined as $\Gamma = \{n, \alpha, \phi, rt, \xi, p\}$. The most sensitive parameters are ϕ and ξ as they define the boundaries of Ω. rt just defines if a region may be refined or not. n and α affect spatial complexity, while p causes small drifts on the time complexity.

The O-D matrix is formed as $M[r, i]$ denotes a cell containing information on the mobility flows from the region Ω_r to the region Ω_i. The matrix evolution over time poses constrains when storing this information, because not only should it be maintained incrementally, but it should also be easily *decomposed* in order to follow the splits/merges performed. Incremental Histograms are proposed to meet these constraints, which are described in the following section.

5. Incremental data discretization using histograms

Histograms are a state-of-the-art method in exploratory data analysis. They make it possible to discretize continuous variables into *intervals*. This approach is a common building block of many machine learning algorithms (e.g. Bayesian Learning Domingos and Pazzani, 1997), and for that reason it is proposed as a tool to maintain accurate statistics over a time-evolving O-D matrix.

Let H be defined as the set of all histograms in M (i.e. the histograms describing a variable of interest in each cell of M). Let $h_{o,\,d} \in H$ represent a histogram of q intervals discretizing a continuous spatiotemporal variable of interest $X_{o,d} = \{(x_i, v_i) | v_i \in \Psi_o \forall i\}$ and $|h_{o,\,d}|$ denotes the mass within. $X_{o,\,d}$ describes directional interactions between the O-D regions Ψ_o, $\Psi_d \in \Omega$ (e.g.: x_i may represent a value of any variable of interest). $h_{o,d} = (B, F)$ can be defined as a set of breakpoints $B = \{b_1, \ldots, b_{q-1}\}$ and a set of frequency counts $F = \{f_1, \ldots, f_q\}$. This section describes a fully incremental strategy to maintain histograms on $X_{o,\,d}$ in real-time on distinct dimensional levels.

5.1. The partition incremental discretization `PiD`

The `PiD` is a fully incremental algorithm capable of maintaining accurate histograms of never-ending streams of data (Gama & Pinto, 2006). This paper proposes the employment of the algorithm to maintain histograms of *equal width*, such as $(b_i - b_{i-1}) = (b_l - b_{l-1}) = \delta_q$, $\forall i, l$.

This algorithm works on two different layers. Let q, q_1 be two user-defined number of bins and $[\nu_1 \colon \nu_2]$ be the range of $X_{o,\,d}$. q stands for the *desired* number of bins, while q_1 is used as input parameter to the `layer1` defined as $q_1 \gg q$. The `layer1` is initialized as $F = \{f_i | f_i = 0, \forall i\}$ and $B = \{\nu_1, \ldots, \nu_2\} : (b_i - b_{i-1}) = \delta_{q_1}, \forall i$. Then, the algorithm runs continuously, incrementing f_i every time a sample (x_a, v_a) is added, where $v_a \in \Psi_o$. If $x_a < \nu_1 \vee x_a \geq \nu_2$, a new bin is added to such extremity with the step δ_{q_1}. The `split` operator is triggered on a bin if $f_i > \eta$, where η is a user-defined parameter usually defined as a ratio of the histogram mass. Consequently, two bins are created, each one comprising *half* of the interval $[b_i, b_{i+1}]$ and containing the same frequency $f'_i = f_i/2 : f'_i \in \mathbb{N}$.

The `layer2` is launched every time the user needs to analyze the data. It iteratively merges the bins in `layer1` to meet the desired q in terms of size intervals δ_q. The main advantage of maintaining these layers is that it is possible to easily produce histograms of different sizes each time it is necessary to discretize the domain variable. Additional details on the `PiD` algorithm are provided in (Gama & Pinto, 2006).

5.2. Following the O-D matrix evolution

One of the major issues of building histograms is the definition of q. There is not a well-established general strategy to do so. Different strategies may be employed depending on the user's purposes. The main contribution of `PiD` is that q does not need to be constant: it can be either time or sample dependent. Whenever Ψ, Ω and M change over time, H must follow the merge and split operations. Let δ_{min} be the minimum interval size in H. The interval widths in H must be subjected to the following constraint:

$$H = \{h_i(B, F) \,|\, \exists\, a \in \mathbb{N} : (b_l - b_{l-1}) = \delta_{min} \times 2^{a-1}, \ \forall i, l\} \tag{10}$$

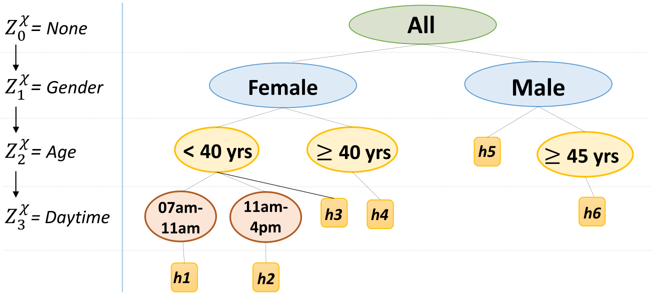


Fig. 2. Example of a multidimensional hierarchy to discretize attributes. Note that the zoom level and the discretization intervals may not be constant.

Consequently, the problem of merging two histograms $h_{o_1,d}, h_{o_2,d}$ into a single one $h_{o,\,d}$ can be defined as

$$q_{o,d} = \frac{\max b_l - \min b_l}{\delta_{o,d}} : b_l \in \{B_{o_1,d} \cup B_{o_2,d}\} \tag{11}$$

where $\delta_{o,d} = \max(\delta_{o_1,d}, \delta_{o_2,d})$. Then, the `layer2` is employed to turn the histograms $h_{o_1,d}, h_{o_2,d}$ into equal-width histograms as $q_{o_1,d} \equiv q_{o_2,d} \equiv q_{o,d}$. Finally, the frequency set is defined as $F_{o,d} = \{f_i | f_i = f_{io1,d} + f_{io2,d}, \forall i \in \{1, \ldots, q_{o,d}\}\}$.

The constraint defined in Eq. 10 makes the `layer2` task *easier* by guaranteeing that $\delta_i \mod \delta_{min} = 0, \forall i$. This property guarantees that all the histograms in H are *additive* between each other (Moreira-Matias, Gama, Ferreira, Mendes-Moreira, & Damas, 2013a). The division of $h_{o,\,d}$ into $h_{o_1,d}, h_{o_2,d}$ is a simple operation where $B_{o_1,d} = B_{o_2,d} = B_{o,d}$ and $F_{o_1,d} = F_{o_2,d} = \{f_i \in \mathbb{N} | f_i \simeq f_{io_d}/2\,, \forall i\}$.

5.3. Dimensions and hierarchies

The histograms are a well-known approach to provide sample-based discrete approximations of a Probability Density Function (p.d.f.) on the value of a continuous variable $X_{o,\,d}$. However, it is known that the mobility dynamics (such as the number of taxi pick-ups/drop-offs (Yue et al., 2009) in a region, or a bus round-trip time Mendes-Moreira, Moreira-Matias, Gama, and de Sousa, 2015), follow a *bimodal* distribution (e.g. peak/non-peak hour) throughout the day. Mobility dynamics can even be multimodal if a larger time span is considered, such as one week (workday/weekend). This p.d.f. can be difficult to learn online. To overcome this problem, **Dimensional Hierarchies** are proposed as a *flexible* method to discretize other dimensions describing $X_{o,\,d}$ (e.g. the **temporal**).

Let Z be a set of χ dimensions related to $X_{o,\,d}$, where $Z_i \in Z$ denotes a hierarchized set of $|Z_i|$ dimensional attributes. (Chen et al., 2005) firstly propose it as a method to discretize $X_{o,\,d}$ on multiple χ dimensional axis. Depending on the amount of data available on $X_{o,\,d}$, the discretization layers on each axis may have different *zoom* values.

This work adapts this definition by redefining Z as a hierarchical set of dimensions. Let $Z^\chi = \bigcup_{i=1}^{\chi} Z_i$ be an **ordered** set of *multidimensional* attributes where the order is user-defined (depending on the purpose of the histogram). The discretization intervals in each zoom level may also be user-defined or data-driven (e.g. breakpoints on average values and/or quartiles). The proposed framework maintains distinct histograms $h_{o,\,d,\,i}$ on every zoom level i by continuously running the `layer1` over the histograms. The `layer2` is triggered prior to each statistical analysis of $h_{o,\,d,\,i}$ only if $|h_{o,d,i}| > \epsilon_i = 2 \times \epsilon_0$. ϵ_0 denotes a user-defined parameter for the minimum amount of available data points to trigger the `layer2` on the zero-level dimensional hierarchy (i.e. base histogram; without dimensional discretization). An illustrative example of this framework is provided in the Fig. 2. In this example, h_1 stands for a histogram of the maximum instant speed of a taxi driven from region o to region d by a 40-year-old female subject between 07am and 11am.

Z^χ establishes relationships between attributes of distinct dimensions. Conversely to Z, initially proposed in (Chen et al., 2005), Z^χ does not allow different levels of discretization in different dimensions. It is necessary to maintain additional histograms if this analysis is intended. This step works as a threshold search for the *nearest neighbor*, which tries to build statistics using past samples where the descriptive variables are similar to the present variables. It does so by maintaining a decision tree of each O-D where the goal is to find the histogram which gives the best approximation to the present scenario. Consequently, the goal is to describe $X_{o,\,d}$ using multiple attribute-based histograms which are more likely to approximate unimodal p.d.f. (rather than multimodal p.d.f.).

6. Case study

A taxi company operating in the city of Porto, Portugal, was used as the case study. This city is the center of a medium-sized urban area (consisting of 1.3 million inhabitants), where passenger demand is lower than the number of vacant taxis running, resulting in a huge competition between companies. The data were acquired using the telematics installed in each of the 441 running vehicles of the company fleet. The data refer to a non-stop period of nine months between August 2011 and April 2012. Each data chunk firstly arrives with the following six attributes: the driver's ID, a Julian timestamp, the taxi status (zero/one for vacant/busy), and the latitude/longitude coordinates.

The variable of interest in this study is the **travel time** between two O-D locations. This dataset was processed to obtain a stream containing two million O-D locations. They correspond to one million taxi trips performed during this period. Fig. 3 represents a sample-based estimation of the p.d.f.. The *lognormal* form indicates that the taxi services in the city are usually *short* timed (such as 50% < 10m). However, at this granularity, it is not possible to infer more than this since a specific route between an O-D pair is not explained.

7. Experiments

This section presents the experimental work performed in this context. It starts by describing a *naive* online learning model built over the proposed framework to perform Travel Time Estimation (TTE). Secondly, the experimental setup and the evaluation metrics are described. Finally, the results obtained are presented.

It is important to highlight that the authors do not want to claim this induction model as a contribution to the TTE problem *per se*. The literature on this topic is extensive (Mendes-Moreira et al., 2012). The results obtained thorough this model work as a proof of concept on the applicability of this framework to maintain accurate real-time statistics on urban dynamics.

7.1. An application for travel time estimation

TTE aims to predict the cruise time of a given trip between an O-D pair of locations. It can be defined as short or long-term depending on the predicting horizons (Mendes-Moreira et al., 2012). The most common is the short-term one. It is commonly employed in Automatic Traveler Information Systems (ATIS) and Navigational GPS devices (Carrascal, 2012; Chien, Ding, & Wei, 2002). Producing online predictions on this stochastic variable is a difficult problem. Typically, these systems employ batch regression models along with online models (such as time-series analysis and/or state-based induction models) to update the initial predictions using the real-time vehicle trace (Bin, Zhongzhen, & Baozhen, 2006; Chen, Liu, Xia, & Chien, 2004; Chien et al., 2002).

This work considers the TTE in a more classical approach: given a pair of O-D locations (v_o, v_d) at time instant t, the target variable is the cruise time between these locations, expressed as $\beta_{o,\,d,\,t}$. Let $h_{o,\,d,\,z}$ be

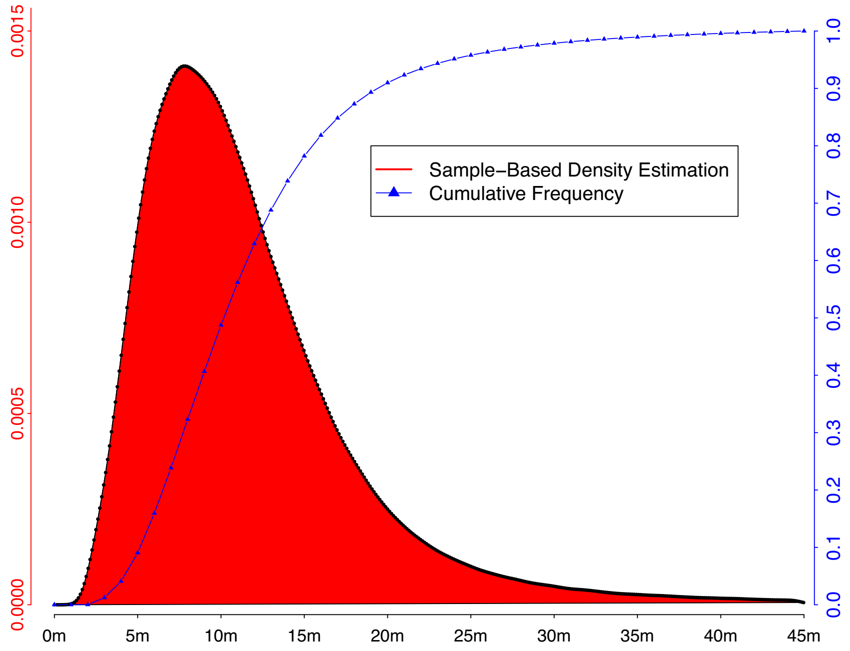


Fig. 3. Kernel Density Estimation (Gaussian) of the Travel Time (in minutes). Note the *lognormal* form and the low density values (< 0.0014).

the most suitable histogram to describe the present scenario given the values of the dimensional attributes defined in Z^{χ}. Let $\bar{b_i} = (b_i - b_{i-1})/2$ denote the center of the interval corresponding to the upper and lower bounds on the bin *i*. $\beta_{o,\,d,\,t}$ can be obtained as follows:

$$\beta_{o,d,t} = \frac{1}{\Upsilon} \times \sum_{i=1}^{q_{o,d,z}} \left[\left(\frac{f_{o,d,z,i}}{\max\left(f_{o,d,z,i}\right)} \right)^2 \times b_{o,d,z,\bar{i}} \right] \tag{12}$$

$$\Upsilon = \sum_{i=1}^{q_{o,d,z}} \left(\frac{f_{o,d,z,i}}{\max\left(f_{o,d,z,i}\right)} \right)^2 \tag{13}$$

where *T* denotes the time elapsed between *t* and *t* − 1. The quadratic normalization of the frequencies in the Eq. 12 aims at minimizing the well-known vulnerability of equal-width discretization techniques: outliers (Gama & Pinto, 2006).

7.2. Experimental setup

To define $q_{o,\,d,\,i}$, it was necessary to have a rule capable of dealing with large volumes of samples as long as it meets the constraint expressed in the Eq. 11. It can be expressed as follows:

$$q_{o,d,i} \simeq |h_{o,d,i}|^{(\frac{2}{3})} : b_{o,d,i} - b_{o,d,i-1} \leq 120 \wedge q_{o,d,i} \in \mathbb{N} \tag{14}$$

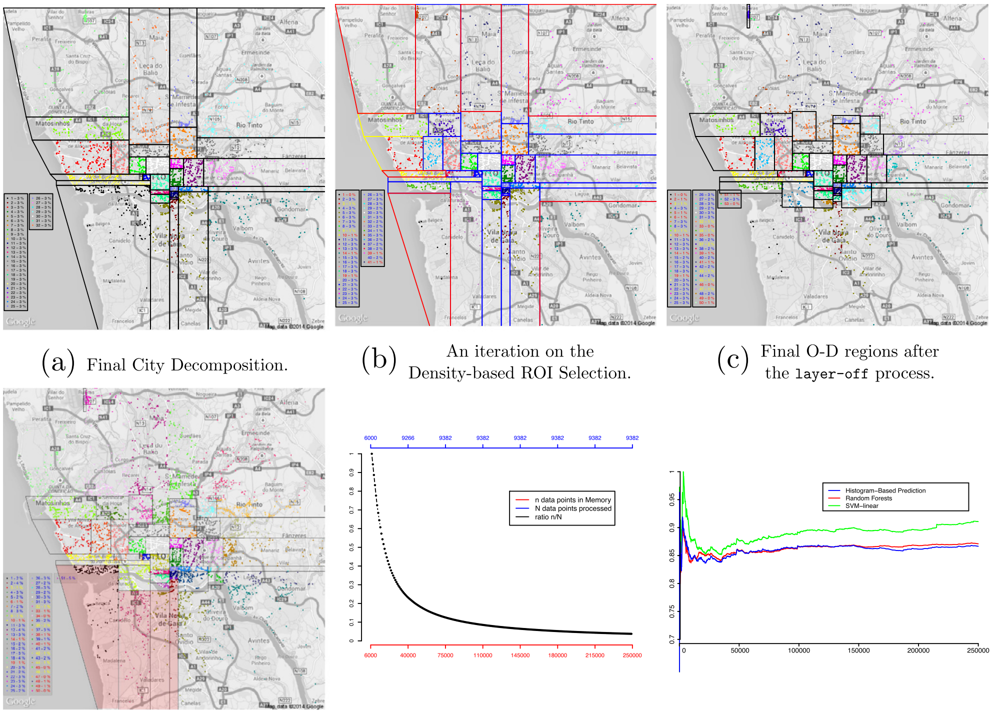
This rule is merely user defined and it was chose after carrying out a sensitivity analysis on five different equations to set the number of bins $q_{o,\,d,\,i}$. These methods were developed following closely the Section 2 in (Birge & Rozenholc, 2006). A sensitivity analysis was performed for the parameters *n, rt,* ϕ, ξ and α based on a simplified version of Sequential Monte Carlo method over an older dataset (the reader can consult the survey in (Cappé, Godsill, & Moulines, 2007) to know more about this topic). These parameters were chose because are the ones which variations reflected some changes on the method outcome during such analysis. The tested values were all the *admissible* combinations (i.e. $\alpha > \xi$) on the following ranges: $n = \{2000, 4000, 6000, 10000\}$, $\alpha = \{0.02, 0.05, 0.10\}$, $rt = \{0.1, 0.2, 0.3\}$, $\xi = \{0.005, 0.01\}$ and $\phi = \{0.3, 0.5, 0.8\}$. The best combination of values for this set of parameters was then selected to conduct these experiments. This combinations is detailed in Table 2, along with the remaining ones.

Table 2
Parameter Setting used in the experiments.

Parameter	Value	Description
n	6000	Number of O-D points used on the `layer-off`
α	$0.05 \times N$	Maximum mass ratio contained in a O-D subregion
rt	0.1	Minimum excessive mass ratio to refine a O-D subregion
ξ	$0.01 \times N$	Minimum mass ratio contained in a O-D subregion
ϕ	0.5 of the avg. mass density	Minimum mass density in a O-D subregion
p	each sample	Split/merging test periodicity on the `layer-on`
$q_{o,\,d,\,i}$	Eq. 14	Desired number of bins on `layer2`
q_1	270: width (δ_{q_1})=10s	Initial number of bins in `layer1`
ϵ_0	100	Minimum number of samples to build a zero-level histogram
δ_{min}	2s	Minimum interval width for the histograms
η	0.30	Maximum total mass ratio contained on a single bin in `layer1`

In the experiments, the time dimension is expressed in seconds. They were conducted using the R Software (R Core Team, 2012). The graphical representations of the city O-D regions were obtained using the package `[RGoogleMaps]`. The `layer-off` was just triggered once to start the algorithm. $Z = \{Distance, Time\}$ were the dimensions considered, while $Z^{\chi} = \{\texttt{haversineDistance}, \texttt{dayTime}, \texttt{weekType}, \texttt{dayType}\}$ was defined as the multidimensional hierarchy set. The (1) `haversineDistance` has an unique breakpoint based on historical data (the average distance in the trips described by $h_{o,\,d,\,0}$). The remaining three attributes have breakpoints for their intervals defined as (2) {07h-11h, 11h-16h, 16h-21h, 21h-07h}, (3) {*Workday,Weekend*} and the (4) seven days of the week, respectively. The sea was considered a constraint to compute a region area. This was done by defining a minimum longitude along the coast. The areas were calculated by approximating the constraints using trapezoids.

The histograms built were used as input for the prediction model presented here to infer the travel times of the most recent 250,000



(a) Final City Decomposition.

(b) An iteration on the Density-based ROI Selection.

(c) Final O-D regions after the `layer-off` process.

(d) O-D Regions after processing 200,000 data points.

(e) Ratio between the samples seen and the memory used.

(f) Normalized RMSE for the tested prediction algorithms.

Fig. 4. Illustration of the Time-Evolving OD-Matrix Estimation Process. Note the density refinement in the northwest airport area discovered in C), the ability to adapt to a large increase in the region's mass in (D), and the low memory requirements to maintain a time-evolving framework in (E).

trip samples (i.e. O-D trip pairs (v_o, v_d)). The attribute values of each sample were used to select the most suitable histogram. The option chosen was to build histograms in every region $\Psi_i \in \Psi$, maintaining a quadratic $k \times k$ O-D matrix over the entire city. However, the histograms were not employed if $\Psi_o \notin \Omega$ in the current time instant. Whenever there is no zero-level histogram available, a naive approach is followed by assuming a constant cruising velocity of 30 km/h. Predictions were also produced on the travel time interval by selecting the minimum number of consecutive bins containing, at least, 75% of the mass $|h_{o,\,d,\,i}|$.

To demonstrate robustness, the model was tested in three distinct scenarios: maintaining the histogram framework over an O-D matrix built on a grid-based City Decomposition (by dividing the city into 7×7 equally sized areas) and comparing it with the mass-based approach; employing zero-level histograms vs. the proposed multidimensional discretization, and monitoring the performance of the induction algorithm over time against two state-of-the-art offline regression methods on TTP: the Random Forests (Mendes-Moreira et al., 2012) and the Support Vector Machines (Bin et al., 2006; Mendes-Moreira et al., 2012). The regression features were defined as follows: (1) day, coded as a sequence of integer numbers; (2) starting Time (in seconds) and (3) day of the week. The packages `[randomForest]`, `[e1071]` provided the methods' implementations used in the experiments. They were executed using their default parameter setting. Each O-D pair was treated as an independent regression problem (as in the induction model proposed).

7.3. Results

Fig. 4 illustrates the multiple stages of estimating the O-D matrix using HS Trees. The first four subfigures report the Offline Estimation process, while the fifth reports a `layer-on` iteration. The fifth subfigure compares the memory used during the online estimation with the number of data points processed. The last subfigure reports the evolution of the algorithms' prediction error throughout time. This report is based on a normalized RMSE. This metric is calculated firstly by computing the average RMSE throughout time for each predictive method. Then, all the series obtained are divided by the same maximum value. The aggregated results for all the tested samples are presented in Table 4. The effects of the multidimensional discretization framework are exemplified in Fig. 5. In average, the `layer-off` took 92 sec. of computational time on each run, while the `layer-on` just took 0.01 s. per iteration.

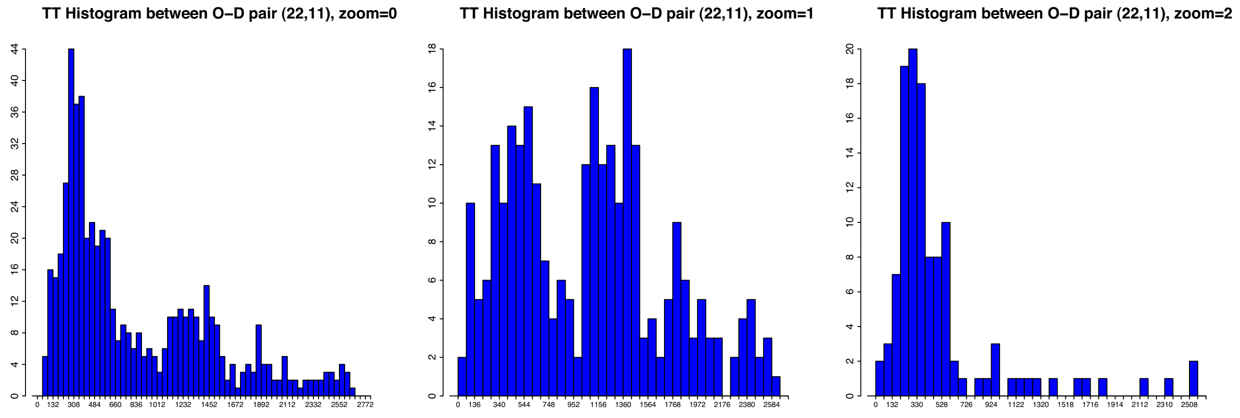
8. Discussion

This section provides a critical overview of our framework. Fisrtly, an overview on the conclusions drawnable from our experimental results are provided. Then the advantages of our methodology regarding related works are discussed, along with its limitations. Finally, an overview of the resulting insights is presented.

8.1. Results analysis

Five main conclusions can be drawn from the results presented: the proposed O-D matrix estimation method is able to discover **dense** ROI. Note the evolution from Fig. 4a to c. The area uncovered in the northwest area is the city's airport. This ROI was initially contained in a vast area, but the density refinement staged uncovered its true shape. However, such refinement is only performed by launching the `layer-off`. This is one of the main drawbacks of this methodology. Setting an adequate periodicity to launch this *layer* prepares the system's ability to react to the formation of highly dense zones. Yet, a high periodicity will largely increase the computational effort in the processing task.

The system is able to maintain a (2) **flexible O-D matrix** over time by updating the **low** levels of memory required. Fig. 4d highlights the framework's flexibility to sudden changes in the cluster's masses.



(a) Zero-Level Bimodal Histogram.

(b) Zoom Level 1 of dimensional discretization.

(c) Zoom Level 2 of dimensional discretization.

Fig. 5. Example of the multidimensional discretization effects of the travel time density function. Note that the zero-level histogram approximates a bimodal p.d.f while the zoom=2 in (C) highlights a unimodal p.d.f. by selecting the trips occurred in 11am–4pm.

Table 3
TTE Prediction Evaluation comparing a Grid-Based City Decomposition and a Mass-Based City Division.

	RMSE	MAE	Average interval width	In interval (%)
Grid-Based	349.33	222.22	466.06	66.54
Mass-Based	306.34	198.66	531.47	79.10

Table 4
Comparison of different online/batch predictive models on TTE.

	RMSE	MAE	Average interval width	In interval (%)
Random forests	307.94	209.89	*Not applicable*	
SVM-linear	321.96	**189.87**	*Not applicable*	
Histogram-Based MaxZoom=0	316.69	210.60	557.38	79.13
Histogram-Based MaxZoom=4	**306.34**	**198.66**	**531.47**	**79.10**

Fig. 4e shows that the algorithm maintains a *logarithmic* space complexity. Note that this complexity is not affected by the `layer-off` launching periodicity.

The mass-based city decomposition (3) **outperforms** the grid-based one. It is not only able to discover equal-mass ROI, but also to maintain equally-sized cells on the O-D matrix. It is not surprising to find that the grid-based histograms are less suitable than the histogram proposed in this paper for TTE (as observed in Table 3). The grid-based simplicity is its best quality as well as a strong drawback. The proposed HS trees are also simple but **data driven**, which strengthens the distribution of data in their leaves.

This incremental approach (4) is more **suitable** than the state-of-the-art batch regression models in the present TTE scenario. Since the models obtained from the training set are not updated using the newly arrived samples, their performance decreases throughout time (see Fig. 4f and Table 4). Even if the SVM-linear presents the lower MAE in Table 4, it is highly questionable to claim that it would be able to maintain such performance, especially if we see its evolution in Fig. 4f. The mean deviation (i.e. $\simeq$ 200sec.) also reflects the stochasticity of the variable, demonstrated in Fig. 3.

It is also important to highlight the histogram's ability to produce accurate **intervals** in the domain of the target variable. The accuracy of these intervals can be partially user-defined by setting a minimum mass ratio, similarly to what was done in these experiments. However, it also depends on the quality of the histograms provided. Table 4 denotes (5) that the multidimensional discretization of the explanatory variables has a considerable effect on the prediction's quality. This reduction of the target variable's variability is explained on the example provided in Fig. 5 (where it is possible to reduce the initial number of modes to just one).

8.2. Advantages

This framework possesses multiple advantages regarding the existing state-of-the-art. Regarding the ROI selection stage, there is no other research work which proposes an incremental method to do it so. The aplicational advantages are clear: the discovery of regions which have a *seasonal* (des)interest – instead of a permanent one – due to a particular event (e.g. a big congress or a large-scale road work). The most difficult issue on doing it so is to reflect such spatial merges/splits along the remaining links of the dimensional chain set in each O-D pair. The histograms, due to their *additivity*, are natural solutions for this problem. The main issue that arise is on the splits, where the histograms are basically split into two equal masses for each bin. The result of this would be the same of assuming that both feature subspaces discovered by such split will follow a probability distribution of similar shapes – which is not necessarily true. However, the arrival of more samples on a (sufficiently) fast rate will easily enforce a convergence to the true distribution – as fast as such samples arrive.

Offline Regression is a problem formulation commonly applied to solve any mobility modeling problem – such as TTE. In this paper, we compared two different types of these methods against our own approach: RF and SVM. The main difference between these two algorithms and our own approach lies on their goal: despite they shared aim, our framework is still an unfinished learning framework – as it requires some type of induction algorithm to be operate over the multiple summarizations provided by the distinct histograms. On this paper, we used a very simplistic approach (Eq. 12) to this issue- our goal was to highlight the predictive power of the incremental approximations established through the distinct feature dimensions by using a very simple method of point forecasting. Hence, other predictive frameworks can be built upon this framework. These possible overlaps are discussed along this subsection.

SVMs (or, more precisely, SVR - Support Vector Regression for this particular application) differ largely than our approach. The typical version of SVR (i.e. offline) aims to find an ϵ-based function which describes the data behavior by guaranteeing that it has, at most, a residual of ϵ for all the examples in the training set while keeping the function as flat as possible. However, the issues of this particular applications is about allowing distinct approximations/solutions for different contexts/feature subspaces. The optimality of SVR main enforce it to ignore samples that deviate largely from the remaining ones – which may represent the majority. Moreover, it requires the apriori selection of a kernel – which the typical techniques may lead to overfit the training data, especially in problems with non-stationary distributions like our own (Jebara, 2004). On the other hand, solutions for incremental Support Vector Learning are still not that popular among practitioners. One of the most known bottlenecks on computing the support vectors incrementally lies on the matrix-based operations (Laskov, Gehl, Krüger, & Müller, 2006). Moreover, the most common solutions for this problems still requires the storage of the entire support vectors in memory during the learning stage – which may be unaffordable for our aplicational case. A possible solution would be to use our discretization trees as model trees (Landwehr, Hall, & Frank, 2005) – where SVMs could be locally trained for finite training sets. However, this hypothesis would require further experiments to be validated.

RF are an approach more similar to our own as they also produce trees trained in feature subspaces. However, these samples subsets are randomly selected – which may enforce that trees trained from samples that explain other non-relevant contexts (e.g. peak-hours versus late evening trips) can have a superior weight on the final decision output. On this way, our tree could be used to set distinct weights of the trees output regarding the context where they are operating, given the estimated apriori probability distributions. Moreover, the main advantage of Random Forests is that the learning process can be easily turned into an incremental one (Gama et al., 2004). Hence, its main drawback facing our own approach is the necessity of having the feature dimensions set beforehand - consequently, novel dimensions cannot be added up during training stage. Again, model trees could be a solution to this problem by enforcing different trees from distinct dimensional chains and subsets, in a meta-learning fashion solution. However, such hypothesis also require some experimental validation before being considered.

Another way of turning any offline model on an online one is by modeling the noise produced by their output. At this level, there were multiple well-known successful applications to the TTE using the popular state-based model Kalman Filter to do it so (Chen et al., 2004; Yu, Yang, Chen, & Yu, 2010; Zaki, Ashour, Zorkany, & Hesham, 2013). However, the Kalman filter bounds the noise to a Gaussian Distribution. Despite it handles most of the typical drifts functional forms, it may be faced as a major limitation as well for this aplicational scenario, where drifts may appear with a very high *slope* (e.g. car accident). Bayesian Inference is also used to accommodate both historical and real-time samples into a single learning framework (typically, from different sources). Indeed, Bayesian statistics can be a good start point to leverage our framework's predictive power by the possibilities that it has on estimating probabilities departing from a certain *belief* (i.e. probability distribution). The author's believe that such theory can provide advances to this framework by merging the resulting estimations on the probability distributions with other type of learners. Yet, this is an open research topic.

8.3. Limitations

Despite its contributions to the estimation of urban dynamics and related problems, the proposed methodology also presents three drawbacks: the aforementioned need to launch the `layer-off` from time to time, the need of have a learning component added on its lose end and the large amount of parameters. A sensitivity analysis was carried out on the most sensible subset of parameters, which strengthens its values. It is claimed that most parameters only have an impact on the granularity or reactiveness of the model. However, the truth is that its setting, even considering some *apriori* parameter fitting methodology, requires some previous experience on this problem. As learning component, we considered simply a point forecasting method based on weighted mean of each bin's mass. However, different applications may require different type of learners. This is still a topic totally open to further research.

It is also important to sustain that this framework does not address the presence of constraints (i.e. the river). This may cause clusters containing the two unconnected river margins to be formed. Fig. 4 exemplifies this undesirable effect, especially for ROI downtown. However, its effects are minimal in this specific study, which happens due to the high number of bridges in these regions (four), and due to their high density levels. To learn more about this topic, go to (Tung, Han, Lakshmanan, & Ng, 2001).

8.4. Overview

This experimental setup considered only probe car data standalone. However, it could consider several data types which could be defined in multiple different dimensions in additional to the typical spatiotemporal ones. To do it so, it just requires to extend the cardinality of Z to accommodate such dimensions. As the split criterion on the multidimensional discretization is related mainly with the number of samples available in each feature subspace, we do not need to constrain the usage of any type of samples depending on their original source. Moreover, we can leverage on such dimensions to achieve context-aware estimations, thus modeling an artificial concept of *neighborhood*. This characteristic is key to increase the reliability of our framework regarding the existing state-of-the-art on this topic.

This framework is applicable and/or adjustable to any urban analysis problem. Yet, it may not present a meaningful contribution to problems where the expected sample rate is large enough to employ batch learning models. Hence, this is not the case of real-time decision support systems, such as the recommendation models. Typically, their ability to produce accurate recommendations for the passenger finding problem depends on the production of **reliable** predictions on some dependent variables, such as the spatiotemporal distribution of the demand (Moreira-Matias et al., 2013b) and the regions' profitability (Yuan, Zheng, Zhang, Xie, & Sun, 2011). The authors want to claim this work as a **straightforward contribution** to maintain statistics of interest and/or induction models about the decision variables of real-time recommendation models on this topic, regardless of their target variable.

9. Final remarks

This application paper proposes a novel technique to maintain statistics regarding the relationships between Regions of Interest (ROI) in a urban area. Its final aim is to approximate scenario-oriented probability distributions through the incremental construction of histograms on multiple and distinct dimensional chains – which correspond to different discretization levels. This methodology presents three straightforward contributions regarding the existing state-of-the-art: (1) it presents an incremental approach to the ROI selection problem through a merge/split schema which is able to propagate itself adequately along the multiple dimensional hierarchies while the state-of-the-art is mainly based on offline clustering methods defined over static timespans; (2) it is able to do the approximation of probability distributions using a fully nonparametric approach. This increases its reactivability and its capacity of forgetting outdated information to provide concepts which are adequately approximate to those who are about to happen in a short-term horizon within each given O-D pair. Such ability is an advance over the two common approaches to this problem: (2-i) basic (parametric) Kalman Filters, which assume that the noise within follows a Gaussian distribution, and (2-ii) Bayesian Inference, which assume a prior belief, independently on how reliable it maybe on the present context. Finally, it is able to (3) accommodate samples from multiple heterogeneous sources arriving with a very high rate. The most remarkable characteristic is that these samples may be defined on distinct dimensions and still, they can be considered on the learning process without discarding any of the relevant information that any of those dimensions may contain. This ability is key to guarantee that any type of suitable induction algorithm that built upon this framework will output values approximate to the real ones. Consequently, this framework presents a solid advance to any Expert and Intelligent system developed to mine urban mobility patterns, independently on its final goal or scope. Experiments conducted in a real-world case study validated its contributions in different aspects of this problem. Its **incrementality** represents a relevant contribution for those interested in inferring the future values of urban dynamic variables in **real-time** using high-speed GPS data streams.

As many other incremental frameworks (Moreira-Matias et al., 2013a), the error introduced by the continuous approximations performed by the different discretization levels make it necessary to maintain an **offline** operator which may be triggered from time to time to reduce the error. The most relevant aspect of the error introduced is the absence of an **online density refinement** of the mass-based clusters obtained through split/merge operations. Density-based spatial clustering algorithms are seen as promising approaches to address this issue. However, it is not possible to confirm if they are directly applicable to this specific context. This framework may also be considered alongside a predictive schema using the state-of-the-art Bayesian Inference, by using the context-aware histograms to model reliable apriori beliefs through probability distributions for time periods where, given any criteria (i.e. a given state such a *obstructed lane*), a stationary distribution is assumed to be in place (which is unknown and not necessarily similar to the apriori one). These problems comprise open research questions.

Acknowledgments

This work was supported by Sibila project “NORTE-07-0124-FEDER-000059”, a project financed by the North Portugal Regional Operational Programme (ON.2 O Novo Norte), under the National Strategic Reference Framework (NSRF), through the European Regional Development Fund (ERDF); by the project “I-City for Future Mobility” (NORTE-07-0124- FEDER-000064); by the Portuguese Funds through the FCT (Portuguese Foundation for Science and Technology) within project FCOMP-01-0124-FEDER-037281 and also by European Commission through the project MAESTRA (Grant number ICT-2013-612944).

References

Barceló, J., Montero, L., Marqués, L., & Carmona, C. (2010). Travel time forecasting and dynamic origin-destination estimation for freeways based on bluetooth traffic monitoring. *Transportation Research Record: Journal of the Transportation Research Board, 2175*, 19–27.

Bentley, J. L. (1975). Multidimensional binary search trees used for associative searching. *Communications of the ACM, 18*(9), 509–517.

Bera, S., & Rao, K. (2011). Estimation of origin-destination matrix from traffic counts: the state of the art. *European Transport / Trasporti Europei, 49*, 3–23.

Bin, Y., Zhongzhen, Y., & Baozhen, Y. (2006). Bus arrival time prediction using support vector machines. *Journal of Intelligent Transportation Systems, 10*(4), 151–158.

Birge, L., & Rozenholc, Y. (2006). How many bins should be put in a regular histogram. *ESAIM: Probability and Statistics, 10*, 24–45.

Bristol City Council (2015). *Open data bristol*. https://opendata.bristol.gov.uk/. Accessed 15.08.15.

Cappé, O., Godsill, S., & Moulines, E. (2007). An overview of existing methods and recent advances in sequential monte carlo. *Proceedings of the IEEE, 95*(5), 899–924.

Carrascal, U. (2012). A review of travel time estimation and forecasting for advanced traveler information systems. Master's thesis. Universidad del Pais Vasco.

Castro, P., Zhang, D., Chen, C., Li, S., & Pan, G. (2013). From taxi GPS traces to social and community dynamics: a survey. *ACM Computing Survey, 46*, 17:1–17:34.

Ceder, A. (2002). Urban transit scheduling: framework, review and examples. *Journal of Urban Planning and Development, 128*(4), 225–244.

Chen, B.-C., Chen, L., Lin, Y., & Ramakrishnan, R. (2005). Prediction cubes. In *Proceedings of the 31st international conference on very large data bases* (pp. 982–993). VLDB Endowment.

Chen, M., Liu, X., Xia, J., & Chien, S. (2004). A dynamic bus-arrival time prediction model based on APC data. *Computer-Aided Civil and Infrastructure Engineering, 19*(5), 364–376.

Chien, S., Ding, Y., & Wei, C. (2002). Dynamic bus arrival time prediction with artificial neural networks. *Journal of Transportation Engineering, 128*(5), 429–438.

CKAN (2015). *Ottawa bus gtfs dataset*. http://data.ottawa.ca/en/dataset/oc-transpo-schedules/resource/c460cbb0-f388-4006-8756-2ef63df60830. Accessed 15.08.15.

Cortes, C., & Vapnik, V. (1995). Support-vector networks. *Machine learning, 20*(3), 273–297.

Djukic, T., van Lint, J., & Hoogendoorn, S. (2012). Application of principal component analysis to predict dynamic origin-destination matrices. *Transportation Research Record: Journal of the Transportation Research Board, 2283*, 81–89.

Domingos, P., & Hulten, G. (2000). Mining high-speed data streams. In *Proceedings of the 6th ACM SIGKDD international conference on knowledge discovery and data mining* (pp. 71–80). ACM.

Domingos, P., & Pazzani, M. (1997). On the optimality of the simple bayesian classifier under zero-one loss. *Machine learning, 29*(2-3), 103–130.

Dublinked (2015). *Dublin bus gtfs data*. http://dublinked.com/datastore/datasets/dataset-254.php. Accessed 15.08.15.

Fraley, C., & Raftery, A. (2002). Model-based clustering, discriminant analysis, and density estimation. *Journal of the American statistical Association, 97*(458), 611–631.

Gama, J. (2010). *Knowledge discovery from data streams*. Chapman and Hall/CRC.

Gama, J., Medas, P., & Rocha, R. (2004). Forest trees for on-line data. In *Proceedings of the 2004 ACM symposium on applied computing* (pp. 632–636). ACM.

Gama, J., & Pinto, C. (2006). Discretization from data streams: applications to histograms and data mining. In *Proceedings of the 2006 ACM symposium on applied computing* (pp. 662–667). ACM.

Ge, Y., Xiong, H., Tuzhilin, A., Xiao, K., Gruteser, M., & Pazzani, M. (2010). An energy-efficient mobile recommender system. In *Proceedings of the 16th ACM SIGKDD international conference on knowledge discovery and data mining* (pp. 899–908). ACM.

Giannotti, F., Nanni, M., Pedreschi, D., Pinelli, F., Renso, C., Rinzivillo, S., & Trasarti, R. (2011). Unveiling the complexity of human mobility by querying and mining massive trajectory data. *The VLDB Journal - The International Journal on Very Large Data Bases, 20*(5), 695–719.

Hazelton, M. (2008). Statistical inference for time varying origin–destination matrices. *Transportation Research Part B: Methodological, 42*(6), 542–552.

Huang, J., Qian, F., Gerber, A., Mao, Z., Sen, S., & Spatscheck, O. (2012). A close examination of performance and power characteristics of 4g lte networks. In *Proceedings of the 10th international conference on mobile systems, applications, and services, MobiSys '12* (pp. 225–238). ACM.

Iqbal, M., Choudhury, C., Wang, P., & González, M. (2014). Development of origin–destination matrices using mobile phone call data. *Transportation Research Part C: Emerging Technologies, 40*, 63–74.

Jebara, T. (2004). Multi-task feature and kernel selection for SVMS. In *Proceedings of the 21st international conference on machine learning* (p. 55). ACM.

Ji, Y., Mishalani, R., & McCord, M. (2015). Transit passenger origin–destination flow estimation: efficiently combining onboard survey and large automaticpassenger count datasets. *Transportation Research Part C: Emerging Technologies, 58*(Part B), 178–192.

Kamga, C., Yazici, M., & Singhal, A. (2013). Hailing in the rain: Temporal and weather-related variations in taxi ridership and taxi demand-supply equilibrium. In *Proceedings of transportation research board 92nd annual meeting*. In *13-3131*.

K. Ting, & Wells, J. (2010). Multi-dimensional mass estimation and mass-based clustering. In *Proceedings of IEEE 10th international conference on data mining (ICDM)* (pp. 511–520). doi:10.1109/ICDM.2010.49.

Kuwahara, M., & Sullivan, E. C. (1987). Estimating origin-destination matrices from roadside survey data. *Transportation Research Part B: Methodological, 21*(3), 233–248.

Landwehr, N., Hall, M., & Frank, E. (2005). Logistic model trees. *Machine Learning, 59*(1-2), 161–205.

Laskov, P., Gehl, C., Krüger, S., & Müller, K. (2006). Incremental support vector learning: Analysis, implementation and applications. *The Journal of Machine Learning Research, 7*, 1909–1936.

Lee, J., Shin, I., & Park, G. (2008). Analysis of the passenger pick-up pattern for taxi location recommendation. In *Proceedings of the 4th international conference on networked computing and advanced information management (NCM'08): 1* (pp. 199–204). IEEE.

Li, B. (2005). Bayesian inference for origin-destination matrices of transport networks using the em algorithm. *Technometrics, 47*(4).

Liu, L., Andris, C., Biderman, A., & Ratti, C. (2009). Uncovering taxi drivers mobility intelligence through his trace. *IEEE Pervasive Computing, 160*, 1–17.

Lu, C., Zhou, X., & Zhang, K. (2013). Dynamic origin–destination demand flow estimation under congested traffic conditions. *Transportation Research Part C: Emerging Technologies, 34*, 16–37.

Mendes-Moreira, J., Jorge, A., de Sousa, J., & Soares, C. (2012). Comparing state-of-the-art regression methods for long term travel time prediction. *Intelligent Data Analysis, 16*(3), 427–449.

Mendes-Moreira, J., Moreira-Matias, L., Gama, J., & de Sousa, J. F. (2015). Validating the coverage of bus schedules: a machine learning approach. *Information Sciences, 293*(0), 299–313.

Moreira-Matias, L., Azevedo, J., Mendes-Moreira, J., Ferreira, M. Gama, J. (2015). The geolink taxi service prediction challenge at ecml/pkdd 2015. https://archive.ics.uci.edu/ml/datasets/Taxi+Service+Trajectory+-+Prediction+Challenge,+ECML+PKDD+2015. Accessed 15.08.15.

Moreira-Matias, L., Gama, J., Ferreira, M., Mendes-Moreira, J., & Damas, L. (2013a). On predicting the taxi-passenger demand: a real-time approach. In *Progress in artificial intelligence*. In *LNCS: 8154* (pp. 54–65). Springer.

Moreira-Matias, L., Gama, J., Ferreira, M., Mendes-Moreira, J., & Damas, L. (2013b). Predicting taxi-passenger demand using streaming data. *IEEE Transactions on Intelligent Transportation Systems, 14*(3), 1393–1402.

Moreira-Matias, L., Gama, J., Mendes-Moreira, J., & de Sousa, J. (2014). An incremental probabilistic model to predict bus bunching in real-time. In *Advances in intelligent data analysis XIII* (pp. 227–238). Springer International Publishing.

Munizaga, M., & Palma, C. (2012). Estimation of a disaggregate multimodal public transport origin–destination matrix from passive smartcard data from Santiago, Chile. *Transportation Research Part C: Emerging Technologies, 24*, 9–18.

Park, J., Murphey, Y., McGee, R., Kristinsson, J., Kuang, M., & Phillips, A. (2014). Intelligent trip modeling for the prediction of an origin–destination traveling speed profile. *IEEE Transactions on Intelligent Transportation Systems, 15*(3), 1039–1053.

Parry, K., & Hazelton, M. L. (2012). Estimation of origin–destination matrices from link counts and sporadic routing data. *Transportation Research Part B: Methodological, 46*(1), 175–188.

Perrakis, K., Karlis, D., Cools, M., & Janssens, D. (2015). Bayesian inference for transportation origin-destination matrices: the poisson-inverse gaussian and other poisson mixtures. *Journal of the Royal Statistical Society: Series A (Statistics in Society), 178*(1), 271–296.

Phithakkitnukoon, S., Veloso, M., Bento, C., Biderman, A., & Ratti, C. (2010). Taxi-aware map: identifying and predicting vacant taxis in the city. *Ambient Intelligence, 6439*, 86–95.

Qi, G., Li, X., Li, S., Pan, G., Wang, Z., & Zhang, D. (2011). Measuring social functions of city regions from large-scale taxi behaviors. In *Proceedings of IEEE international conference on pervasive computing and communications workshops (percom workshops)* (pp. 384–388). IEEE.

R Core Team (2012). R: A language and environment for statistical computing. R Foundation for Statistical Computing. Vienna, Austria. ISBN 3-900051-07-0.

Robusto, C. (1957). The cosine-haversine formula. *The American Mathematical Monthly, 64*(1), 38–40.

Spiess, H. (1987). A maximum likelihood model for estimating origin-destination matrices. *Transportation Research Part B: Methodological, 21*(5), 395–412.

Toledo, T., & Kolechkina, T. (2013). Estimation of dynamic origin-destination matrices using linear assignment matrix approximations. *IEEE Transactions on Intelligent Transportation Systems, 14*(2), 618–626.

Tung, A. K., Han, J., Lakshmanan, L. V., & Ng, R. T. (2001). Constraint-based clustering in large databases. In *Database theory - ICDT* (pp. 405–419). Springer.

U.S. Department of Transportation (2015). *Multimodal data set for portland oregon region test data set for the FHWA connected vehicle initiative real-time data capture and management program*. http://portal.its.pdx.edu/Portal/index.php/fhwa. Accessed 15.08.15.

van der Hurk, E., Kroon, L., Maróti, G., & Vervest, P. (2015). Deduction of passengers' route choices from smart card data. *Intelligent Transportation Systems, IEEE Transactions on, 16*(1), 430–440.

Verbas, İ., Mahmassani, H., & Zhang, K. (2011). Time-dependent origin-destination demand estimation: Challenges and methods for large-scale networks with multiple vehicle classes. *Transportation Research Record: Journal of the Transportation Research Board, 2263*, 45–56.

Yu, B., Yang, Z., Chen, K., & Yu, B. (2010). Hybrid model for prediction of bus arrival times at next station. *Journal of Advanced Transportation, 44*(3), 193–204.

Yuan, J., Zheng, Y., Zhang, L., Xie, X., & Sun, G. (2011). Where to find my next passenger?. In *Proceedings of the 13th ACM international conference on ubiquitous computing (Ubicomp)*.

Yue, Y., Zhuang, Y., Li, Q., & Mao, Q. (2009). Mining time-dependent attractive areas and movement patterns from taxi trajectory data. In *Proceedings of the 17th international conference on geoinformatics* (pp. 1–6). IEEE.

Zaki, M., Ashour, I., Zorkany, M., & Hesham, B. (2013). Online bus arrival time prediction using hybrid neural network and Kalman filter techniques. *International Journal of Modern Engineering Research, 3*(4), 2035–2041.

Zhang, D., Li, N., Zhou, Z.-H., Chen, C., Sun, L., & Li, S. (2011). ibat: detecting anomalous taxi trajectories from GPS traces. In *Proceedings of the 13th international conference on ubiquitous computing* (pp. 99–108). ACM.

Zheng, Y., Liu, Y., Yuan, J., & Xie, X. (2011). Urban computing with taxicabs. In *Proceedings of the 13th international conference on ubiquitous computing*. In *UbiComp '11* (pp. 89–98). ACM.

Zhou, X., & Mahmassani, H. (2006). Dynamic origin-destination demand estimation using automatic vehicle identification data. *IEEE Transactions on Intelligent Transportation Systems, 7*(1), 105–114.