The VLDB Journal
DOI 10.1007/s00778-011-0244-8

SPECIAL ISSUE PAPER

Unveiling the complexity of human mobility by querying and mining massive trajectory data

Fosca Giannotti · Mirco Nanni · Dino Pedreschi · Fabio Pinelli · Chiara Renso · Salvatore Rinzivillo · Roberto Trasarti

Received: 29 September 2010 / Revised: 1 June 2011 / Accepted: 29 June 2011

Abstract The technologies of mobile communications pervade our society and wireless networks sense the movement of people, generating large volumes of mobility data, such as mobile phone call records and Global Positioning System (GPS) tracks. In this work, we illustrate the striking analytical power of massive collections of trajectory data in unveiling the complexity of human mobility. We present the results of a large-scale experiment, based on the detailed trajectories of tens of thousands private cars with on-board GPS receivers, tracked during weeks of ordinary mobile activity. We illustrate the knowledge discovery process that, based on these data, addresses some fundamental questions of mobility analysts: what are the frequent patterns of people's travels? How big attractors and extraordinary events influence mobility? How to predict areas of dense traffic in the near future? How to characterize traffic jams and congestions? We also describe M-Atlas, the querying and mining language and system that makes this analytical process possible, providing the mechanisms to master the complexity of transforming raw GPS tracks into mobility knowledge. M-Atlas is centered onto the concept of a *trajectory*, and the mobility knowledge discovery process can be specified by M-Atlas queries that realize data transformations, data-driven estimation of the parameters of the mining methods, the quality assessment of the obtained results, the quantitative and visual exploration of the discovered behavioral patterns and models, the composition of mined patterns, models and data with further analyses and mining, and the incremental mining strategies to address scalability.

Keywords Spatio-temporal data mining · Trajectories · Mobility patterns · Movement analysis

1 Introduction

The analysis of movement has been fostered by the widespread diffusion of wireless technologies, such as the satellite-enabled Global Positioning System (GPS) and the mobile phone networks. These network infrastructures, as a by-product of their normal operations, allow for sensing and collecting massive repositories of spatio-temporal data, such as the call detail records from mobile phones and the GPS tracks from navigation devices, which represent society-wide proxies of human mobile activities. These big mobility data provide a new powerful social microscope, which may help us understand human mobility, and discover the hidden patterns and models that characterize the trajectories humans follow during their daily activity. This direction of research has recently attracted scientists from diverse disciplines, being not only a major intellectual challenge, but also given its importance in domains such as urban planning, sustainable mobility, transportation engineering, public health, and economic forecasting. The European project GeoPKDD (Geographic Privacy-aware Knowledge Discovery and Delivery [16,18]), started in 2005, is a precursor in mining human mobility data, which developed various analytical and mining methods for spatio-temporal data. On this basis, we show in this paper how to support the complex knowledge discovery process from raw data of individual trajectories up to high-level collective mobility

S. Rinzivillo (✉) · F. Giannotti · M. Nanni · F. Pinelli · C. Renso · R. Trasarti
KDD Lab, ISTI-CNR, Pisa, Italy
e-mail: rinzivillo@isti.cnr.it

D. Pedreschi
KDD Lab, University of Pisa, Pisa, Italy

F. Giannotti · D. Pedreschi
CCNR, Northeastern University, Boston, MA, USA

Published online: 30 July 2011

knowledge, capable of supporting the decisions of mobility and transportation managers, thus revealing the striking analytical power of big mobility data. It should be noted that analysts reason about high-level concepts, such as systematic vs. occasional movement behavior, purpose of a trip, and home-work commuting patterns. Accordingly, the mainstream analytical tools of transportation engineering, such as origin/destination matrices, are based on semantically rich data collected by means of field surveys and interviews. It is therefore not obvious that big, yet raw, mobility data can be used to overcome the limits of surveys, namely their high cost, infrequent periodicity, quick obsolescence, incompleteness, and inaccuracy. On the other extreme, automatically sensed mobility data are ground truth: real mobile activities, faithfully and continuously sampled as they occur, in real time, but clearly without any semantics annotation or context.

The first contribution of this paper is to show how the semantic deficiency of big mobility data can be bridged by their size and precision. To this purpose, we describe the key results obtained on a large-scale experiment conducted with the mobility analysts of the cities of Milan and Pisa, on the basis of real life GPS tracks sensed from tens of thousands private cars. We show how it is possible to find answers to the challenging analytical questions about mobility behavior, which are not supported by the current generation of commercial systems, such as: What are the most popular itineraries followed by people's travels and what is the spatio-temporal distribution of such travels? How do people behave when approaching a key attractor, such as a big station or airport? How do people reach and leave the site of an extraordinary event, such as an important football match? How to predict areas of dense traffic in the near future? How to characterize traffic jams and congestions? More than just examples, these questions are paradigmatic representatives of the analysts' need to disentangle the huge diversity of individual whereabouts and discover the subgroups of travels characterized by some common behavior, or purpose. It is no surprise, then, that finding answers to these questions is beyond the limits of the current generation of commercial systems, and cannot even be accomplished by simply applying known research prototypes, such as the mobility data mining methods developed within GeoPKDD by the authors of this paper [17,27,29] or by other authors [14,24,25,46]. There is a long way to go from raw GPS data to useful representations of mobility behaviors: we need a *mobility knowledge discovery process*.

The second contribution of this paper is to show how to master the complexity of the mobility knowledge discovery process by means of an integrated querying and mining system, centered onto the concept of a *trajectory*, i.e., a sequence of time-stamped locations, sampled from the itinerary of a moving object. The entire analytical process able to create the answers to the high-level questions can be specified as SQL-like queries in our system, which supports the following: the needed data transformations, the data-driven estimation of the parameters of the mining methods adopted, the evaluation of the quality and accuracy of the obtained results, the quantitative and visual exploration of the resulting behavioral patterns and models, the storage of mined patterns and models, the seamless composition of patterns, models and data with further analyses and mining, and the incremental mining strategies needed to overcome the scalability issues that emerge when dealing with big data. We called our system M-Atlas, for *mobility atlas*, to stress that it can be used to create and navigate a comprehensive catalog of the mobility behaviors of a territory. Indeed, all the analyses, both quantitative and visual, presented in this paper were entirely realized within M-Atlas. We present the key design principles underlying M-Atlas, emphasizing its compositionality of querying and mining, and the novel parameter estimation and incremental mining techniques that, as a further contribution, we are introducing in this paper. To this end, we discuss how to realize in M-Atlas some known techniques for empirical estimation of the parameter of density-based trajectory clustering [6] and propose new analogous techniques for trajectory pattern and flock mining. Finally, we show how progressive sampling techniques can be specified, which address effectively the scalability challenges and are essential to achieve the analyses over the GPS data sets analyzed in this paper. To better emphasize this issue, we consider not only the Milan data set, consisting of ≈17,000 cars performing ≈200,000 travels over a week, but also a one-order-of-magnitude larger data set about coastal Tuscany, the region around the city of Pisa, consisting of ≈40,000 cars performing ≈1,500,000 travels over 5 weeks. From our collaboration with a mobility agency, we learned that the most interesting and challenging analytical questions about mobility (that are not supported by the current generation of commercial systems) are exactly aimed at *discovering interesting subgroups of vehicles and travels characterized by some common movement behavior*. To perform this kind of analysis, a complete querying, analysis and mining system is needed, able to support the overall knowledge discovery process centered around the trajectory concept.

Plan of the paper follows. Section 2 presents some statistics that validate the GPS data sets used in the experiments and introduces the mobility questions that drove the analysis through the paper. Then, Sect. 3 introduces the design principles of the data mining query language of the M-Atlas system. In Sect. 4, we show how the data mining query language can be practically used to build complex knowledge discovery processes on mobility data. Afterward, Sect. 5 exposes the experiments we have carried out using M-Atlas on two different GPS data sets that answer the mobility questions. Section 6 illustrates the system architecture and summarizes the performance evaluation. The essential literature review is reported in Sect. 7. Finally, Sect. 8 draws conclusions and highlight the future developments.

2 GPS data as a microscope of urban mobility

We concentrate in this paper on massive real-life GPS data sets, obtained from tens of thousands private vehicles with on-board GPS receivers. The owners of these cars are subscribers of a *pay-as-you-drive* car insurance contract, under which the tracked trajectories of each vehicle are periodically sent (through the GSM network) to a central server for anti-fraud and anti-theft purposes. This data set has been donated for research purposes by *Octo Telematics Italia S.r.l* [31], the leader for this sector in Europe. We use two GPS data sets: the first, *Milano2007*, is about ≈17,000 cars tracked during one week (from April 1st through April 7, 2007) of ordinary mobile activity in the urban area of the city of Milan (a 20 km × 20 km square). The second, *Pisa2010*, is about ≈40,000 cars tracked during 5 weeks (from June 14th through July 18, 2011) in coastal Tuscany, a 100 km × 100 km square centered on the city of Pisa.

The average sampling rate of the GPS receivers is 30 s. Globally, Milano2007 consists of ≈2 Million observations and Pisa2010 of ≈20 Million observations, each consisting of a quadruple $(id, lat, long, t)$, where id is the car identifier, $(lat, long)$ are the spatial coordinates, and t is the time of the observation. The car identifiers are pseudonymized, in order to achieve a basic level of anonymity.[1] The resolution of the spatial coordinates is at 10^{-6} degrees, and the error of the positioning system is estimated at 10–20 m in normal conditions. The temporal resolution is in seconds. All the observations of the same car id over the entire observation period are chained together in increasing temporal order into a global *trajectory* of car id. The global trajectory is then split into several sub-trajectories, corresponding to *trips* or *travels*, by using a cut-off threshold of 30 min: if the time interval between two subsequent observations of the car is larger than 30 min, the first observation is considered as the end of a travel and the second observation is considered as the start of another travel; using this reconstruction procedure, we obtained ≈200,000 different travels in Milano2007 and ≈1,500,000 different travels in Pisa2010.

2.1 Comparison with survey data

In order to asses the significance of this data set as a proxy of the real mobility phenomena within a metropolitan area of 2 million inhabitants, we compared the Milano2007 data set against the survey data (MilanoSurvey) collected in 2005-2006 by the mobility agency of Milano municipality,[2] which are used to produce a periodic mobility report [3]. An important aspect to be considered in this comparison is that both the sample population and the form of collected data are different. First, the Milano2007 data set covers only vehicular movements, whereas MilanoSurvey includes public transportation and pedestrians. Second, the automatic collection procedure applied for GPS data ensures that all movements are correctly captured, whereas surveys leave space to omissions or distortions. Finally, GPS data provide no explicit semantic information about the purpose of movements, the final destination, and profiles of the citizens involved, whereas surveys explicitly collect this information. Significant differences hold also for the mere size of the sample: 17,000 vehicles versus 45,000 vehicles and 210,000 physical persons covered by the survey, although the number of GPS-equipped cars is continuously increasing (today, more than 50,000 cars are sensed on the same area in one week). Concerning the periodicity of the sample, the difference is striking: near real time for GPS tracks vs years for the surveys: MilanoSurvey is conducted every 5 years. Finally, GPS data are produced at a very low cost as a by-product of a sensing infrastructure which is operational for the car insurance industry, while surveys require large ad hoc investments.

In our assessment of the Milano2007 data set, we replicated a set of statistics published in MilanoSurvey; the comparison has been carried out by analyzing the distribution of movements and presence of people, and the obtained results, as discussed below, bring strong evidence to the validity and coherence of GPS data. An important outcome of this experiment is that GPS data contain detailed information about occasional (as opposed to systematic) mobility, an important trait of reality, which is known to be underestimated by surveys.

Movement distribution: We measured the number of moving vehicles in every hour of the day and created a histogram over the entire week. The result is shown in Fig. 1.

The two distributions match significantly, especially for the days from the second to the fifth of the week, that actually represent *regular* working days, from Monday to Thursday. Friday, April 6, is Easter Friday, which explains the significant difference in the shape of the distribution w.r.t. previous weekdays. Within working days, the most relevant deviation from the survey data is a higher volume of movements between the two peaks in the rush hours and (to a minor extent) the later part of the day. Actually, the assessment with the Mobility Agency revealed not only that the results are coherent, but also that the survey distribution is known to underestimate the movements where the mismatch

[1] It is well know that de-identification with pseudonyms offers a very weak protection of anonymity (see, e.g., [28]); for this purpose, M-Atlas offers primitive for trajectory anonymity [1,28].

[2] AMA—http://www.ama-mi.it/english.

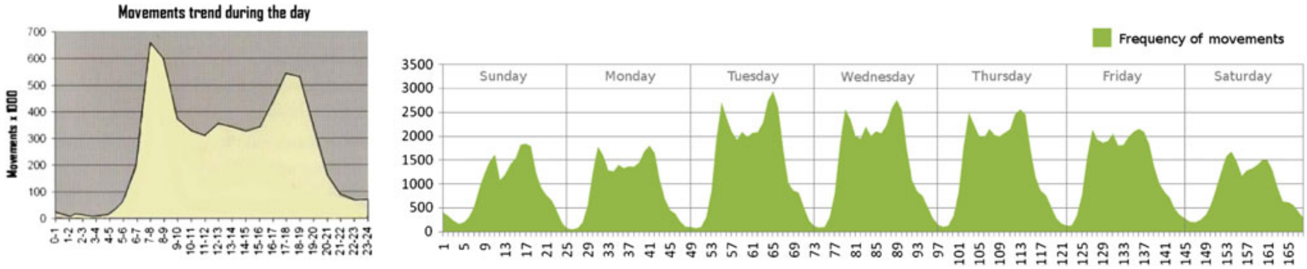


Fig. 1 Movement distribution by hour: representative weekday in MilanoSurvey (*left*) and entire week in Milano2007 (*right*)

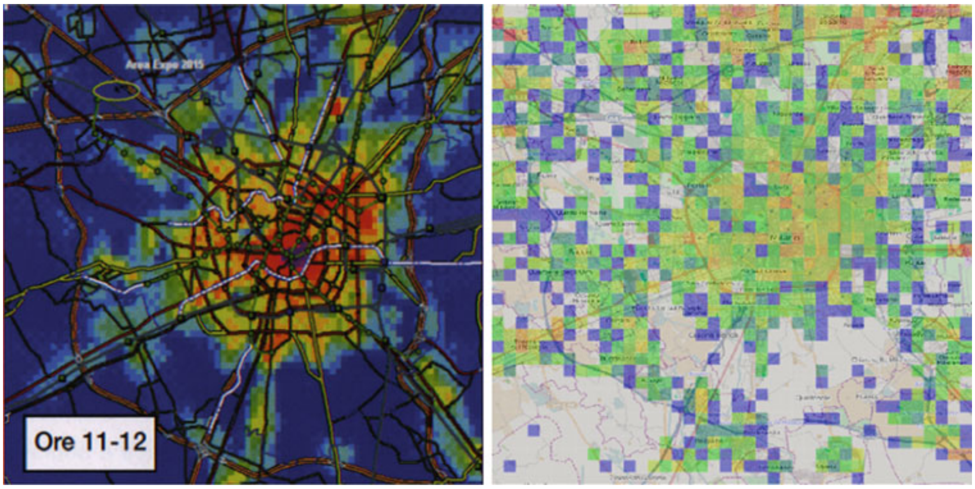


Fig. 2 Presence distribution between 11am and noon, survey (*left*), GPS data (*right*); frequent locations plotted in *red*, less frequent locations plotted in *green*, infrequent locations in *violet-blue*

occurs. The explanation of this phenomenon is that GPS data also capture nonsystematic movements, while survey data do not, as interviewed people tend not to report their occasional mobility, such as going to the dentist or visiting a friend. Also, GPS data contain mobile activity of people that do not live in the greater metropolitan area, while the survey focuses on Milano residents.

Presence distribution: We measured the number of people present and stationary (not moving) at the various locations at every hour of the day, as reported in Fig. 2(left) for a specific time slot. A similar estimate was obtained on Milano2007 by (i) partitioning the space into a regular grid and (ii) counting for each cell the number of vehicles that were stationary in the cell for each time interval. Such values were averaged over all (regular) working days available. Fig. 2(right) shows the results.

The two distributions match well in most locations, including some particular areas along main streets and suburban residential areas, confirming again the coherence of results obtained with survey and mobility data. The main deviation occurs in the inner city center, where a high-density spot found by surveys is significantly lower in Milano2007: this is explained by the strong access restrictions to private cars in the city center, as well as by the limited capacity of roads and traffic, which causes an underrepresentation in the GPS data of the people that reach their workplaces in the center with public transportation.

2.2 Basic statistics

We measured some basic quantities describing the travels represented in the trajectory data sets: the length of a trip, the duration of a trip, the correlation of length and speed of trips, the radius of gyration of a vehicle (the average distance of a vehicle from its most likely location), and the density of (moving and stationary) vehicles in space and time.

Trip length and duration: Figure 3(left) shows the distribution of trip length (in km), where the length $l(T)$ of a trip $T = \langle (x_0, y_0, t_0), \ldots, (x_n, y_n, t_n) \rangle$ is estimated by the formula $\sum_{i=1,n} \delta((x_{i-1}, y_{i-1}), (x_i, y_i))$; here, δ denotes Euclidean distance. The heavy-tailed distribution of trip length highlights how there are many short trips of a few kilometers, and few, but non negligible very long trips of tens or even hundreds of kilometers; a similar consideration applies to the distribution of trip duration (i.e., $t_n - t_0$), shown in Fig. 3(right). The lesson learned here confirms how mobility is a complex phenomenon that cannot be characterized by any simple notion of *average behavior*. The skewed distributions indicate a huge variability and heterogeneity of trips, spanning over 3-4 orders of magnitude of duration and length:

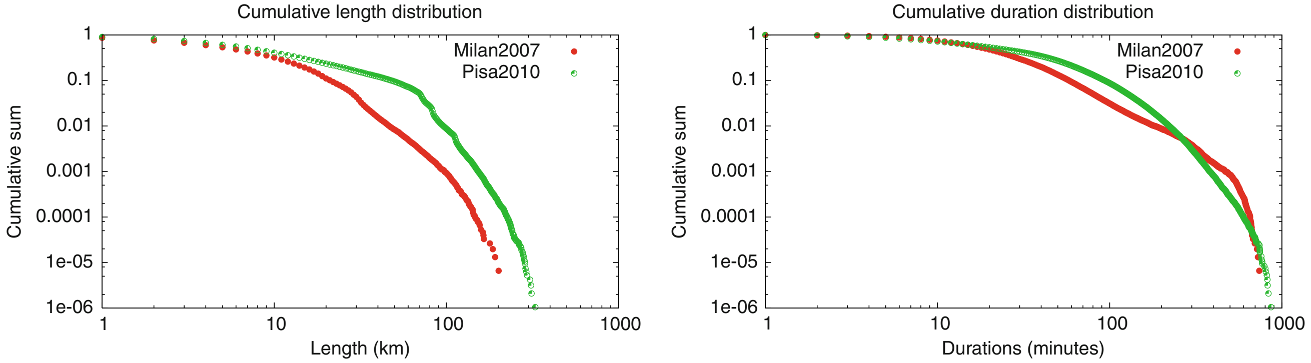


Fig. 3 Trip length cumulative distribution in log–log scale (*left*), trip duration cumulative distribution in log–log scale (*right*). *Red lines* for Milano2007 and *green lines* for Pisa2010

novel analytic methods are needed to disentangle such complexity.

Correlation of length and speed of trips: Figure 4 shows the correlation plots of average trip length (in km) and speed (in km/h) for both the data sets. Each plot, for each speed value s, reports the average distance traveled by all trips with average speed (in s). In the Milan2007 data set, the plot shows how the distance traveled grows linearly with speed, as expected, only up to 80 km/h, while it decreases for higher speed. In the Pisa2010 data set, the distance traveled grows linearly up to 110 km/h, with a low slope between 20 and 40 km/h. The plots show also the number of trips for each speed value: the high diversity of lengths for speeds beyond 130 km/h (the highest speed limit in Italy) is due to the low number of travels with that velocity and can be considered as noise, coherently with the intuition that very fast trips take place in particular situation of light traffic, typically at night.

Radius of gyration: Figure 5(left) shows how the movements of a typical trajectory insist over a preferred location, most likely the home place or the work place of the vehicle's owner. The radius of gyration of each vehicle can be hence computed as its average distance from the preferred location, or center of mass.

Given the entire trajectory $T = \langle (x_0, y_0, t_0), \ldots, (x_n, y_n, t_n)\rangle$ of a specific vehicle, its center of mass is defined as $cm(T) = (\frac{1}{n}\sum_{i=0,n} x_i, \frac{1}{n}\sum_{i=0,n} y_i)$ and its radius of gyration is $rg(T) = \sqrt{\frac{1}{n}\sum_{i=0,n} \delta((x_i, y_i) - cm(T))^2}$. Figure 5(right) has been created computing the radius of gyration of each vehicle and represents the general law of the power of attraction of the most likely location on each individual, confirming the results obtained in [19].

Spatio-temporal analysis of density: Figure 6 illustrates the distribution of vehicles in the urban area in three different time slots; space has been discretized into rectangular grids and time into regular intervals. Not surprisingly, density increases in rush hours.

Penetration of GPS-enabled vehicles: Figure 7 shows the correlation between the resident population and the number of tracked cars in Milano2007 and Pisa2010. The number of resident people in both the regions has been provided by the Italian Institute for Statistics (ISTAT) census data. The GPS-enabled vehicles have been partitioned into residential, i.e., belonging to people who spend regularly the night in their preferred location within the areas covered by the two data sets and visitors. We observe an evident correlation between residential tracked cars and general population. Also, we get an experimental confirmation that GPS-enabled cars are about 1% of population in Pisa2010 and 0.25% in Milan2007. Considering only the registered cars, Pisa2010 represents the 2% and Milano2007 the 0.5%.

2.3 Analysis of movement behavior

Besides convincing ourselves that the Milano2007 data are a valuable proxy of real mobility at the urban scale, we learned two lessons from our basic analytical explorations. First, all statistics confirmed that there is a huge complexity represented in the data, a wide variability of individual mobility behaviors that cannot be fully understood in its diversity by looking only at macroscopic, global measures and laws. Second, we realized that the basic spatio-temporal statistics are not well suited to support the discovery and analysis of *movement patterns*, because the very nature of a trajectory—a time-stamped sequence of spatial locations—is factored out by the basic statistics.

Collaborating with the analysts of the Milano mobility agency, we learned that the most interesting and challenging analytical questions about mobility (that are not supported by the current generation of commercial systems) are exactly aimed at *discovering interesting subgroups of vehicles and travels characterized by some common movement behavior*. Five paradigmatic questions of this kind are the following.

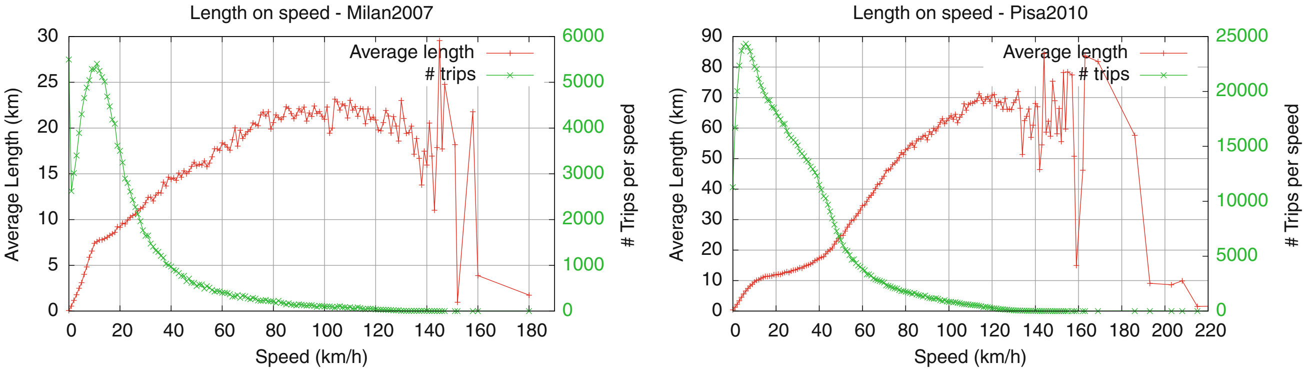


Fig. 4 Correlation plot of length and average speed of trips and number of trips per speed for the Milan2007 (*left*) and Pisa2010 (*right*)

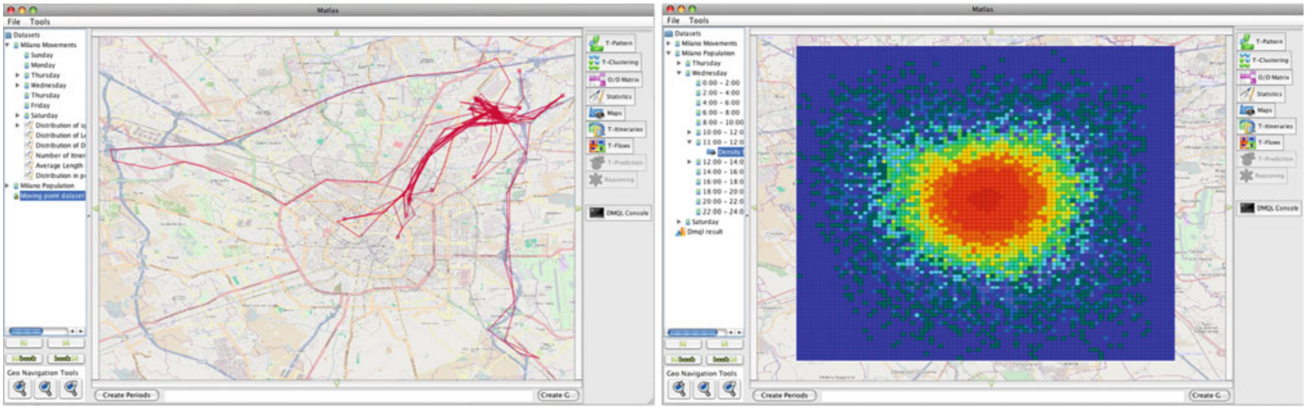


Fig. 5 The complete weekly trajectory of a single vehicle; its most likely location emerges clearly (*left*); plot over a regular grid of the probability of finding a user in a location, normalized in each vehicles intrinsic reference system (*right*)

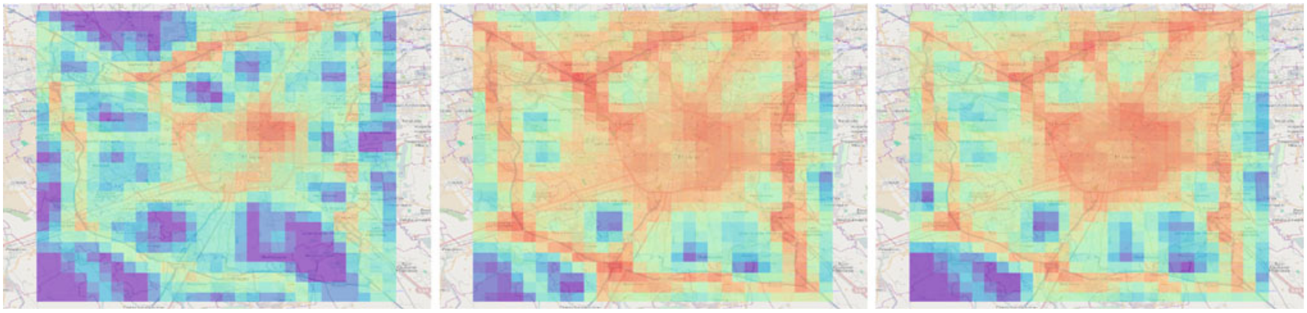


Fig. 6 Aggregated density moving vehicles from midnight to 2 am (*left*); from 6 am to 8 am (*center*); from 6 pm to 8 pm (*right*)

1. What are the most popular itineraries followed from the origin to the destination of people's travels? What routes, what timing, what volume for each such itinerary? How do people leave the city toward suburban areas (or vice-versa)? What is the spatio-temporal distribution of such trips?
2. How to understand the accessibility to key mobility attractors, such as large facilities, railway stations or airports? How do people behave when approaching an attractor?
3. How to detect an extraordinary event and understand the associated mobility behavior? How and when do people reach and leave the event's location? What is the spatio-temporal distribution of such (portion of) trips?
4. What will be the areas with highest traffic volume in the next hour(s)? To what extent are our predictions accurate?
5. How to characterize a traffic jam? How to detect where and when traffic jams occur?

To answer these questions, a complete querying, analysis and mining system is needed, able to support the overall knowledge discovery process centered around the trajectory concept. Such a system is expected to master all the phases

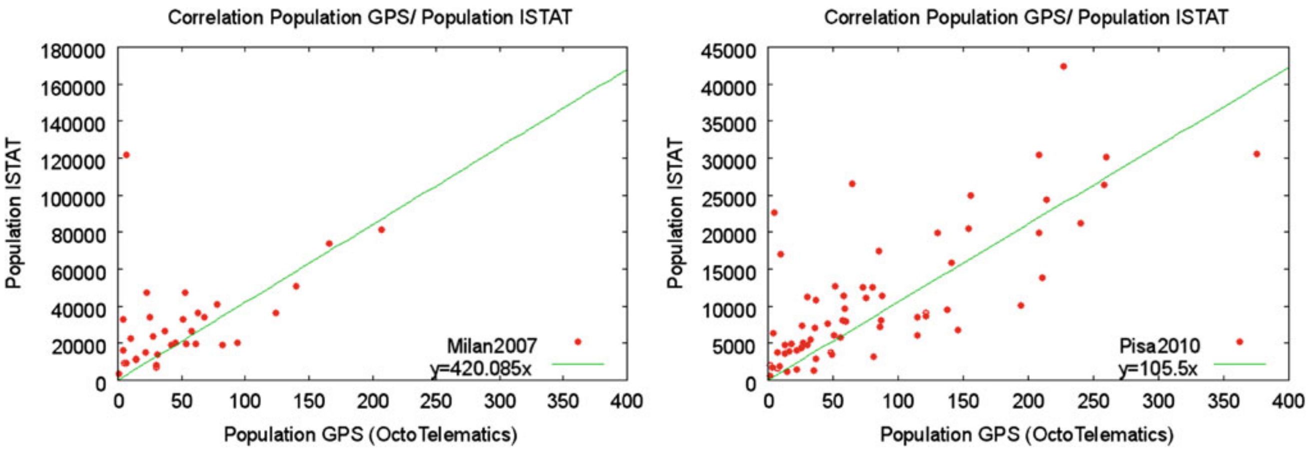


Fig. 7 Correlation of GPS-enabled vehicles with resident population in Milan2007 (*left*) and Pisa2010 (*right*)

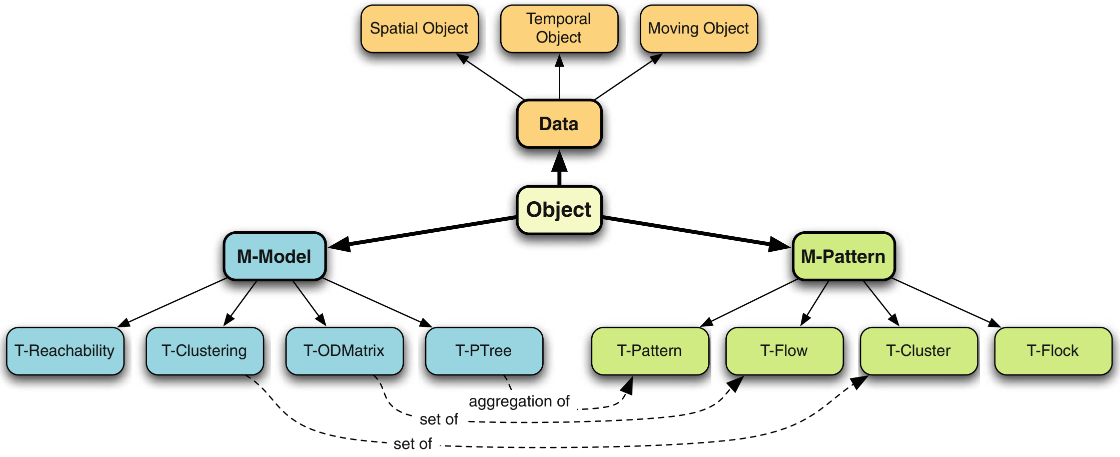


Fig. 8 The M-Atlas type hierarchy. M-Model, M-Pattern and Data are the basic types of data. We can notice the relationship between M-Models and M-Patterns. For example, T-Clustering model is represented by a set of T-Cluster patterns, while T-PTree model is an aggregation of T-Patterns

of such process, to the aim of supporting interactive, iterative visual exploration of the analytical results, thus enabling the analyst to combine different forms of knowledge and drive the analysis toward the discovery of interesting movement patterns.

This ambitious goal is precisely what we pursue with M-Atlas, initially designed and developed within the GeoPKDD project [18,39,40] and continuously expanded with new mobility mining features. In fact, all the analyses presented so far were entirely performed using M-Atlas; in Sect. 5, we will show how M-Atlas is able to provide answers to the questions above, using the ideas and methods of mobility data mining and their integration into a logically coherent querying and mining framework—but, before that, we need to describe the design principles of M-Atlas and their realization into a usable and robust system.

3 Design principles of M-Atlas

M-Atlas[3] is a mobility data mining query language, i.e., a querying and mining system centered onto the concept of trajectory. Besides the mechanisms for storing and querying trajectory data, M-Atlas has mechanisms for mining trajectory patterns and models that, in turn, can be stored and queried. The basic design choice is *compositionality*, i.e., querying and mining of trajectory data, patterns and models may be freely combined, in order to provide the expressive power needed to master the complexity of the mobility knowledge discovery process. The formal compositional framework underlying M-Atlas has been defined in [33,40] and is referred to as the Two-Worlds model. This model views

[3] Available for download at the URL: http://m-atlas.eu.

the knowledge discovery process as the interaction between two worlds: the data world and the model world. The former is a database of entities, trajectories in our case; the latter is a database of models and patterns extracted from the data, representing the result of mining tasks. Two kinds of operators connect the two worlds: the mining operators and the entailment operators. Mining operators map data into models or patterns, while entailment operators map models, patterns, and data into the data that satisfy the property expressed in the given models or patterns. This view supports compositionality, in that data can be mapped onto models and vice versa, and is coherent with inductive databases [22]. Another design choice in the Two-Worlds model is that all entities are represented in the object-relational data model, which is more suitable to tackle the structural complexity of spatio-temporal data wrt. tabular data.

Architecturally, M-Atlas has three high-level components: (i) a persistent store for trajectory data, models, and patterns, (ii) a spatio-temporal query language for trajectory data, models, and patterns, and (iii) a repertoire of constructors of spatio-temporal models and patterns.

3.1 Data, models, and patterns

M-Atlas adopts state-of-the-art moving object database design principles for its trajectory store, extended with mechanisms for managing and querying models and patterns. There are three main object types in M-Atlas: Data, M-model, and M-pattern depicted in Fig. 8. We distinguish between *models* and *patterns*: a pattern is a representation of a local property that holds over a sub-group of mobility data, e.g., a flock of trajectories; on the other hand, a model is a representation of a global property that holds over an entire data set: accordingly, a model is either a global aggregate (e.g., speed distribution in a trajectory data set) or a collection of patterns (e.g., the clustering that partitions an entire data set into separate clusters).

Practically, the system adds new object-relational types to the database in order to represent the new types of data, patterns, and models. The advantage of having an object-relational representation is threefold: (i) it allows the definition of complex data such as lists and trees, (ii) yields a compact representation of the data, and (iii) makes it possible to use classical indexing techniques already in the database on complex objects.

3.1.1 Data types

M-Atlas supports three types of data: purely spatial data, purely temporal data, and moving points or trajectories.

Spatial objects have a geometric shape and a position in space and are represented as $S = (type, < p_1, \ldots, p_n >)$ where $type \in \{point, line, polygon\}$ defines the meaning of the list of points $< p_1, \ldots, p_n >$: if $type = point$, then the list is composed by only one point with its coordinates; if $type = line$, then the list represents a broken line; if $type = polygon$, then the list represents the contour of the polygon.

Temporal objects are represented as $T = (t, d)$ where t is an absolute temporal value (w.r.t. a time reference system) and d is a duration expressed in seconds. When t is equal to the special value *null*, then the temporal object represents a relative time period. An *interval* object is a pair of temporal objects $I = (T_{min}, T_{max})$.

Moving objects are the spatio-temporal evolution of the position of a spatial object. There are three different types of moving objects: moving point, moving line, and moving polygon. In this paper, we concentrate on moving points, which represent trajectories. A moving point is defined as $Mo =< p_1, t_1 >, \ldots, < p_n, t_n >$, where p_j is a spatial object representing a point, t_j is a temporal object representing an absolute time point, and $t_i < t_j$ for $1 \leq i < j \leq n$. To the purpose of this paper, the terms *trajectory* and *moving point* are synonyms.

Data Constructors can be associated with each data type, allowing, e.g., to construct data objects by acquiring and preprocessing raw data. As an example, the following construction query builds a table *Travels* of reconstructed travels from the raw observations contained in the table *RawData*. By setting a maximum space gap (in km) and time gap (in seconds) between any two consecutive observations in a trajectory, we can specify the end of a travel and the beginning of a new one.

```
CREATE DATA Travels BUILDING MOVING_POINTS
FROM (SELECT userid,lon,lat,datetime FROM RawData
   ORDER BY userid,datetime)
SET MOVING_POINT.MAX_SPACE_GAP = 0.2 AND
  MOVING_POINT.MAX_TIME_GAP = 1800
```

3.1.2 M-Pattern Types

A mobility pattern, M-Pattern in short, represents the common behavior of a (sub-)group of trajectories, obtained as a result of a data mining algorithm. The types of M-Patterns currently supported by M-Atlas are shown in Fig. 9.

T-Cluster: A T-Cluster (Fig. 9a) is defined as a set $S = \{(\tau_1, l), (\tau_2, l), \ldots\}$ of labeled trajectories, which share the same membership tag l. The trajectories of a T-Cluster are grouped on the basis of their similarity according to a specified similarity function, chosen from a repertoire of possible choices.

T-Pattern: it is represented as $tp = (R, T, s)$ where $R =< r_0, \ldots, r_k >$ is a sequence of regions, $T =< t_1, \ldots, t_k >$ is a

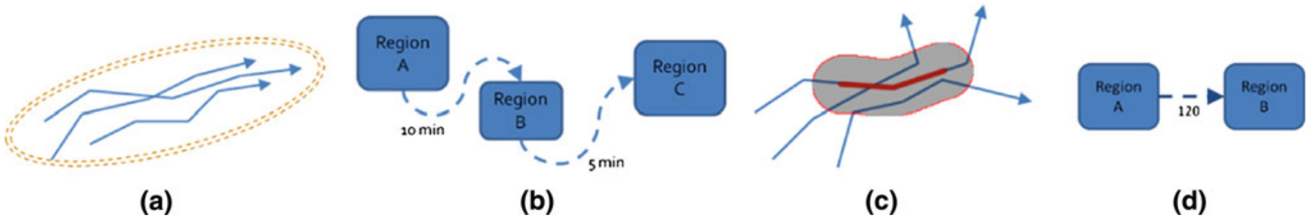


Fig. 9 M-Pattern types: **a** T-Cluster, **b** T-Pattern, **c** T-Flock, **d** T-Flow

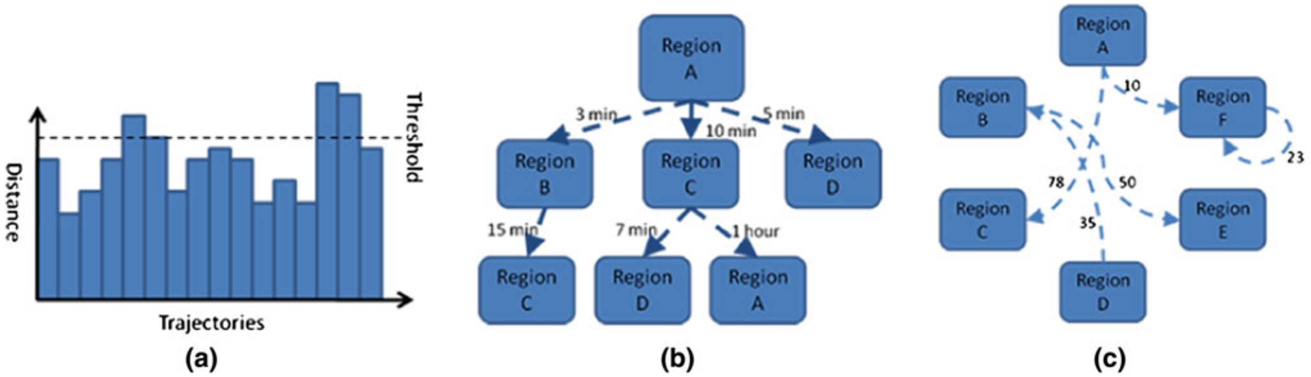


Fig. 10 M-Models types: **a** Reachability plot, **b** T-PTree and **c** T-ODMatrix

sequence of relative time intervals $t_j = [t_j^s, t_j^e]$ associated to each region, and s is the support of tp, i.e., the number of trajectories that are compatible with tp in space and time. Informally, a T-Pattern can be represented as $r_0 \xrightarrow{t_1} r_1 \cdots \xrightarrow{t_k} r_k$. Originally introduced in [17], a T-Pattern (Fig. 9b) is a concise description of frequent behaviors, in terms of both space (i.e., the regions of space visited during movements) and time (i.e., the duration of movements).

T-Flock: A T-Flock $f = (I, r, b)$ represents a spatio-temporal coincidence of a group of moving points, where $I = [t_{min}, t_{max}]$ is the time interval of the coincidence, b is the base moving point, and r is the spatial buffer around b which is used to determine the coincidence. This spatio-temporal coincidence defines a common behavior of the people which move together for a certain time interval (Fig. 9c).

T-Flow: The T-Flow $tf =< R_1, R_2, w >$ represents a flow of $w \geq 0$ trajectories which move from region R_1 to region R_2 (Fig. 9d).

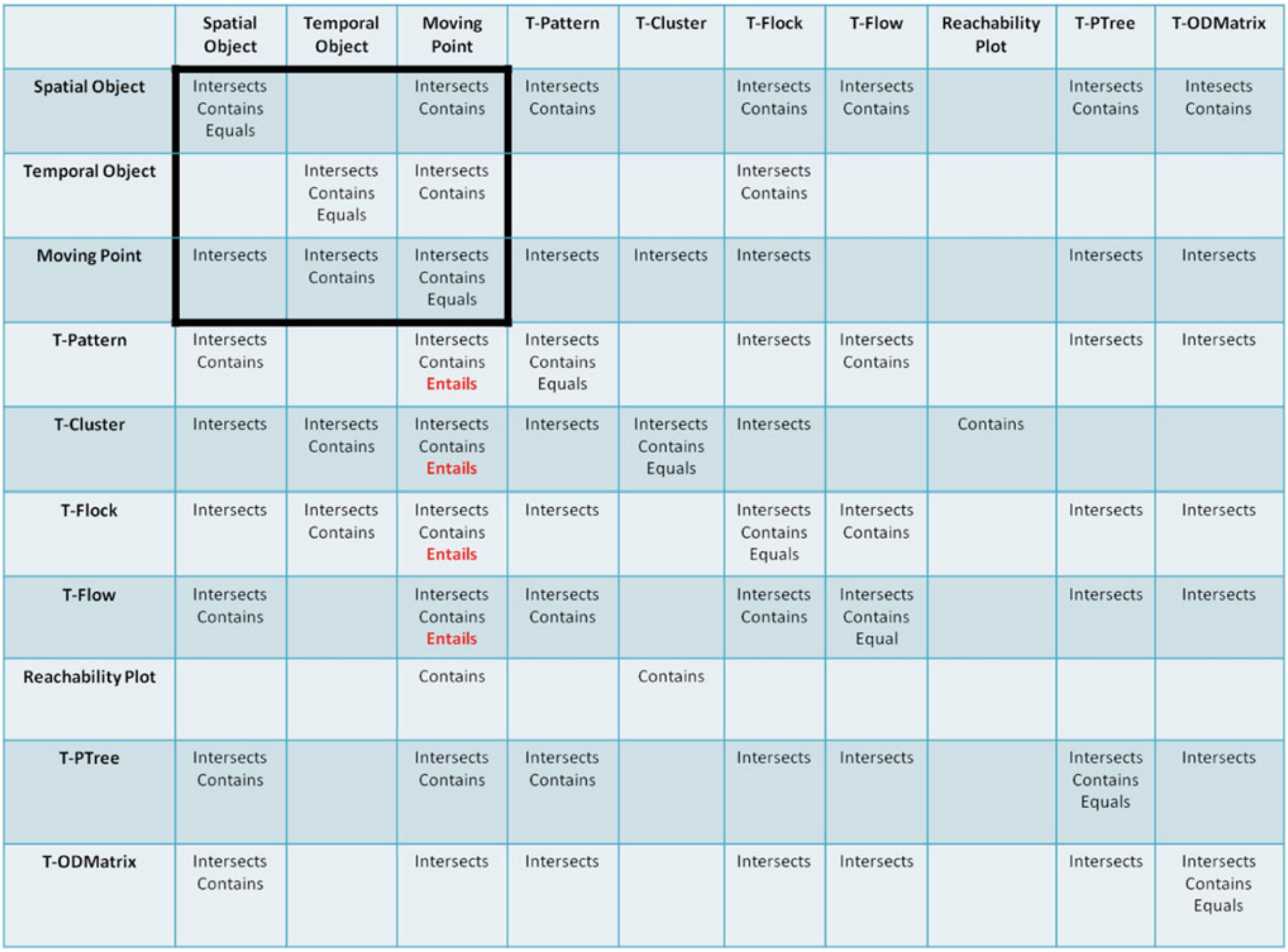
3.1.3 M-Model Types

Mobility models, M-Models in short, are the global models extracted by a data mining algorithm, where the adjective *global* indicates the fact that each such model describes the entire input data set. Figure 10 illustrates some of the available M-models in M-Atlas; other M-Models are simply the entire collection of T-Patterns, T-Clusters, and T-Flocks mined over a trajectory data set.

Reachability plot: is a histogram of distances between trajectories, obtained considering a specific distance function (Fig. 10a). More precisely, it is a sequence of pairs $Rp = < (t_1, d_1) \ldots (t_n, d_n)) >$ where t_j is a trajectory and d_j is the distance between t_j and t_{j+1}, where t_{j+1} is the nearest neighbor of t_j which does not occur in $\{t_1, \ldots, t_j\}$. Using a threshold ϵ for distance, the reachability plot identifies a set of T-Clusters representing the partition of the whole data set into labeled groups of similar trajectories.

T-PTree. A T-Pattern Tree, T-PTree in short, is a compact representation of a set of T-Patterns (Fig. 10b). It is a prefix tree $PT = \{root, N, E\}$, where N is the set of nodes of the tree, E is the set of edges, and $root$ is the root of the tree. Each node $n_i = \{r, supp\}$ contains a spatial region r and a support value $supp$; each edge $e_{i,j} = \{t_{min}, t_{max}\}$ connects the nodes i and j specifying a relative time interval. The support label on the nodes represents the maximum support value of the T-Patterns which have the path $root, \ldots, n_i$ as prefix. The formal definition of prefix of a T-Pattern is in [27]; intuitively, a T-pattern tp_1 is prefix of another T-Pattern tp_2 if every region and interval of the first pattern are included in the region and interval of the second, in the specified order.

T-O/DMatrix: A T-O/DMatrix (Fig. 10c) is defined as a labeled graph $odm = \{O, D, E\}$ where $O = \{o_1 \ldots o_n\}$ are the nodes which identify the origins, $D = \{d_1 \ldots d_k\}$ are the nodes which identify the destinations, and E are the



	Spatial Object	Temporal Object	Moving Point	T-Pattern	T-Cluster	T-Flock	T-Flow	Reachability Plot	T-PTree	T-ODMatrix
Spatial Object	Intersects Contains Equals		Intersects Contains	Intersects Contains		Intersects Contains	Intersects Contains		Intersects Contains	Intesects Contains
Temporal Object		Intersects Contains Equals	Intersects Contains			Intersects Contains				
Moving Point	Intersects	Intersects Contains	Intersects Contains Equals	Intersects	Intersects	Intersects			Intersects	Intersects
T-Pattern	Intersects Contains		Intersects Contains Entails	Intersects Contains Equals		Intersects	Intersects Contains		Intersects	Intersects
T-Cluster	Intersects	Intersects Contains	Intersects Contains Entails	Intersects	Intersects Contains Equals	Intersects		Contains		
T-Flock	Intersects	Intersects Contains	Intersects Contains Entails	Intersects		Intersects Contains Equals	Intersects Contains		Intersects	Intersects
T-Flow	Intersects Contains		Intersects Contains Entails	Intersects Contains		Intersects Contains	Intersects Contains Equal		Intersects	Intersects
Reachability Plot			Contains		Contains					
T-PTree	Intersects Contains		Intersects Contains	Intersects Contains		Intersects	Intersects		Intersects Contains Equals	Intersects
T-ODMatrix	Intersects Contains		Intersects	Intersects		Intersects	Intersects		Intersects	Intersects Contains Equals

Fig. 11 M-Atlas spatio-temporal primitives

edges which connect an origin node with a destination node. Each node (both origins and destinations) contains a spatial region and the label on the edges represent the number of movements which start in the origin region and end in destination node. This model results from the composition of a set of T-Flows, each representing the trajectories from the origin to the destination region.

Model and Pattern constructors: A generic constructor for M-Models (and M-Patterns) is defined as a function $T_d \rightarrow (T_m, T_p)$ where T_d is a data table, T_m is a model table (containing a single M-Model object), and T_p is a table containing a set of M-Patterns objects. This operator realizes the construction of M-Models and M-Patterns through the execution of a data mining method with a specified parameter setting. M-Atlas provides a mining constructor for each method in its *data mining library*, presented in Sect. 3.3. An example of mining constructor query is the following, which generates a step of density-based trajectory clusters under specific parameters:

```
CREATE MODEL ClusteringTable MINE AS T-CLUSTERING
FROM (Select t.id, t.trajobj from TrajectoryTable t)
SET T-CLUSTERING.FUNCTION = ROUTE_SIMILARITY AND
  T-CLUSTERING.EPS = 100 AND
  T-CLUSTERING.MIN_PTS = 20
```

3.2 Spatio-temporal query primitives

The querying primitives over data, models, and patterns are summarized in Fig. 11; the upper left square contains the *data* × *data* primitives, corresponding to the classical spatio-temporal primitives defined in [21]. All the other primitives have been specifically designed for M-Atlas, in that they involve models and patterns (*data* × *model/pattern*, *model/pattern* × *data*, or *model/pattern* × *model/pattern*).

Each primitive is defined as a function $r(T_1, T_2) \rightarrow (T_{rel})$, where T_1 and T_2 are two sets of objects and $T_{rel} = \{\langle o_1, o_2\rangle | o_1 \in T_1 \wedge o_2 \in T_2 \wedge rel(o_1, o_2)\}$. Here, rel is a predicate defined between the types of objects in T_1 and T_2, which specifies the relation that should hold over the pairs of objects that are kept in the resulting table T_{rel}.

Albeit there are apparently only a few kinds of spatio-temporal primitives (*contains*, *intersects*, *equals*), a large variety comes from the different combinations of types of objects to which such primitives are applied, as illustrated in Fig. 11. Each combination depends on the semantics of movement represented by the types of the involved objects; for instance, the definition of *intersects* between a T-pattern and a Moving Point is completely different from that between a T-Flock and a Moving point. The expressive power of M-Atlas derives exactly from the comprehensive repertoire of spatio-temporal primitives over all combinations of data, patterns, and models; the entire repertoire is reported in [39].

A *pattern* × *pattern* primitive is the *contains* relation between two T-Patterns $tp^1 = (R^1, T^1, s^1)$ and $tp^2 = (R^2, T^2, s^2)$, defined as follows:

$$contains(tp^1, tp^2) \equiv \exists k > 0 \mid contains(R^1_k, R^2_k) \wedge \cdots \wedge contains(R^1_{k+n}, R^2_{k+n}) \wedge \quad contains(T^1_k, T^2_k) \wedge \cdots \wedge contains(T^1_{k+n}, T^2_{k+n}), n = |R^2|$$

where the *contains* operator between regions and temporal intervals (*data* × *data*) is defined as in [21]. To construct the table of pairs of objects that satisfy a generic relation, we use the query syntax `CREATE RELATION`, as in the following example, where a table of pairs of T-patterns (tp_1, tp_2) is created, such that tp_1 contains tp_2:

```
CREATE RELATION TPatternContains USING CONTAINS
FROM (SELECT t1.id, t1.tpattern, t2.id, t2.tpattern
   FROM TPatternTable t1, TPatternTable t2
   WHERE t1.id <> t2.id)
```

A distinctive *pattern* × *data* primitive is the *entails* relation. $entails(p, o)$ holds if the data object o is an instance of pattern p. The definition of *entails* is specific for each M-Pattern, and the details are given in Sect. 3.3. An example of query is the following, which creates a table containing the trajectories belonging to a specific T-Cluster:

```
CREATE RELATION TrajectoriesInCluster USING ENTAILS
FROM (SELECT t.id, t.traj, c.id, c.cluster
   FROM TrajectoryTable t, ClustersTable c)
```

Transformation primitives: Transformations are a class of primitives which uses external methods to perform complex data pre-processing and model/pattern post-processing operations in the knowledge discovery process.

```
CREATE TRANSFORMATION TransformedData USING
TRANSFORMATION_ALGORITHM
FROM (SELECT t.id, t.trajobj FROM TrajectoryTable t)
SET PARAM.K = N
```

3.3 M-Models and M-Patterns constructors

The models and patterns of M-Atlas are constructed by a `CREATE MODEL` query, which refers to a specific method available in the spatio-temporal data mining library. The main such methods are sketched below.

T-Pattern

Input: D, a set of trajectories; R, a set of spatial objects denoting regions of interest; s_{min}, a minimum support threshold; τ, a time tolerance threshold.

Output: the set of all T-Patterns $TP = r_0 \xrightarrow{[t^s_1, t^e_1]} r_1 \cdots \xrightarrow{[t^s_n, t^e_n]} r_n$ such that TP *entails* at least a fraction s_{min} of the input trajectories in D, where each r_i is a region from R and each $[t^s_j, t^e_j]$ is a temporal annotation specifying the minimum and maximum duration of the transition from region r_{i-1} to region r_i.

Entailment: A T-Pattern TP entails a trajectory T if the latter contains an instance of the former, i.e., a sequence of points that are contained in the regions that compose the T-Pattern, and such that their time gaps are contained in the corresponding transition time intervals of the T-Pattern with tolerance τ. In formula, there exists a subsequence T' of T, $T' = \langle (x'_0, y'_0, t'_0), \ldots, (x'_n, y'_n, t'_n) \rangle$ such that:

1. $\forall_{0 \leq j \leq n}.\ (x'_j, y'_j) \in R_j$, and
2. $\forall_{1 \leq j \leq n}.(t'_j - t'_{j-1} \pm \tau) \in [t^s_j, t^e_j]$

Complexity: The algorithm for T-Pattern mining (see [17]) has both space and time complexity linear on the number of input trajectories, while complexity grows exponentially with the average length of the input trajectories.

T-Clustering

Input: D, a set of trajectories; $d(T_1, T_2)$, a distance function between trajectories, selected from a repertoire, which includes the following instances:

- *Common destination*: $d_d(T_1, T_2)$ is given by the Euclidean distance $\delta(p_1, p_2)$ between the last point p_1 of T_1 and the last point p_2 of T_2
- *Common origin*: $d_o(T_1, T_2)$ is given by the Euclidean distance between the first point of T_1 and the first point of T_2
- *Common origin and destination*: $d_{od}(T_1, T_2) = d_o(T_1, T_2) + d_d(T_1, T_2)$
- *Route similarity*: This considers the entire spatial path of the two trajectories T_1 and T_2 and assigns the average Euclidean distance between any two points of T_1 and T_2 within a spatial neighborhood [4]
- *Colocation Similarity*: synchronized spatio-temporal distance

$$d_{st}(T_1, T_2) = \sum_{t \in I} \delta(T_1(t), T_2(t))/|I|$$

where $T_i(t)$ denotes the (interpolated) position of trajectory T_i at time t; the distance at each time is averaged over the length of the considered time interval

Eps, a distance threshold; *minPts*, the minimum number of points contained in a neighborhood of radius *Eps*.

Output: *Reachability plot*, a high-level description of the clustering structure of the input trajectories, obtained using the density-based trajectory clustering method of [29]. A reachability plot, given a distance threshold ϵ, generates a partition of the input data set into a set of T-clusters. The adopted algorithm is a variant of the well-known OPTICS [6] method. We remark that, while M-Atlas also includes different other clustering methods (and new ones can be easily integrated into the system), our experience suggests that density-based clustering best suits trajectory data, due to the abundance of noise and irregularly shaped clusters.

Entailment: A T-Cluster C, obtained from a reachability plot, entails a trajectory T simply if $T \in C$.

Complexity: T-Clustering has a space complexity $O(m)$, where m is the number of input trajectories, and a time complexity $O(mK)$, if the computational cost of a single neighborhood query is $O(K)$. In the case that the execution of neighborhood query can be optimized using an index with a query time of $O(logm)$, then T-clustering is $O(mlogm)$; otherwise, the whole complexity is $O(m^2)$.

T-Flock

Input: D, a set of trajectories; τ, re-sampling time period; m, minimum number of objects in a flock; k, minimum duration of a flock (time unit is τ); r, maximum radius of a flock.

Output: The set of $(m, k, r) - flocks$ [8,20,42] discovered in D. An $(m, k, r) - flock$ is defined as a group of at least m trajectories that fall within a disk of radius r for a time interval I of duration $|I| \geq k$. Before flock extraction, the original trajectories are re-sampled with constant rate τ. The heuristics applied to extract flocks [42] is based on a bottom-up, time slice merging procedure that starts from single-point flocks and is iterated to build flocks of longer duration. This approach differs from others in literature, for instance [20], that follows an earliest/longest occurrence-first policy, and [8], that is based on approximated range queries over all candidate time intervals of sufficient duration.

Entailment: A T-Flock (I, r, b) entails a trajectory T if the positions of T at the time instants in interval I fall within distance r from the base trajectory b of the T-Flock.

Complexity: T-Flock discovery has a $O(n^2 l)$ computational cost and $O(nl)$ space complexity, where $n = |D|$ is the data set size and l is the average length of input trajectories.

T-O/DMatrix

Input: D, a set of trajectories; R_O, a set of origin regions; R_D, a set of destination regions (R_O and R_D may overlap).

Output: A T-O/D Matrix, an M-Model representing the origin/destination matrix M for the trajectories in D, where $M_{i,j} = n$ if there are n trajectories $T \in D$ such that T starts in the origin region $R_i \in R_O$ and T ends in the destination region $R_j \in R_D$. In other words, $M(i, j)$ is the *flow* from R_i to R_j.

Complexity: The space complexity of T-O/D Matrix is $O(mn)$, where m and n are the cardinality of the two region data sets. The computational cost is $O(l)$, where l is the number of input trajectories. M-Atlas provides other model constructors, including the T-PTree (see Fig. 10b), a structure designed to support the next-location prediction method in [27].

4 Mastering the knowledge discovery process

Each visual interaction of the analyst with the M-Atlas interface is compiled into a sequence of M-Atlas queries. Alternatively, an expert data miner can directly submit queries to the M-Atlas engine, to exploit its full expressiveness. In either cases, an analytical process is created by combining data and model constructors with spatio-temporal primitives within the querying and mining language.

One of the key objectives of M-Atlas is to enable the mobility data analyst to master the complexity of the knowledge discovery process even in its more critical issues, such as the definition of complex interactive and iterative analysis, the estimation of algorithm parameters, and the validation of models. The rest of this section is dedicated to highlight how M-Atlas supports the subtleties of the KDD process, also providing a fertile ground to create and realize novel analytical methods.

4.1 Clustering by sample

A clustering-based analytical process requires several user interactions, aimed at refining and adjusting the parameters while a better insight into the extracted models is reached. Therefore, the system reaction time during such iterative process is crucial to allow the user to actively interact. To this aim, in [5], an interactive clustering method is proposed, based on the idea that firstly, a clustering partition is computed over a sampled data set and secondly, such partition is used as a classifier over the entire data set. More precisely, the method is composed by the following steps: (1) a sampling of the entire data set is computed, and a clustering analysis is performed over the sampled data until a satisfactory cluster-

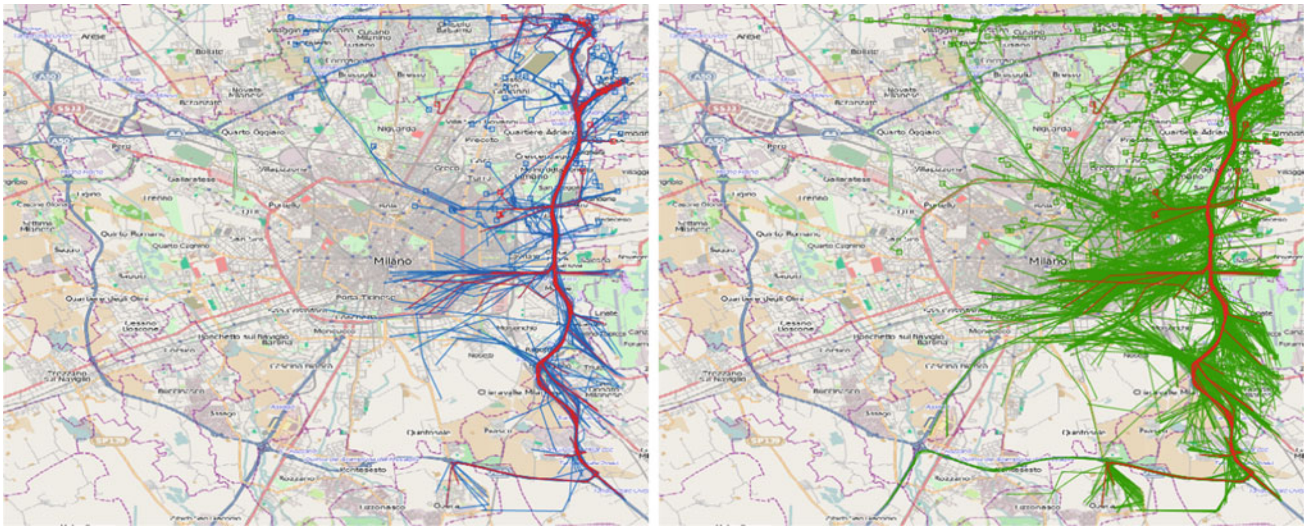


Fig. 12 Classification of new trajectories using a set of specimens from `WednesdaySpecimens`. *Left*, *blue lines* represent the trajectories of a single cluster of Wednesday, April 4, and the *red lines* are the specimens learned for the selected cluster. *Right*, *green lines* are the trajectories of the entire week classified by the same set of specimens. Visual inspection confirms that cluster shape is preserved, albeit the size of the second data set is 7 times larger. Quantitative measures of cluster quality, such as silhouette coefficients, can be easily computed

ing partition is reached; (2) one or more representatives for each cluster are computed; and (3) such representatives are used to classify the data of the entire data set by assigning each data its best fitting representative.

Such complex analytical process, based on the interleaving of a unsupervised method with a supervised one, may be entirely expressed in M-Atlas by combining query and mining primitives as follows. The first query focuses on the trajectories of a single day (other sampling approaches may be used as well):

```
CREATE TABLE WednesdayTrajectories AS
  SELECT * from TrajectoryTable
  WHERE day = '04/04/2007';
```

The second query performs the clustering analysis on the selected trajectories using *Route Similarity* as distance functions, 750 meters as distance threshold, and 5 trajectories as the density threshold (parameter setting can be assisted by the estimation method illustrated in Sect. 4.4):

```
CREATE MODEL ClustersWednesday AS MINE T-CLUSTERING
  FROM (SELECT t.id, t.trajectory FROM
   WednesdayTrajectories t)
  SET T-CLUSTERING.FUNCTION = ROUTE_SIMILARITY AND
    T-CLUSTERING.EPS = 750 AND
    T-CLUSTERING.MIN_PTS = 5
```

In the third step, the trajectories entailed by the newly extracted T-Clusters are selected and then used to compute the set of representatives, named *specimens*, for each cluster:

```
CREATE RELATION WedTrajectoriesToClusters USING ENTAIL
  FROM(SELECT t.id, t.trajectory, c.id AS cid
     FROM WednesdayTrajectories t, ClustersWednesday c)

CREATE MODEL WednesdaySpecimens AS MINE SPECIMENS
  FROM (SELECT id, trajectory, cid FROM
   WedTrajectoriesToClusters)
  SET SPECIMENS.MAX_DISTANCE = 750 AND
    SPECIMENS.METHOD = ROUTE_SIMILARITY
```

`SPECIMENS` is a new mining primitive that creates, for each original cluster, a set of specimens, i.e., a condensed representation of a set of trajectories according to a selected distance function.

The final step is the classification of every new (unseen) trajectory T, by assigning T either to one of the clusters or to *noise*. To this aim, we check for each trajectory T, its closest specimen S, and assign T to the cluster of S. This is a complex algorithm that is specified as a transformation primitive, which takes as input a set of specimens, a set of trajectories, and a distance function and constructs a table where each trajectory is tagged with its assigned cluster/set of specimens.

```
CREATE TRANSFORMATION ClassifiedTrajectories USING
 SPECIMENS_CLASSIFIER
  FROM (SELECT id, trajectory FROM TrajectoryTable)
  SET SPECIMENS_CLASSIFIER.SPECIMENS = (SELECT *
   FROM WednesdaySpecimens) AND
    SPECIMENS_CLASSIFIER.METHOD = ROUTE_SIMILARITY
```

Figure 12 shows the result of classifying the trajectories of the entire week using the set of specimens found in `WednesdaySpecimens`.

4.2 Temporal analysis of T-Patterns

An important task is to study the stability of a set of extracted T-Patterns over time. We show a method to accomplish this task, using the trajectories of the Pisa2010 data set, partitioned into five consecutive weeks of data. We extract a set of 274 T-Patterns from the first week, and we want to analyze the variation of the support of these T-Patterns in the four

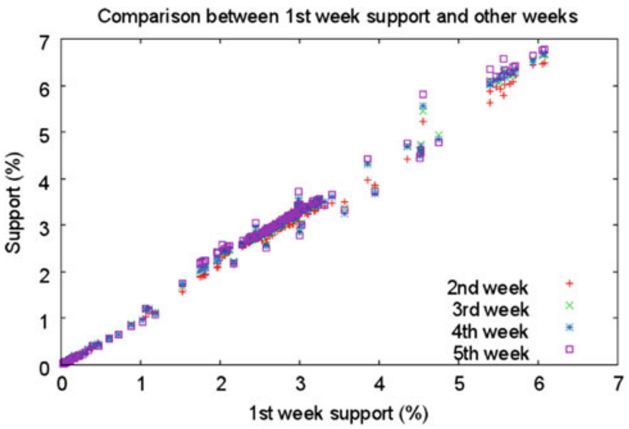


Fig. 13 Stability of support of 274 T-Patterns mined in week 1 of Pisa2010 over the remaining 4 weeks of Pisa2010. Each point (x, y) in the scatter plot is associated with a specific T-Pattern TP, where x is the (relative) support of TP in week 1 and y is the support of TP in one of the weeks from 2 to 5 (four different points are plotted for each of the 274 T-Patterns)

subsequent weeks. To this purpose, we count the trajectories that entail each T-Pattern in weeks 2 through 5 (see query below for week 2).

```
CREATE RELATION tp_on_2week USING ENTAIL
  FROM (Select p.id, p.tpattern, t.id, t.traj
  FROM TPTable p, Traj2Week t)
SELECT pid, count(*) FROM tp_on_2week group by pid.
```

Figure 13 compares the original support values found in week 1 with the support in weeks 2–5, highlighting that almost all the T-patterns maintain a similar support over the observation period.

4.3 T-Pattern parameter estimation

The basic step of the T-Pattern algorithm is the detection of frequent regions in the area under analysis. Therefore, the support threshold is the most influent parameter for the whole process. We present a heuristics data-driven method to estimate the value for this threshold. The cumulative frequency distribution of trajectories in the spatial grid cells is shown in Fig. 14(left). We claim that the points of significant slope change in this distribution are the best candidates for the support threshold, because these points separate groups of grid cells that have a rather uniform frequency internally but the frequency between the different groups is very different. Our heuristic detects this slope-changing points as candidates for the support threshold of T-Pattern algorithm.

Another crucial parameter for the extraction of T-Pattern is the time tolerance τ. In Fig. 14(right), we plot all the time distances for every possible pair of points in each trajectory. These represent all the possible transition time candidates in the T-Pattern mining algorithm. The sharp steps in the zoomed inset are the artifact of the average sampling rate, ≈33 s. This is the minimum admissible value for the τ parameter. We note that with a high value of τ, the T-pattern computation aggressively merges the transition times. For instance, with a 130 s the 10% of transition times are merged. An adequate candidate for the τ parameter is around the 50th percentile (14 min) and, in any case, between the 10th and the 90th percentiles (2–45 min).

4.4 Density parameter estimation

A recurrent parameter type required by the mining algorithms of M-Atlas is the density threshold. For example, T-Clustering uses a density threshold to separate noise (sparse groups of trajectories) from the clusters (highly dense groups). In T-Flock mining, the density threshold is used to prune the search space for the candidate generation of possible flock extensions. In general, the density of the neighbor-

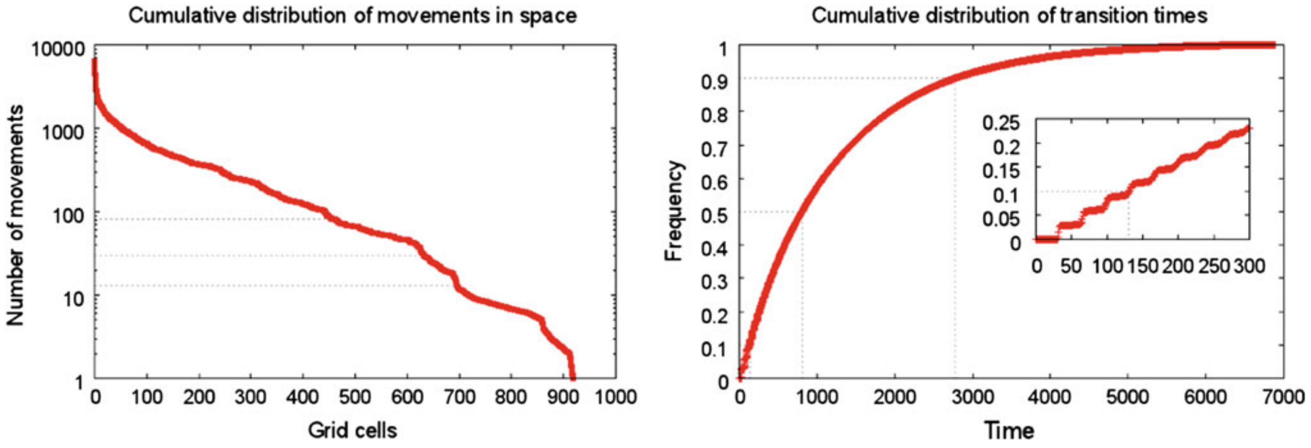


Fig. 14 Cumulative frequency distribution of trajectories in space: the system proposes a ranked list of three candidate values for the T-Pattern support threshold (13, 24, 82) based on detected points of significant slope variation (*left*) Cumulative distribution of transition times between each pair of points in each trajectory (*right*)

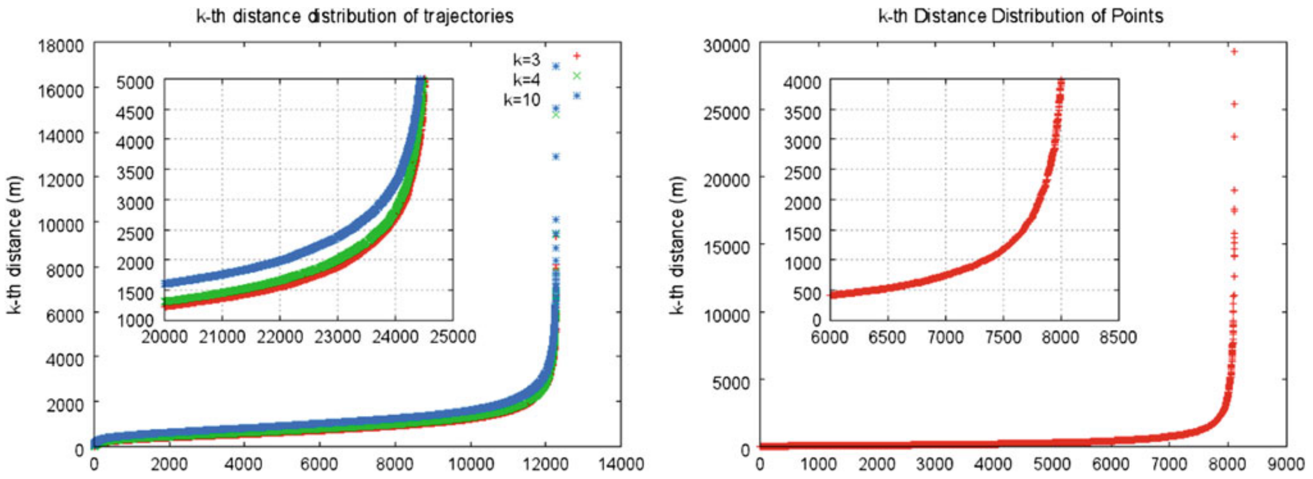


Fig. 15 The kth nearest neighbor distance for different objects: *left* distribution of distances for trajectories in the Milan2007 data set using the *Route similarity* distance function and different values of k; *right* distribution of the second nearest neighbor ($k = 2$) distances for points in the Pisa2010 data set

hood of an object is determined by counting the number of distinct objects within a given radius. For the T-Flock algorithm, the radius depends on both space and time. In the T-Clustering, the semantics of distance, and hence of radius, depends on the distance function selected by the analyst. In general, both methods use a radius threshold r and a minimum number of point k, which jointly define the density threshold. Setting r and k with no prior knowledge is difficult, but the analyst can be assisted by a heuristic methodology that, given a choice for k, suggests the empirically best value for r. This estimation method, proposed originally by [13] for density-based clustering, is extended here also for T-Flock mining and can be fully supported by M-Atlas queries and basic statistics. Given a candidate value for k freely guessed by the analyst (the rule of thumb from [13] is to pick a small value around 4-10), the radius parameter r can be estimated as follows. We measure the distance between each trajectory T in the data set and the k-th nearest neighbor of T, and plot all such distances in increasing order. The distribution of such distances can give us a meaningful overview of how to separate trajectories with a dense neighborhood from those with a sparse neighborhood. In particular, if the plot has a point of sharp increase in the derivative (slope change), then the distance value at that point is a suitable candidate to separate "dense" trajectories and noise. Such process is supported by means of an ad hoc transformation, named `DENSITY_ANALYSIS`. The following query supports the density analysis for the Milan2007 data set with $k = 10$ and the similarity function set to *Route Similarity*):

```
CREATE TRANSFORMATION density_analysis_route
  USING DENSITY_ANALYSIS
  FROM (SELECT * FROM TrajectoryTable)
  SET REACHABILITY_ANALYSIS.MIN_PTS = 10 AND
      REACHABILITY_ANALYSIS.METHOD = ROUTE
```

Figure 15(left) shows the density distribution as obtained from the previous query using distinct values for k (i.e., $k = 3$, $k = 4$, $k = 10$). It is clear from the plot that a suitable value for the radius r is 3,000 m for $k = 3$ and 4,000 m for $k = 10$. In the case of T-Flocks, the plot reported in Fig. 15(right) shows the distances of the second point ($k = 2$) for the Pisa2010 data set. In the given figure, a clear knee of the curve occurs at around 1,600 m, which can be set as candidate r. This high value also indicates that the data set is quite sparse and thus requires a large radius value to find density-based clusters.

5 Discovery of mobility behavior with M-Atlas

We now address the questions of Sect. 2.3 with analytical processes supported by M-Atlas.

5.1 Most popular itineraries from the city center to suburban areas

To characterize the main flows from the city center toward the suburbs, we start by considering the administrative borders of Milan and its adjacent municipalities (see Fig. 16(left)). Such regions are used as input for the T-O/D Matrix model constructor, obtaining a high-level description of the flows between each pair of regions. The visual interface enables the analyst to interact with the model (see Fig. 16(right)). In our analysis, we focus on the T-Flows leaving the city of Milan toward the north-east suburbs (the NE satellite municipalities of Monza, Sesto San Giovanni, Cinisello Balsamo, Cologno Monzese, and Brugherio). We select the trips entailed by the

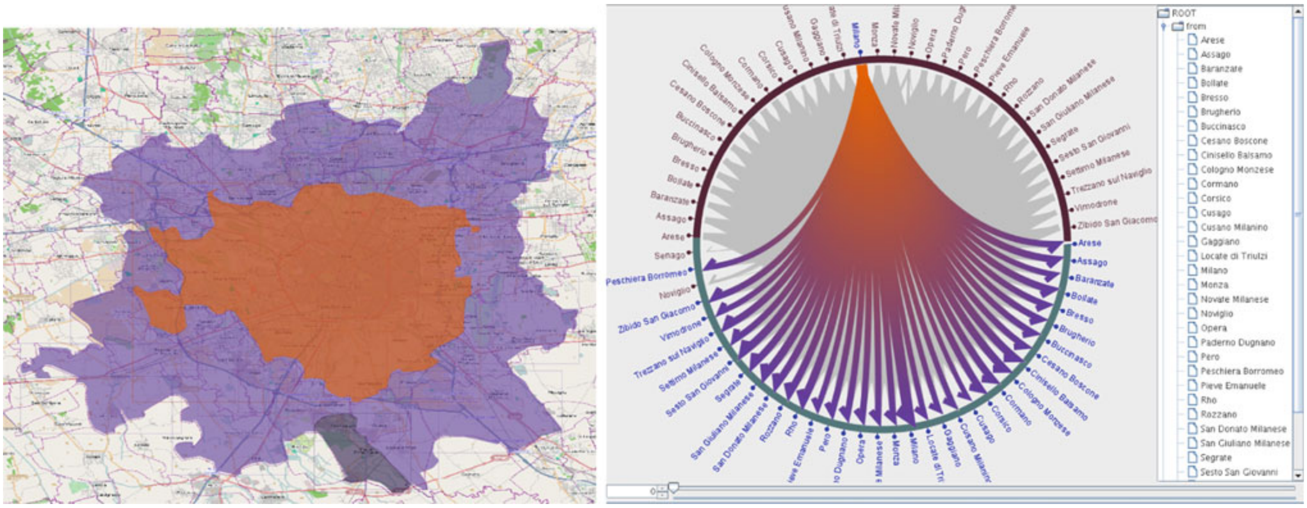


Fig. 16 The resulting T-O/D Matrix model for Milano2007 on a specific weekday (Wednesday, April 3). *Left* The regions used as input the model: the center region (in *orange*) contains the administrative borders of Milan; the *purple cells* represent the adjacent cities. *Right* The visual interface to browse the O/D Matrix: each region is represented with a node, nodes are displayed in a circular layout. The arc connecting two nodes represents the flow, i.e., the number of trips from the origin to the destination node; the arc width is proportional to the flow. The analyst browses visually the O/D Matrix either selecting some specified origins and/or destinations, or highlighting the main flows by setting a minimum support threshold

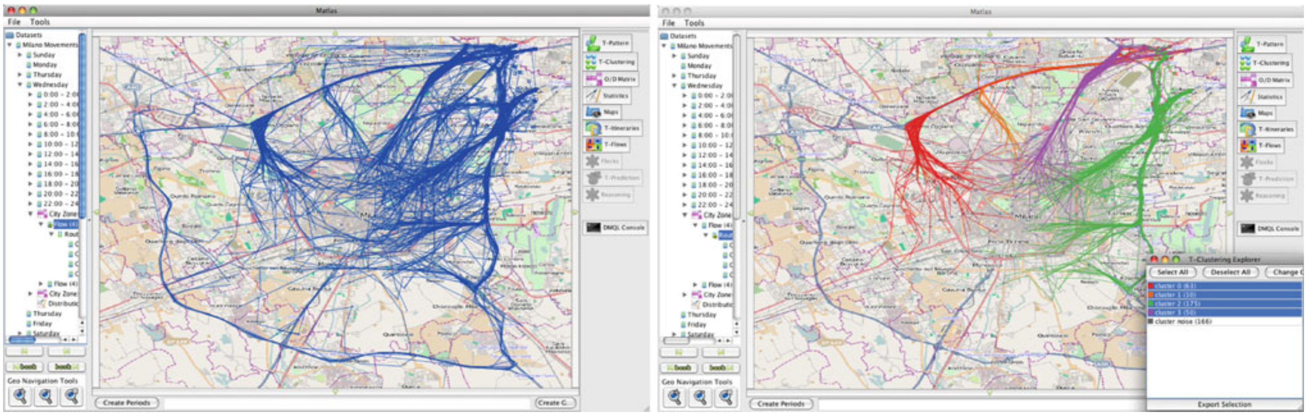


Fig. 17 The result of T-Clustering from the trajectories moving from the center to the North-East area. *Left* The input data set for the clustering algorithm: the trajectories moving from the center to the North-East area. *Right* The resulting clusters using the *Route Similarity* distance function. The cluster are visualized using a *themed color*, where the trajectories in the same cluster are visualized with the *same color*. The analyst can browse the different clusters. In this example, the three largest clusters are visualized: *cluster 2 (green)* shows the most popular route, which heads east toward the outer ring and then north; *cluster 0 (red)* is the second most popular route, north and then east; *cluster 3 (purple)* heads straight toward north-east

selected T-Flows. The M-Atlas queries that realize this tasks, automatically generated as a product of the visual interaction with the analysts, are the following:

```
CREATE MODEL MilanODMatrix AS MINE ODMATRIX
FROM (SELECT t.id, t.trajectory FROM TrajectoryTable t),
(SELECT orig.id, orig.area FROM MunicipalityTable orig),
(SELECT dest.id, dest.area FROM MunicipalityTable dest)

CREATE RELATION CenterToNESuburbTrajectories USING ENTAIL
FROM (SELECT t.id, t.trajectory FROM TrajectoryTable t,
MilanODMatrix m
WHERE m.origin = Milan AND
m.destination IN (Monza,...,Brugherio))
```

The resulting trajectories are presented to the analyst as in Fig. 17(left). Despite all these trips originate in the city center and end in the NE suburbs, a broad diversity is still evident. To discover the most popular itineraries followed by the selected travels, we use the T-Clustering model constructor with the *Route Similarity* distance function, and parameters *Eps* and *MinPts* estimated with the method of Sect. 4.4. Behind the scenes, M-Atlas generates and executes the model constructor query:

```
CREATE MODEL ClusteringTable AS MINE T-CLUSTERING
FROM (Select t.id, t.trajectory from
CenterToNESuburbTrajectories t)
SET T-CLUSTERING.FUNCTION = ROUTE_SIMILARITY AND
```

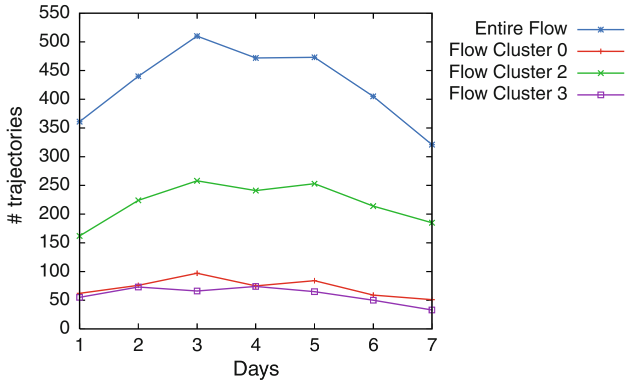


Fig. 18 Distribution of estimated cardinality of three main clusters 0 (*red*), 2 (*green*), 3 (*purple*), and number of all travels from the city center to NE suburbs (*blue*) over the week April 1st (Sat)–7th (Sun). Clusters 0 (*red*) and 3 (*purple*) are essentially constant with a small decrease during the weekend (days 1 and 7), while cluster 2 (*green*) has a shape similar to the general flow, with a significant decrease during the weekend

```
    T-CLUSTERING.EPS = 400 AND
    T-CLUSTERING.MIN_PTS = 5
```

As a result, the analyst obtains a list of T-Clusters, each of which can be visualized by means of an underlying entail query that selects the trajectories belonging to the T-Cluster. Figure 17(right) shows how the most popular clusters highlight the main routes used by drivers to leave the center toward NE.

In order to assess the validity of the discovered clusters, we need to check if they reflect episodic events of the specific weekday analyzed, or whether the clusters systematically repeat during the observation period. To this aim, we need to measure how the population of the clusters distributes on the days of the week, and this task can be accomplished using the clustering-by-sampling process illustrated in Sec. 4.1. For each day from Sunday, April 1st through Saturday, April 7, we classified each trajectory as either a member of one of the discovered clusters or noise according to its distance from a cluster prototype. Figure 18 shows how the distribution of the estimated population of the three clusters varies during the week. The figure highlights that clusters 0 and 3 are stable over the entire week, while the most popular cluster 2 (green) is stable over weekdays only, suggesting that it is composed mainly by outbound commuters who travel during working days.

The next question is to determine if the commuters of cluster 2 travel from home to work or vice versa. The answer is obtained by analyzing the temporal distribution of the trips of the cluster over the hours of a weekday (see Fig. 19(center)).

5.2 Accessibility to key mobility attractors

To understand how users access big mobility attractors, we focus on the travels ending in the most crowded parking lots of the city. A T-O/D Matrix between the entire city as origin and the individual parking lots as destinations can be constructed, to the purpose of selecting the highest flux toward the top accessed parking lot with its associated trajectories. The following queries perform this task, yielding the visualization of Fig. 20(left).

```
  CREATE MODEL ParkODMatrix AS MINE ODMATRIX
   FROM (SELECT t.id, t.trajectory FROM
    TrajectoryTable t),
      (SELECT orig.id, orig.area FROM
       MunicipalityTable orig
       WHERE orig.id = Milan),
      (SELECT dest.id, dest.area FROM
       ParkingLotTable dest)

  CREATE RELATION TopParkTrajectories USING ENTAIL
    FROM (SELECT t.id, t.trajectory FROM
     TrajectoryTable t,
ParkODMatrix m
       WHERE m.weight = MAX(m.weight))
```

The Linate airport parking lot emerges as the top destination. Figure 20 shows the set of trajectories that start in Milan and end in the airport parking lot. It is evident that vehicles start from a broad diversity of locations, but converge toward the parking lot. Our goal is to characterize the typical behav-

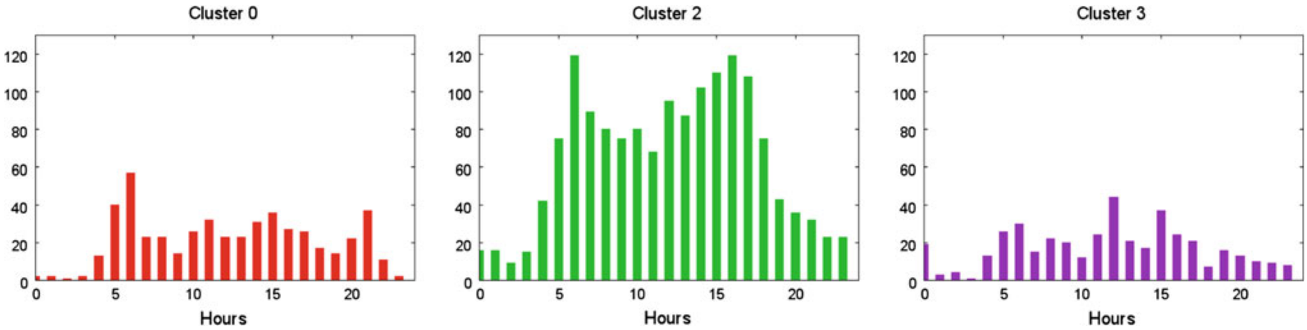


Fig. 19 Temporal distribution of the trajectories in the clusters of Fig. 17(right) on the hours of weekdays. Cluster 0 and Cluster 3 (*left, right*) do not exhibit significant peaks, while cluster 2 (*center*) has a peak in the morning and one in the afternoon. The temporal profile of Cluster 2 captures two commuting behaviors: a group leaving the city in the morning (commuters going to work outside), and a larger group leaving the city in the late afternoon (commuters coming back home in the suburbs after work)

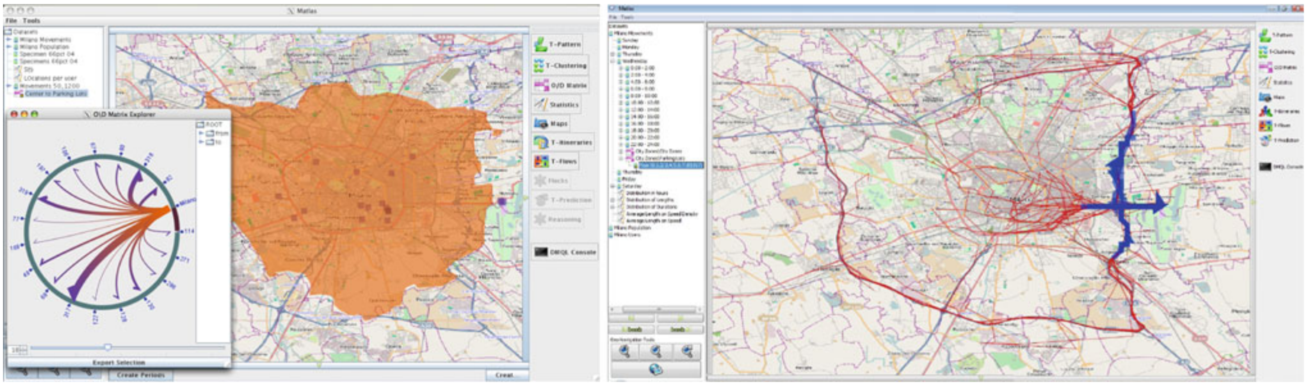


Fig. 20 Accessibility to parking lots. *Left* Asymmetric T-O/D Matrix from Milan (origin) toward parking lots (destinations). The highest fluxes to parking lots are highlighted by adjusting the frequency threshold slide bar (*bottom left*). The biggest attractor is parking lot 317 (Linate airport). *Right* Travels (*red*) from Milan to the Linate airport parking lot, and summary of associated T-Patterns (*blue*), characterizing how the travels approach the final destination

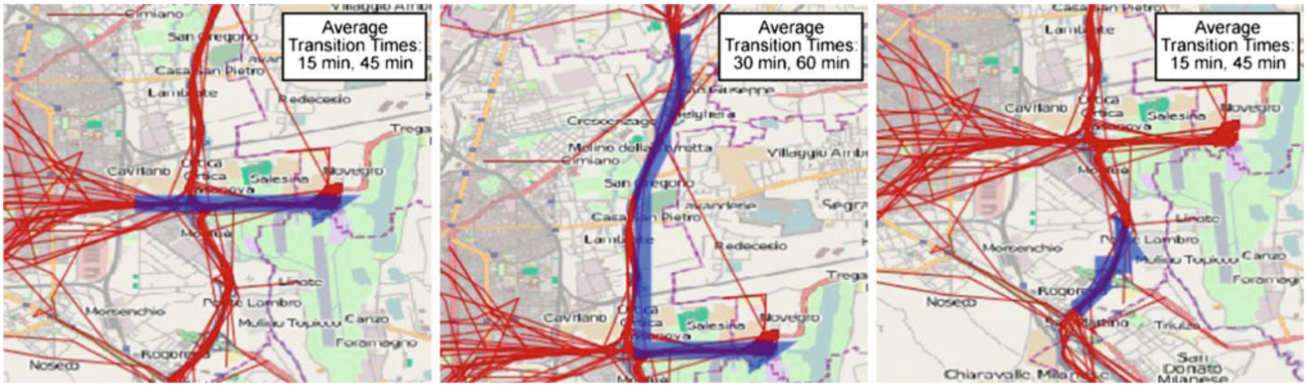


Fig. 21 Most significant T-Patterns for traffic directed to Linate airport: from the city center (*left*), from north ring (*center*), from south ring (*right*) Transition times are reported in the *insets*

iors of vehicles when approaching the attractor, a task that cannot be directly addressed by T-Clustering, due to fact that travels follow similar routes only in their final parts (whose length is not known a priori). An effective way to detect frequent segments of trips that are followed by a significant volume of vehicles is T-Pattern mining. The following model constructor query realizes this task, generating the T-Patterns supported by at least 5% of the travels to Linate (parameters are chosen following the methodology of Sect. 4.3).

```
CREATE MODEL LinateTPatterns AS MINE T-PATTERN
 FROM (SELECT t.id, t.trajectory FROM
  TopParkTrajectories t)
 SET T-PATTERN.side = 50 AND T-PATTERN.time = 900
   AND T-PATTERN.support = 0.05
```

Figure 20(right) is a visual summary of the discovered T-Patterns, which allow us to characterize the three main routes to approach the attractor, together with the different travel times. Figure 21 focuses on the three most frequent T-Patterns. Observe how the T-Patterns approaching the airport from north are longer than those from south, highlighting that the northern travels tend to concentrate on the outer ring earlier than the southern travels, which instead use a small segment of the ring. This behavior suggests the presence of more alternative routes to get in the proximity of the airport from south and city center than from north.

5.3 Extraordinary events

Extraordinary events have large impact on mobility. Big rendezvous, such as concerts and sport competitions, set the destination of many individual trips toward a small area (the event location), where many people concentrate for the event duration. At the end of the event, the same area is the origin of many return trips. Even if not known a priori, big events can be easily detected by localizing exceptionally high concentrations of presence in specific areas at specific time intervals. Density maps for stationary cars, analogous to the maps of Fig. 6 for density of moving cars, can be used for visual

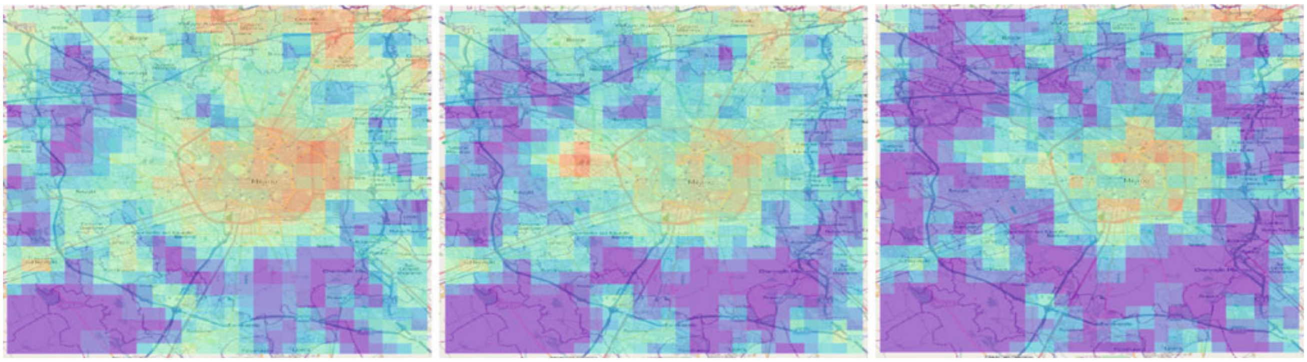


Fig. 22 Distribution of presence on Tuesday, April 3rd, in three contiguous time slots of 2 h: (*left*) from 6 pm to 8 pm, (*center*) from 8 pm to 10 pm, (*right*) from 10 pm to midnight. An evident hot (*red*) spot emerges between 8 pm and 10 pm, and disappears afterwards. The location (immediate west of city center) is that of Stadio Meazza, the main soccer arena

exploratory analysis of abnormal concentration of presence. The following query creates the presence density maps for the intervals and spatial cells defined by the tables *IntervalTable* and *GridTable*. In our analysis, we use a 0.5 km × 0.5 km grid and compute, for each grid cell and for every interval of two hours of each day, the number of cars that are stationary in the cell.

```
CREATE RELATION PresenceTable Map USING INTERSECT
 FROM (SELECT stop.id, stop.trajectory FROM
  PresenceTable stop),
   (SELECT i.id AS iid, i.interval FROM
    IntervalTable i)
   (SELECT g.id AS gid, g.area FROM
   GridTable g)

CREATE TABLE PresenceMap AS
 SELECT pt.iid, pt.gid, count(*)
 FROM PresenceTable pt
 GROUP BY pt.iid,pt.gid
```

The result obtained from Milano2007 is shown in Fig. 22. The location of the hot spot—the main soccer arena and surrounding parking areas—suggests that a big sport event occurred in such location. It's easy to check that a quarter-final match of the UEFA Champions League took place in the exact location and time, attended by ≈77,700 spectators.[4] The same result is obtained automatically, by a query that selects every cell C and time interval h (8–10 pm in our case) such that the population of cell C during h is above the 90th percentile in the distribution of the population of (C, h) over the entire observation period.

The next step is the analysis of the trips associated with the detected event, i.e., when and how attendees reached and left the event location. First, the arrival and departure time of the each car v parked in the arena area during the day is approximated considering, respectively, the ending point of the incoming trajectory and the starting point of the outgoing trajectory of v. The distribution of arrivals and departures during the day is depicted in Fig. 23(left). We further analyze the return travels of the attendees after the match, in order to detect the main escape routes. We apply T-Clustering to the trajectories leaving the arena area between 10pm and 00am, obtaining the T-Clusters shown in Fig. 23. The detected escape routes are relevant information for a mobility manager to enact countermeasures to prevent possible congestion.

5.4 Mobility prediction

The prediction of traffic congestions represent a challenging task for urban mobility managers. The following experiments are aimed at showing how to exploit M-Atlas to predict future areas of dense traffic, which may lead to traffic congestions. The T-PTree tool has been used to predict the location of areas particularly dense of trajectories. We run this experiment on the Pisa2010 data set which covers a larger area and a longer temporal interval compared with Milan2007 data set. This is particularly useful in prediction tasks since the training and test phases use a richer data set. In fact, the longer temporal duration allows to use a coarse granularity for the prediction (e.g., the training set include several days and can be tested on a larger temporal interval). Here, we selected a subset of the entire Pisa2010 data set which includes trajectories from 5 working days (from Monday July 5th to Friday July 9) restricted to the morning peak hours (8–10 am). This selection resulted in about 10,000 trajectories for the training set. Then, we selected, as test set, the trajectories of Monday July 12th (in the same temporal interval) leading to a total of around 4,000 trajectories. From them, the algorithm was able to predict the next location of about 3,000 trajectories focused on 29 regions. Five of them contain more than 150 trajectories. Scaled to the global number of circulating vehicles (see Sect. 2), this corresponds to about 7,500 vehicles predicted to converge to these areas in the two-hours interval. The M-Atlas query is:

[4] Milan A.C. versus Bayern Munich, source http://en.wikipedia.org/wiki/UEFA_Champions_League_2006-2007.

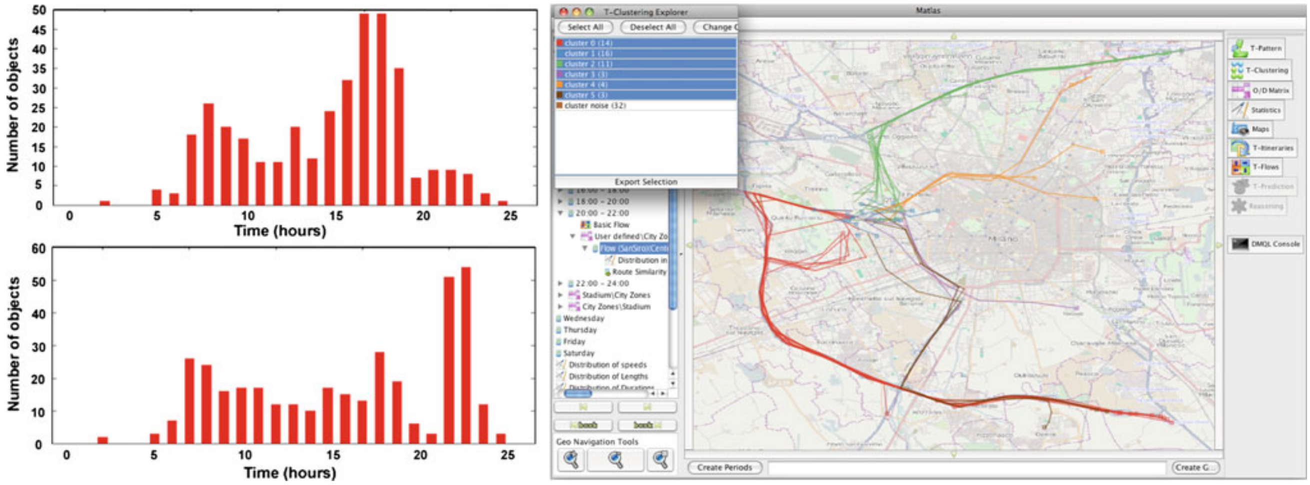


Fig. 23 *Top-left* Temporal distribution of arrivals to and *bottom-left* departures from the arena area: arrivals peak from 5 pm to 8 pm, and departures peak from 10 pm to midnight. Arrivals are spread over several hours, while departures occur soon after the end of the match. *Right* Clusters of trips leaving the arena after the football match. The largest clusters perform short range trips (*blue*) or take the road ring, either NE (*green*) or SE (*red*)

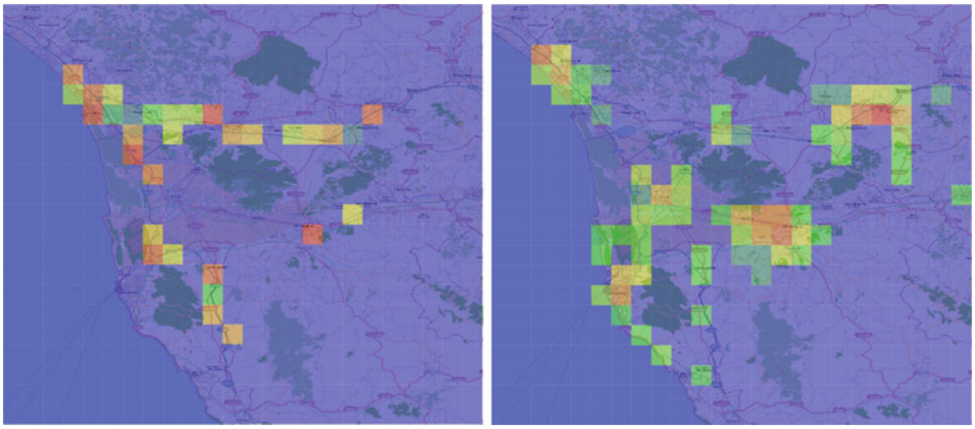


Fig. 24 Distribution of presence: with predicted trajectories (*left*), with the real trajectories (*right*). As an overall overview, we note that the locations of darker areas reasonably correspond in both pictures. However, we can observe that the ground truth areas appear larger then the predicted and this is due to the way the T-PTree uses the regions extracted by means of T-Pattern algorithm

```
CREATE TRANSFORMATION PredictionsTable USING PREDICTOR
  FROM (Select t.id, t.object from TrajectoryTable t)
   SET PREDICTOR.T-PATTERN_TABLE = TpatternTable AND
     PREDICTOR.TH_S = 10 AND
     PREDICTOR.TH_T = 3600 AND
     PREDICTOR.TOLERANCE = 1000
```

Figure 24 reports the results of the prediction compared with the ground truth obtained by computing the density map of the trajectories moving during the predicted period.

It is worth pointing out that the interpretation of the predicted zones suggests further deeper analysis. Indeed, the dense regions does not necessarily indicate traffic problems in that areas. These regions represent dense movement of cars, which can hint the possibility of traffic jams or congestions. Further analysis, focussed on these specific areas, are needed to have a more precise indication of possible traffic problems.

5.5 Traffic jams detection

This experiment is aimed at finding the possible traffic jams that occurred in the monitored area. We considered as traffic jam a group of cars moving close together slowly for a certain amount of time. We experimented the use of T-Flock to find cars moving together thus detecting possible traffic jams selecting the slow flocks. Similar to the previous experiment, we use the Pisa2010 data set which is richer in terms of number of trajectories and larger in the spatial and temporal extent.

We run the T-Flock algorithm on M-Atlas using the following query:

```
CREATE MODEL flock_table AS MINE FLOCK
 FROM (SELECT t.id, t.object FROM TrajectoryTable t)
   SET FLOCK.TIME_GRANULARITY = 60 AND
     FLOCK.RADIUS = 500 AND
     FLOCK.MIN_POINTS = 3 ANS
     FLOCK.MIN_TIME_SLICES = 4
```

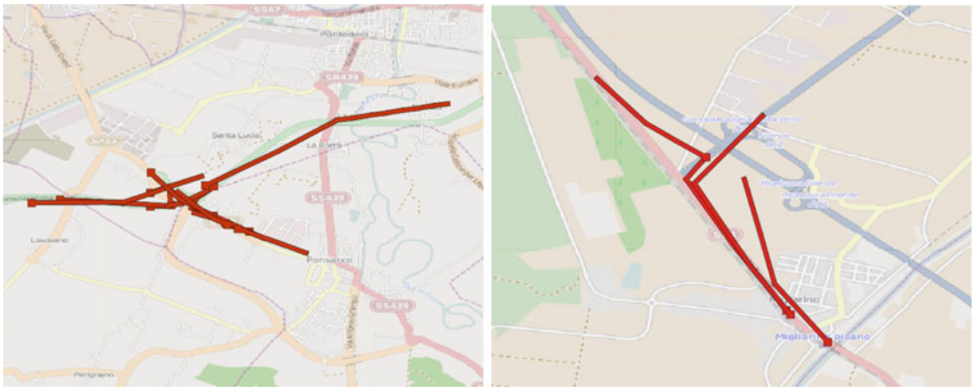


Fig. 25 The results of T-Flock from Pisa2010 data set. 13 Flocks found in a highway near the city of Pontedera, the average speed of each flock ranges from 15 to 37 km/h and the temporal duration of each flock ranges from 3 to 10 min (*left*). 4 flocks found at the tollhouse of the highway close to the city of Pisa, the average speed of flocks vary from 16 to 24 km/h with a duration of 3 min (*right*)

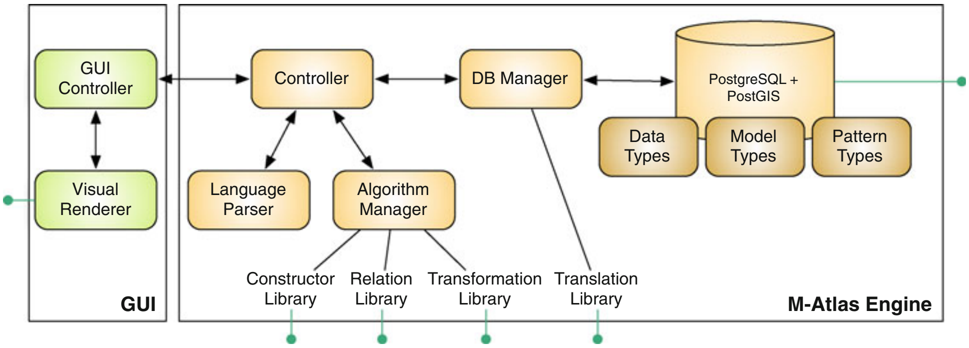


Fig. 26 M-Atlas system architecture. A query is submitted through the graphical interface to the *Controller* module, which coordinates the tasks performed by all other modules. The *Language Parser* analyzes the input query. Standard SQL queries are directly sent to the *Database Manager* and executed by the *Object-Relational DBMS*. All other M-Atlas queries are translated by the *Language Parser* into an execution plan, which combines both DB queries and calls to the methods provided by the *Algorithm Manager*. The results of a query is stored into the *ORDBMS* and possibly displayed, through the *Controller*, by the *Graphical User Interface*. The pins represent the modules which can be extended by the plug-in system

We found several flocks, some of them are depicted in Fig. 25. Most of the found flocks have three members. However, we have to recall that the number of trajectories belonging to a flock should be reported to a global scale (see Sect. 2) to have a measure of the real size of the car group. For example, a flock of three vehicles can be estimated as a group of about 150 cars at the global scale.

These results suggest that some traffic jams occurred in these areas, since the average velocity of the flocks is much less the normal speed in the roads where the flocks are located (highways, in this specific case). When several flocks are found in the same location, as in the case of the Pontedera area, this may indicate that these locations are usually interested by traffic congestions.

6 System architecture and performance evaluation

The architecture of M-Atlas is composed of two main components: the *Graphical User Interface*, supporting the visual analytic process, and the *M-Atlas Engine*, providing the full power of the data mining query language (see Fig. 26).

The architecture has been designed as a plug-in environment, where new models and patterns can be easily added, together with their mining algorithms. Extending the system requires four steps: (i) the new model/pattern type is introduced in the DB; (ii) the Translation Library of the DB Manager is extended with the access methods for the new type; (iii) the mining method associated with the new type is added to the Constructor Library; and (iv) the spatio-temporal primitives associated with the new type are added to the Relation Library. M-Atlas is being continuously extended with new functionalities; examples of system extensions are presented in [39]. A basic requirement for the architecture is minimizing memory usage during query execution. To this purpose, query results are managed, as far as possible, by reference in streaming fashion, i.e., by processing iteratively one set of rows of fixed size at a time, both during loading and storing. However, the system adapts to the memory policy of the various mining algorithms. Therefore, the memory consumption of most M-Atlas queries is constant, with the remarkable exception of the mining algorithms, which require multiple passes over data. While time complexity of the various mining methods is reported in Sect. 3.3, we

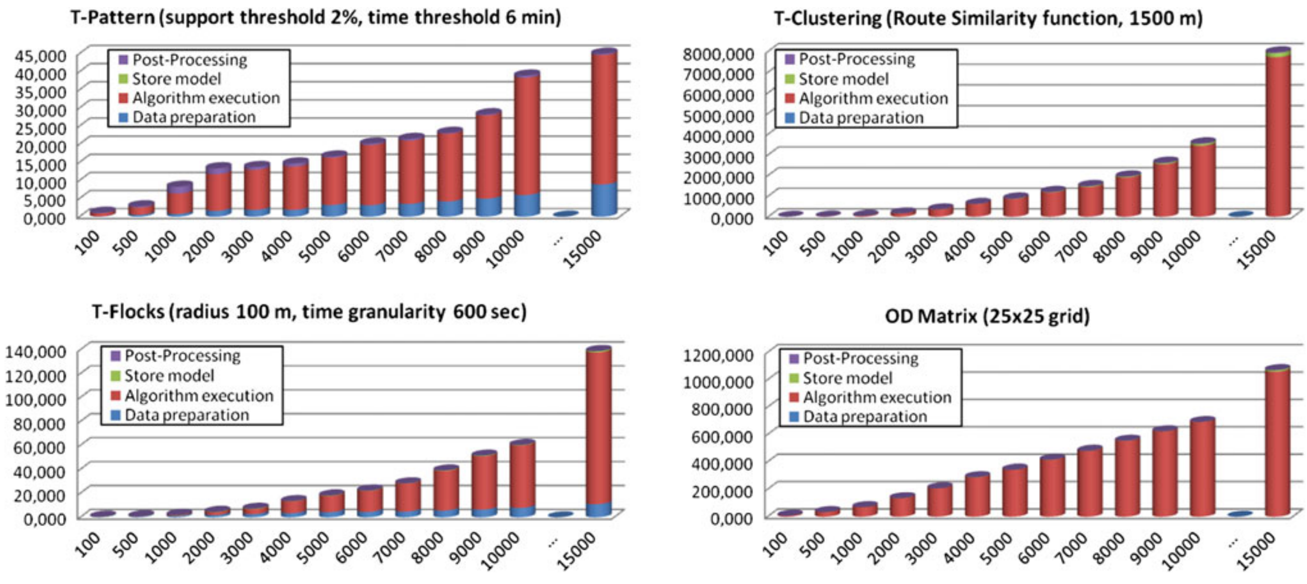


Fig. 27 Total execution time (in seconds) of model constructor queries for T-Patterns, T-Clustering, T-Flocks and T-O/D Matrix. *Different colors* of each *bar* indicate the fraction of time taken by: data retrieval and preparation (*blue*), mining algorithm execution (*red*), model storage (*green*), and post-processing for visualization (*purple*)

report here an empirical evaluation of the performance of system, to assess the real scalability of the execution time of the various model constructor queries. Figure 27 shows the runtime for each model constructor query execution (in seconds). Each chart is obtained as the average of 10 experiments, each repeated on 13 input trajectory data sets of increasing size and equal average trajectory length. In every case, performance scales in accordance with the theoretical complexity, indicating that the overhead introduced by the system is not predominant. This is confirmed by the observation that most of time is taken by the execution of the mining algorithm.

7 Related literature review

The theoretical model at the basis of M-Atlas is called Two-Worlds [40], and it has been inspired by the inductive database vision proposed by Mannila in [12]. Here, the main idea is that the results of the mining process, the models or patterns, are materialized in the database for further analysis. The Two-Worlds model is also inspired by the Three-Worlds model proposed in [23]. In this model, the Data World (representing the data to be mined) and the Models World (representing the extracted patterns) are linked by relations which connect Data to Patterns (representing the mining process) and Patterns to Data (representing the data belonging to the extracted model). The common aspect of these approaches is that there is a need to model the mining results at the same level as data objects to manipulate them further. The Two-Worlds theoretical framework and the associated query language are detailed in [40]. Based on this theoretical framework, M-Atlas is the result of the extension and the proper integration of several components presented separately in other works. The core of the M-Atlas architecture has its ancestor in Daedalus [33], evolved in the GeoPKDD system [30] along the duration of the GeoPKDD project. Daedalus was a first prototype of a system based on the Two-Worlds model for progressive querying and mining trajectory data; GeoPKDD system was an engineered version where we ran preliminary experiments on mobility data. However, the present work enhances previous prototypes considerably in several aspects. First of all, in M-Atlas a new language grammar has been defined and implemented, thus giving more expressive power in defining the mining queries. Moreover, an enhanced architecture has been designed with the objective of improving efficiency in the queries computation. Furthermore, Daedalus and GeoPKDD were built on top of the Hermes moving object database [34], while M-Atlas is based on PostGIS [38], extended with functions to manipulate trajectories. Another improvement is in the number of both data mining algorithms that are now plugged into the system and new ad hoc tools for trajectory statistics (such as the T-O/DMatrix tool). Finally, M-Atlas provides an improved graphical user interface where the query language is hidden to the user who may interact with the system by using visual metaphors. Other systems have been proposed in the literature to support the knowledge discovery process. Among them, it is worth mentioning the ATLaS system proposed by Zaniolo et al. in [44]. This system introduced a new programming language as a Turing-complete extension of SQL for mining operations. However, the two systems differ in several aspects. First of all, M-Atlas is specialized for trajectory data, while ATLaS is targeted to relational case. Secondly, ATLaS requires that mining algorithms are programmed directly using the internal language—thus, it is likely they become extremely complex, whereas M-Atlas allows an easy plug-in of mining algorithms and a query language to call them. An environment that shares with M-Atlas the objective of supporting the knowledge discovery form trajectory data is MoveMine introduced in [26] where authors realized a system that connects different trajectory mining algorithms. The main added value of M-Atlas respect to MoveMine is that M-Atlas is not only a platform for connecting different mining tools, but it is based on a theoretical framework where data and models mined from different algorithms may be manipulated and combined together. Furthermore, M-Atlas offers a data mining query language where progressive and interactive knowledge discovery processes can be defined. Another recent and interesting project related to mobility data mining is GeoLife [15]. It aims at building a sort of location-based social network considering the typical mobility experiences of the users. The construction of the social network is based on efficient retrieval of similar trajectories [10], on spatio-temporal data mining algorithms [46] and a recommender system [35]. The framework thus is more oriented to a direct interaction between the GeoLife systems and the end-user who may query directly his mobile devices. Moreover, the GeoLife tools do not furnish an advance methodology for traffic analysis, as M-Atlas. Instead, GeoLife techniques are mainly focussed to a mobile user that may query the provided system for directions or suggestions. In fact, the envisaged scenario for GeoLife is to provide a set of services accessible through the mobile user portable device. On the contrary, M-Atlas provides a platform and a methodology for movement analysis more addressed to a traffic analyst. A complementary research direction, related to the analysis of huge quantity of movement data, comes from the field of networks science. The main difference between the network science methods and the data mining relies in the fact that complex networks mainly analysis data from a global point of view, trying to find some general law that represent the movement. On the other hand, the data mining community is interested in finding local behaviors and patterns extracted from the data. The first proxy of human mobility used in this area was the data from a popular banknote tracking web site [9]. Later, large data sets of mobile phone call records were analyzed, to the purpose of discovering and validating the macro-level laws of human mobility, such as the law governing the distribution of traveled distances [19,36]. Applications of these findings concern the spreading patterns of phone viruses [45] and the analysis of the entropy and predictability of human mobility [37]. Compared with the work reported in this paper, network scientists did not address so far the problem of finding mobility patterns, or clusters, concerning subgroups of people or travels that exhibit specific behavior or deviate from typical behavior. Also, the GPS data sets studied in our paper, albeit smaller than typical phone call record data sets, is unique in its ability to represent travels, at the urban scale, with extremely fine spatio-temporal resolution.

8 Conclusions

We shared, in this paper, the lesson learned in our multi-year project on mobility data mining. In a nutshell, massive data sets of human trajectories are indeed a powerful basis for understanding mobility patterns at society-wide scale, provided that the complex analytical process needed to transform such raw data into high-level knowledge is adequately supported. We designed our querying and mining language and system M-Atlas precisely as the platform for the mobility knowledge discovery process and showed in this paper how it enables to answer challenging questions posed by the analysts of movement behavior.

Other important facets of M-Atlas have not been discussed in this paper, including (i) the privacy-preserving data publishing and mining techniques designed to transform trajectory data sets into anonymous forms in such a way that strong privacy-protection guarantees can coexist with high data utility [1,2,28]; (ii) the semantic annotation and interpretation of trajectory data and patterns with reference to domain ontologies specifying the background knowledge in particular contexts [7]; (iii) the analysis of different forms of mobility data, such as mobile phone call records, characterized by complementary weaknesses and strengths with respect to GPS trajectories [32].

Finally, many fascinating directions remain open for further research. One is the neverending quest for richer semantics in mobility data, sustained by the enhanced sensing capabilities of smart phones and next-generation mobile devices. Novel mining models and techniques are needed for semantic trajectories and associated background information, such as the underlying road network where movements take place. A second aspect is the emergence of data capturing not only movement but also the social relations between people, such as the mobile phone call records that allow to reconstruct, besides trajectories, also the "who-calls-whom" social network. Another example are the participatory location-based social networks, such as Gowalla and Foursquare. These data allow to begin studying the interplay between mobility patterns and the structure of social ties (see, e.g., [43]), and call for challenging extensions of our mining and querying framework. A third aspect is simulation: once the mobility patterns and profiles of a whole population have been learned (see, e.g., [41]), it is natural to investigate how to build on this basis large-scale simulations, capable of predicting realistic evolutions of complex social phenomena. As a final direction, we observe that mobility data are huge and come in a streaming fashion, so it is urgent to scale M-Atlas accordingly, overcoming the limitations of current spatio-temporal database systems. We need to create the equivalent of the trajectory database underlying M-Atlas *in the cloud*, with appropriate map-reduce primitives for mobility data mining.

Acknowledgments The authors wish to thank Rebecca Ong and Lorenzo Gabrielli for their technical support. This work has been possible with the scientific contributions of all researchers involved in the GeoPKDD European project. We also acknowledge Octotelematics S.p.A for providing the data sets. This work has been partially supported by the European FET-Open project LIFT (ICT-2009.8.0, grant no. 255957). Moreover, the valuable suggestions from the anonymous reviewers allowed us to reach a higher quality of work.

References

1. Abul, O., Bonchi, F., Nanni, M.: $\mathcal{N}$ever $\mathcal{W}$alk $\mathcal{A}$lone: Uncertainty for anonymity in moving objects databases. In: Proceedings of the 24nd IEEE International Conference on Data Engineering (ICDE'08) 2008
2. Abul, O., Bonchi, F., Nanni, M.: Anonymization of moving objects databases by clustering and perturbation. Inf. Syst. **35**(8), 884–910 (2010)
3. Agenzia Milanese Mobilità e Ambiente. Indagine sulla mobilità delle persone dell'area milanese (2006)
4. Andrienko, G., Andrienko, N., Wrobel, S.: Visual analytics tools for analysis of movement data. SIGKDD Explor. Newslett. **9**(2), 38–46 (2007)
5. Andrienko, G., Andrienko, N., Rinzivillo, S., Nanni, M., Pedreschi, D., Giannotti, F.: Interactive visual clustering of large collections of trajectories. In: IEEE Visual Analytics Science and Technology (VAST 2009) 3–10 (2009)
6. Ankerst, M., Breunig, M.M., Kriegel, H.P., Sander, J.: Optics: ordering points to identify the clustering structure. SIGMOD , 49–60 (1999)
7. Baglioni, M., de Macedo, J., Renso, C., Trasarti, R, Wachowicz, M. Towards semantic intepretation of movement data. In: AGILE Conference (2009)
8. Benkert, M., Gudmundsson, J., Hübner, F., Wolle, T.: Reporting flock patterns. Comput. Geom. Theory Appl. **41**(3), 111–125 (2008)
9. Brockmann, D., Hufnagel, L., Geisel, T.: The scaling laws of human travel. Nature **439**, 462 (2006)
10. Chen, Z., Heng Tao, S., Zhou, X., Zheng, Y., Xie, X.: Searching trajectories by locations: an efficiency study. In: Proceedings of the 2010 International Conference on Management of data, SIGMOD '10, pp. 255–266
11. Cudré-Mauroux, P., Wu, E., Madden, S.T.: An adaptive storage system for very large trajectory data sets. In: International Conference on Data Engineering, pp. 109–120 (2010)
12. De Raedt, L., Jaeger M., Lee, S.D., Mannila, H.: A theory of inductive query answering. In: IEEE International Conference on Data Mining (2002)
13. Ester, M., Kriegel, H-P., Sander, J., Xu, X.: A density-based algorithm for discovering clusters in large spatial databases with noise. In: Proceedings on the Knowledge Discovery in Databases Conference, pp. 226–231 (1996)
14. Gaffney, S., Smyth, P.: Trajectory clustering with mixture of regression models. In: Proceedings of the 5th International Conference on Knowledge Discovery and Data Mining, pp. 63–72. ACM (1999)
15. GeoLife Web Site http://research.microsoft.com/en-us/projects/geolife
16. GeoPKDD website. http://www.geopkdd.eu
17. Giannotti, F., Nanni, M., Pinelli, F., Pedreschi, D.: Trajectory pattern mining. In: Proceedings of the International Conference on Knowledge Discovery and Data Mining, pp. 330–339 (2007)
18. Giannotti, F., Pedreschi, D. (Eds.) Mobility, Data Mining and Privacy—Geographic Knowledge Discovery. Springer, Berlin (2008)
19. Gonzalez, M., Hidalgo, C.A., Barabási, A.-L.: Understanding individual human mobility patterns. Nature **453**, 779–782 (2008)
20. Gudmundsson, J., van Kreveld, M.: Computing Longest Duration Flocks in Trajectory Data. In: 14th Annual ACM International Symposium on Advances in Geographic Information Systems, pp. 35–42. New York: ACM
21. Güting, R.H., Böhlen, M.H., Erwig, M., Jensen, C.S., Lorentzos, N.A., Schneider, M., Vazirgiannis, M.: A foundation for representing and querying moving objects. ACM Trans. Database Syst. **25**(1), 1–42 (2000)
22. Imielinski, T., Mannila, H.: A database perspective on knowledge discovery. Commun. ACM **39**(11), 58–64 (1996)
23. Johnson, T., Lakshmanan, L.V.S., Ng, R.T.: The 3W model and algebra for unified data mining. In: VLDB '00: Proceedings of the 26th International Conference on Very Large Data Bases, pp. 21–32. Morgan Kaufmann, San Francisco (2000)
24. Kalnis, P., Mamoulis, N., Bakiras, S.: On discovering moving clusters in spatio-temporal data. In Proceedings of 9th International Symposium on Spatial and Temporal Databases (SSTD'05), pp. 364–381. Springer, Berlin (2005)
25. Lee, J.-G., Han, J., Whang, K.-Y.: Trajectory clustering: a partition-and-group framework. In: SIGMOD Conference, pp. 593–604 (2007)
26. Li, Z., Ji, M., Lee, J.-G., Tang, L.A., Yu, Y., Han, J., Kays, R.: Movemine: mining moving object databases. In: SIGMOD Conference, pp. 1203–1206 (2010)
27. Monreale, A., Pinelli, F., Trasarti, R, Giannotti, F.: Wherenext: a location predictor on trajectory pattern mining. In: 15th ACM SIGKDD Conference on Knoledge Discovery and Data Mining (KDD'09) (2009)
28. Monreale, A., Andrienko, G., Andrienko, N., Giannotti, F., Pedreschi, D., Rinzivillo, S., Wrobel, S.: Movement data anonymity through generalization. Trans. Data Privacy **3**(2), 91–121 (2010)
29. Nanni, M., Pedreschi, D.: Time-focused clustering of trajectories of moving objects. J. Intell. Inf. Syst. **27**(3), 267–289 (2006)
30. Nanni, M., Trasarti, R., Renso, C., Giannotti, F., Pedreschi, D.: Advanced knowledge discovery on movement data with the Geo-PKDD system. In: Proceedings of the 13th International Conference on Extending Database Technology, pp. 693–696 (2010)
31. Octotelematics http://www.octotelematics.it/
32. Olteanu, A.-M., Trasarti, R., Couronn, T., Giannotti, F., Nanni, M., Smoreda, Z., Ziemlicki, C.: GSM data analysis for tourism application. In: Proceedings of the 7th International Symposium on Spatial Data Quality (ISSDQ) (2011)
33. Ortale, R., Ritacco, E., Pelekis, N., Trasarti, R., Costa, G., Giannotti, F., Manco., Renso, C., Theodoridis, Y.: The DAEDALUS framework: progressive querying and mining of movement data. In: 16th ACM SIGSPATIAL International Symposium on Advances in Geographic Information Systems, ACM-GIS, p. 52 (2008)
34. Pelekis, N., Theodoridis, Y., Vosinakis, S., Panayiotopoulos, T.: Hermes: a framework for location-based data management. In: Proceedings of the International Conference on Extending Database Technology, pp. 1130–1134 (2006)

35. Quannan, L., Zheng, Y., Xing, X., Yukun, C., Wenyu, L., Wei-Ying, M.: Mining user similarity based on location history. In: Proceedings of the 16th ACM SIGSPATIAL International conference on Advances in Geographic Information Systems, GIS '08, vol. 34, pp. 1–34:10 (2008)
36. Song, C., Koren, T., Wang, P., Barabási, A.-L.: Modelling the scaling properties of human mobility. Nat. Phys. **7**, 713 (2010)
37. Song, C., Qu, Z., Blumm, N., Barabási, A.-L.: Limits of predictability in human mobility. Science **327**, 1018–1021 (2010)
38. The PostGIS spatial database http://postgis.refractions.net/
39. Trasarti, R.: Mastering the Spatio-Temporal Knowledge Discovery Process. PhD in Computer science, University of Pisa (2010)
40. Trasarti, R., Giannotti, F., Nanni, M., Pedreschi, D., Renso, C.: A query language for mobility data mining. Int. J. Data Warehousing Mining (IJDWM) **7**(1), 24–45 (2011)
41. Trasarti, R., Pinelli, F., Nanni, M., Giannotti, F.: Mining mobility user profiles for car pooling. In: 17th ACM SIGKDD Conference on Knoledge Discovery and Data Mining (KDD'11) (2011)
42. Wachowicz, M., Ong, R., Renso, C., Nanni, M.: Discovering moving flock patterns among pedestrians through spatio-temporal coherence. Int. J. Geograph. Inf. Sci. (in press)
43. Wang, D., Pedreschi, D., Song, C., Giannotti, F., Barabási, A.-L.: Human mobility, social ties and link prediction. In: 17th ACM SIGKDD Conference on Knoledge Discovery and Data Mining (KDD'11) (2011)
44. Wang, H., Zaniolo C., Atlas, L.C.: A small but complete sql extension for data mining and data streams. In: Proceedings of International Conference of Very Large Data Base, pp. 1113–1116 (2003)
45. Wang, P., Gonzalez, M., Hidalgo, C.A., Barabási, A.-L.: Understanding the spreading patterns of mobile phone viruses. Science **324**, 1071–1076 (2009)
46. Zheng, Y., Zhang, L., Xie, X., M, W.-Y.: Mining interesting locations and travel sequences from GPS trajectories. In: Proceedings of the 18th International Conference on World Wide Web, WWW '09, pp. 791–800