Detecting behavior types of moving object trajectories

Feda Almuhisen, Nicolas Durand, Mohamed Quafafou

Noname manuscript No.
(will be inserted by the editor)

Detecting Behavior Types of Moving Object Trajectories

Feda AlMuhisen · Nicolas Durand · Mohamed Quafafou

Received: date / Accepted: date

Abstract Trajectory mining is a challenging and crucial problem especially in the context of smart cities where many applications depend on human behaviors. In this paper, we characterize such behaviors by patterns, where each pattern type represents a particular behavior, e.g. emerging, latent, lost, etc. From GPS raw data, we introduce algorithms that allow computing a formal concept lattice which encodes optimal correspondences between hidden patterns and trajectories. In order to detect behaviors, we propose an algorithm that analyses the evolution of the discovered formal concepts over time. The method generates tagged city maps to easily visualize the resulting behaviors at different spatio-temporal granularity values. Refined or coarse analysis can thus be performed for a given situation. Experimental results using real-world GPS trajectory data show the relevance of the proposed method and the usefulness of the resulting tagged city maps.

Keywords Formal concepts · Frequent patterns · Behavior · Trajectories · Visualization

1 Introduction

The concept of Smart City has become essential due to its concern of improving everyday technology in urban life, developing city network infrastructures, building better intelligent transportation systems to optimize traffic and transportation flows, and providing innovative services for citizens. In this context, a lot of positioning technologies, sensor networks, and online social media are used and deployed in cities. These devices and applications generate, by nature, spatio-temporal data which are collected from smartphones carried by people, GPS tracking systems on cars, airplanes, and location-based services provided by social media. Analyzing such data is very important to understand user behaviors for applications such as human mobility understanding, smart transportation, and urban planning [28]. Trajectory analysis [40] is one of the mining tools that are used for this purpose. The analysis of trajectories is suited for a wide range of applications, for instance, monitoring, tracking, territorial management, and security. The handled spatio-temporal data are complex, rich and huge. They also inherit the geolocalization imprecision. Thus, they need to be analyzed by suitable methods based on clustering, classification, frequent patterns, etc. [23]. Moreover, these methods generate reports and visualize the results for easily understanding.

In this paper, we propose a new method based on formal concept analysis to detect behaviors over time and visualize them on tagged city maps automatically generated. In order to achieve this goal, we segment and label the GPS trajectories according to the specified spatio-temporal granularity values. Then, each trajectory is transformed and represented in a georeference area as a set of grids. After that, this discrete representation of trajectory is used to extract the hidden patterns. From these data, we construct a formal concept lattice to encode the correspondence between the discovered frequent patterns and trajectories. Thus, a formal concept is a pair (Y, X), where X is a pattern and Y is the set of trajectories that pass through the grids belonging to X. Using this relationship between trajectories and grids, we define different types of behaviors (latent, emerging, decreasing, jumping, and lost) to statistically characterize the dynamics and diversity

Aix Marseille Univ, Université de Toulon, CNRS, ENSAM, LSIS, Marseille, France
E-mail: firstname.lastname@lsis.org

of mobility in relation to urban spaces. Finally, we use these results to automatically generate maps which are tagged by different colors, whereas each color represents a given behavior. These maps can be generated at different spatio-temporal level to refine the users mobility analysis in a given area of the city. We have evaluated our method using the T-drive data set, which is a GPS real taxi trajectories collected by Microsoft Research Asia [36,34]. Our method is generic regarding the application domain of trajectories and can be applied to smart city problems related to moving objects information, e.g., smartphones, bicycles, animals, etc.

The rest of the paper is organized as follows. Section 2 defines notations and basic notions necessary for understanding the paper. Related works are discussed in Section 3. The proposed method that allows an automatic tagging of city maps using patterns is detailed in Section 4. The experimental results are presented in Section 5. Finally, we conclude and present some future work in Section 6.

2 Background

A trajectory is a sequence of geographical locations of a moving object on a defined time period. A set of n moving object trajectories $Trjs$={Trj_1, Trj_2, ..., Trj_{n-1}, Trj_n}, where each trajectory Trj_i has a different length, and can be defined as a sequence of geographical points returned at a specific time stamp t [39]. A trajectory Trj_i is equal to {p_1,p_2,...,p_{k-1},p_{k_i}} and each point $p_k \in Trj_i$ is defined in three-dimensional space, i.e., p_k=($latitude_k$, $longitude_k$, t_k), where ($latitude_k$, $longitude_k$) defines the geographical coordinate location pair at specific time t_k.

2.1 Frequent Patterns

The discovery of frequent patterns is an important task in data mining which has been discussed in [1]. It corresponds into finding the sets of items (i.e. attribute values) which appear simultaneously in at least a certain given number of transactions (i.e. objects) recorded in a database. The latter is represented by the formal context $\mathcal{D} = (\mathcal{T}, \mathcal{I}, \mathcal{R})$, where $\mathcal{T}$ consists of a set of transactions, $\mathcal{I}$ is a set of items, and $\mathcal{R} \subseteq \mathcal{T} \times \mathcal{I}$ is a binary relation. Each pair $(t, i) \in \mathcal{R}$ means that transaction $t \in \mathcal{T}$ has a relation with item $i \in \mathcal{I}$. A pattern is a subset of $\mathcal{I}$. We used a string character notation for the patterns, for example AB for $\{A, B\}$. They are implicitly sorted according to the lexicographic order. The cover of a pattern X is the set of transactions of $\mathcal{D}$ that are related to the items of X (i.e., transactions that contain X): $cover(X) = \{t \in \mathcal{T} \mid \forall i \in X, (t, i) \in \mathcal{R}\}$. The support of pattern X, i.e., support count, represents the number of transactions that contain X: $support(X) = |cover(X)|$. Let us remark that we can also express the support as a percentage of the total number of transactions in $\mathcal{D}$, i.e., $\frac{|cover(X)|}{|\mathcal{T}|} \times 100$. Pattern X is *frequent* if its support exceeds (or is equal to) a minimum threshold denoted *minsup*. The set of all-frequent patterns is denoted $\mathcal{FI}$ and is equal to $\{X \subseteq \mathcal{I} \mid support(X) \geq minsup\}$. The notion of frequent closed pattern has been proposed by [30]. The pattern X is closed if none of its supersets have the same support as X: $\forall Y \supset X$, $support(Y) < support(X)$. The frequent closed pattern is a pattern both closed and frequent. Let us note that the set of frequent closed patterns (denoted $\mathcal{FCI}$) can be used to generate all the frequent patterns and their supports, and $\mathcal{FCI} \subseteq \mathcal{FI}$.

Table 1: Example of transactional database

tid	items				
t_1	A	B		D	E
t_2	A	B		D	
t_3	A	B	C	D	
t_4		B	C	D	
t_5	A	B	C		

Table 2: Example of transactional data in different datasets

tid	items					tsid
t_1	A	B		D	E	$\mathcal{D}_1$
t_2	A	B		D		
t_3	A	B	C	D		$\mathcal{D}_2$
t_4		B	C	D		
t_5	A	B	C			

Example 1 *Table 1 shows an example of a transactional database representing a formal context with 5 transactions and 5 items (denoted as $A \ldots E$). If the minsup value is set to 3 (i.e., 60%) then pattern D is frequent because its support is 4 (t_1, t_2, t_3 and t_4). AE is not frequent since its support is 1 (t_1). We have $\mathcal{FI}$={A, B, C, D, AB, AD, BC, BD, ABD}. C is frequent but not closed. BC is frequent and closed because it corresponds to the maximum set of items shared by t_3, t_4 and t_5. $ABDE$ is closed but not frequent. We have $\mathcal{FCI} = \{B, AB, BC, BD, ABD\}$.*

2.2 Emerging Patterns

Emerging patterns have been introduced in [8] as patterns whose support increased in a significant way from one dataset to another. The data represented in Table 2 contains 5 items (denoted $A \ldots E$) and 5 transactions (denoted $t_1 \ldots t_5$) distributed in 2 datasets ($\mathcal{D}_1$

and $\mathcal{D}_2$). Let us remark that these datasets can be seen as classes. The support of a pattern in a dataset $\mathcal{D}$, denoted by $supp(X, \mathcal{D})$, is the number of transactions which contain X in $\mathcal{D}$. The quantitative evaluation of the contrast between data sets brought by a pattern is measured by its growth rate. The *growth rate* of a pattern X from $\mathcal{D}_j$ to $\mathcal{D}_i$ is defined by (1).

$$GR_{j,i}(X) = \frac{|\mathcal{D}_j|}{|\mathcal{D}_i|} \times \frac{supp(X, \mathcal{D}_i)}{supp(X, \mathcal{D}_j)}. \quad (1)$$

The more $GR_{j,i}(X)$ is high, the more X characterizes $\mathcal{D}_i$ compared to $\mathcal{D}_j$. Given a threshold value $mingr > 1$, a pattern X is an *emerging pattern* from $\mathcal{D}_j$ to $\mathcal{D}_i$ if $GR_{j,i}(X) \geq mingr$. If $GR_{j,i}(X) = \infty$, X is said to be a *jumping emerging pattern* (JEP) from $\mathcal{D}_j$ to $\mathcal{D}_i$.

Example 2 *Let us consider the example of Table 2 with* $mingr = 3$ *and verify if the D, ABD and ABDE are emerging from* $\mathcal{D}_2$ *to* $\mathcal{D}_1$ *or not. D is not an emerging pattern (*$GR_{2,1}(D) = 1.5$*). ABD is an emerging pattern because* $GR_{2,1}(ABD) = 3$*. ABDE is a JEP (*$GR_{2,1}(ABDE) = \infty$*).*

2.3 Formal Concepts

Formal concept analysis (FCA) is a theory of data analysis identifying the conceptual structures within data sets. The closure properties and its capability of discovering inherent hierarchical structures give it an advantage to be used to analyze the different pattern relationships and build better mining algorithms [16,31].

Given a formal context $\mathcal{D}$, there is a unique ordered set which describes the inherent lattice structure defining natural groupings and relationships among the transactions and their related items. This structure is known as a concept lattice or Galois lattice [12]. Each element of the lattice is a couple (T, I) which consists of a set of transactions (i.e., the *extent*) and a set of items (i.e., the *intent*). Each couple (called *formal concept*) must be a complete couple with respect to $\mathcal{R}$, which means that the following mappings (noted f and g) hold. For $T \subseteq \mathcal{T}$ and $I \subseteq \mathcal{I}$, we have: (1) $f(T) = \{i \in \mathcal{I} | \forall t \in T, (t, i) \in \mathcal{R}\}$ and (2) $g(I) = \{t \in \mathcal{T} | \forall i \in I, (t, i) \in \mathcal{R}\}$. $f(T)$ returns items common to all transactions $t \in T$, while $g(I)$ returns transactions that have at least all items $i \in I$. The idea of maximally extending the sets is formalized by the mathematical notion of *closure* in ordered sets. The operators $h_1 = f \circ g$ and $h_2 = g \circ f$ are the Galois closure operators. Let X be a pattern of items, if $h_1(X)=X$, then X is a *closed pattern*. A formal concept is composed of a closed pattern of items and of the set of transactions containing this closed pattern.

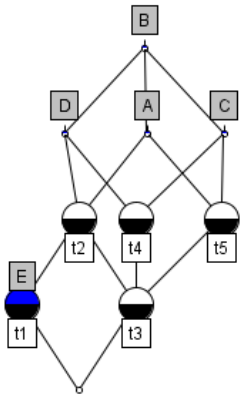

Fig. 1: Example of concept lattice.

The concept lattice, denoted by $\mathcal{L}$, is the set of formal concepts organized by the subsumption relation $\leq$. Given two formal concepts (T_1, I_1) and (T_2, I_2), $(T_1, I_1) \leq (T_2, I_2) \Longleftrightarrow T_1 \subseteq T_2$ et $I_2 \subseteq I_1$. (T_1, I_1) (respectively (T_2, I_2)) is a subconcept (respectively superconcept) of (T_2, I_2) (respectively (T_1, I_1)). The *frequent concept lattice* is formed using the formal concepts that have at least *minsup* transactions in their extent.

Example 3 *The line diagram in Figure 1 is an example of visual representation for the concept lattice computed from Table 1. Each node corresponds to a formal concept. The nodes upwards in this diagram are labeled by the attributes used to name the concept, and the nodes downwards are labeled by the objects used to name the concept. Note that the information reading for this diagram can be as follow: an object o is described by an attribute m if and only if there is an ascending path from the node named o to the node named by m. For example, the node labeled by* t_5 *corresponds to the formal concept* $(\{t_3, t_5\}, ABC)$*. The node labeled by D represents* $(\{t_1, t_2, t_3, t_4\}, BD)$*.*

In this paper, we propose a new method to detect and visualize user behavior by analyzing trajectories over time. The originality of our method is to leverage formal concepts, frequent patterns and emerging patterns. Before presenting our method, we present the related work in the next section.

3 Related Work

In this section we first review some of the existing work about trajectory analysis by focusing on pattern mining methods. Then, in order to encode the correspondence between the discovered patterns and trajectories, we study existing work on formal concepts in the context

of trajectory analysis. Finally, we review some work related to emerging patterns which capture trends over time, and are considered as particular behaviors.

3.1 Trajectory Analysis

Recently, a lot of studies focused on GPS trajectory analysis and providing extensive details about it [40, 39]. Other works [15,3] benefit from trajectory analysis and mining techniques to build better social networks and travel recommendation systems. Authors in [25] provided a survey that discusses recent results on mining mobility patterns from GPS data. In [7], the authors give a general view for recent publications that discuss trajectory analysis methods and patterns mining issues, by reviewing the methods and algorithms recently used in this field in detail and comparing their different approaches. From literature, trajectory analysis methods can be summarized as clustering method, which concern to find a group of trajectories that share common sub-traces to measure the closeness using distance measures between trajectories, then use density-based clustering methods in order to cluster the trajectories [20, 19]. In [14], the authors defined T-patterns (Trajectory patterns) as concise descriptions 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). They also proposed a method for extracting them using a sequential pattern mining algorithm. Alternatively, there is a growing interest in using semantic trajectory analysis techniques and methods which give a meaning to moving objects trajectory in terms of why, how, and for which purpose they move. [29] provided a survey that focuses on this term. Many tools and frameworks have also been developed to help in analyzing trajectory. An interesting tool, called $MoveMine$ introduced in [24], integrates functions that include moving objects pattern mining and trajectory mining.

In the literature there are many work proposed extracting trajectory patterns using spatial information and the aim differs according to the research goal. In [17], the authors used a graph-based mining algorithm to extract the frequent trajectory patterns in a spatial-temporal database by scanning the database to generate a mapping graph and trajectory information list. Then, they use a depth first algorithm to mine frequent patterns. We can note that this method outperforms two classes of algorithms: Apriori-based and PrefixSpan algorithms. In another way, the authors of [4] discussed mining flock patterns which are a type of spatio-temporal patterns that represent moving object groups close to each other in a certain time segment. They discussed algorithms related to extract this type of patterns [13] and proposed a modified algorithm called FPM. However, the limitation in such research was that it depends on the type of trajectory that we deal with as the users do not know which type of patterns they deal with. One of the works that has to be mentioned and solves this limitation is the work of [18] where the authors proposed a unifying framework for extracting trajectory patterns from asynchronous movement. It has the potential of navigating the patterns at all different levels of temporal tightness based on location, time, or both and defines what they called a $UT - pattern$ that helps in understanding the interaction between moving objects and group dynamics. In another way, in [6], the authors mine the frequent user's behavior from the socio-geographic data by finding frequent regions and change its representation from movements between points into movement between dense regions. Then, they extract trajectory patterns from these dense regions in the form of associative rules. It is important to mention that most of the previous discussed research use clustering and pattern discovery methods to extract frequent behavior, while in our method we extract the proposed behavior types by leveraging formal concept analysis and emerging pattern discovery.

In order to discretize the trajectory data and to establish areas, i.e., regions, we use the grid partitioning concept. Several studies have been conducted to process spatial data using grids. For example, in [22], the authors proposed a trajectory data clustering method based on dynamic grid division-merger limit. They map the trajectories into equal sized squares determined by experimental analysis. The authors of [33] divided the map of a city into uniform grids and compare their partitioning method within other methods. They conclude that the best prediction result is given by k-d tree based partitioning strategy due to the even distribution of points. Furthermore, the authors of [38] proposed a spatial iterative grid partition algorithm. The space is partitioned into uniform grids multiple times recursively until a desired grid granularity is reached, resulting in a more balanced number of points in each cell and better predictions. Unlike the previous work, we use uniform grids because we perform a spatio-temporal analysis by extracting the frequent closed patterns to detect changes and evolution between two time periods.

Finally, the paper [35] introduces a new concept called latent activity trajectory which captures different urban socio economic activities in different locations and a chronological order. The authors segment the regions according to the major roads, then cluster them into functional zones based on a topic modeling approach. More precisely, they used a kernel density estimation to identify the intensity for each functional

zone. Thus, only latent behaviors are discussed. Our work defines new types of behaviors, e.g., emerging, decreasing, lost, etc.

3.2 Formal Concepts and Trajectories

Formal concept analysis has been used in many applications and disciplines, such as knowledge discovery, information retrieval, etc. In [31], the authors give an extensive survey of all the papers that apply FCA in different domains such as biology, chemistry, etc. Our concern in these works is trajectory analysis. In [5], the authors provide a framework for analyzing sequential data structure based on FCA and they applied it on a data set of patient trajectories in a health-care system. The work in [16] also presented an approach for clustering trajectories of care for breast cancer using FCA. However, this work does not concern moving object trajectory analysis.

To the best of our knowledge, there is no research work that is based on FCA for analyzing moving objects. Our proposition is based on FCA in order to capture the maximal relations between trajectories and regions that they pass through.

3.3 Emerging Patterns and Applications

Emerging patterns have different applications. They are highly used for building classifiers. In [21], the authors proposed the $DeEP$ classifier, which is an instance-based classifier using the concept of emerging patterns. Researchers try to define new notions related to emerging patterns. For example, in [32], the authors define an exact condensed representation for the emerging patterns in a data set. They proposed their method of extracting emerging patterns that has the best growth rates and they called them strong emerging patterns (SEPs). In [11], the authors proposed adopting closed emerging patterns (CEPs) for characterizing classes. This method has been applied on hepatitis data to characterize the stage of liver fibrosis. Authors in [2] presented a hybrid classification method in the context of FCA, where each formal context has been augmented according to a class of objects. The class information is used to characterize the concepts whose extents include objects of a single class which allows extracting closed JEP. This approach has been applied in chemistry for classifying biological inhibitors. Our method is more general in terms of using the class of objects to specify the type of patterns. We also define other types of patterns and study the corresponding moving object behaviors.

4 FCA-based Behavior Discovery

In this section we present a novel method of trajectory analysis that generates moving object behavior tagged maps over time. The proposed method is based on frequent concept extraction and pattern evolution type detection. It is composed of three main steps: (1) Spatio-temporal preprocessing, (2) Pattern and behaviors extraction, and (3) Visualization of geolocalized behaviors (see Figure 2). In Step 1, trajectories are segmented, labeled, and mapped on a raster area G using local georeference according to time and spatial granularity values specified by the user. These discretized data are then used to extract hidden patterns and detect behaviors in the next step. In Step 2, for each time windows Δt (i.e., a pair of time granularity values (t_{i-1}, t_i)), we compute the frequent concept lattice $\mathcal{L}$ according to the minimal support threshold value $minsup$. Each node of $\mathcal{L}$, i.e., formal concept, encodes the correspondence between a pattern (set of grids) and the set of trajectories that pass through the grids of this pattern. Then, for each formal concept, we analyze the evolution of the pattern by computing an indicative value K. By given K, a minimal threshold value of emergence θ, and an error tolerance value ϵ, we detect the type of the pattern that belongs to a predefined set of types: $\{latent, emerging, decreasing, jumping, lost\}$. Each pattern type corresponds to a behavior of trajectories. Let us note that Steps 1 and 2 correspond to the off-line process (i.e., precomputed). Step 3 corresponds to the online process, where for user specified time windows Δt, we use the previous results to automatically generate maps which are tagged using colors, whereas each color represents a given behavior. Table 3 summarizes the important symbols.

4.1 Spatio-Temporal Preprocessing

Preprocessing the spatial trajectories is an important step in our methodology. It is performed on raw trajectory data to overcome the size problem. It is also needed to simplify and reduce the number of points with negligible error. These reduced trajectories are used later for improving the efficiency of the visualization process, also for the purpose of mining and analyzing trajectories to extract hidden patterns. In addition to that, the preprocessing step includes detecting and removing incomplete, inaccurate, incorrect, and out of range data points to generate vectors for the moving objects. It is also important to detect a certain time if this object stopped or moved. Therefore, stopping detection can be defined as, the observing of where an object spends a relatively large time without leaving a spatial area,

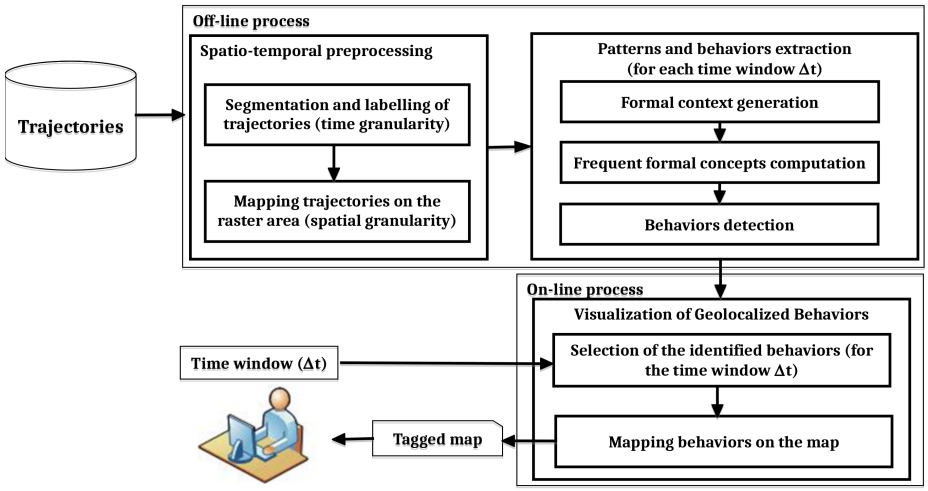

Fig. 2: Overview of the proposed approach.

Table 3: Important symbols

Symbol	Explanation
$Trjs$	Set of trajectories
trj	A trajectory
p	Geographical point
$long_p$, lat_p	Longitude and latitude of p
$LongD$ $LatD$	Longitude and latitude distance covered in degrees
$Longk$ $Latk$	Longitude and latitude distance covered in kilometers
C	Set of time granularity values
G	Raster area
G_{height} G_{width}	Dimensions of one grid (according to spatial granularity)
Δt	Time window (a pair of time granularity values)
G_{trj}	Set of grids corresponding to trj
$\mathcal{D}$	Formal context
$minsup$	Minimum support threshold value
θ	Minimal emergence threshold value
ϵ	Error tolerance value
$\mathcal{L}$	Frequent concept lattice
X	A pattern of grid cells
$FC = (Y, X)$	A formal concept
$K_i(Y, X)$	Evolution indicator value of X between time i-1 and time i

which is detected by having a sequential redundant spatial position on the generated vector over large period of time. A redundancy of spatial location (i.e., grid index, see Section 4.1.2) means either that the object is not moving or it is moving through the spatial defined coordinates. Thus, in this case we normalize the redundant spatial grid by one grid index in the generated vector. Our main motivation behind the elimination of the redundant grid index is that, in most cases, analysts are interested in the actual movement of an object. In addition, it will affect the pattern extraction later on.

The main steps to process the raw data are (1) Segmentation and labeling trajectories according to time granularity and (2) Mapping trajectories on a raster area according to spatial granularity.

4.1.1 Segment and Label Trajectories according to Time Granularity

Generally, GPS trajectory data are returned every 5-10 seconds and saved in servers. The problem is that the data to be analyzed are huge. So, we perform a trajectory segmentation by predefined time granularity, and we label each segment by its corresponding time class.

The time granularity value is chosen in order to segment each trajectory on set $Trjs$. It can be for instance minutes, hours, days, year, etc. Each value of granularity is called a class. We note $C = \{c_1, c_2, ..., c_{k-1}, c_k\}$ the set of classes values where $c_j < c_{j+1}$ for $1 \leq j \leq k-1$. Using the specified time granularity, each trajectory is segmented according to the defined classes. The trajectory of object n after segmentation by C can be defined as follow: $Trj = \{Trj_n^{c_1}, Trj_n^{c_2}, ..., Trj_n^{c_k}\}$. If $Trj_n^{c_j}$ is empty then this trajectory is deleted.

Example 4 *Assume that we have $n = 100$ moving objects in trajectory data $Trjs = \{Trj_1, Trj_2, Trj_3, \ldots, Trj_{100}\}$ collected for a duration of a month, for example in October. The time granularity value is the day. The trajectories by class obtained will be $Trj_n^{c_j}$ where $(1 \leq n \leq 100)$ and $1 \leq j \leq 31$. For instance, the class c_3 corresponds to the 3rd of October, which means that we keep the natural order of time sequence between the days of the month.*

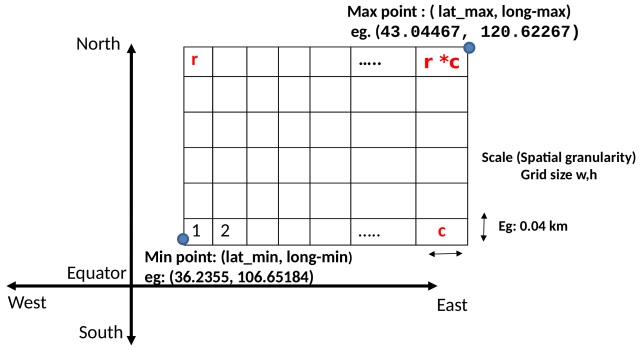

Fig. 3: Georeference system area according to spatial granularity.

Algorithm 1 Georeference area generation according to spatial granularity

Require:
$point_{min}$: $(latitude_{min}, longitude_{min})$ // Minimum geographical coordinate that covers the area we want to map
$point_{max}$: $(latitude_{max}, longitude_{max})$ // Maximal geographical coordinate that covers the area we want to map
G_{height} : Spatial granularity grid height dimension in kilometers
G_{width} : Spatial granularity grid width dimension in kilometers
Ensure:
G: Raster area containing ($Columns$*$Rows$) cells, also its latitude, longitude distance covered in degree (LatD and LongD)

1: Create a new raster area G ;
2: G.LongD ← $longitude_{max}$ - $longitude_{min}$; //Longitude distance covered in degree
3: G.LatD ← $latitude_{max}$ - $latitude_{min}$; // Latitude distance covered in degree
4: Latk ←$LatD * 111.320$; // Latitude distance covered in kilometer
5: Longk ← $111.320 * longD * cos((latD * \pi)/180)$; // Longitude distance covered in kilometer
6: G.Columns ← $Longk/G_{width}$; // Total number of columns in the grid area.
7: G.Rows ← $Latk/G_{height}$; // Total number of rows in the grid area
8: **return** G;

4.1.2 Mapping Trajectories on a Raster Area according to Spatial Granularity

The conversion of the geographic coordinates (*latitude*, *longitude*) values for the collected trajectories to an image of trajectory area map scale grid points can be achieved by applying two steps: (1) Generate a local georeference system area G for the purpose of mapping (see Algorithm 1), and (2) Map each trajectory to G (see Algorithm 2). Figure 3 illustrates the generated area.

Algorithm 2 Mapping a trajectory on a raster area according to spatial granularity

Require:
Trj : $\{p_1, p_2, ..., p_{k-1}, p_k\}$ // each point contains $(latp_k, longp_k)$
$point_{min}$: $(latitude_{min}, longitude_{min})$ // Minimum geographical coordinate in the raster area.
G : Raster area
Ensure:
G_{trj} : Set of grid indexes corresponding to Trj

1: $long_{step}$ ← $G.LongD/G.Columns$; // The longitude distance between each column in raster area in degree
2: lat_{step} ← $G.LatD/G.Rows$; // The latitude distance between each row in raster area in degree
3: G_{trj} ← {};
4: **for** $i = 1$ to k **do**
5: $LatDp_i$ ← $latp_i - latitude_{min}$; // Latitude distance in degree from the minimum point in raster area to $latp_i$
6: $LongDp_i$ ← $longp_i - longitude_{min}$; // Longitude distance in degree from the minimum point in raster area to $longp_i$
7: g_i ← $\lfloor LatDp_i/lat_{step} \rfloor * G.Columns + \lfloor LongDp_i/long_{step} \rfloor + 1$; // Grid index in the raster area for the corresponding geographical point
8: G_{trj} ← $G_{trj} \cup g_i$;
9: **end for**
10: **return** G_{trj};

Algorithm 1 has the following input values: two points ($point_{min}$ and $point_{max}$) that cover the geographical area, we want to map (these points are diagonal and can be extracted from the trajectory real data set), also a spatial granularity width (G_{width}) and height (G_{height}) for the grid dimension in kilometers. Lines 2 and 3, calculate the latitude and longitude degrees that cover the mapping area. A conversion of degrees into kilometer map scale is performed in lines 4 and 5 in order to divide the raster area by applying general calculation for distance measure above or below the equator. For example, the general calculation for distance measure in kilometers for country like China which is above east the equator as shown in Figure 3 are as follow: $Latk = LatD * 111.320$; $Longk = 111.320 * LongD * cos((LatD * \pi)/180)$. Finally, lines 6 and 7 calculate the number of columns and rows in the generated raster area according to a given grid dimension (spatial granularity) in kilometers, width and height. The algorithm returns the detail of the G raster area ($Columns$ * $Rows$) cells and its position.

Algorithm 2 shows the way to map a trajectory that contains geographical points into spatial grids in the generated raster area. As input, we need a $point_{min}$ which is the minimum point in the generated raster area, and also a trajectory Trj which we want to map. First, the algorithm calculates the grid step distance $long_{step}$ and lat_{step} value in degree for the covered lon-

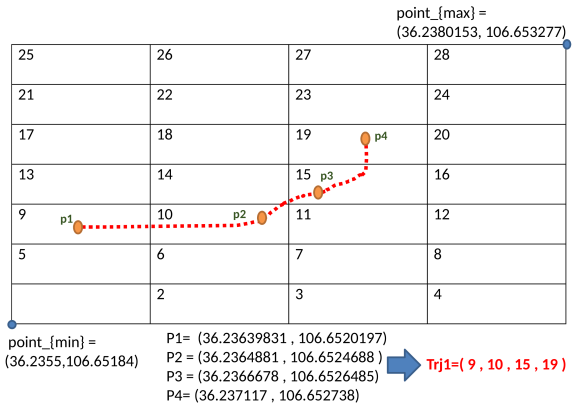

Fig. 4: Geo-mapped trajectory example.

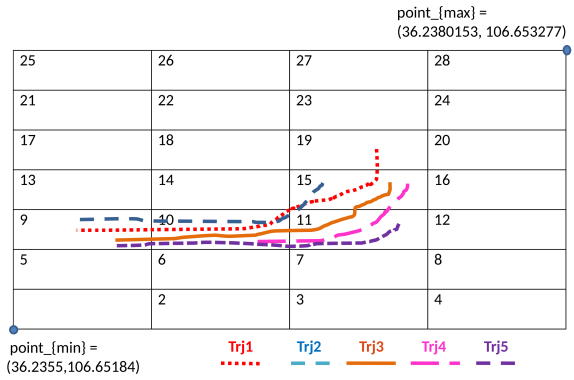

Fig. 5: Example of mapping.

gitude and latitude in the raster area (lines 1 and 2). Assuming that $(lat_{pi}, long_{pi})$ is a point from trajectory Trj, lines 5 and 6 show how to calculate the distances (in degree) that cover the area of mapping, based on the minimum point in the original local georeference discussed above. Then, line 7 calculates the grid index based on the columns and rows number of the whole raster area. The same steps are applied for each point of the trajectory. Finally, the algorithm returns the set of grid indexes G_{trj}.

Example 5 *Let us assume that we want to generate a local georeference for the following inputs:* $point_{min}$ = (36.2355, 106.65184), $point_{max}$ = (36.2380153, 106.653277), *and a spatial granularity* $G_{height} = G_{width}$ = *0.04km. Using Algorithm 1, we obtain 4 columns and 7 rows for the raster area as shown in Figure 4. Now, let us consider 4 points from* Trj_1 *as shown in Figure 4:* p_1 = (36.23639831, 106.6520197), p_2 = (36.2364881, 106.6524688), p_3 = (36.2366678, 106.6526485), *and* p_4 = (36.237117, 106.652738). *In order to return the corresponding grid indexes for these points we follow the steps in Algorithm 2. Let us take* p_1 *and* p_2. *We know that* $G.Columns = 4$, $G.Rows = 7$, $G.LongD = 0.001437$ *and* $G.LatD = 0.0025153$, *which are previously calculated using Algorithm 1. From these values, we calculate* lat_{step}, $long_{step}$ *which are the values of distance in degree according to the spatial granularity given in our georeference system. Here,* lat_{step} = 0.000359324 *and* $long_{step}$ = 0.000359329. *Then, we calculate the latitude and longitude degrees that cover the area corresponding to the given points* p_1 *and* p_2: $LatD_{p1}$=*0.000898311*, $LongD_{p1}$=*0.000179664*, $g_1 = 9$; $LatD_{p2}$=*0.000898311*, $LongD_{p2}$=*0.000538993*, $g_2 = 10$. *By applying the same procedure to the rest of the points we will get the following values:* $g_3 = 15$ *and* $g_4 = 19$. *The trajectory will be defined according to the generated grid indexes as follow* $Trj = \{9, 10, 15, 19\}$. *Let us note that in case of having redundancy occurrence of the grid index in the same trajectory sequentially, this means that either the object is stopped or its moving in the same spatial granularity coordinates. In this case, we keep only one of the redundant grid index.*
Figure 5 and Table 4 show an example of 5 trajectories after mapping using the local georeference system, within its corresponding class time in two specified labels as $\{Sunday, Monday\}$.

Table 4: Example of trajectory data presented within grid indexes and their corresponding classes (the time granularity is the day)

Trajectory	Grid indexes (G)				Class (C)
trj_1	9	10	15	19	*Sunday*
trj_2	9	10	15		*Sunday*
trj_3	9	10	11	15	*Monday*
trj_4		10	11	15	*Monday*
trj_5	9	10	11		*Monday*

4.2 Patterns and Behaviors Extraction

The behaviors detection associated to the trajectories over the time windows is performed in two main steps: (1) Extraction of the frequent concept lattice and (2) Detection of behavior type for each frequent formal concept. Algorithm 3 summarizes these steps. The next two sections discuss in detail the algorithms.

4.2.1 Frequent Concept Lattice Computation

The computation of the frequent concept lattice (see section 2.3) allows to obtain both the grid closed patterns and their corresponding sets of trajectories. The first step to compute this lattice is to create the formal context $\mathcal{D} = (\mathcal{T}, \mathcal{G}, \mathcal{R})$ for a specified time windows Δt (see Algorithm 3, line 1). $\mathcal{T}$ is the set of trajectories for the studied time window. $\mathcal{I}$ a set of grid index

Algorithm 3 Extraction of frequent concept lattice and detection of patterns type

Require:
Δt: Time window
$\mathcal{G} : \bigcup_{j=1}^{l} G_j$ which is consists of grid indexes, where $l = 1 \ldots n*m$ (m is the number of classes, and n is the number of initial trajectories)
minsup: Minimum support threshold value
θ: Minimal emergence threshold value
ϵ: Error tolerance value
Ensure:
$\mathcal{L}$: Frequent concept lattice
St: Set of detected types for the formal concepts of $\mathcal{L}$

1: Creation of formal context $\mathcal{D} = (\mathcal{T}, \mathcal{G}, \mathcal{R})$ according to the time window Δt;
2: Computation of frequent concept lattice $\mathcal{L}$ for $\mathcal{D}$ according to *minsup*;
3: Detection of the type of each pattern contained in the formal concepts of $\mathcal{L}$ according to θ and ϵ;
4: **return** $\mathcal{L}$ and St ;

$\mathcal{G} = \bigcup_{i=1}^{m} G_i$. For each trajectory t and each grid index $g \in G_t$, (t, g) is added to $\mathcal{R}$ to indicate that the trajectory t has a relation with grid index g. Then, the frequent concept lattice is computed according to the specified minimal threshold *minsup* (see Algorithm 3, line 2). Let us remark that $CHARML$ algorithm [37] coming from the pattern community is especially adapted for this computation.

Example 6 *Let us consider the example of Table 4 as the binary relation $\mathcal{R}$ of a formal context $\mathcal{D} = (\mathcal{T}, \mathcal{G}, \mathcal{R})$ where $\mathcal{T} = \{trj_1, trj_2, trj_3, trj_4, trj_5\}$, $\mathcal{G} = \{9, 10, 11, 15, 19\}$. The time granularity value is one day (Sunday, October 11, 2016 and Monday, October 12, 2016), and each trajectory is labeled by a time class $C \in \{Sunday, Monday\}$. Table 5 presents the formal concepts of the frequent concept lattice obtained with minsup = 2 (i.e., 40%). For example, the formal concept (trj_1, trj_2, trj_3, trj_5, ;9 10) means that the trajectories trj_1, trj_2, trj_3, and trj_5 all pass through the grid indexes 9 and 10 (9 10 is a frequent closed pattern).*

4.2.2 Detection of Behavior Types

After the computation of the frequent concept lattice for the specified time window, we detect the type of the closed pattern (i.e., the intent) of each frequent formal concept by studying the time classes of the corresponding trajectories (i.e., the extent). The type of a closed pattern reflects the behavior of users who have generated the corresponding trajectories. We define five types: *emerging*, *decreasing*, *latent*, *jumping* and *lost*. During the studied time window, the *emerging* type means that the presence of the pattern increased in the trajectories. The *decreasing* type means that the presence of the pattern decreased in the trajectories. The *latent* type means that the presence of the pattern is quite similar. The *jumping* type means the pattern which is absent, appeared. The *lost* type means that the pattern disappeared.

Let $FC = (Y, X)$ be a formal concept, θ be the minimal threshold value of emergence, and ϵ be the error tolerance. For the time window $[t_{i-1}, t_i]$, the type of the grid index pattern X (i.e., the intent of FC) is detected by computing K_i (see Equations 2 and 3).

$$K_i(Y, X) = \frac{count(Y, t_i)}{count(Y, t_{i-1})} \tag{2}$$

Where $count(Y, t_j)$ is the number of trajectories of Y (i.e., the cardinality if the extent of FC) labeled by the class corresponding to t_j.

$$\text{Type} = \begin{cases} \text{If } K_i(Y,X) = 0 \pm \epsilon \\ \quad \text{then the type of } X \text{ is } \textit{lost}. \\ (\text{If } ((K_i(Y,X) > \theta) \wedge (\theta = 1)) \\ \quad \vee ((K_i(Y,X) \geq \theta) \wedge (\theta > 1))) \\ \quad \text{then the type of } X \text{ is } \textit{emerging}. \\ \text{If } K_i(Y,X) < \theta \\ \quad \text{then the type of } X \text{ is } \textit{decreasing}. \\ \text{If } K_i(Y,X) = 1 \pm \epsilon \\ \quad \text{then the type of } X \text{ is } \textit{latent}. \\ \text{If } K_i(Y,X) = +\infty \\ \quad \text{then the type of } X \text{ is } \textit{jumping}. \end{cases} \tag{3}$$

Example 7 *Table 5 presents the result of computing K_i and of the behavior type detection for the frequent formal concepts obtained from the example of Table 4 with minsup = 2 (i.e., 40%), $\theta = 1$ and $\epsilon = 0$. Here, t_i corresponds to Monday and t_{i-1} to Sunday. Let us consider the first frequent formal concept (trj_1,trj_2,trj_3, trj_5 ; 9 10). Two trajectories are in the Sunday class (trj_1 and trj_2) and the other two are in the Monday class (see Table 4). Thus, K_i is equal to 1 and the type of the pattern {9 10} is latent. There are two latent patterns ({9 10} and {10 15}), three jumping patterns ({10 11}, {9 10 11} and {10 11 15}), one emerging pattern ({10}), and one decreasing pattern ({9 10 15}).*

4.3 Visualization of Geolocalized Behaviors

This section explains the method that we proposed in order to visualize the behaviors identified for a specified time window (see Section 4.2). The visualization method is based on the spatial granularity value in kilometers, the number of columns and rows in the raster area generated in Algorithm 1.

Table 5: Frequent formal concepts and type obtained for the example of Table 4 ($minsup = 2$ (40%), $\theta = 1$, $\epsilon = 0$)

Formal Concept	#Sunday	#Monday	K	Type
(trj_1 trj_2 trj_3 trj_5 ; 9 10)	2	2	1	LATENT
(trj_1 trj_2 trj_3 trj_4 ; 10 15)	2	2	1	LATENT
(trj_3 trj_4 trj_5 ; 10 11)	0	3	$+\infty$	JUMPING
(trj_3 trj_5 ; 9 10 11)	0	2	$+\infty$	JUMPING
(trj_3 trj_4 ; 10 11 15)	0	2	$+\infty$	JUMPING
(trj_1 trj_2 trj_3 trj_4 trj_5 ; 10)	2	3	1.5	EMERGING
(trj_1 trj_2 trj_3 ; 9 10 15)	2	1	0.5	DECREASING

Algorithm 4 Grid geographical dimension extraction in raster area

Require:
fc_{ty} : Formal concept containing three parts ($intent_{fc}$, $extent_{fc}$,$Type$)
G: Raster area contain ($columns$*$rows$) cells, also its latitude, longitude distance covered in degree ($LatD$ and $LongD$)
Ensure:
LG_{dim}: Ordered list containing the grid dimension min, max for formal concept intent values

1: $LG_{dim} \leftarrow \{\}$;
2: $long_{step} \leftarrow G.LongD/G.Columns$; //longitude distance between each column in raster area in degree
3: $lat_{step} \leftarrow G.LatD/G.Rows$; //latitude distance between each row in raster area in degree
4: **for all** $g \in intent_{fc}$ **do**
5: $row^g \leftarrow \lfloor (g-1)/G.Columns \rfloor + 1$; //calculate the grid row position
6: $col^g \leftarrow ((g-1) \mod G.Columns) + 1$; //calculate the grid column position
7: $lat^g_{min} \leftarrow G.lat_{min} + (lat_{step} * (row^g - 1))$; //calculate the grid g latitude min value
8: $lat^g_{max} \leftarrow lat^g_{min} + lat_{step}$; //calculate the grid g latitude max value
9: $long^g_{min} \leftarrow G.long_{min} + (long_{step} * (col^g - 1))$; //calculate the grid g longitude min value
10: $long^g_{max} \leftarrow long^g_{min} + long_{step}$; //calculate the grid g longitude max value
11: $add(LG_{dim}, \{(lat^g_{min}, long^g_{min}), (lat^g_{max}, long^g_{max})\})$; // add the pair grid min max longitude, latitude values to the end of the list LG_{dim}
12: **end for**
13: **return** LG_{dim};

Algorithm 4 shows the steps that we follow in order to extract the geographical dimensions for certain grid patterns of the formal concept (see Section 4.2), which we will use later for the visualization method. Let fc_{ty} be a frequent formal concept extended by its type, ($intent_{fc}$, $extent_{fc}$, $Type$). In order to start the visualization method we need to extract the real dimension for each grid in the formal concept. Each grid g in the $intent_{fc}$ is defined by min, max dimensions based on the raster area. Then each detected type is assigned to a specific color. Hence, when reading the grid index and the type, the color is given to the mapped grid dimension area in real map. It is important to note that the type of each grid item is chosen according to the majority vote value, which is calculated by considering all the formal concepts that contains this item in its intent. In the case of equality, several strategies can be used. For instance, we can randomly choose from the obtained types or we can give a special type color. Let us remark that we can use the neighbors of the grid item to pick the type according to the majority vote, but we may also face the equality problem within the item neighbors. We chose to use a special color to indicate ambiguous type.

Example 8 *In order to compute the grid dimensions for the concept* (trj_1, trj_2, trj_3, trj_5 ; 9 10) *in Table 4 in the raster area of Figure 5, we do: Grid* 9 *:* $row^g = 3$, $col^g = 1$. *The* $(lat^g_{min}, long^g_{min})$, $(lat^g_{max}, long^g_{max})$ *values for* 9 *are: {(36.23621865,106.65184), (36.23657797, 106.6521993)} and for grid* 10 *we have: {(36.23621865, 106.6521993), (36.23657797,106.6525587)}. So,* LG_{dim} *is equal to {(36.23621865, 106.65184), (36.23657797, 106.6521993)}, {(36.23621865, 106.6521993), (36.23657797, 106.6525587)}.*

5 Experiments

In this section, we summarize the experimental results obtained on a real world data set. We present information about this data and about the followed protocol. The results and discussion are then exposed.

5.1 Data

We have used a real data set (*T-drive*) which contains a large amount of taxi GPS trajectories collected by Microsoft Research Asia. These data have been used in [34,36]. The original data set contains the GPS trajectories for 10,357 taxis during the period of one week from Feb2 to Feb8 in 2008. Each file of the data set contains the trajectories of one taxi. Each line in a file has the following fields separated by comma: TaxiID, Date, Time, Longitude and Latitude. The total number of points in this data set is about 15 million and

the total distance of the trajectories reaches 9 million kilometers. The (longitude,latitude) was returned every 5 to 10 seconds and the coordinate system used for the returned longitude latitude was the Decimal Degree coordinate system (DD). In order to perform our experiments, we choose 222 taxis randomly.

5.2 Experimental Protocol

For the experiments, the methodology has been applied based on three factors, time granularity, minsup value, and spatial granularity. The protocol in general can be summarized as follows, the trajectories have been segmented according to time granularity, all the inaccurate, redundant data have been deleted, then the trajectories were mapped in raster area G by applying Algorithm 1 using the spatial granularity value. Each trajectory has been mapped on the raster area for finally having a vector which contains information about the grid numbers that each trajectory passes through its time window journey (see Algorithm 2). For each pair of time Δt, and for certain *minsup* value, we extracted the frequent formal lattice using $CHARML$ algorithm [37]. Then, the type of the pattern contained in each formal concept has been identified (see Algorithm 3). Let us recall that the type of a pattern corresponds to a certain behavior. Finally, we visualized the moving object behavior on a real map by applying Algorithm 4.

Several experiments were applied by changing time granularity (24 hours, 12 hours, 2 hours), spatial granularity (20, 40, 60 meters), and minsup (10, 20, 30, 40). For the type detection we set ϵ to 0 and θ to 1 in Algorithm 3.

In order to show the relevance and the interest of our method, we compare our results with those of a baseline method which uses only statistics instead of patterns. The baseline method computes, for each grid, the total number of the trajectories passing through it. The type detection and the visualization are performed as our method.

All the algorithms were implemented in Java, and all the experiments were performed on a Intel Core i7 2.50 GHz with 16 GB of memory.

5.3 Experimental Results

As mentioned before, a study according to different spatial granularities and zones within different pairs of days or half a day and different minsup values has been done to evaluate the proposed method. We present the quantitative and qualitative results of our method, and the comparison with the baseline method. Let us note that the corresponding days in the data are as follow: Feb2 = Saturday, Feb3 = Sunday, Feb4 = Monday, Feb5 = Tuesday, Feb6 = Wednesday, Feb7= Thursday, and Feb8 = Friday.

5.3.1 Quantitative Results

Tables 6, 7, and 8 and Figures 7, 6 and 8 show the statistical and graphical representation for our experimental results using different spatial granularities 20, 40, 60 meters respectively, for $minsup = 10$ (i.e, 2.3%). We observe that when we have high spatial granularity dimensions, we have a high number of patterns. This is clearly shown in Figure 9 where the total number of patterns increases whenever we have a high granularity value. Thus, if the spatial granularity value increases, the total number of grids decreases and the data density increases. Consequently, there are more possible common grids between the trajectories, so the probability to find patterns increases. We can conclude that there is a relation between spatial granularity and the total number of generated patterns. Moreover, we observe that the number of lost and jumping patterns is low compared to those of the other types. This is due to the time granularity which is coarse and does not allow to capture fast evolution of behaviors. A pattern is lost or jumping if it is not detected during both the two compared time intervals.

In order to study the effect of *minsup* values and types, we did another experiment where we change the minsup value for same spatial granularity for one week as shown in Tables 9, 10, and 11. We note from these statistics that when ever the *minsup* value increases the number of generated pattern decrease and some types may disappear. For example, if we compare the results in Table 8 with those in Table 9, we can notice that whithin minsup=20 (i.e., 4.5%) we have no jumping or lost patterns.

We have also applied the same protocol by changing the time segmentation. Instead of using 24 hours, we have set the time granularity value to 12 hours and 2 hours, while keeping spatial granularity value as 60 meters and *minsup* value as 2.3%. This percentage corresponds to 10 trajectories in the case of 12 hours (same as 24 hours), and 6 for 2 hours (because the total number of trajectories is lower). Tables 12 and 13 show the statistical results for 12 hours and 6 hours, respectively. In Table 13, Feb3(0:00-2:00) means from 0:00:00 to 1:59:59. If we compare these statistics within the ones in Table 8, we can conclude that when ever we decrease the time granularity, we get less total number of patterns. For example, in time window (Feb3,Feb4) Table 8, the total number of patterns is equal to 88,719 while

Table 6: Number of patterns according to the type, one week, granularity 20 meters and $minsup = 10$ (2.3%)

Time window	Emerging	Decreasing	Latent	Lost	Jumping	Total
(Feb2,Feb3)	1,699	363	252	0	15	2,329
(Feb3,Feb4)	2,810	2,011	846	1	3	5,671
(Feb4,Feb5)	1,674	2,755	819	2	0	5,250
(Feb5,Feb6)	323	1,594	238	4	0	2,159
(Feb6,Feb7)	105	289	74	1	0	469
(Feb7,Feb8)	7	95	18	0	0	120

Table 7: Number of patterns according to the type, one week, granularity 40 meters and $minsup = 10$ (2.3%)

Time window	Emerging	Decreasing	Latent	Lost	Jumping	Total
(Feb2,Feb3)	8,184	1,195	1,074	0	41	10,494
(Feb3,Feb4)	11,439	9,072	3,642	4	14	24,171
(Feb4,Feb5)	7,664	11,221	3,094	19	5	22,003
(Feb5,Feb6)	1,422	7,605	1,107	21	0	10,155
(Feb6,Feb7)	555	2,043	278	8	1	2,885
(Feb7,Feb8)	109	685	88	0	0	882

Table 8: Number of patterns according to the type, one week, granularity 60 meters and $minsup = 10$ (2.3%)

Time window	Emerging	Decreasing	Latent	Lost	Jumping	Total
(Feb2,Feb3)	20,729	2,220	2,041	0	312	25,302
(Feb3,Feb4)	40,442	34,797	13,431	9	40	88,719
(Feb4,Feb5)	20,767	40,327	10,118	55	3	71,270
(Feb5,Feb6)	2,658	15,668	1,933	90	0	20,349
(Feb6,Feb7)	1,290	3,105	524	1	1	4,921
(Feb7,Feb8)	237	1,421	148	1	0	1,807

for the same period of time on Table 12 it is equal to 18,444. The decreasing of the number of patterns occurs because the data density is lower. Let us remark that decreasing the *minsup* leads to more patterns.

5.3.2 Qualitative Results

In order to analyze the results, we captured a different interesting areas shown in Figure 10 in a real map, and visualized the detected behaviors. We give explanation for the results. For the purpose of clarity, we give each pattern type a key letter in the figures where each letter related to a color and type of pattern as follows: (R:Red, *decreasing*, (G:Green, *emerging*), (B:Blue, *latent*), (Y:Yellow, *lost*), (GR:Gray, *jumping* and (P:Pink, *ambiguous*). It is important to say that the percentage of ambiguous patterns (see Figure 11) which are the grids that hold two types of behavior at the same time was very low. For example, the percentage of ambiguous type is 2.24% for (Feb2(pm),Feb3(am)).

Some emerging patterns have been captured in the airport area (see Figure 12) and in the Administration area (see Figure 13) during the period of (Feb2,Feb3), and (Feb3,Feb4) respectively. This means that these regions face a higher taxi movement in the second day compared to the first day. The result in airport area can be clarified by the fact that Chinese New Year was in this week, where most of Chinese go back to celebrate the holiday. Taxi movement in administrative area was also higher in Feb4 compared to Feb3 and this can be explained by the fact that Feb4 is a working day while Feb3 is weekend. The majority of patterns shown in the temple of heaven area (see Figure 16) during (Feb6,Feb7) were also emerging, which means that in Feb7 more taxis passed this area comparing to the one in Feb6 where there was a worshiping ceremony at this temple for the Chinese New Year celebration.

In another way, some decreasing patterns have been identified in the Beijing railway station area during (Feb4,Feb5) (see Figure 14). We got the same observation for the Chaoyang golf club area during (Feb5,Feb6) (see Figure 15). It means that these areas have been facing a higher taxi movement in the first day compared to the one in the second day. For example, this behavior can be clarified in Beijing railway area by the fact that Feb4 is the first day of the week so the transportation used will be higher than any other day through the week.

Finally, a latent pattern has been discovered in the region of Temple of heaven during (Feb7,Feb8) (see Figure 17). It means that the taxis movement is approximately similar in these two days. This can be explained by the fact that in these two days temple of heaven (Tiantan park) which is the largest building for religious

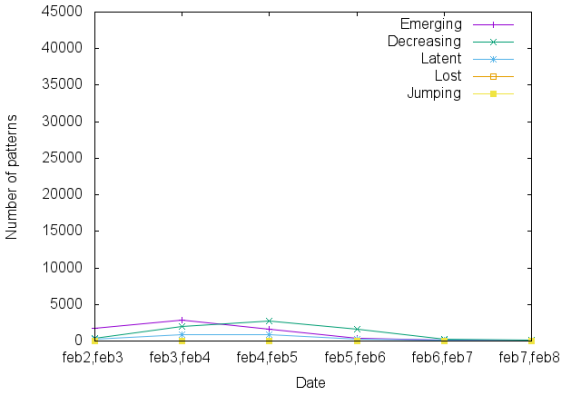

Fig. 6: Number of patterns for one week and spatial granularity = 20 meters.

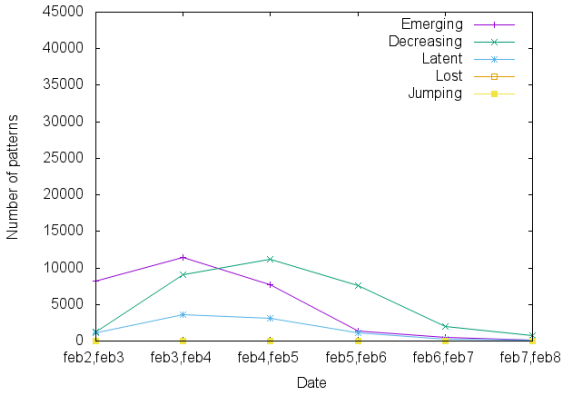

Fig. 7: Number of patterns for one week and spatial granularity = 40 meters.

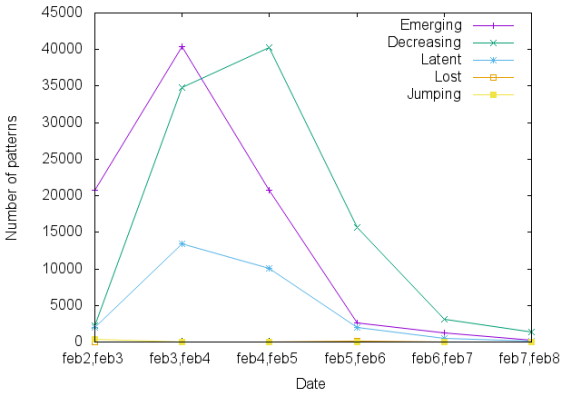

Fig. 8: Number of patterns for one week and spatial granularity = 60 meters.

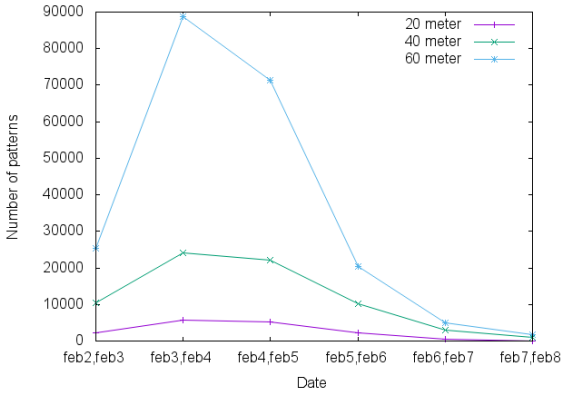

Fig. 9: Total number of patterns for one week and different spatial granularities.

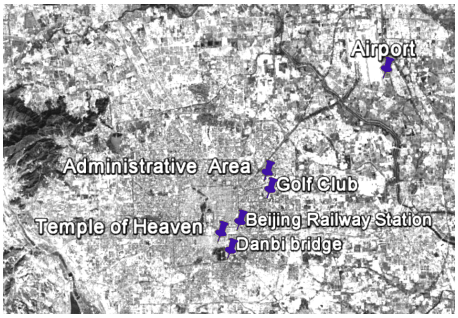

Fig. 10: Beijing areas of interest.

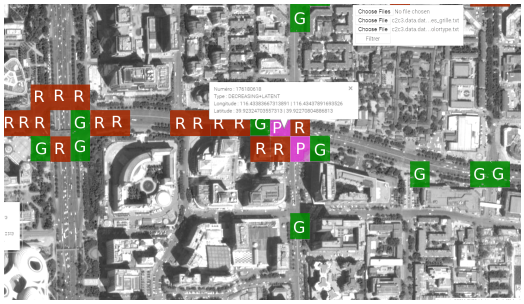

Fig. 11: Ambiguous pattern example.

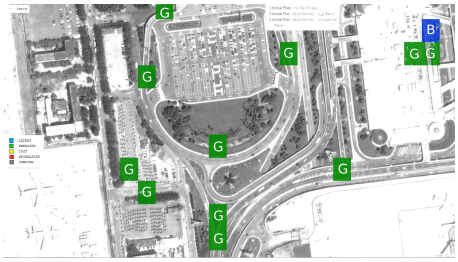

Fig. 12: Tagged map for the Airport area in (Feb2,Feb3).

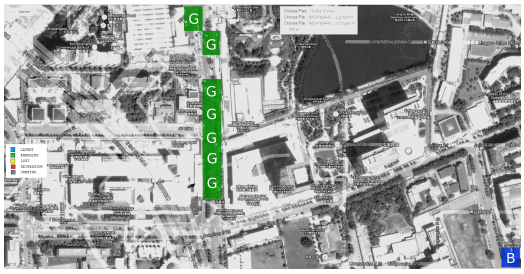

Fig. 13: Tagged map for the Administrative area (Feb3,Feb4).

Table 9: Number of patterns according to the type, one week, granularity 60 meters and $minsup = 20$ (4.5%)

Time window	Emerging	Decreasing	Latent	Lost	Jumping	Total
(Feb2,Feb3)	1,678	99	57	0	0	1,834
(Feb3,Feb4)	2,304	1,872	447	0	0	4,623
(Feb4,Feb5)	1,328	2,516	444	0	0	4,288
(Feb5,Feb6)	148	1,544	96	0	0	1,788
(Feb6,Feb7)	115	300	30	0	0	445
(Feb7,Feb8)	12	124	10	0	0	146

Table 10: Number of patterns according to the type, one week, granularity 60 meters and $minsup = 30$ (6.8%)

Time window	Emerging	Decreasing	Latent	Lost	Jumping	Total
(Feb2,Feb3)	389	9	5	0	0	403
(Feb3,Feb4)	587	422	75	0	0	1,084
(Feb4,Feb5)	261	618	76	0	0	955
(Feb5,Feb6)	26	303	10	0	0	339
(Feb6,Feb7)	23	36	3	0	0	62
(Feb7,Feb8)	0	13	0	0	0	13

Table 11: Number of patterns according to the type, one week, granularity 60 meters and $minsup = 40$ (9%)

Time window	Emerging	Decreasing	Latent	Lost	Jumping	Total
(Feb2,Feb3)	103	1	0	0	0	104
(Feb3,Feb4)	197	140	22	0	0	359
(Feb4,Feb5)	69	207	18	0	0	294
(Feb5,Feb6)	4	66	1	0	0	71
(Feb6,Feb7)	2	4	0	0	0	6
(Feb7,Feb8)	0	0	0	0	0	0

worshiping in China also a tourist attraction where having the chinese new year ceremonies.

To study the effect of time granularity, we compare Figure 18 which shows the behavior in the region of Beijing Railway during the period of (Feb3,Feb4) where time granularity equals day, within Figures 19 and 20 where the time granularity value is 12 hours. We do not visualize the results obtained for 2 hours because the total number of discovered patterns is too low with no important conclusion.

It is clear that when ever we have a smaller time granularity value, we will get more homogeneous behaviors. For example, in Figure 18, the whole behavior of the area is not really clear compared to Figure 19 where the majority of the patterns are emerging. This means that the area face higher taxi movements afternoon compared to the morning period. This can also be verified by Figure 20. We can observe a decreasing behavior which confirms that the traffic on Feb3 pm was higher compared to the morning period of Feb3 and Feb4. We can explain that by, Feb3 is Sunday pm and all the citizens use transportation at the end of weekend or holiday.

5.3.3 Comparative Results

Tables 14, 15 and 16 present the distribution of the grids according to behavior types computed by the baseline method. We observe that the number of jumping and lost grids is very high compared to the number of the other types. These two types are dominant and do not allow to identify the others. Moreover, the total number of grids to visualize is high, which does not help to analyze the results. As we can see in Figure 21, a lot of grids are tagged as "lost" and the other behavior types are hidden. If the number of labeled grids is too high, the map can be impractical to compute and exploit.

In fact, our method avoids these problems as the discovery of frequent closed patterns, i.e., contained in formal concepts, allows us to tune the importance of the considered trajectories by using the *minsup* threshold value. The more the *minsup* value is high, the more the discovered patterns are important and in the same time their number is smaller. In conclusion, our method thus detect only the important behavior types and not all types.

6 Conclusion

Trajectory analysis is a challenging problem especially in the urban context as the world's population will in-

Table 12: Number of patterns according to the type, one week, spatial-granularity 60 meters, time granularity 12 hours and $minsup = 10$ (2.3%)

Time window	Emerging	Decreasing	Latent	Lost	Jumping	Total
Feb2(am),Feb2(pm)	0	0	0	0	797	797
Feb2(pm),Feb3(am)	394	2,135	206	33	0	2,768
Feb3(am),Feb3(pm)	4,276	267	171	0	128	4,842
Feb3(pm),Feb4(am)	833	6,087	653	69	1	7,643
Feb4(am),Feb4(pm)	4,557	796	515	2	89	5,959
Feb4(pm),Feb5(am)	135	3,767	174	125	1	4,202
Feb5(am),Feb5(pm)	4,339	144	161	2	48	4,694
Feb5(pm),Feb6(am)	24	3,358	35	153	0	3,570
Feb6(am),Feb6(pm)	78	889	66	5	0	1,038
Feb6(pm),Feb7(am)	11	695	17	21	0	744
Feb7(am),Feb7(pm)	356	24	15	0	12	407
Feb7(pm),Feb8(am)	6	275	7	24	0	312
Feb8(am),Feb8(pm)	63	6	8	0	1	78

Table 13: Number of patterns according to the type, Feb3, spatial-granularity 60 meters, time granularity 2 hours and $minsup = 6$ (2.3%)

Time window	Emerging	Decreasing	Latent	Lost	Jumping	Total
Feb3(0:00-2:00),Feb3(2:00-4:00)	10	38	2	0	4	54
Feb3(2:00-4:00),Feb3(4:00-6:00)	7	36	5	2	0	50
Feb3(4:00-6:00),Feb3(6:00-8:0)	4	7	0	0	0	11
Feb3(6:00-8:00),Feb3(8:00-10:00)	13	0	0	0	4	17
Feb3(8:00-10:00),Feb3(10:00-12:00)	14	8	1	0	0	23
Feb3(10:00-12:00),Feb3(12:00-14:00)	23	12	5	0	0	40
Feb3(12:00-14:00),Feb3(14:00-16:00)	44	24	7	0	0	75
Feb3(14:00-16:00),Feb3(16:00-18:00)	47	47	13	0	0	107
Feb3(16:00-18:00),Feb3(18:00-20:00)	23	23	11	0	0	57
Feb3(18:00-20:00),Feb3(20:00-22:00)	21	13	3	0	0	37
Feb3(20:00-22:00),Feb3(22:00-00:00)	41	53	7	0	0	101

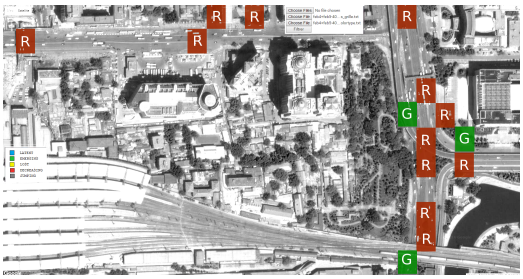

Fig. 14: Tagged map for the Beijing Railway station area (Feb4,Feb5).

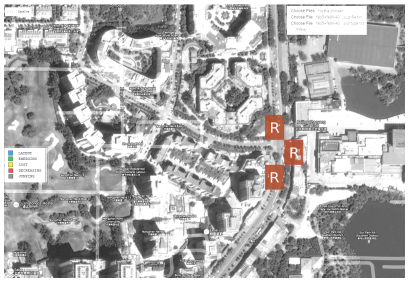

Fig. 15: Tagged map for the Chaoyang Golf club area (Feb5,Feb6).

crease significantly in the next few decades. The automatic generation of on demand reports, that aggregate results of trajectory analysis and the visualization of geolocalized information, is an important issue that will support urban management, adapted policy definition, public security, etc.

In this paper, we have proposed a new method based on frequent formal concept extraction and the analysis of their evolution over time in order to detect behaviors. We have proposed five types of behavior: emergent, latent, decreasing, jumping, and lost. Our method generates tagged city maps at different spatio-temporal granularity values. Therefore, refined or coarse analysis can be performed for a given situation. Experimental results using real-world GPS trajectory data have shown the relevance of the proposed method and the usefulness of the resulted tagged maps.

In future work, we will extend our system by predicting user behavior in a given area during the next time window. We will also use our recent works on approximation of frequent itemset borders [10,9] and on probabilistic topic models [27,26] for enrichment and trajectory recommendation purposes.

Fig. 16: Tagged map for the Temple of heaven area (Feb6,Feb7).

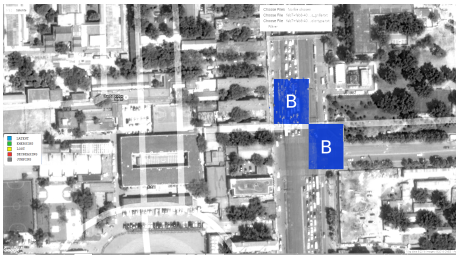

Fig. 17: Tagged map for the Temple of Heaven and Danbi Bridge area (Feb7,Feb8).

Table 14: Distribution of the grids according to the type, one week, spatial-granularity 60 meters, and time granularity 24 hours (Baseline method)

Time window	Emerging	Decreasing	Latent	Lost	Jumping	Total
(Feb2,Feb3)	12,687	3,613	6,437	11,650	26,465	60,852
(Feb3,Feb4)	10,388	9,546	8,830	20,438	21,590	70,792
(Feb4,Feb5)	7,514	11,538	9,150	22,152	25,244	75,598
(Feb5,Feb6)	3,501	11,357	7,477	31,111	18,827	72,273
(Feb6,Feb7)	3,885	6,157	7,096	24,024	18,445	59,607
(Feb7,Feb8)	2,517	5,063	5,659	22,344	16,747	52,330

Table 15: Distribution of the grids according to the type, one week, spatial-granularity 60 meters, and time granularity 12 hours (Baseline method)

Time window	Emerging	Decreasing	Latent	Lost	Jumping	Total
Feb2(am),Feb2(pm)	0	0	0	0	34,372	34,372
Feb2(pm),Feb3(am)	2,055	7,156	4,454	20,707	10,808	45,180
Feb3(am),Feb3(pm)	8,997	1,523	4,248	9,705	24,791	49,264
Feb3(pm),Feb4(am)	2,214	9,555	4,984	22,806	11,373	50,932
Feb4(am),Feb4(pm)	9,145	2,277	4,871	11,833	22,339	50,465
Feb4(pm),Feb5(am)	1,504	8,681	4,256	24,191	11,173	49,805
Feb5(am),Feb5(pm)	7,837	1,736	4,659	11,382	27,908	53,522
Feb5(pm),Feb6(am)	658	6,506	3,294	31,682	8,961	51,101
Feb6(am),Feb6(pm)	1,104	3,993	4,096	21,827	10,226	41,246
Feb6(pm),Feb7(am)	458	3,316	2,975	24,271	7,144	38,164
Feb7(am),Feb7(pm)	2,984	530	2,779	7,600	21,707	35,600
Feb7(pm),Feb8(am)	414	2,721	2,568	22,297	7,655	35,655
Feb8(am),Feb8(pm)	1,715	558	2,452	8,633	16,761	30,119

Acknowledgments

The authors would like to thank the anonymous reviewers for comments and suggestions on earlier drafts of this paper.

References

1. Agrawal, R., Imielinski, T., Swami, A.: Mining Association Rules between Sets of Items in Large Database. pp. 207–216. Washington DC, USA (1993)
2. Asses, Y., Buzmakov, A., Bourquard, T., Kuznetsov, S.O., Napoli, A.: A hybrid classification approach based on fca and emerging patterns-an application for the classification of biological inhibitors. In: Int. Conf. on Concept Lattices and Their Applications, pp. 211–222. Fuengirola, Spain (2012)
3. Bao, J., Zheng, Y., Wilkie, D., Mokbel, M.: Recommendations in location-based social networks: a survey. GeoInformatica **19**(3), 525–565 (2015)
4. Benkert, M., Gudmundsson, J., Hübner, F., Wolle, T.: Reporting flock patterns. Computational Geometry **41**(3), 111 – 125 (2008)
5. Buzmakov, A., Egho, E., Jay, N., Kuznetsov, S.O., Napoli, A., Raïssi, C.: Fca and pattern structures for mining care trajectories. In: Int. Workshop on What Can FCA Do for Artificial Intelligence, pp. 7–14. Beijing, China (2013)
6. Cesario, E., Comito, C., Talia, D.: Trajectory Data Analysis Over a Cloud-Based Framework for Smart City Analytics, pp. 143–162. Springer (2014)
7. Chen, L., Liu, L., Chen, B., Jia, H.: Research on patterns mining method for moving objects. Advanced Science and Technology Letters **123**, 200–205 (2016)
8. Dong, G., Li, J.: Efficient Mining of Emerging Patterns: Discovering Trends and Differences. In: ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining, pp. 43–52. San Diego, California, USA (1999)

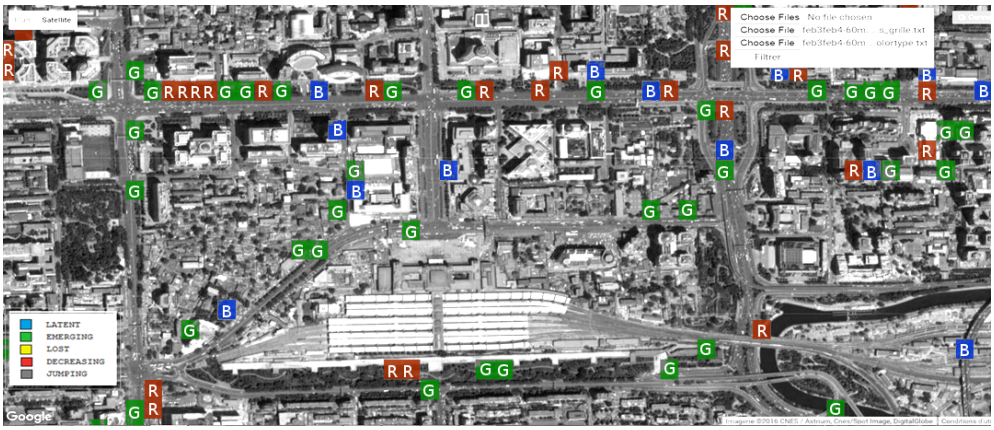

Fig. 18: Tagged map for Beijing Railway station area in (Feb3,Feb4).

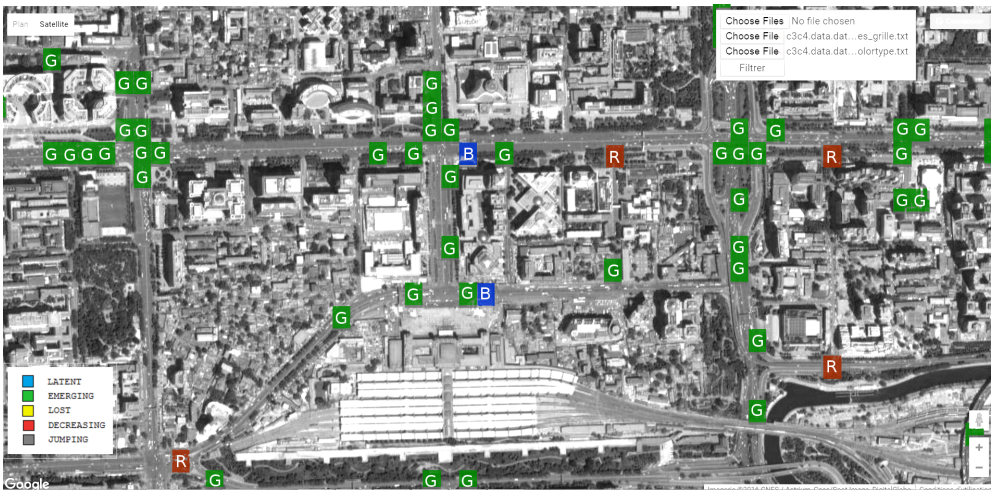

Fig. 19: Tagged map for Beijing Railway station area in (Feb3(am),Feb3(pm)).

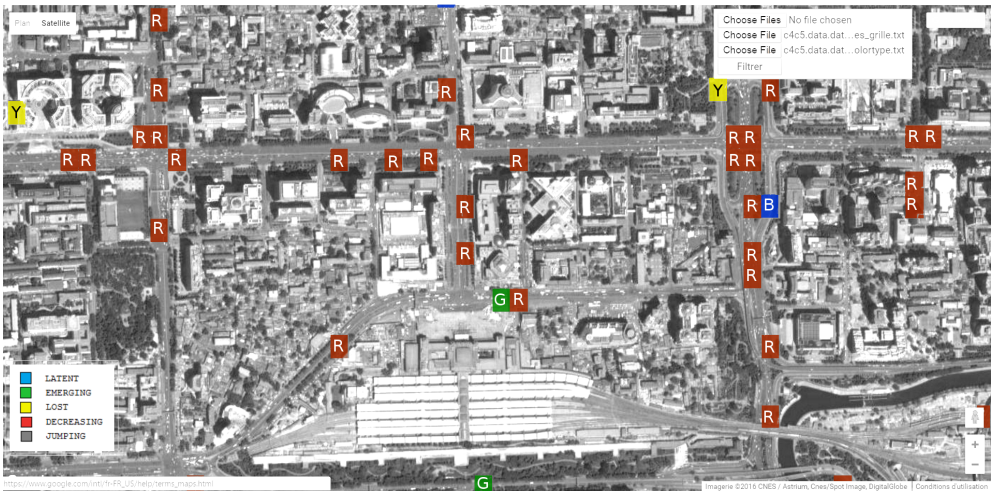

Fig. 20: Tagged map for Beijing Railway station area in (Feb3(pm),Feb4(am)).

Table 16: Distribution of the grids according to the type, Feb3, spatial-granularity 60 meters, and time granularity 2 hours (Baseline method)

Time window	Emerging	Decreasing	Latent	Lost	Jumping	Total
Feb3(0:00-2:00),Feb3(2:00-4:00)	188	463	796	6,794	3,142	11,383
Feb3(2:00-4:00),Feb3(4:00-6:00)	64	154	430	3,941	2,022	6,611
Feb3(4:00-6:00),Feb3(6:00-8:0)	41	57	297	2,275	1,972	4,642
Feb3(6:00-8:00),Feb3(8:00-10:00)	110	39	391	1,827	5,409	7,776
Feb3(8:00-10:00),Feb3(10:00-12:00)	571	267	1,098	4,013	9,185	15,134
Feb3(10:00-12:00),Feb3(12:00-14:00)	1,035	816	2,087	7,183	8,348	19,469
Feb3(12:00-14:00),Feb3(14:00-16:00)	1,229	1,050	2,253	7,754	7,924	20,210
Feb3(14:00-16:00),Feb3(16:00-18:00)	1,352	1,177	2,222	7,705	8,197	20,653
Feb3(16:00-18:00),Feb3(18:00-20:00)	1,280	1,388	2,466	7,814	8,487	21,435
Feb3(18:00-20:00),Feb3(20:00-22:00)	1,274	1,297	2,387	8,663	7,934	21,555
Feb3(20:00-22:00),Feb3(22:00-00:00)	1,015	1,153	2,039	8,685	6,701	19,593

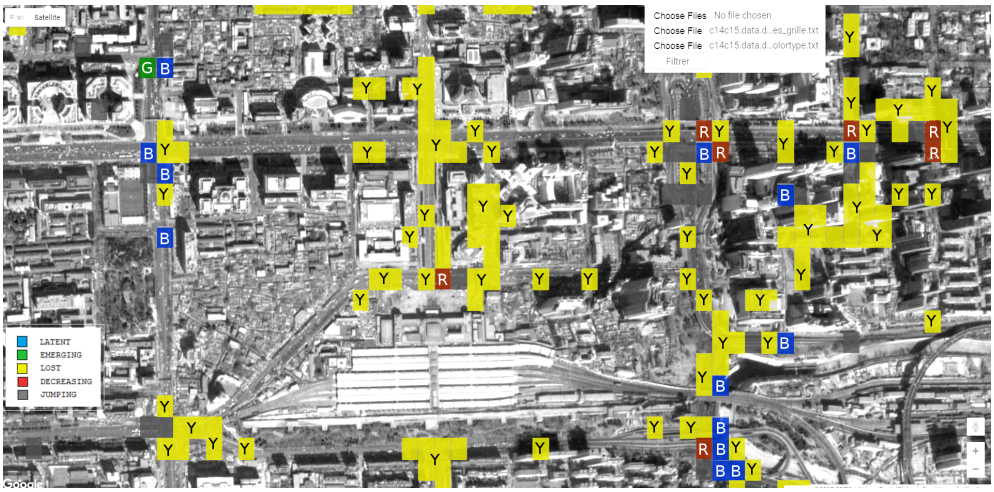

Fig. 21: Tagged map for Beijing Railway station area in (Feb3(14:00-16:00), Feb3(16:00-18:00)).

9. Durand, N., Quafafou, M.: Approximation of Frequent Itemset Border by Computing Approximate Minimal Hypergraph Transversals. In: Int. Conf. on Data Warehousing and Knowledge Discovery, pp. 357–368. Munich, Germany (2014)
10. Durand, N., Quafafou, M.: Frequent Itemset Border Approximation by Dualization. Transactions on Large-Scale Data- and Knowledge-Centered Systems (TLDKS) **26**, 32–60 (2016)
11. Durand, N., Soulet, A.: Emerging Overlapping Clusters for Characterizing the Stage of Liver Fibrosis. In: ECML/PKDD'05 Discovery Challenge on Hepatitis Data, pp. 139–150. Porto, Portugal (2005)
12. Ganter, B., Wille, R.: Formal concept analysis: mathematical foundations. Springer Science & Business Media (2012)
13. Geng, X., Uno, T., Arimura, H.: Trajectory pattern mining in practice - algorithms for mining flock patterns from trajectories. In: Int. Conf. on Knowledge Discovery and Information Retrieval and the Int. Conf. on Knowledge Management and Information Sharing, pp. 143–151. Vilamoura, Algarve, Portugal (2013)
14. Giannotti, F., Nanni, M., Pinelli, F., Pedreschi, D.: Trajectory pattern mining. In: ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining, pp. 330–339. San Jose, California, USA (2007)
15. Hu, Y., Yang, Y., Huang, B.: A comprehensive survey of recommendation system based on taxi gps trajectory. In: Int. Conf. on Service Science, pp. 99–105. Weihai, China (2015)
16. Jay, N., Nuemi, G., Gadreau, M., Quantin, C.: A data mining approach for grouping and analyzing trajectories of care using claim data: the example of breast cancer. BMC Medical Informatics and decision making **13**(1), 130 (2013)
17. Lee, A.J., Chen, Y.A., Ip, W.C.: Mining frequent trajectory patterns in spatial-temporal databases . Information Sciences **179**(13), 2218 – 2231 (2009)
18. Lee, J.G., Han, J., Li, X.: A unifying framework of mining trajectory patterns of various temporal tightness. IEEE Trans. on Knowl. and Data Eng. **27**(6), 1478–1490 (2015)
19. Lee, J.G., Han, J., Li, X., Gonzalez, H.: Traclass: Trajectory classification using hierarchical region-based and trajectory-based clustering. Proc. VLDB Endowment **1**(1), 1081–1094 (2008)
20. Lee, J.G., Han, J., Whang, K.Y.: Trajectory clustering: A partition-and-group framework. In: ACM SIGMOD Int. Conf. on Management of Data, pp. 593–604. Beijing, China (2007)
21. Li, J., Dong, G., Ramamohanarao, K.: Instance-based classification by emerging patterns. In: European Conf. on Principles of Data Mining and Knowledge Discovery, pp. 191–200. Lyon, France (2000)

22. Li, J., Yang, M., Liu, N., Wang, Z., Yu, L.: A trajectory data clustering method based on dynamic grid density. Int. Journal of Grid and Distributed Computing **8**(2), 1–8 (2015)
23. Li, Z.: Spatiotemporal Pattern Mining: Algorithms and Applications, pp. 283–306. Springer International Publishing (2014)
24. Li, Z., Ji, M., Lee, J.G., Tang, L.A., Yu, Y., Han, J., Kays, R.: Movemine: Mining moving object databases. In: ACM SIGMOD Int. Conf. on Management of data, pp. 1203–1206. Indianapolis, Indiana, USA (2010)
25. Lin, M., Hsu, W.J.: Mining GPS data for mobility patterns: A survey. Pervasive and Mobile Computing **12**, 1 –16 (2014)
26. Naim, H., Aznag, M., Durand, N., Quafafou, M.: Semantic Pattern Mining Based Web Service Recommendation. In: IEEE Int. Conf. on Service Oriented Computing, pp. 417–432. Banff, Alberta, Canada (2016)
27. Naim, H., Aznag, M., Quafafou, M., Durand, N.: Probabilistic Approach for Diversifying Web Services Discovery and Composition. In: IEEE Int. Conf. on Web Services, pp. 73–80. San Francisco, CA, USA (2016)
28. Pan, G., Qi, G., Zhang, W., Li, S., Wu, Z., Yang, L.T.: Trace Analysis and Mining for Smart Cities: Issues, Methods, and Applications. IEEE Communications Magazine **51**(6), 120–126 (2013)
29. Parent, C., Spaccapietra, S., Renso, C., Andrienko, G., Andrienko, N., Bogorny, V., Damiani, M.L., Gkoulalas-Divanis, A., Macedo, J., Pelekis, N., Theodoridis, Y., Yan, Z.: Semantic trajectories modeling and analysis. ACM Computing Surveys (CSUR) **45**(4), 1–32 (2013)
30. Pasquier, N., Bastide, Y., Taouil, R., Lakhal, L.: Efficient Mining of Association Rules Using Closed Itemset Lattices. Information Systems **24**(1), 25–46 (1999)
31. Poelmans, J., Ignatov, D.I., Kuznetsov, S.O., Dedene, G.: Formal concept analysis in knowledge processing: A survey on applications. Expert Systems with Applications **40**(16), 6538 –6560 (2013)
32. Soulet, A., Crémilleux, B., Rioult, F.: Condensed Representation of Emerging Patterns. In: Pacific-Asia Conf. on Knowledge Discovery and Data Mining, p. 127132. Sydney, Australia (2004)
33. Xue, A.Y., Qi, J., Xie, X., Zhang, R., Huang, J., Li, Y.: Solving the data sparsity problem in destination prediction. Int. Journal on Very Large Data Bases **24**(2), 219–243 (2015)
34. Yuan, N.J., Zheng, Y., Xie, X., Sun, G.: Driving with knowledge from the physical world. In: ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining, pp. 316–324. San Diego, California, USA (2011)
35. Yuan, N.J., Zheng, Y., Xie, X., Wang, Y., Zheng, K., Xiong, H.: Discovering Urban Functional Zones Using Latent Activity Trajectories. IEEE Trans. on Knowl. and Data Eng. **27**(3), 712–725 (2015)
36. Yuan, N.J., Zheng, Y., Zhang, C., Xie, W., Xie, X., Sun, G., Huang, Y.: T-drive: Driving directions based on taxi trajectories. In: Int. Conf. on Advances in Geographic Information Systems, pp. 99–108. San Jose, California, USA (2010)
37. Zaki, M.J., Hsiao, C.J.: Efficient algorithms for mining closed itemsets and their lattice structure. IEEE Trans. on Knowl. and Data Eng. **17**(4), 462–478 (2005)
38. Zhang, L., Liu, L., Xia, Z., Li, W., Fan, Q.: Sparse trajectory prediction based on multiple entropy measures. Entropy **18**(9), 1–14 (2016)
39. Zheng, Y.: Trajectory Data Mining: An Overview. ACM Trans. Intell. Syst. Technol. **6**(3), 29–41 (2015)
40. Zheng, Y., Zhou, X.: Computing with spatial trajectories. Springer Science & Business Media (2011)