



HAL
open science

Sequential Formal Concepts over Time for Trajectory Analysis

Feda Almuhsen, Nicolas Durand, Mohamed Quafafou

► **To cite this version:**

Feda Almuhsen, Nicolas Durand, Mohamed Quafafou. Sequential Formal Concepts over Time for Trajectory Analysis. IEEE/WIC/ACM International Conference on Web Intelligence (WI 2018), Dec 2018, Santiago, Chile. hal-01952227

HAL Id: hal-01952227

<https://hal.science/hal-01952227>

Submitted on 18 Dec 2018

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Sequential Formal Concepts over Time for Trajectory Analysis

Feda Almuhsen
Aix Marseille Univ,
Université de Toulon, CNRS, LIS
Marseille, France
feda.almuhsen@univ-amu.fr

Nicolas Durand
Aix Marseille Univ,
Université de Toulon, CNRS, LIS
Marseille, France
nicolas.durand@univ-amu.fr

Mohamed Quafafou
Aix Marseille Univ,
Université de Toulon, CNRS, LIS
Marseille, France
mohamed.quafafou@univ-amu.fr

Abstract—Tracking technologies and location-acquisition have led to the increase of the availability of trajectory data. Many efforts are devoted to develop methods for mining and analysing trajectories due to its importance in lots of applications such as traffic control, urban planning etc. In this paper, we present a new trajectory analysis and visualisation framework for massive movement data. This framework leverages formal concepts, sequential patterns, emerging patterns, and analyses the evolution of mobility patterns through time. Tagged city maps are generated to display the resulting evolution analysis and directions at different spatio-temporal granularity values. Experiments on real-world dataset show the relevance of the proposition and the usefulness of the resulting tagged city maps.

Index Terms—Sequential patterns, Concept lattice, Spatio-temporal data, Trajectories, Visualisation.

I. INTRODUCTION

With the maturity of positioning and tracking technologies in cities, global positioning system (GPS) sensors have been deployed to collect citizen’s movement, where these movements represent the moving objects spatio-temporal sequences records. Analysing and visualising such information is very important to understand the users behaviours. Moreover, visualising the direction of movements in geographical space gives more detailed analysis. For example, direction describes how moving objects redistribute when passing through an intersection node in a traffic network. This analysis plays a major role in smart city applications such as human mobility understanding, vehicle and pedestrian traffic control, smart transportation, and urban planning [1] [2]. Our research work aims at finding a solution for mining and analysing movement data. Trajectory Trj can be defined as a sequence of geographical points returned at specific time stamp t [3] [4]. Each Trj contains a set of points $\{p_1, p_2, \dots, p_{k-1}, p_k\}$ in which each point $p_k \in Trj$ is defined in a three-dimensional space, i.e., $p_k = (longitude_{p_k}, latitude_{p_k}, t_{p_k})$.

Sequential pattern mining technique [5] is one of the mining tools applied to trajectories. It helps in discovering important sequential ordering patterns and relationships between events and elements in the dataset. Most of the proposed approaches are based on two steps: Sequence transformation and pattern mining. In the first step, trajectories are transformed into sequences where each element of a sequence can represent either grid ID [6], line segment [7], or cluster [8]. While

the second step focuses on how to mine patterns in the set of generated sequences. A large amount of effort has been devoted to mining patterns with a closeness constraint, which significantly reduces the size of the output. Examples of such algorithms are CloSpan [9] and Moving Trajectory CloSpan Algorithm (MTCloSpan Algorithm) [10].

This paper presents a novel aspect of analysing and visualising massive movement data, where we propose a new trajectory analysis framework. Our main contribution corresponds to leveraging formal concepts [11], sequential patterns [5], and emerging patterns [12] for the purpose of understanding the characteristics and the evolution of mobility movements patterns through time and visualising their corresponding directions. To the best of our knowledge, there is no research work based on these three notions for analysing moving objects. This work extends our previous work [13] by considering sequences and movement directions visualisation. We propose a method to build a spatial-temporal trajectory discrete representation in order to extract hidden closed sequential patterns using uniform grids. We construct the sequential concept lattice to encode the maximal correspondence between the discovered closed sequential patterns (ordered visited spatial regions) and trajectories that pass through these regions. This usage will reduce the search space to reveal the underlying movement patterns. We propose an algorithm that analyses the evolution of the discovered sequential formal concepts over time and define different evolution types (*emerging*, *decreasing*, *latent*, *jumping*, and *lost*). Finally, we provide a visual representation of the results in maps which are tagged by different evolution types and show their corresponding movement directions. One practical advantage of these maps is that they can be generated in different spatio-temporal level to provide a description of the movements in the city. We evaluated the proposed method using a real-world dataset composed of taxi movements in San Francisco. Let us remark that the proposed method is generic and is not limited to taxi data and a smart city context.

The rest of the paper is organised as follows. Section II gives some definitions about sequential patterns and concept lattice. The proposed method is detailed in Section III. The experimental results are presented in Section IV. Finally, we conclude in Section V.

II. PRELIMINARIES

Sequential pattern mining [5] is an important data mining task with a wide range of applications such as text analysis, market basket analysis, and trajectory analysis. Let $\mathcal{I} = \{i_1, \dots, i_k\}$ be a set of items. A subset $I \subseteq \mathcal{I}$ is called an itemset. $|I|$ denotes the number of items of I . A **sequence** is an ordered list of itemsets. It can be represented as $s = \langle (I_1)(I_2)\dots(I_k) \rangle$ where each I_i is a subset of \mathcal{I} and I_i comes before I_j if $i \leq j$. The aim is to find all the frequent subsequences in a sequential database. Let $\mathcal{S} = \langle s_1, \dots, s_n \rangle$ be a list of sequences in a database. A sequence $s_1 = \langle a_1, \dots, a_m \rangle$ is a subsequence of another sequence $s_2 = \langle b_1, \dots, b_n \rangle$, denoted by $(s_1 \subseteq s_2)$, if there are integers $1 \leq i_1 \leq i_2 \leq \dots \leq i_m \leq n$, such as $a_1 \subseteq b_{i_1}, a_2 \subseteq b_{i_2}, \dots$, and $a_m \subseteq b_{i_m}$. s_2 is called a supersequence of s_1 (s_2 contains s_1). The support of a sequence s_n , noted $support(s)$, is the number of occurrences of s_n in \mathcal{S} . A sequence s is said to be a **sequential pattern** (also called a frequent sequence) if $support(s) \geq minsup$, where $minsup$ is the minimal support threshold value set by the user. A sequential pattern s_1 is **closed** if $\nexists s_2$ with $s_1 \subset s_2$ and $support(s_1) = support(s_2)$. Closed sequential patterns are interesting for us because they represent the largest frequent subsequences common to sets of sequences.

Formal concept analysis (FCA) [11] is a theory of data analysis identifying the conceptual structures within datasets. The closure properties and its capability of discovering inherent hierarchical structures give it the advantage to be used to analyse the different pattern relationships and build better mining algorithms [14]. A **formal context** is composed of a set of objects \mathcal{O} , a set of items \mathcal{I} , and a binary relation \mathcal{R}_1 between \mathcal{O} and \mathcal{I} (i.e., $\mathcal{R}_1 \subseteq \mathcal{O} \times \mathcal{I}$). While by dealing with ordered data, each object represents a sequence in \mathcal{S} . Thus, items in each object are partially ordered through the time. A **sequential context** (or *order context*) \mathcal{C} [15] is defined as $(\mathcal{R}, \mathcal{O}, \mathcal{I})$, where $\mathcal{R} \subseteq \mathcal{O} \times \mathcal{I} \times \mathbb{N}$. For an entry $(o, i, k) \in \mathcal{R}$, k represents the order of occurrence of i with respect to the other attributes in the same object. Sequential context can be represented by a cross table where rows are labeled by objects and the columns are labeled by the items. The entry in row j and column i corresponds to the order of item i in the equivalent input sequence s_j . Given a sequential context, there is a unique ordered set which describes the inherent lattice structure defining natural groupings and relationships among the objects and their related sequences [15]. This structure is called a **sequence concept lattice**. Each element of the lattice is a couple (O, S) composed of a set of sequences (the intent) and a set of objects (the extent). Each couple (called **sequence formal concept**) must be a complete couple with respect to \mathcal{R} , which means that the following mappings (noted ϕ and ψ) hold. For $O \subseteq \mathcal{O}$ and $S \subseteq \mathcal{S}$, we have: (1) $\phi(O) = \{s \in \mathcal{S} \mid s \text{ is maximally contained in } s_i, \forall i \in O\}$ and (2) $\psi(S) = \{i \in \mathcal{O} \mid s \subseteq s_i, \forall s \in S\}$. $\phi(O)$ returns sequences common to all objects $o \in O$, while $\psi(S)$ returns objects that have at least all sequences $s \in S$. The idea of maximally extending the

sets being formalised by the mathematical notion of closure in ordered sets. ϕ and ψ form a Galois connection and the compositions $\widehat{\Delta} = \psi \circ \phi$ and $\Delta = \phi \circ \psi$ are the Galois closure operators. Let S be a set of sequences if $\Delta(S) = S$ then each sequence of S is a closed sequence. A sequence formal concept is composed of a set of closed sequences and of the set of objects containing this closed sequences. We call **sequential formal concepts**, the sequence of formal concepts that have at least $minsup$ objects in their extent, and **sequential concept lattice**, the lattice formed using the sequential formal concepts.

III. TRAJECTORY EVOLUTION TYPES DETECTION

In this section, we present a new trajectory analysis framework based on sequential patterns, FCA, and emerging patterns. The main steps are: (1) Spatio-temporal preprocessing ; (2) Sequential concept lattice and evolution type detection ; and (3) Visualisation of geolocalised evolution type.

In step 1, trajectories are segmented by time value Δt , labeled by time granularity value τ , and mapped on a raster area G using local georeference according to spatial granularity value α . These discretised data are then used to extract hidden sequential patterns and the corresponding evolution types in the next step. In step 2, For each time windows T (i.e., a pair of time granularity values (t_{i-1}, t_i)), we compute the sequential concept lattice \mathcal{L} according to the minimal support threshold value $minsup$. Each node of \mathcal{L} , i.e., sequential formal concept, encodes the maximal correspondence between a set of closed sequential patterns (where each sequence contains an ordered set of grids), and the set of trajectories passing through these grids. For each sequential formal concept, we analyse the evolution of the closed sequential pattern by computing an indicative value K . By given K , minimal threshold value of emergence θ , and an error tolerance value ϵ , we detect the evolution type and the corresponding movement direction for each grid included in the closed sequential pattern. Let us remark that evolution type belongs to a predefined set: $\{latent, emerging, decreasing, jumping, lost\}$. Finally, in step 3 we automatically generate maps which are tagged by the detected evolution types.

A. Spatio-Temporal Preprocessing

Preprocessing is an important step in our framework. This step aims at simplifying and reducing the number of points processed with a negligible error for the purpose of mining trajectories. The main steps are (1) Trajectory segmentation according to time values ; (2) Mapping trajectories on a raster area according to spatial granularity value.

1) *Segmenting and Labeling Trajectories*: GPS data is huge in size to be analysed directly. Therefore, we perform two level of trajectory time segmentation. First step is to segment trajectories by Δt value. Δt is a given value that compared the time difference between two cleaned ordered GPS points. We perform this step to get an equal intervals of moving objects sequences. This value is in minutes. In the second step, we segment and label trajectories by a given time granularity τ , (can be minutes, hours, days, etc.) specified according to the

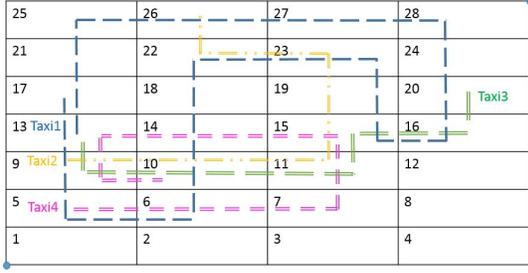


Fig. 1: Example of trajectory mapping.

type of analysis required. Each value of granularity is called a label. We note $\mathcal{L}a = \{l_1, l_2, \dots, l_{k-1}, l_k\}$ as the set of time granularity values where $l_j < l_{j+1}$ for $1 \leq j \leq k-1$. Each trajectory is segmented according to the defined labels. The trajectory of object n after segmentation by $\mathcal{L}a$ can be defined as follow: $Trj = \{Trj_n^{l_1}, Trj_n^{l_2}, \dots, Trj_n^{l_k}\}$. If $Trj_n^{l_j}$ is empty then this trajectory is deleted.

2) *Mapping Trajectories on a Raster Area*: Location coordinates in a spatio-temporal series are real numbers, which do not repeat themselves exactly in every pattern instance [7]. We proposed to use a solution that represents the continuous movement of an object as a sequence of traversed grid cells along with its respective traversal times. To achieve this, we split the geographical region as a raster area G where grid width and height dimension has an equal size spatial granularity α in kilometers. Then, we map trajectories using local georeference technique on G . This mapping includes conversion of the geographic coordinates (*latitude, longitude*) values in trajectories to grid index that represents these points. For each trajectory trj we obtain G_{trj} , the set of grids corresponding to trj traversal points. Figure 1 and Table I show an example of 4 taxi trajectories after mapping using local georeferenced system, within the corresponding time label (Day1, Day2).

B. Sequential Concept Lattice and Evolution Type Detection

Evolution type detection associated to trajectories over time windows is performed in two steps presented next.

1) *Sequential Concept Lattice Computation*: The computation of the sequential lattice allows to obtain the closed sequential patterns of grids (in our case here it maybe spatial zones, routes, etc.) and their corresponding sets of trajectories. The first step to compute this lattice is to create the ordered formal context $\mathcal{C} = (\mathcal{R}, \mathcal{O}, \mathcal{I})$ for a specified time window. \mathcal{O} is the set of trajectories for the studied time window. \mathcal{I} is the set of grid indexes. Then, the sequential concept lattice is computed according to the specified minimal threshold value $minsup$. For example, Table I represents a sequence dataset of 4 taxis trajectories. Each trajectory has been segmented by Δt , and labeled by the corresponding time granularity value (see Table II). Some sequential formal concepts computed with $minsup = 2$ are presented in Table III.

2) *Detection of Evolution Types*: We detect the evolution type of each sequence in sequential concept lattice by studying

TABLE I: Example of taxis data represented by their grid sequences and corresponding time labels

trajseq-id	Sequence	label
T_{axi1}	$\langle (13)(17)(21)(25)(26)(27)(28)(24) \rangle < (20)(16)(15) \rangle$ $\langle (19)(23)(22)(18) \rangle < (14)(10)(6) \rangle < (5)(9)(13)(17) \rangle$	Day1
T_{axi2}	$\langle (9)(10)(11)(15) \rangle < (19)(23)(22)(26) \rangle$	Day1
T_{axi3}	$\langle (20)(16) \rangle < (15)(11)(10)(9)(13) \rangle < (14)(15)(19)(23)(27) \rangle$	Day2
T_{axi4}	$\langle (5)(6)(7)(11)(15) \rangle < (14)(13)(9)(10) \rangle$	Day2

TABLE II: Trajectory data sequences in different days represented by grids

seq-id	Sequence	label
s_1	$\langle (13)(17)(21)(25)(26)(27)(28)(24) \rangle$	Day1
s_2	$\langle (20)(16)(15) \rangle$	Day1
s_3	$\langle (19)(23)(22)(18) \rangle$	Day1
s_4	$\langle (14)(10)(6) \rangle$	Day1
s_5	$\langle (5)(9)(13)(17) \rangle$	Day1
s_6	$\langle (9)(10)(11)(15) \rangle$	Day1
s_7	$\langle (19)(23)(22)(26) \rangle$	Day1
s_8	$\langle (20)(16) \rangle$	Day2
s_9	$\langle (15)(11)(10)(9)(13) \rangle$	Day2
s_{10}	$\langle (14)(15)(19)(23)(27) \rangle$	Day2
s_{11}	$\langle (5)(6)(7)(11)(15) \rangle$	Day2
s_{12}	$\langle (14)(13)(9)(10) \rangle$	Day2

the support variation through a pair of time windows (t_{i-1}, t_i) . Let us remark that the evolution type belongs to a predefined set of labels: $\{latent, emerging, decreasing, jumping, lost\}$. *Emerging* label means that the presence of the pattern increased in t_i compared to t_{i-1} ; *Decreasing* means that the presence of the pattern decreased in t_i compared to t_{i-1} ; *Latent* means that the presence of the pattern is quite similar in both time. *Jumping* means the pattern which was absent in t_{i-1} , appeared in t_i . *Lost* means that the pattern disappeared in t_i .

Let $FC = (O, S)$ be a sequential formal concept, θ be the minimal threshold value of emergence, and ϵ be the error tolerance. The evolution type of each sequence contained in S is detected by computing K_i as follow:

$$K_i(O, S) = \frac{count(O, t_i)}{count(O, t_{i-1})}$$

Where $count(O, t_j)$ is the number of trajectories of O labeled by the corresponding t_j . We apply the following rules based on K_i value. If $K_i(O, S) = 0 \pm \epsilon$, then the type of S is *lost*. If $((K_i(O, S) > \theta) \wedge (\theta = 1)) \vee ((K_i(O, S) \geq \theta) \wedge (\theta > 1))$ then the type of S is *emerging*. If $K_i(O, S) < \theta$, then the type of S is *decreasing*. If $K_i(O, S) = 1 \pm \epsilon$, then the type of S is *latent*. If $K_i(O, S) = +\infty$, then the type of S is *jumping*. Table III presents some results of computing K_i and the detected evolution types from the sequential dataset presented in Table II.

C. Visualisation

In this step, we present two methods for visualising the detected evolution types in tagged city maps. The first method generates maps tagged by different colors, whereas each color represents a given evolution type. The second method is to represent the results using colors and arrows, where arrows represent the flow directions (North, South, East, etc.) and the

TABLE III: Some sequential formal concepts and their types obtained with $minsup = 2$, $\theta = 1$, $\epsilon = 0$

Extent	Intent	day1	day2	K	Evolution type
$\{s_2, s_6, s_9, s_{10}, s_{11}\}$	$\{< (15) >\}$	2	3	1.5	<i>emerging</i>
$\{s_2, s_8\}$	$\{< (20)(16) >\}$	1	1	1	<i>latent</i>
$\{s_3, s_7\}$	$\{< (19)(23)(22) >\}$	2	0	0	<i>lost</i>
$\{s_3, s_7, s_{10}\}$	$\{< (19)(23) >\}$	2	1	0.5	<i>decreasing</i>

color of the arrow represents the corresponding type detected. This representation is important because trajectories of moving objects are directed by nature. Therefore, the direction information is highly relevant and important to display in interactive visualisation systems to provide a fine analysis.

Given a sequential concept lattice and the corresponding type values. For each sequence in the intents, we extract the real coordinates of the ordered grids. 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 area in the real map. The type of each grid item is chosen according to the majority vote value, which is computed by considering all the sequential formal concepts that contain this grid in its intent. The second method is a variant of the previous method where we detect the direction of a grid in a sequence. For that, we consider the real coordinates of the grid and of the next grid in the sequence. Let us note that if there is no next grid, there is no detected direction. The type of each direction (or none) for a grid item is chosen according to the majority vote value, which is computed by considering all the sequential formal concepts that contain this grid with this direction in its intent.

IV. EXPERIMENTS

A. Data and Protocol

We analysed the data collected from the CRAWDAD website [16]. This dataset consists of a 500 taxis trajectories in San Francisco which was collected during the period from 17 May 2008 to 10 June 2008 (24 days). We selected 455 taxis during 23 days. Several experiments were performed by varying time granularity τ (24, 12, 6 hours), spatial granularity α (20, 40, 60 meters), and $minsup$ (0.03, 0.05, 0.07). We set Δt to 60 minutes, ϵ to 0 and θ to 1. Let us note that the chosen Δt value does not segment again the data. We also conducted a comparative study between the results obtained from sequential concept lattice (noted here as "sequences") with directions or without, and the ones obtained from frequent concept lattice (noted here as "itemsets"). We calculated the similarity between the results by using Jaccard measure ($Sim(A, B) = \frac{|A \cap B|}{|A \cup B|}$). The intersection of two results A and B is the set of grids having the same detected type in the two results A and B . In the case of using directions, if the type of a grid corresponds to the type of one direction or more, then the result is considered as the same. The more the value is closer to 1, the more the compared results are similar. All the algorithms were implemented in Java, and all the experiments were performed on an Intel Xeon X5560 2.8GHz with 16GB of memory.

B. Results

Table IV presents the characteristics of the computed sequential datasets. In the tables, May17 corresponds to Saturday (Armed Forces Day), May18 to Sunday, ..., May26 to Monday (Memorial Day). The types are: LA (*latent*), E (*emerging*), D (*decreasing*), J (*jumping*), and LO (*lost*).

1) *Quantitative Results*: Tables V, VIII, and VII show part of statistical analysis for the results obtained from using different spatial granularities 20, 40, 60 meters respectively ($minsup = 0.05$ and time granularity = one day). We can observe that the total number of patterns increases whenever we have a high spatial 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 a sequential formal concept increases. We can conclude that there is a relation between spatial granularity and the total number of generated sequential formal concepts. Moreover, we observe that the number of lost and jumping types are 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. A set of patterns is *lost* or *jumping* if it is not detected during both the two compared time windows. We have also applied the same protocol by changing time granularity value. Instead of using 24 hours, we have set the time granularity value to 12 hours and 6 hours, while keeping spatial granularity value of 60 meters and $minsup$ value as 0.05. Tables VI and IX show the statistical results for 12 hours and 6 hours, respectively for May17, May18, and May19. If we compare these statistics with the ones in Table VII, we can conclude that whenever we decrease the time granularity we get less total number of sequential formal concepts. For example, in time window (May18, May19) Table VII, the total number of sequential formal concepts is equal to 156,138 while for the same period of time on Table VI it is equal to 12,502 and to 3,597 in Table IX. The decreasing of the number of sequential concept lattice occurs because the data density is lower. Finally, we have applied the protocol by changing the $minsup$ value, for the same spatial granularity and time as presented in Tables VII, X and XI. We note from these results that whenever the $minsup$ value increases the number of generated sequential formal concepts decreases, and some evolution types may disappear. For example, if we compare the results in Table X with those in Table XI, we can notice that with $minsup=0.07$ we have no *jumping* or *lost* types.

2) *Qualitative Results*: Emerging sequential formal concepts have been captured in the Mission District area (see Figure 2) during the period of (May24, May25). This indicates

TABLE IV: Computed sequential datasets

Exp. No.	Parameters	#Sequences (avg.)	#Items (avg.)
1	$\alpha = 0.02$ km, $\tau = 24$ h, $minsup = 0.05$	895	34,475
2	$\alpha = 0.04$ km, $\tau = 24$ h, $minsup = 0.05$	895	14,255
3	$\alpha = 0.06$ km, $\tau = 24$ h, $minsup = 0.05$	895	8,074
4	$\alpha = 0.06$ km, $\tau = 12$ h, $minsup = 0.05$	865	7,198
5	$\alpha = 0.06$ km, $\tau = 6$ h, $minsup = 0.05$	820	6,122
6	$\alpha = 0.06$ km, $\tau = 24$ h, $minsup = 0.03$	895	8,074
7	$\alpha = 0.06$ km, $\tau = 24$ h, $minsup = 0.07$	895	8,074



Fig. 2: Tagged map for Mission District (May24,May25).

TABLE V: Number of sequential formal concepts

$\alpha = 0.02$ km, $\tau = 24$ h, $minsup = 0.05$

Time windows	#E	#D	#LA	#LO	#J	Total
(May17,May18)	1,390	35	3	0	0	1,428
(May18,May19)	751	1,854	71	0	0	2,676
(May19,May20)	760	1,075	78	0	0	1,913
(May20,May21)	905	825	86	0	0	1,816
(May21,May22)	1,300	727	110	0	0	2,137
(May22,May23)	1,434	938	132	0	0	2,504
(May23,May24)	1,157	1,273	107	0	0	2,537
(May24,May25)	1,430	1,197	131	0	0	2,758
(May25,May26)	1,548	864	100	0	0	2,512

TABLE VII: Number of sequential formal concepts

$\alpha = 0.06$ km, $\tau = 24$ h, $minsup = 0.05$

Time windows	#E	#D	#LA	#LO	#J	Total
(May17,May18)	53,956	271	55	0	35	54,317
(May18,May19)	29,644	122,901	3,593	0	0	156,138
(May19,May20)	26,263	48,478	3,524	0	0	78,265
(May20,May21)	32,515	30,648	3,517	0	0	66,680
(May21,May22)	53,050	25,736	4,010	0	0	82,796
(May22,May23)	75,987	33,585	5,753	0	0	115,325
(May23,May24)	69,702	62,714	6,175	0	0	138,591
(May24,May25)	115,022	65,441	8,935	0	0	189,398
(May25,May26)	114,786	50,599	6,551	0	0	171,936

that those regions face a higher taxi movement on the second day compared to the first day. This results can be clarified by the fact that these days are part of Memorial Day weekend. In fact, there was a parade in May25 morning which runs along Mission Street for three hours or more. For that reason traffic in May25 was higher in this area compared to May24. Figure 4 shows a decreasing evolution type in the crossroad of Townsend street and 7th street, within a majority of *emerging* direction flow to North East and North West, while a majority of *decreasing* to South East. This precise information gives a detailed analysis compared to the *decreasing* type without direction captured in Figure 3. This example is typical and shows that the proposed visualisation method using evolution type with directions provides a more fine analysis of movements, a precise description for each region in the city, and a better understanding of traffic flow at street levels.

3) *Comparative Results*: Table XII presents the similarity values obtained from comparing (1) Itemsets and sequences without direction, $Sim(it,seqnod)$; (2) Itemsets and sequences with direction, $Sim(it,seqd)$; (3) Sequences with direction and without direction, $Sim(seqnod,seqd)$. The purpose of this

TABLE VI: Number of sequential formal concepts

$\alpha = 0.06$ km, $\tau = 12$ h, $minsup = 0.05$

Time windows	#E	#D	#LA	#LO	#J	Total
May17(am),May17(pm)	856	1	0	0	137	994
May17(pm),May18(am)	4,600	1,384	85	0	1	6,070
May18(am),May18(pm)	1,376	4,945	126	0	0	6,447
May18(pm),May19(am)	1,483	1,817	105	0	0	3,405
May19(am),May19(pm)	1,306	1,291	53	0	0	2,650

TABLE VIII: Number of sequential formal concepts

$\alpha = 0.04$ km, $\tau = 24$ h, $minsup = 0.05$

Time windows	#E	#D	#LA	#LO	#J	Total
(May17,May18)	9,817	125	19	0	2	9,963
(May18,May19)	5,046	19,738	606	0	0	25,390
(May19,May20)	5,258	7,991	642	0	0	13,891
(May20,May21)	5,827	5,956	644	0	0	12,427
(May21,May22)	10,287	4,548	688	0	0	15,523
(May22,May23)	12,031	7,247	1,044	0	0	20,322
(May23,May24)	11,031	10,674	1,026	0	0	22,731
(May24,May25)	16,952	11,299	1,424	0	0	29,675
(May25,May26)	16,847	8,016	954	0	0	25,817

comparison is to examine if our extension to direction and the use of majority voting affects the obtained results. As we can see, $Sim(seqnod,seqd)$ has the maximum value among all the other. Hence the results obtained from sequential concept lattice with direction covered the results without direction. Furthermore, this validates that the extension to directions provides more details to the model without affecting the final results. Additionally, we can conclude that the results obtained from our proposed method do not vary a lot from the results obtained from frequent concept lattice, which prove the correctness of both methods. In order to study the effect of different parameters in our framework, we have used different parameter values (see Exp. No. in Table IV). We can observe the following relation, spatial and temporal granularities and $minsup$ values do not affect the comparative results. The proposed method is robust. We always have high similarity values between the two results.

V. CONCLUSION

We have proposed a new trajectory analysis framework based on sequential patterns, emerging patterns and formal concept analysis. It allows to detect and characterise the

TABLE IX: Number of sequential formal concepts
 $\alpha = 0.06$ km, $\tau = 6$ h, $minsup = 0.05$

Time windows	#E	#D	#LA	#LO	#J	Total
May17(0:00-6:00)-May17(6:00-12:00)	0	0	0	0	54	54
May17(6:00-12:00)-May17(12:00-18:00)	184	6	0	0	2	192
May17(12:00-18:00)-May17(18:00-23:59)	312	152	2	0	1	467
May17(18:00-23:59)-May18(0:00-6:00)	493	155	13	0	0	661
May18(0:00-6:00)-May18(6:00-12:00)	367	398	25	1	0	791
May18(6:00-12:00)-May18(12:00-18:00)	142	435	8	3	0	588
May18(12:00-18:00)-May18(18:00-23:59)	310	188	12	0	2	512
May18(18:00-23:59)-May19(0:00-6:00)	263	290	15	0	0	568
May19(0:00-6:00)-May19(6:00-12:00)	312	120	8	0	0	440
May19(6:00-12:00)-May19(12:00-18:00)	143	160	6	1	0	310
May19(12:00-18:00)- May19(18:00-23:59)	281	94	9	0	4	388

TABLE X: Number of sequential formal concepts
 $\alpha = 0.06$ km, $\tau = 24$ h, $minsup = 0.03$

Time windows	#E	#D	#LA	#LO	#J	Total
(May17,May18)	320,457	2,097	508	0	3998	327,060
(May18,May19)	187,308	748,810	31,919	44	0	968,081
(May19,May20)	152,627	272,393	25,594	1	0	450,615
(May20,May21)	176,022	171,445	23,406	0	0	370,873
(May21,May22)	288,536	150,320	28,110	0	0	466,966
(May22,May23)	404,990	196,577	42,724	0	0	644,291
(May23,May24)	397,210	353,401	48,511	0	0	799,122
(May24,May25)	728,667	421,345	71,019	0	0	1,221,031
(May25,May26)	714,969	359,008	58,396	0	0	1,132,373

TABLE XI: Number of sequential formal concepts
 $\alpha = 0.06$ km, $\tau = 24$ h, $minsup = 0.07$

Time windows	#E	#D	#LA	#LO	#J	Total
(May17,May18)	16,049	73	10	0	1	16,133
(May18,May19)	8,894	36,883	836	0	0	46,613
(May19,May20)	8,041	15,140	914	0	0	24,095
(May20,May21)	10,395	9,708	940	0	0	21,043
(May21,May22)	15,934	7,434	1,061	0	0	24,429
(May22,May23)	24,270	9,969	1,594	0	0	35,833
(May23,May24)	21,702	19,544	1,560	0	0	42,806
(May24,May25)	34,439	19,396	2,205	0	0	56,040
(May25,May26)	36,879	14,508	1,831	0	0	53,218

evolution of patterns through time, and to visualise the results in tagged city maps at different spatio-temporal granularities. Experiments on real-world data have shown the applicability of the proposed method and its relevance. Moreover, we have shown that the use of sequential patterns compared to frequent itemsets provides more precise analysis with the direction information. In future work, we will extend our system to build a probabilistic model to predict the pattern evolution during the next time window.

REFERENCES

- [1] G. Pan, G. Qi, W. Zhang, S. Li, Z. Wu, and L. T. Yang, "Trace analysis and mining for smart cities: Issues, methods, and applications," *IEEE Communications Magazine*, vol. 51, no. 6, pp. 120–126, June 2013.
- [2] M. Lin and W.-J. Hsu, "Mining GPS data for mobility patterns: A survey," *Pervasive and Mobile Computing*, vol. 12, pp. 1–16, 2014.
- [3] Y. Zheng, "Trajectory data mining: An overview," *ACM Trans. Intell. Syst. Technol.*, vol. 6, no. 3, pp. 29–41, 2015.
- [4] Y. Zheng and X. Zhou, *Computing with spatial trajectories*. Springer Science & Business Media, 2011.
- [5] P. Fournier-Viger, J. C.-W. Lin, R. U. Kiran, Y. S. Koh, and R. Thomas, "A survey of sequential pattern mining," *Data Science and Pattern Recognition*, vol. 1, no. 1, pp. 54–77, 2017.
- [6] F. Giannotti, M. Nanni, F. Pinelli, and D. Pedreschi, "Trajectory pattern mining," in *KDD'07*, 2007, pp. 330–339.
- [7] H. Cao, N. Mamoulis, and D. W. Cheung, "Mining frequent spatio-temporal sequential patterns," in *ICDM'05*, 2005, pp. 82–89.
- [8] Y. Ye, Y. Zheng, Y. Chen, J. Feng, and X. Xie, "Mining individual life pattern based on location history," in *MDM'09*, 2009, pp. 1–10.
- [9] X. Yan, J. Han, and R. Afshar, "Clospan: Mining closed sequential patterns in large datasets," in *SDM'03*, 2003, pp. 166–177.
- [10] L. Wang, K. Hu, T. Ku, and J. Wu, "Discovering closed frequent patterns in moving trajectory database," in *ICCT'12*, 2012, pp. 567–572.
- [11] B. Ganter and R. Wille, *Formal concept analysis: Mathematical foundations*. Springer, 1999.
- [12] G. Dong and J. Li, "Efficient mining of emerging patterns: Discovering trends and differences," in *KDD'99*, 1999, pp. 43–52.
- [13] F. Almuhsen, N. Durand, and M. Quafafou, "Detecting behavior types of moving object trajectories," *Int. Journal of Data Science and Analytics*, vol. 5, no. 2, pp. 169–187, March 2018.
- [14] J. Poelmans, D. I. Ignatov, S. O. Kuznetsov, and G. Dedene, "Formal concept analysis in knowledge processing: A survey on applications," *Expert Systems with Applications*, vol. 40, no. 16, pp. 6538–6560, 2013.
- [15] G. C. Garriga, "Lattice theory for sequences," in *Formal Methods for Mining Structured Objects*. Springer, 2013, pp. 21–37.
- [16] M. Piorkowski, N. Sarafijanovic-Djukic, and M. Grossglauser, "CRAWDAD dataset epfl/mobility (v. 2009-02-24)," <https://crawdad.org/epfl/mobility/20090224/cab>, Feb. 2009, traceset: cab.



Fig. 3: Tagged map for Townsend street without direction.

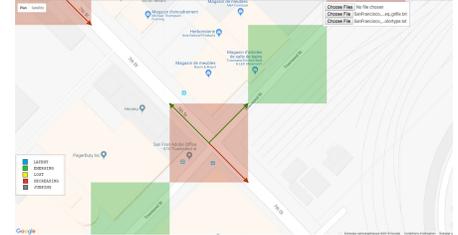


Fig. 4: Tagged map for Townsend street with direction.

TABLE XII: Comparative study between itemsets and sequences experiments

Exp. No.	Sim(it,seqnod)	Sim(it,seqd)	Sim(seqnod,seqd)
1	0.8875	0.9040	0.9467
2	0.8147	0.8530	0.8755
3	0.8232	0.8515	0.8813
4	0.8802	0.8896	0.9387
5	0.8515	0.8550	0.9837
7	0.8274	0.8564	0.8943
Average	0.8474	0.8682	0.9200