

Discovering Local and Global Co-Location Patterns in Trajectories with Different Properties

Fernando José Braz¹, Vania Bogorny²

¹Instituto Federal de Educação Ciência e Tecnologia – IFC - Campus Araquari - Santa Catarina

²Departamento de Informática e Estatística – Universidade Federal de Santa Catarina
Florianópolis - Santa Catarina

fernando.braz@ifc-araquari.edu.br, vania@inf.ufsc.br

Abstract. *Most trajectory mining approaches consider a very small set of properties to extract patterns from trajectories. Besides, a lot of them consider those properties separately. In this paper we present a method to find co-location patterns based on different properties of trajectories along time. The proposal allows to identify a sequence of co-locations composed by different properties (distance, acceleration, speed, time etc) that represents the behavior of a set of trajectories. The sequences of co-locations patterns present the evolution of an event, for example traffic jam. By using this knowledge is possible to anticipate the phenomenon occurrence, and to take actions to solve problems regarding to the event.*

1. Introduction and Motivation

The recent development and price reduction of mobile and location aware technologies (like cellular phones equipped with *GPS*), allows tracking the movement of objects. This fact contributes for increasing the amount of data related to moving objects. A singular characteristic of a moving object is how its position changes over time. The time ordered set of positions of a moving object is what is called *trajectory of a moving object*.

The analysis of trajectories will lead to the understanding of the behavior of objects in a given environment. In general, a trajectory has an identifier (t_{id}), a set of x,y coordinates representing the spatial location and a time t for every point (x,y) .

Trajectory data do not have a special meaning, have a lot of noise, and are difficult to understand and to interpret from the user's point of view. By plotting a set of trajectories in a bi-dimensional space, not much information or knowledge can be extracted. Several trajectory mining methods have been developed in the last few years to extract interesting patterns from trajectories, with the intent to discover behavior patterns from groups of trajectories, like for instance, [Laube 2005, Cao 2006, Giannotti 2007, Lee 2008, 2009, Bogorny 2010].

Most trajectory mining approaches consider a very small set of variables to extract patterns from trajectories, basically considering distance, direction and density. Another problem is that most of existing works consider these objective measures separately, without combining one or more measures. For instance, to identify a pattern of low traffic or traffic jam in a set of trajectories for a transportation management application, it may be necessary to consider speed reduction, density of presence, distance, the road speed and so on. To increase the problem, lets imagine that a traffic jam is not just a place with some

characteristics, but a sequence of movements (events) with different characteristics that will result in a traffic jam, as for instance, *decreasing in the average speed* and *increasing in the number of objects* into a given area.

In other domains, like the movement analysis of ships in real time, the pattern could be used to prevent or to advertise a damage of engines as well as the occurrence of an attack of pirates. Both events could be discovered by considering different sets of properties of ship trajectories. In a traffic air control, airplanes entering or crossing a storm area, as well as pane problems could be rapidly detected. The military strategy can be identified by mining the co-location patterns of a set of vehicles in a battle field.

Considering the above examples we turn up with a problem: how to discover the sequence of events that occurred at different locations and at different time periods, characterized by different attributes? How to previously infer the behavior changing of individual or groups of trajectories that result in a certain event?

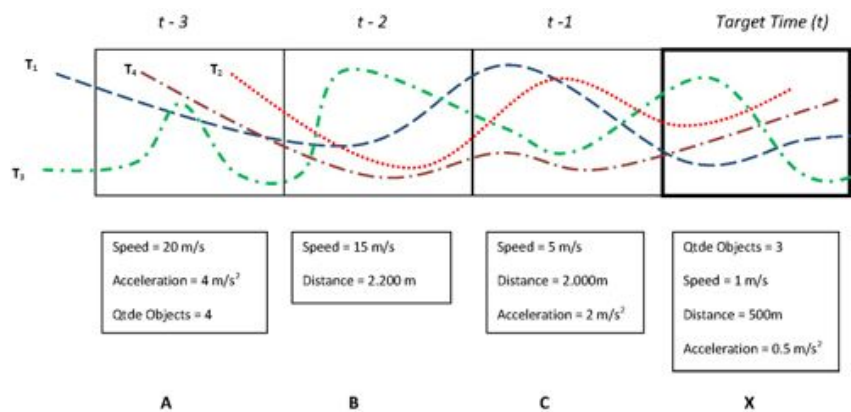


Figure 1. Example of trajectories with properties changing over time

Figure1 shows an example of four trajectories (T_1, T_2, T_3, T_4) of moving objects traveling in a spatio-temporal area. There are four different time windows ($t - 3$, $t - 2$, $t - 1$ and t) represented by the four boxes. Different properties are considered for each time window (*average speed, average acceleration, traveled distance, number of trajectories*). The combination of these values can be used to identify the behavior of a set of trajectories to characterize different events. For example, the behavior of the trajectories at time window t is different of the behavior at time window $t - 3$. A set of slow moving objects, with reduced acceleration and traveling a distance of 500 meters at a speed of 1 m/s is the behavior of the objects at time window t . However, at time window $t - 3$ happens a totally different behavior for the same set of trajectories: in that spatio-temporal window the objects are moving at a higher speed and with higher acceleration. In this example, the sequence of events at time window $t - 3$, $t - 2$ and $t - 1$ will result in the final event at time t .

Considering the different characteristics of the same set of trajectories at different time windows, in a traffic management application, a sequence of different behaviors may result in the pattern *traffic jam*, for instance. What we have to define is: which are the characteristics of the behavior at time window t ? How was the behavior at previous time windows $t - 1$, $t - 2$ and $t - 3$? Is it possible to identify this evolution considering differ-

ent characteristics in each time window? We address this problem proposing a co-location pattern mining algorithm to discover different sequences of behaviors, that we call *local co-location patterns*, that generate/result in a final event, called *global co-location pattern*.

2. Related Works and Contribution

Several data mining methods for trajectories have been proposed in the recent years, but only a few words propose mining co-location patterns. Most existing co-location mining algorithms were developed for spatial data, as for instance, the works of [Shekhar and Huang 2001], [Huang et al. 2004] and [Yoo and Shekhar 2006]. Spatial co-location patterns represents a subset of spatial features frequently located together.

One of the earliest works for minign co-location patterns from trajectories was the approach of [Cao et al. 2006]. In this work, the proposal is to find *co-location episodes* in order to represent the inter-movement regularities among different types of moving objects. A *co-location episode* is as a sequence events, where events are sets of objects moving close to each other for a given time period. Besides, there is a particular object type (*centric feature*) which participates in a sequence of *co-locations*. In this approach the trajectories are split in time windows, and the algorithm looks for objects that move close to each other in each time window.

In [Celik et al. 2007] the authors presented a proposal to find *local co-location patterns* named *Zonal co-location patterns*. The proposal considers that a pattern is not uniformly distributed over the space. Therefore, different regions of the space may present different patterns. The idea is to delimit an area in order to find *local co-location patterns*. *Zonal co-location patterns* represent subsets of feature types that are frequently located in a subset of the space.

[Celik et al. 2008] proposed to mine patterns that represent subsets of object-types that are located together in space and time. This work introduces the use of time by using a *time prevalence* measure. The environment of the proposal is a mixed group of moving objects, considering only distance between the objects as the measure to identify the co-location of the objects. The locations of a set of objects are recorded along time. The proposal considers a time interval in order to investigate the co-location patterns. The time interval is divided into a set of fractions of time, and, for each one of them, co-location patterns of objects are identified. The distance among the objects is the measure to identify a co-location. Thereafter, the persistence of each pattern is evaluated along the complete time interval, in order to identify whether is eventual or a persistent pattern. In summary, the goal is to discover persistent patterns that co-occur im most, but not all, spatio-temporal intervals. Therefore, consecutive occurrences are not mandatory.

The work presented in [Zheng and Xie 2010] uses the co-location of places considering human behavior. Basically, the proposal is to identify locals where objects stayed over a given time and, to find correlations among these places. The proposal uses both travel experience of a set of users, and the sequence that locations have been visited. in order to investigate the location correlation on a set of user-generated GPS trajectories. A set of records that store the position of objects over time represents the trajectories of objects. By using measures of distance and time, stay ponts may be identified for each trajectory. A stay point is a region where an object remains for a minimum time interval.

The algorithm identifies stay points for each trajectory of the set, and reveals the movement pattern between them. Thus, the local correlation is established considering human behavior.

A proposal to find periodic behaviors for moving objects was presented in [Li et al. 2010]. The algorithm uses the location of the objects in order to mine patterns of movement. The proposal assumes the receipt of data in the traditional format of trajectories: space and time representing the positions of objects. A sequence of those records is named as location sequence. The proposed algorithm considers the existence of reference spots, that are dense areas visited frequently during the movement, into a location sequence. The periodic movement is represented by a probability of a given object is at a reference spot at a timestamp.

The previous works could be used to identify the different types of co-location patterns into a set of several trajectories. However, none of them consider a co-location pattern composed of different characteristics of trajectories. Besides, the generation of co-location sequences along the time, is not considered by the previous proposals. Therefore, the main contributions of this paper in relation to existing works include:

a) We present a trajectory co-location pattern mining method that generates every co-location pattern based on different characteristics of trajectories (e.g. speed, acceleration, density of presence, etc), while existing works in general consider a simple measure like distance between the objects. For example, one co-location pattern may occur in regions with similar speed and acceleration, while another co-location pattern may occur in regions with similar density of presence or similar direction change.

b) The main contribution of our work is that from a set of co-location patterns, we generate sequences of co-locations in time (in other words sequences of events) that cause a pattern event called fact or target event. For instance, a sequence of co-locations A, B and C will generate a target fact X, where for instance A is a co-location of deceleration and low density of presence, B is a co-location of low speed and high density of presence, C is a co-location of low speed and low acceleration and X is the resultant event/fact caused by the three previous co-locations, having speed zero and acceleration zero, named traffic jam.

c) We consider a set of spatial and non-spatial properties of trajectories to generate co-location patterns. These properties must be co-located in space and time, use the concept of time window.

The current proposal considers the work presented in [Braz 2008], where the authors discuss the possibility of usage of the data mining techniques to reveal knowledge in a large data volume of trajectories.

2.1. Scope and Outline

The scope of this paper is limited to the discovery of spatio-temporal co-location patterns in trajectories. More specifically, we find sequences of local co-location patterns that lead to a given global co-location pattern.

The remaining of the paper is organized as follows: Section 2 presents the basic concepts and definitions. Section 3 presents the proposed algorithm. Section 4 presents the experiments with synthetic and real trajectory data and, finally, section 5 concludes

the paper.

3. Basic Concepts and Definitions

In this section we present a set of basic concepts and definitions that are necessary to understand the paper. The most basic definition is a trajectory.

Definition 3.1 *Trajectory: A trajectory is a set of (tid, x, y, t) , where tid is the identifier of the moving object, x and y are the spatial position at time t .*

In this work we consider that the trajectories are located in a spatio-temporal area, and this area is split in time windows, similar to the idea proposed in [Cao et al. 2006].

Definition 3.2 *Time Window: A time window w is the smallest spatio-temporal area in which the space is divided.*

Considering that each time window represents the smallest spatio-temporal area, every subarea has a set of characteristics called *properties*. For example, the *average speed* property, represents a characteristic of movement of a set of trajectories crossing the time window.

Definition 3.3 *Property: A property p is the smallest item / characteristic of a time window. A set of properties p is given as $P = \{p_1, p_2, \dots, p_n\}$.*

The properties low *average speed* and small *traveled distance*, for instance, could characterize an event, a traffic jam. A set of one or more properties characterize a candidate set of properties for a local co-location pattern.

Definition 3.4 *Candidate local co-location: A Candidate local co-location c is a subset of P , such that $c = \{p_i, p_{i+1}, \dots, p_{i+m}\}$, where $0 \leq i$ and $(i + m) \leq n$*

Only candidates that occur frequently, in different time-windows, i.e.; more times than a given number or percentage, will be analyzed. Those candidates, that are frequent, are named *local co-locations*. By looking our example in Figure1, A, B, and C could be candidate local co-locations. When a candidate is frequent, it occurs more than a number of times, and this minimal number is called *minSup*.

Definition 3.5 *Local Co-location: A local co-location l is a candidate co-location c , such that $l = \{p_i, p_{i+1}, \dots, p_{i+m}\}$, where $0 \leq i$ and $(i + m) < n$, and the proportion of trajectories which contain c , in a given time window, is greater than a minimum value, this value is named support.*

A sequence of local co-locations, characterized by a set of properties, will result in a *Global co-location*. A global co-location represents the occurrence of an event.

Definition 3.6 *Global Co-location: A Global co-location g is the sequence of local co-locations l , such that $g = l_j, l_{j+1}, \dots, l_k$, where k is the number of local co-locations that compose g .*

Local co-locations will be searched after the target event has been defined. For instance, when a traffic jam is identified, previous local co-locations are computed in order to find the groups of trajectory properties that frequently occur together, before the global event is formed (global event in this example is the traffic jam). For discovering the global event, we look at a *target time window*.

Definition 3.7 *Target time window:* A Target time window (W) is the time window in which the occurrence of a Global co-location g will be searched.

The goal of this work is to present a method to identify the evolution of a subset of properties along the time, where every set of properties characterizes a *local co-location*. The result of this evolution is the *global co-location*. This evolution happens along the time, starting at a time $(t - i)$, and finishing at time t , as shown in the example in Figure 1 where one global co-location X happens at time window t . That global co-location (X) is composed by a sequence of local co-locations $\{A, B, C\}$, occurring at time windows $\{t - 3, t - 2, t - 1\}$ respectively.

4. T-COLOC: an algorithm for Trajectory Sequential Co-Location Mining

In this section we present the T-Coloc algorithm. The algorithm takes as input a set of trajectories T , the Target time window W where the global co-location occurs, the number of time windows Δw , where the local co-locations will be searched, and the minimum support α to validate the co-location patterns.

Algorithm 1 T-Coloc Algorithm

```

1: Input:
2:  $T$  : set of trajectories
3:  $W$  : the Target Time window
4:  $\Delta w$  : number of time windows
5:  $\alpha$  : minimum support for co-location patterns
6:
7: Output:
8:  $L$ : A Global co-location
9:
10: Method:
11: //Generate the Global co-location  $g$  related to the event occurring at Target Time Window  $W$ .
12:  $g = GENcolocation(T, W, \alpha)$ 
13:
14: //Generate local co-locations  $l$  related to each time window
15: for Each Time Window ( $\sigma$ ) in  $(w-1, w-2, \dots, w-\Delta w)$  do
16:    $L = GENcolocation(T, \sigma, \alpha)$ 
17:   Add  $l$  to  $L$ 
18: end for
19: Add  $g$  to  $L$ 
20: return  $L$ 

```

The algorithm has two main steps. The first one is to find the co-location pattern that represents the event occurring at the target time window t (line 12). The Global co-location g related to the event occurring at the Target Time Window W is generated with the method GENcolocation (detailed in algorithm 2). For generating the global co-location, this method takes as input the set of trajectories (T) at time window (W), the set of properties (P) and the value (α) of minimum support for co-location patterns. The output of this is the Global co-location G .

In the second step we have to find the sequence of local co-locations l in each time window. This sequence is defined in the loop (line 15), by starting at the time window with the lowest time, therefore keeping the sequence of the local co-locations in time. For each time window (w), the algorithm generates local co-locations l also using the method

GENcolocation (line 16). Each local co-location is added to the set L (line 17), in the temporal order to build the sequence of local co-locations along the time interval. Finally, after computing the sequence of local co-locations, the last step is to add to the set of local co-locations L , the global co-location g (line 19).

Algorithm 2 GENcolocation(PT, w, α)

```

1: Output:
2:  $l$ : Local co-locations related to the time window  $\sigma$ .
3:
4: Method:
5:  $j = 1$  // index of local co-locations
6: Generate the set of candidate local co-locations  $c_j$ .
7:  $l_j = \emptyset$ 
8: repeat
9:   for Each candidate local co-location  $c_j$  do
10:    Count the proportion ( $r_c$ ) of trajectories which contain  $c_j$ 
11:    if  $r_c \geq \alpha$  then
12:       $l_j = l_j \cup c_j$ 
13:    end if
14:  end for
15:   $j = j + 1$ 
16:  Generate the set of candidate local co-locations  $c_j$ 
17: until  $c_j \neq \emptyset$ 
18: return  $l$ 

```

The method shown in algorithm 2 is the GENcolocation. This method generates the co-locations for each time window, including the global co-location at target time window W . For generating local co-locations, this method receives as parameters the set (T) of trajectories, the trajectory properties P , the time window (w) and a value of minimum support (α), in order to check the frequency of co-location in the specified time window. The first step is to find the subset of properties P of the trajectories occurring at the time window w (line 6). The method generates the set of candidate co-locations c_j (line 6), where j represents the size of the candidate co-location. For example, suppose $\{a, b, c, d\}$ represent the set of properties, this is the set of candidate co-locations with size 1 (c_1). This candidate generation can be done by any of the existing well known frequent pattern mining methods, as Apriori [Agrawal and Srikant 1995] or the closed frequent itemset approaches that are more efficient and generate less candidates [Zaki 2001].

For each candidate co-location c , the method computes the proportion of trajectories (in time window σ) which contain that candidate co-location (line 9). The candidate co-location with proportion r_c of the trajectories above minsupport ($r_c \geq \alpha$) (line 11), will be associated to the sequence of local co-locations l (lines 12). The loop will be executed for all combination of properties in the candidate c_j . Thereafter, the value of j will be increased in 1 in order to identify the next level of co-locations (line 15). With $j \geq 1$ the set of candidates represent the pairwise combination to generate the set of candidate co-locations with size 2 (c_2), as $ab;ac;ad;bc;bd;cd$, and $abc;abd;bcd$ represent the candidate with size 3 (c_3) (line 16). The output will be the last valid sequence of local co-locations defined in l (line 18).

5. Experiments and Evaluation

Two datasets were considered to evaluate the algorithm: a synthetic and a real dataset. The first one presents a previously defined environment to simulate a traffic jam occurrence. The second dataset has real car trajectories of the city of Rio de Janeiro.

5.1. Discretization

It is well known that for data mining and knowledge discovery the user may need to perform the discovery process several times, and therefore to discretize the data at different granularities [Han 1995]. In trajectories, this problem increases because the information of space and time has to be discretized [Bogorny et al. 2009]. All properties like speed, acceleration, distance, have to be discretized in intervals. Without discretization or generalization of the data, no patterns may be found. Therefore, we performed some experiments with different discretizations of trajectory properties that will be detailed in the following.

5.2. Experiments with the Synthetic Dataset

We have generated a synthetic dataset using *Brinkhoff - Trajectory Generator* [Brinkhoff 2000]. The dataset simulates the occurrence of several *traffic jam* phenomena. The dataset was generated based on the road network of the City of Oldenburg, which contains 6105 nodes and 7035 edges, having 73520 Moving Objects traveling on the network, with a total of 657281 Points representing the number of trajectory points.

We defined a spatial area to evaluate the algorithm, and considered on this dataset, the following properties (p) in the experiment: **Time** (representing the time taken by trajectories into a time window), **Presence** (number of trajectories), **Traveled Distance** (representing the traveled distance by trajectories into a time window), **Speed** and **Acceleration**.

In this experiment, we considered 10 different time windows (w): [0-1],[1-2],[2-3],[3-4],[4-5],[5-6],[6-7],[7-8],[8-9],[9-10]. We have simulated an occurrence of Traffic Jam (Global co-location - G) at time window 7 (target time window - w). The simulation considers that the traffic jam event starts at time window 4, evolves along the time and, at time window 7, it is totally formed.

The goal is to verify whether the algorithm is able to identify the global co-location (G) at time window 7 as well as to discover how this event starts and evolves along the time before time window 7. We considered a support of 0.70 for the local co-locations (l), i.e. at least 70% of the trajectories in every time window must have the same property of time, presence, etc, to be consider a valid co-location pattern. Each local co-location represents characteristics of a set of trajectories crossing the time window. Table 1 presents the local co-locations obtained with this dataset.

Table 1. Co-locations - Synthetic Dataset

Time Window	Co-location
4	time 14 + distance 6 + acceleration 6 + presence 2 + speed 3
5	time 16 + distance 4 + acceleration 2 + presence 4 + speed 2
6	time 18 + distance 2 + acceleration 2 + presence 6 + speed 1
7	time 19 + distance 0 + acceleration 0 + presence 7 + speed 0

The results clearly present the evolution of the traffic jam phenomenon along time windows. The emphasized area of the Table 1 presents the occurrence of a traffic jam - the global co-location - where speed, distance and acceleration became zero with a presence of 7 trajectories.

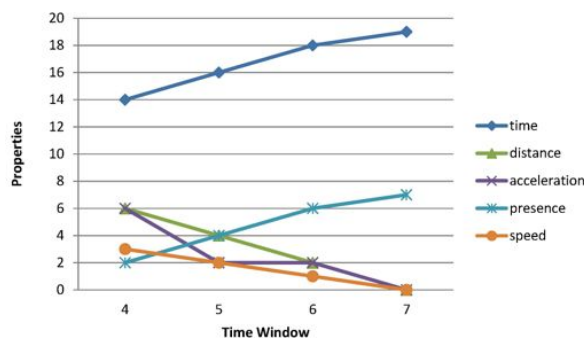


Figure 2. Time Window x Properties - Synthetic Dataset

At time window 4, the time value is at level 14 (time 14), meanwhile at time window 7, the same property is 19. Therefore, the time spent by set of trajectories, in the same spatial area, increased along time. An inverse behavior happens considering the distance attribute: the traveled distance decreases along the time. The same behavior happen with the properties acceleration and speed, that decrease significantly, reaching zero at time window 7. Finally, the number of distinct trajectories (property **presence**) in the same area increases along the time. The different local co-locations obtained at time windows 4,5 and 6 represent a classical evolution of the global phenomenon at time window 7. Figure 2 presents this behavior through the property evolution, where time duration and presence increase, while all other properties decrease at the last time window.

5.3. Experiments with Real Dataset - Rio de Janeiro

In this experiment we considered a real dataset representing the movement of 100 different car trajectories in the city of Rio de Janeiro (Brazil). Every car had the objective to cross the city at different days and times, equipped with a GPS device collecting points every second. The goal was to find low speed regions and traffic jams in the city. Figure 3 presents a visualization of those trajectories.

In order to execute this experiment we considered time windows of size 5 minutes, and similar properties of the trajectories as the previous experiment: **acceleration**, **time**, **distance** and **speed**. Only the attribute presence was not considered, because this set of trajectories was produced by the same object, but in different days, not having groups of trajectories moving together in time. To find low speed regions in these trajectories we applied the algorithm CB-SMOT [Palma et al. 2008], which is a speed-based clustering algorithm. This algorithm discovered the Global event such that we can evaluate our algorithm. Starting from the global event found with CB-SMOT, we expanded that regions in large rectangles, show in Figure 3.

Having done this, we use these rectangles as the areas to be analyzed and split these rectangles in time windows, to discovery the trajectory property change before resulting in the low speed region. We performed several experiments, and show in this paper that one performed on rectangle 6 in Figure 3.

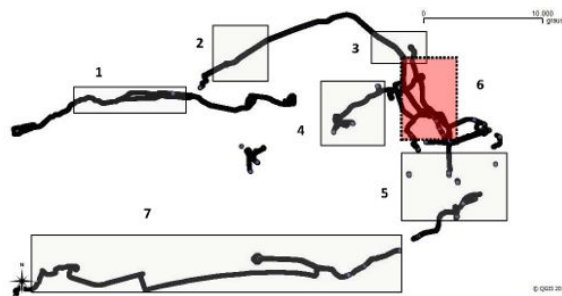


Figure 3. Rio de Janeiro - Trajectories - Rectangles

The discretization intervals of this experiment are presented in Table 2. The attribute acceleration was discretized in three intervals: $\{-1, 0, 1\}$, where -1 is deceleration, 0 represents stable movement and 1 is acceleration. The continuous properties of *speed* and *distance* were discretized in eighth intervals. Table 2 presents these intervals.

Table 2. Discretization 1

Attribute	Intervals	Discretization Intervals
Acceleration	$[-\infty, -1], [-1, 1], [1, +\infty]$	3
Distance	$[0,10],[11,20],[21,30],\dots,[61,70],[71,80]$	8
Speed	$[0,15],[16,31],[32,47],\dots,[80,95],[96,111]$	7

Table 3 presents the results obtained in this experiment. There are 9 different timewindows into the interval between timewindow identifier 01 and 09. Each of them represent a 5 minutes interval. The emphasized area of the table presents the occurrence of the Global co-location. The previous rows present the sequence of Local co-locations that result in the occurrence of the Global co-location. It is possible to identify that the environment remains almost stable between time windows 01 and 03 (15 minutes). Thereafter, during the next 15 minutes (time windows 05, 05 and 06), local co-locations present an environment of decrease of acceleration, distance and speed properties. This behavior suggests that the traffic jam phenomenon is beginning. Finally, at time windows 07, 08 and 09, the traffic jam phenomenon is totally configured and remains stable.

The same behavior of the environment cab identified by Figure 4. It is possible to note that the values of properties *speed* and *distance* decrease along time. After time window 06, the environment remains stable, configuring an environment of traffic jam. Therefore, the co-locations patterns allow to identify the process of occurrence of a traffic jam phenomenon.

Table 3. Co-locations

Time Window	Co-location	Type
01	acceleration 1 + distance 3 + speed 6	<i>local - c1</i>
02	acceleration 1 + distance 3 + speed 1	<i>local - c2</i>
03	acceleration 1 + distance 3 + speed 1	<i>local - c3</i>
04	acceleration 1 + distance 2 + speed 1	<i>local - c4</i>
05	acceleration 1 + distance 0 + speed 1	<i>local - c5</i>
06	acceleration 0 + distance 0 + speed 0	<i>local - c6</i>
07	acceleration 0 + distance 0 + speed 0	<i>local - c7</i>
08	acceleration 0 + distance 0 + speed 0	<i>local - c8</i>
09	acceleration 0 + distance 0 + speed 0	<i>global</i>

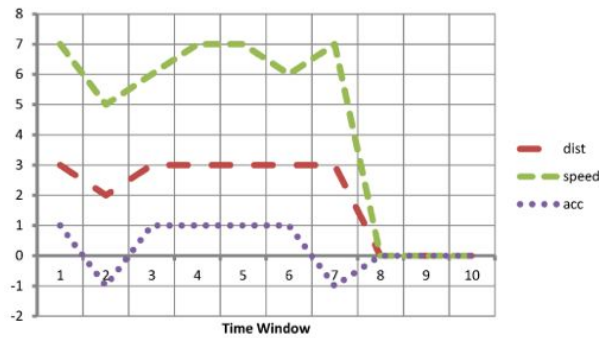


Figure 4. Time Window x Properties - Real Dataset

6. Conclusions

The development of the mobile technology contributes to generate large data volumes. A trajectory of a mobile object can be represented by a ordered sequence of spatial positions. The analysis of these data volumes allows to understand the behavior of a set of objects into a spatio-temporal area. There are some works introducing technics to identify movement patterns of a set of moving objects considering trajectories data. In this work we introduced a method to identify trajectory co-location patterns by using different characteristics of trajectories. Besides, the method allows identify sequences of co-locations in time that cause an event called target event.

We conducted two different experiments: the first one considering a synthetic dataset and, another considering a real dataset. For the two experiments the goal was the same: considering a target event composed by a combination of different property values (speed, distance, presence, time etc), identify the sequence of co-locations that generate the target event.

The results obtained in the two experiments suggest that the method can identify the sequence of co-location patterns that cause the target event. These patterns can be useful in order to anticipate the occurrence of an event (e.g. traffic jam) considering the previous behavior of a set of properties (e.g. distance, speed, acceleration, presence) into a time interval (e.g. 5 minutes). This knowledge could be useful for a traffic management system. Instead of present a description of the environment for a given time, the method allows to understand the formation of the event. Understanding the evolution the event, it is possible to anticipate actions and strategies in order to prevent the occurrence of the phenomenon.

Future research involves the development of OLAP operators for use in environments of trajectory data warehouses. These operators allow the identification and aggregation of trajectories with some level of similarity, considering a spatial-temporal area. Moreover, the development of an information system acting on a database or trajectory data warehouse, also is being developed.

References

- Agrawal, R. and Srikant, R. (1995). Mining sequential patterns. In Yu, P. S. and Chen, A. S. P., editors, *Eleventh International Conference on Data Engineering*, pages 3–14, Taipei, Taiwan. IEEE Computer Society Press.

- Bogorny, V., Kuijpers, B., and Alvares, L. O. (2009). St-dmql: A semantic trajectory data mining query language. *International Journal of Geographical Information Science*, 23(10):1245–1276.
- Braz, F. (2008). Knowledge discovery on trajectory data warehouses: Possible usage of the data mining techniques. In *SBSI 2008: Proceedings of the 4th Brazilian Symposium on Informatics*, Rio de Janeiro, Rio de Janeiro, Brazil. SBC Brazilian Computer Society.
- Brinkhoff, T. (2000). Generating network-based moving objects. In *SSDBM '00: Proceedings of the 12th international Conference on Scientific and Statistical Database Management*, page 253, Washington, DC, USA. IEEE Computer Society.
- Cao, H., Mamoulis, N., and Cheung, D. W. (2006). Discovery of collocation episodes in spatiotemporal data. In *ICDM '06: Proceedings of the Sixth International Conference on Data Mining*, pages 823–827, Washington, DC, USA. IEEE Computer Society.
- Celik, M., Kang, J. M., and Shekhar, S. (2007). Zonal co-location pattern discovery with dynamic parameters. In *ICDM '07: Proceedings of the 2007 Seventh IEEE International Conference on Data Mining*, pages 433–438, Washington, DC, USA. IEEE Computer Society.
- Celik, M., Shekhar, S., Rogers, J. P., and Shine, J. A. (2008). Mixed-drove spatio-temporal co-occurrence pattern mining. In *IEEE Transactions on Knowledge and Data Engineering*, Washington, DC, USA. IEEE Computer Society.
- Han, J. (1995). Mining knowledge at multiple concept levels. In *CIKM*, pages 19–24. ACM.
- Huang, Y., Shekhar, S., and Xiong, H. (2004). Discovering colocation patterns from spatial data sets: A general approach. *IEEE Transactions on Knowledge and Data Engineering*, 16:2004.
- Li, Z., Ding, B., Han, J., Kays, R., and Nye, P. (2010). Mining periodic behaviors for moving objects. In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '10, pages 1099–1108, New York, NY, USA. ACM.
- Palma, A. T., Bogorny, V., Kuijpers, B., and Alvares, L. O. (2008). A clustering-based approach for discovering interesting places in trajectories. *Proceedings of the 2008 ACM symposium on Applied computing SAC 08*, (December):863.
- Shekhar, S. and Huang, Y. (2001). Discovering spatial co-location patterns: A summary of results. In *Lecture Notes in Computer Science*, pages 236–256.
- Yoo, J. S. and Shekhar, S. (2006). A joinless approach for mining spatial colocation patterns. *IEEE Trans. Knowl. Data Eng.*, 18(10):1323–1337.
- Zaki, M. J. (2001). SPADE: An efficient algorithm for mining frequent sequences. *Machine Learning*, 42(1/2):31–60.
- Zheng, Y. and Xie, X. (2010). Learning location correlation from gps trajectories.